IRRIGATION INFORMATICS

Addressing poor irrigation decision support system uptake

Nicholas John Car
ORCID: 0000-0002-8742-7730

Philosophy Doctorate
June, 2019

Melbourne School of Engineering
The University of Melbourne

This thesis is being submitted in total fulfilment of the degree
Abstract

Irrigation Decision Support Systems (DSS), in the first decade and a half of the 21st century, suffered from a lack of uptake by their target audience, namely farmers, according to a number of researchers in Australia and overseas. Reasons for this were proposed and solutions to overcome them explored. These included a “gap” between DSS research/development practice and on-farm practice with a solution being to involve irrigation practitioners in the planning and development of DSS; so-called participatory action research. The wider irrigation research community also undertook forms of social science investigation, such as surveys and networked interviewing, to better understand real-world irrigation decision making.

That work did not evaluate, in any depth, the utility to farmers of new styles of DSS delivery based on the profound telecommunications and information technology changes in the early 2000s. Nor did that work much look to how decision-making in general may be better catered for via DSS.

This thesis does these things by investigating approaches taken to assist irrigation DSS uptake in the field of informatics – the science of information. The fundamental hypothesis being tested is that it is supposed there are real gains to be made in both improving the technical delivery mechanics of irrigation DSS’ advice to farmers and the better staging of that advice through better technical handling of aspects of decision theory.

Related here are several experiments conducted over the decade 2007 – 2017. They look at how:

- information has been, and could be, presented to irrigators to help them use it;
- more sources of knowledge used by irrigators, perhaps available but not currently used, could be incorporated into DSS to make them more relevant to real-world practice;
- the decisions that DSS hope to assist with can be modelled themselves, as distinct from biophysical irrigation modelling, in order to empirically determine irrigation best-practice and communicate decision norms.

Additionally, work over time for this thesis has shed some insights into how changing technology availability, use and general acceptance on Australian farms has shifted the goalposts of DSS uptake.

This thesis shows that:

- there have been, and are likely to continue to be, real gains to be made in the adoption of DSSs through technical system design such as user interface design;
- through both improving DSS design and growing data availability via technology change, the range of data sources available for and used by DSS is growing and this is likely to assist with their uptake. This is due to them being able to incorporate more data sources used in real-world decision making;
some issues preventing DSS uptake have disappeared due to technology change on Australian farms, some remain and new ones have appeared;

- decision modelling, as opposed to biophysical or economic modelling, has not really been undertaken by irrigation DSS designers and yet there are both existing decision modelling systems that designers could use and the potential to create other, better, models which may help with uptake.

Recommended future investigations from this work are:

- addressing the newly emergent issue of ‘data deluge’ which is beginning to plague DSS designers; which source of weather or commodity price should they use? A utility assessment of similar sources of data could be conducted;

- building of collections of real-world irrigation decisions made, modelled using approaches from this thesis, and the testing of automated approaches for assessing their outcome. This would test the practicality of empirical decision-modelling-based DSS;

- testing of the utility to irrigators of decision-modelling-enabled DSS. If current problems facing farmers can be matched to real-world best-practice, does this offer superior utility to calculated, theoretical best-practice?
I, Nicholas John Car, declare that:

♦ This thesis comprises only my original work towards the PhD except where indicated in the preface.
♦ Due acknowledgement has been made in the text to all other material used
♦ This thesis is fewer than the maximum word limit in length (100,000 words), exclusive of tables, maps, bibliographies and appendices.

Signed: _______________ Date: _______________
Preface

Papers

Here follows a description of work towards this thesis carried out in collaboration with others. Indicated are the portions of the work which I, Nicholas John Car, claim as original.

Journal Articles

The following journal articles are related to this PhD:

   ♦ 100% contribution: sole author

   ♦ 100% contribution: sole author

   ♦ 20% contribution: authorship proportion share equally between co-authors

   ♦ 75% contribution: primary author with editorial assistance from co-authors only
Conference Papers

The following conference papers are related to this PhD:

   ♦ 100% contribution: sole author

   *http://www.isess2017.org/res/file/presentations/P04-Car.pdf*
   ♦ 50% contribution: primary author with assistance from co-authors

   *10.1007/978-3-319-15994-2_18*
   ♦ 33.3% contribution: authorship proportion shared equally

   *10.1007/978-3-319-15994-2_9*
   ♦ 25% contribution: authorship proportion shared equally

   ♦ 75% contribution: lead author, editorial assistance from co-author only


♦ 75% contribution: lead author, editorial assistance from co-authors only


♦ 75% contribution: lead author, editorial assistance from co-authors only


♦ 25% co contribution: authorship proportion shared equally


♦ 50% contribution: lead author, with assistance from co-authors


♦ 50% contribution: lead author, with assistance from co-authors
Reports

The following scientific reports are related to this PhD:

   - 25% co contribution: authorship proportion shared equally

   - 25% co contribution: authorship proportion shared equally

Funding

*Here follows an acknowledgement of all sources of funding.*

I, Nicholas John Car, was funded to undertake PhD research through the Cooperative Research Centre for Irrigation Futures between 2006 and 2010. After that date, I received no funding for PhD work.
Acknowledgments

I acknowledge the generous assistance given me anonymously by the very many members of the Open Source Software community. This community has been the largest contributor to my understanding of software and modelling. Many people contribute to Open Source projects for the love of doing something useful but they don’t always get thanks so, thank you.

I acknowledge my University of Melbourne supervisors: Graham Moore and Andrew Western. Graham guided me academically and showed great patience through many years of little action and assisted right up to his retirement. Andrew took me under his wing for the final stages of the PhD, including the difficult task of actually submitting and improving the thesis document. Thank you to both of you.

I acknowledge my CSIRO supervisor, Evan Christen. Evan was both the initiator of the PhD institutional arrangements and a professional and personal support throughout the first few years of the PhD. I hope seeing this final document brings you a happy sense of closure.
# Table of Contents

## Front matter

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Title Page</td>
<td>i</td>
</tr>
<tr>
<td>2</td>
<td>Abstract</td>
<td>iii</td>
</tr>
<tr>
<td>3</td>
<td>Declaration</td>
<td>iv</td>
</tr>
<tr>
<td>4</td>
<td>Preface</td>
<td>v</td>
</tr>
<tr>
<td>5</td>
<td>Acknowledgements</td>
<td>ix</td>
</tr>
<tr>
<td>6</td>
<td>Table of Contents</td>
<td>x</td>
</tr>
<tr>
<td>7</td>
<td>List of Tables and Figures</td>
<td>xii</td>
</tr>
</tbody>
</table>

## Chapters

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td><strong>Chapter 1</strong>: Introduction</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td><strong>Chapter 2</strong>: IrriSatSMS DSS</td>
<td>30</td>
</tr>
<tr>
<td>11</td>
<td><strong>Chapter 3</strong>: IrriSatSMS Extension work</td>
<td>43</td>
</tr>
<tr>
<td>12</td>
<td><strong>Chapter 4</strong>: DSS Case Study</td>
<td>51</td>
</tr>
<tr>
<td>13</td>
<td><strong>Chapter 5</strong>: Data Sources</td>
<td>73</td>
</tr>
<tr>
<td>15</td>
<td><strong>Chapter 6</strong>: Decision Modelling</td>
<td>89</td>
</tr>
<tr>
<td>16</td>
<td><strong>Chapter 7</strong>: Discussion</td>
<td>115</td>
</tr>
<tr>
<td>17</td>
<td><strong>Chapter 8</strong>: Conclusion</td>
<td>120</td>
</tr>
</tbody>
</table>
Appendices

A  PhD Topic Progression, 2009 – 2019  122
B  Thesis topic progression details  131
E  Project Report – SEQ ET0 SMS Trial 1st Year Report  231
F  Experiment Report – Crop Coefficient estimation from mobile phone images  237
G  Conference Poster – Case-Based Reasoning decision support using the DecPROV ontology for decision modelling  256
H  Source Code – for the DecPROV figures in Chapter 1 – Introduction  257
List of Figures and Tables

Many tables and figures in this thesis have already been published in scientific papers. Where this has occurred, the table or figure is numbered sequentially, as it appears in this thesis, however its caption also contains the table or figure's number as per the original publication. Additionally, the page number of the figure in the original publication, if recorded, is indicated in parenthesis below the thesis' page number.

List of Figures

<table>
<thead>
<tr>
<th>No.</th>
<th>Caption</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Chapter 1</strong></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>A graphical version of height = 100sin(x), distance = {0..315}</td>
<td>7</td>
</tr>
<tr>
<td>2</td>
<td>The irriGATE project's evapotranspiration graph</td>
<td>14</td>
</tr>
<tr>
<td>3</td>
<td>WaterSense's more intuitive evapotranspiration graph from (Inman-Bamber &amp; Attard 2007)</td>
<td>14</td>
</tr>
<tr>
<td>4</td>
<td>Proportion of Australian mobile phones that were SmartPhones (left) after (Mackay 2014). Proportion of Australian rural households with internet access (right) from (Australian Bureau of Statistics 2009; Australian Bureau of Statistics 2014; Australian Bureau of Statistics 2011; Australian Bureau of Statistics 2016)</td>
<td>17</td>
</tr>
<tr>
<td></td>
<td><strong>Chapter 2</strong></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Fig. 1: System diagram of the DSS showing components and communication methods.</td>
<td>33 (134)</td>
</tr>
<tr>
<td>6</td>
<td>Fig. 2: Mobile phone screen showing a typical DSS daily SMS message.</td>
<td>33 (134)</td>
</tr>
<tr>
<td>7</td>
<td>Fig. 3. (A) Participating irrigators’ farm sizes, (B) participating irrigators’ ages, and (C) participating irrigators’ irrigation experience.</td>
<td>34 (135)</td>
</tr>
<tr>
<td>8</td>
<td>Fig. 4. Count of irrigators sending in certain numbers of messages per month.</td>
<td>36 (137)</td>
</tr>
<tr>
<td>9</td>
<td>Fig. 5. Partial cumulative CWD graphs for 2 irrigators showing differing responses to the system. The top graph (A) shows the irrigator kept a positive CWD using both rainfall and irrigation while the bottom graph (B) shows the irrigator returning the CWD to zero when it reached a fixed approximate value of about 35 mm.</td>
<td>37 (138)</td>
</tr>
<tr>
<td>10</td>
<td>Fig. 6. Full season CWD lines for 3 irrigators: ‘More’, ‘Close’ and ‘Less’.</td>
<td>38 (139)</td>
</tr>
</tbody>
</table>
Fig. 7. Number of irrigators grouped by utility scores for different time periods in the season.

Fig. 8. Average irrigator assessment of utility of the system for different time periods in the season.

Chapter 3

The areas of IrriSatSMS extensions indicated on a map of Australia showing proportion of irrigated land cover. The blue numbers 1 – 5 indicate the areas in which extensions, as per the number list above, were conducted.

Chapter 4

Figure 1: A system diagram outline of the IrriSatSMS system (after Car et al., 2012).

Figure 2: A system diagram outline of the IrriSat system (after Vleeshouwer, Car and Hornbuckle, 2015).

Figure 3: Three parts of the IrriSat web UI. Clockwise from top left: a user’s new test field marked out ready for analysis; the waterbalance graph of the test field; the test field’s crop coefficient taken from blended satellite imagery (LANDSATs 7 & 8).

Figure 4: Three parts of the IrriSatSMS web UI. Clockwise from top left: a crop’s waterbalance trace; past irrigation (i) and rainfall (r) data inputs for block ‘a’; the web form for adding irrigations and rainfall via the web UI.

Figure 5: Issues with IrriSatSMS commercialisation and the corresponding IrriSat design feature attempting to deal with it.

Figure 6: Issues with IrriSatSMS commercialisation and the corresponding IrriSat institutional arrangement attempting to deal with it.

Figure 7: Proportion of Australian rural households with internet access (Australian Bureau of Statistics 2009, 2011, 2016 & 2016).

Figure 8: Proportion of Australian mobile phones that were SmartPhones (Mackay, 2014).

Figure 9: The human-readable (HTML) web page UI for the Lyrup Flats weatherstation from http://aws.naturalresources.sa.gov.au/samurraydarlingbasin/?aws_id=RMPW05&view=summary

Figure 10: Data from the machine-readable API for the Lyrup Flats weatherstation corresponding to Figure 3 from http://aws.naturalresources.sa.gov.au/api/data/?timestep=minutes&station_ids=RMPW05&start_date=2015-04-17&end_date=2015-04-17.

Chapter 6.1

Fig. 1: A DMN Decision Requirements Graph for "How much should I water today?".
Fig. 2: A more detailed Decision Requirements Graph model of the decision “How much should I water today?” and other decisions and business knowledge that it depends on. Shown also are Knowledge sources.

Fig. 3: The main classes of the DO and their relationships. Note that "Thing" is the base class for everything in OWL modelling.

Fig. 4: DO normative use modelling of the irrigation decision “How much should I water today?” using classes as per Fig. 3. The main diagram is Normative use is with objects in blue being presented to indicate what data-driven use would look like. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Fig. 5: Decision Making Ontology elements shown in a simple UML Class Diagram.

Fig. 6: The Decision-Making Ontology used to model the irrigation question “How much should I water today?” using elements from Fig. 5.

Fig. 7: A UML Object Diagram splitting “How much should I water today?” into multiple, dependent, decisions, modelled using DMO. Elements as per Fig. 6.

Chapter 6.2

Figure 1: Class models of PROV-O (left) and DO (right)

Figure 2: Diagram of the DO example “Bacterial throat infection” based on data from (Nowara 2012)

Figure 3: Core elements of Figure 2 with Question/Answer linking (not in original example), an addition of Patient X & rewording of the Deciding shown in blue and a new Requirement and requirement satisfaction object addition shown in orange. The non-DO Decision Maker object and relationships added are indicated in red

Figure 4: A PROV-O class and property-only representation of Figure 3

Figure 5: Class diagram of the DecPROV ontology (https://promsns.org/def/decprov)

Figure 6: A DecPROV representation of Figure 4

Figure 7: Why questions answered using DecPROV (green outline)

Chapter 6.4

Figure 1: A reproduction of Figure 8 from the DecPROV repository’s examples, see https://promsns.org/def/decprov. Here the decision “How much should I water today?” (HMSIWT), as originally modelled in Section 7.1 using multiple systems is modelled using DecPROV.
List of Tables

<table>
<thead>
<tr>
<th>No.</th>
<th>Caption</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Chapter 1</strong></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Author's categories of DSS, based on the DSS's networking function, as per Literature Review draft, 20019 (see Appendix B)</td>
<td>9</td>
</tr>
<tr>
<td>2</td>
<td>Australian Irrigation DSS, circa 2009</td>
<td>17</td>
</tr>
<tr>
<td>3</td>
<td>Australian Irrigation DSS, circa 2009, continued</td>
<td>17</td>
</tr>
<tr>
<td>4</td>
<td>Research topics mapped to thesis sections</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Chapter 2</strong></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Table 1 Irrigation scheduling method and tool use by the 72 trial irrigators before trial commencement.</td>
<td>35</td>
</tr>
<tr>
<td>2</td>
<td>Table 2 Interactions between trial participants and researchers.</td>
<td>35</td>
</tr>
<tr>
<td>3</td>
<td>Table 3 Days of faulty messages, shown in brackets, [], and days of messages not sent, shown without brackets, for the whole seasons</td>
<td>35</td>
</tr>
<tr>
<td>4</td>
<td>Table 4 Number of irrigators who sent in messages for two major rainfall events</td>
<td>37</td>
</tr>
<tr>
<td>5</td>
<td>Table 5 Number of irrigators per water balance group type</td>
<td>39</td>
</tr>
<tr>
<td>6</td>
<td>Table 6 Number of irrigators per behaviour change type</td>
<td>40</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Chapter 4</strong></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Figure 5: Issues with IrriSatSMS commercialisation and the corresponding IrriSat design feature attempting to deal with it</td>
<td>60</td>
</tr>
<tr>
<td>8</td>
<td>Figure 6: Issues with IrriSatSMS commercialisation and the corresponding IrriSat institutional arrangement attempting to deal with it</td>
<td>61</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Chapter 6.1</strong></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Table 1: A possible DMN Decision Table for the decision &quot;How much should I water today?&quot;, as represented in Figure 2. The Input Data (items in blue) are taken directly from Figure 2. FEEL expressions are used for table cell values' notation</td>
<td>95</td>
</tr>
<tr>
<td>10</td>
<td>Table 2: A DMN Boxed Invocation for the decision &quot;How much should I water today?&quot;, as represented in Fig. 2.</td>
<td>96</td>
</tr>
</tbody>
</table>
Chapter 6.2

Table 1 Classes of DO and their PROV-O classes to which they are associate by being subclasses of them
1. Introduction

Irrigation is the largest use of fresh water globally and in Australia it constitutes about 60% of all human water consumption. Since water resources are already heavily exploited globally, improvements in water use efficiency are required to feed a growing population. One means for doing this is through better irrigation scheduling, that is, the application of appropriate volumes and timing of water to crops. Irrigation Decision Support Systems (DSSs) have long promised to assist with scheduling efficiency by the presentation of digital data and computer calculations to decision makers helping them to make informed, objective decisions, however the poor uptake of DSSs within irrigation and the agricultural sector generally has been noted as a large, multi-faceted, problem (Matthews, 2008). This thesis aims to improve the uptake of DSSs by irrigation decision makers in Australia by the application of the field of informatics, defined below, to irrigation DSS design.

This chapter presents definitions of terms and the general research scope of this PhD and then several stages of literature review that were conducted starting in 2007 in the lead up to the first experiments of this PhD and until before presentation of the papers making the final content chapter (Chapter 6) in 2018. It then lists the research questions ultimately investigated in this thesis as well as presenting a thesis outline. Finally, a description of the science disciplines of this PhD – informatics & irrigation – and its application domain – decision support systems – are given.

1.1. Definitions and Research scope

Irrigation Informatics is a term that has been used, from 2005 onwards, to describe the use of informatics within the irrigation context, by the Cooperative Research Centre for Irrigation Futures (CRC-IF) which was the funding body of this PhD. Informatics, itself a recent word, the author takes to be defined as the science of information use. The Indiana Uni School of Informatics states that

“Informatics is a bridge connecting IT to a particular field of study such as biology, chemistry, fine arts, telecommunications, geography, business”

---


2 By referring to the science of information use itself, rather than mathematics and science that facilitate information use the author differentiates Informatics from Information Theory. The Wikipedia, used for common definitions, describes Information Theory as “a discipline in applied mathematics involving the quantification of data with the goal of enabling as much data as possible to be reliably stored on a medium or communicated over a channel.” (http://en.wikipedia.org/wiki/Information_theory). The author believes this is a standard, uncontroversial, definition of Information Theory.

3 In 2006, the Indiana University School of Informatics authored a web pages titled “What is Informatics?” that contained this definition at the now-defunct web address http://informatics.iupui.edu/overview/what_is_informatics.php.
The Cooperative Research Centre for Irrigation Futures (CRC-IF) stated[4] that informatics is:

“often, though not exclusively, studied as a branch of computer science and information technology and is related to database, ontology and software engineering. It focuses on the use of technology for improving access to, and utilization of, information.”

and, as a concept is about:

“developing the science of storage, retrieval, and optimal use of biophysical information, data, and knowledge for problem solving and decision making in irrigation management, e.g. Yield/$’s per irrigation decision”[5]

The aim of this research fell within the CRC-IF’s vision of:

“making better use of Australia’s water resources ...[by]... having state of the art, robust tools which can be adopted by water managers at all levels for more precisely managing and accounting for water use”[6]

These definitions focussed this thesis on applications of informatics and decision theory to decision support systems to support irrigation decision making.

1.2. Literature Review

1.2.1. Informatics

The choice of specific areas of informatics, within computer science and information technology, investigated for this thesis were not initially set but emerged from needs in the application domain(s).

Initially, much effort was expected to be placed into the areas of Internet data communications to support irrigation Decision Support Systems (DSS) for, in 2006, not many “Internet-enabled” irrigation DSS were known[7] and the possibilities of enabling users to access DSS and for remote sources of data to be provisioned to DSS seemed great.

The next few subsections of this chapter each refer to aspects of informatics, such as software engineering, information systems design and ontology and this is indicated in


[5] Presentation by Evan Christen, this PhD’s industry supervisor, to CRC-IF project leaders, Brisbane, 2005.

[6] Also from the CRC-IF website, see previous Footnote

[7] One notable Internet-enabled irrigation DSS that appeared in 2006 was WaterSense which was billed as “an internet DSS for sugarcane irrigation scheduling”, see (Inman-Bamber et al. 2006).
them. The general informatics concepts are then reviewed with a specific focus on irrigation, starting in Section 1.2.3.7.

1.2.2. Decision theory

Decision theory incorporates a range of disciplines that include philosophy, education, psychology, mathematics (probability, Bayesian, fuzzy logic) and computer science (Boolean logic, artificial intelligence). Decision theory aims at an understanding of what decisions are to be made and how and why people make the decisions they do. Decision theory was investigated in the context of this PhD to see if aspects of it could contribute to better decision making processes generally and then better DSS design. Firstly, general theories of decision making were examined and secondly its relation to the context of irrigation, was explored.

Much work has been done in the area of philosophical decision theory in the last three centuries. Early work on decision quantification, principles for rational decision making and decisions leading to ‘fair’ outcomes, as judged by egalitarian principles, was undertaken by French enlightenment philosophers, notably the Marquis de Condorcet (1743 - 1794) (de Condorcet 1793) in justification of aspects of the constitution for the French Republic of 1793. de Condorcet established processes and principles for decision making. An example: in his ‘stage one’ of the decision-making process, the stage containing an individual’s decision process that leads into further stages of group decision making, an individual:

“discusses the principles that will serve as the basis for decision in a general issue; one examines various aspects of the issue and the consequences of different ways to make the decision”

Explicitly describing a process to be followed that leads to rational decision making seems obvious by today’s standards but was new at the time of de Condorcet’s writing.

According to the American education philosopher James Dewey (Dewey 1910), five consecutive stages of decision making can be followed, one after another, to assist an individual in the decision-making process. These stages are:

1. a felt difficulty
2. the definition of the character of that difficulty
3. suggestion of possible solutions
4. evaluation of the suggestion
5. further observation and experiment leading to acceptance or rejection of the suggestion

Hansson (1994) gives further, more recent, developments of this sequential decision-making process and discusses criticisms of stepwise, sequential (termed ‘linear’ in decision theory literature) decision making processes. It is clear to this author that not much more can be helpfully said about linear decision-making process other than what was said by Dewey – the author feels that Dewey’s process is effectively up to date. The criticisms of linear models of the decision-making process are recent and are neatly explained by (Witte 1972).

“We believe that human beings cannot gather information without in some way simultaneously developing alternatives. They cannot avoid evaluating these alternatives immediately, and in doing this they are forced to a
decision. This is a package of operations and the succession of these packages over time constitutes the total decision-making process.”

The author feels that there is much merit in looking further into linear and non-linear decision-making processes and how decision support system (DSS) design might benefit from work in this area by helping deciders to arrive at successful decisions by undertaking these processes. Both linear and non-linear DSS front end design is considered fully in Section 1.2.5.4.

Within Dewey’s fourth (evaluation of the suggestion) and fifth (further observation and experiment leading to acceptance or rejection of the suggestion) stages: a decider is asked to choose an outcome, potentially from a range of possible alternatives, based on certain criteria. Decision theory has long tried to parameterize possible alternatives to enable a user to weigh them against one another, as rationally as possible, to prevent certain merits or demerits of those alternatives from being overlooked or under emphasized.

In this regard, histories of decision theory, such as (Murphy 1998), rapidly lead to mathematical probability, then to Bayesian decision theory, game theory and other mathematical and scientific theories. Modern usage of probability is commonplace for decision making and Bayesian and other theories are incorporated into many DSS.

The extent to which we can quantify decision parameters, unknowns and risk leads into the philosophical field of philosophical logic and from there into mathematical logic. Using mathematical logic, the upper bound on establishing a complete and consistent set of mathematical axioms that can be used to describe problem parameters is established by limitations given in Gödel’s incompleteness theorems (Hirzel 2000). Additionally, this author has had experience with logic philosophers (John Howse of Melbourne University in particular) who have hotly disputed the results of various philosophical logic problem expressions that leads him to believe that there is no consensus in logical philosophy either.10

Necessarily related to the above theme are attempts to quantify what exactly ‘good’ and ‘bad’ outcomes are. This process, which the author will refer to as the gauging of utility, is fundamental to decision theory for without it one cannot interpret the results of decisions made. The process of gauging utility is a large topic and will not be addressed in this literature review. It will be a major component of methodology and design chapters.

Decision theory has yielded many ‘paradoxes’ by describing scenarios where people seem to make decisions that are counter to their own goals. Recent undergraduate course texts such as (Hansson 1994), cover topics such as ‘Decision making under

---

8 CMU (2019) states that “A decision theory based upon utility is intimately related to theories of probability, which are needed for the calculation of expected consequences.”

9 Pers. comms. with John Howse on the topic of “problems with logical expressions in philosophy”.

10 These rather esoteric mathematical and logical limits were investigated by the author to see whether their bounds may be the limiting factor in decision making. The author feels that he can conclusively say they are not, for the limitations of agricultural understanding and the uncertainty in weather and other environmental patterns place far greater bounds on effective decision making. Further to this, the author hopes that techniques in DSS design employed to circumvent problems posed in the Subsection 2.2.1.4 will also circumvent the bounds mentioned here.
uncertainty’, ‘Decision making under ignorance’ and ‘Decision instability’ in which we can find instances of these ‘paradoxes’, such as Newcomb’s Paradox. Another paradox is known as ‘Death in Damascus’ which is a conundrum exploiting the concept of decision making under instability where the outcome of a decision is not known to the decider until after the decision is made, thus preventing good predictions of decision outcome until it is too late. These paradoxes raise questions about the validity of the decisions that individuals can make in that they challenge what we think of as ‘rational’ and perhaps point out that, in certain scenarios, it may not be possible to make a rational decision or that an individual will usually make a supposed irrational decision.

A recent example of a slightly different paradox is described at length in (Fenton & Neil 2000). This paradox or, more correctly, reasoning fallacy, is a result of lay people, in this case a jury, misunderstanding the results of Bayesian statistics. The lay people then are inclined to believe that people who are likely to be innocent are in fact likely to be guilty and therefore they convict them.

From these readings, the author can see that there is always a possibility that a new theory of statistics or probability will lead to a better understanding of a decision scenario. If a new theory does arise, how is it to be incorporated into an already established decision making process or DSS? In terms of paradoxes and reasoning fallacies known, the author believes that they possess common attributes and that many of them can be avoided, for certain decision arenas (such as irrigation management) at least, with the correct decision situation establishment and good problem presentation. These thoughts will be expanded upon, with examples, in Section 1.2.3 and actual DSS designs that implement them in Section 1.2.5.4.

For new interpretations of data or for new ways to parameterise decision factors, we must look to agricultural and other science, not philosophy or statistics, and that it outside the scope of this PhD.

1.2.3. Decision theory in the context of irrigation

The above relation of decision theory is very broad. This section discusses the specific sorts of decision that we encounter in the irrigation management sector.

Irrigation management decisions are complex, multifaceted decisions that do not seem to be directly comparable to many of the examples given in philosophical papers on decision theory. One reason for this is undoubtedly that, for the sake of making a philosophical point, only simple examples are used so that factors other than those under examination do not cloud the issues examined. This is not the only reason though: another is that the philosophical examples given are usually out of any context and certainly do not consider aspects of real decision making to do with a decision maker’s familiarity with the general area of the current decision to be made. Adding in ‘real life’ detail and placing a decision into a context where the decider has background knowledge should be enough to avoid some of the theory paradoxes given above. The next section elaborates on the differences between theoretical example decisions and irrigation management decisions.

11 There are many discussions of Newcomb’s Paradox on the internet with Wikipedia (https://en.wikipedia.org/wiki/Newcomb’s_paradox) giving a complete discussion.
In all the decision theory paradoxes that the author has seen, the paradoxes are ‘solved’ by establishing accurate expressions of the decision maker’s expected utility (EU) of decision outcome, the quantitative assessment of the decision outcome’s expected value (EV) and, critically, in making sure that these expressions are understood by the decision maker. The author believes that many of the ‘problems’ only exist when either the EU and EV of a decision are not well known or when the decision maker does not understand what the expressions of EU or EV are telling them. An example: it is due to a lack of understanding of what decision makers are likely to see as a return, EV, that leads then to violate the expected utility hypothesis in the case of Ellsberg’s Paradox (Ellsberg 1961). To fully illustrate, Ellsberg’s Paradox states that: suppose you have a box containing 30 red balls and 60 other balls that are either black or yellow. You don’t know what the ratio of black to yellow balls is, only that the total number of black plus yellow balls equals 60. The balls are evenly mixed so that each individual ball is as likely to be drawn out of the box as any other. You are now given two sets of two wagers:

A: You receive $100 if you draw a red ball
B: You receive $100 if you draw a black ball

and

C: You receive $100 if you draw a red or yellow ball
D: You receive $100 if you draw a black or yellow ball

The paradox occurs when one must choose one wager from A or B and then another from C or D. The reason for this is that if you prefer A to B, based on the notion that you think drawing a red ball is more likely than a black ball, then you should prefer C to D for the same reason. Supposing you prefer B to A then, by the same measure, you should prefer D to C. However, when surveyed, people strictly prefer A to B and D to C. This violates expected utility theory (The idea that you have value in an outcome, in this case that $100 is better than $0 for the choices are at odds with a single preference (red over black) either way).

Explanations as to why people strictly prefer A over B but then D over C usually focus on the fact that the probabilistic information available to the decision maker is incomplete and therefore people are exhibiting ambiguity aversion when they make the choices that they do (Gilboa et al. 2007). Put another way, people are “more averse to uncertainty than they are to regular risks of known proportions” (Hansson 1994). A way to elicit a ‘correct’ choice from a decider – one that maximises EV – is simply to explain the above to them and make sure they understand what their natural reaction to the situation is and why it may not lead them to the best outcome. If you want people not to violate the expected utility theory the questioner might discuss with the decider the ins and outs of probability. Another way to look at this problem is to value the ambiguity aversion more highly than material returns of the decision and then say that the deciders are actually making the ‘correct’ decision (i.e. one that most benefits them and does not violate expected utility) it is just that other factors have legitimate claims to holding utility for the decision maker: in this case benefit comes from psychological ease, not just from material gain.

For irrigation decision making, either the EV of a decision is established by calculation (for example an irrigation at such-and-such time with so-much water results in a soil moisture content of such-and-such) and then the EU left to the individual irrigator to assess (they would certainly factor in many decision results, not just water use
efficiency), or the EV is not easily quantifiable (for example should I irrigate tomorrow or go to the coast for a holiday) in which case the ‘correct’ decision is a business decision on behalf of the irrigator and their particular circumstances that no mathematical or other system can effectively ‘solve’. Further, ‘decision instability’ that the author understands to be of particular interest to philosophers (this occurs when the final outcome of a decision is dependent on the decision made) is not of much relevance to irrigation management decisions due to the well-known nature of most irrigation management decision outcomes: we are not dealing with ‘a’ or ‘b’ atomic outcomes in the irrigation sector, as in point making examples in philosophy, but rather with incremental improvements to decision outcomes. Essentially the philosophy of unstable decisions deals with distinct decisions in isolation and this is fundamentally not the case with irrigation management decisions. We may expect that an irrigation decision maker is already making irrigation management decisions that, at least to some degree, bear them utility of outcome. A person using no decision support may, when a decision is judged by qualitative measures, achieve a utility of outcome 99% of that of a decision made with decision support. Qualitatively, a person may achieve 100% of the supported decision utility while not using decision support, particularly if they follow a course of self-fulfilling prophesy whereby the decision they make is the correct one for them, regardless of outcome, as judged by others. Lessons for irrigation decision making, based on these points are given in the Section 1.2.4.

To generalize: the author believes that rather than testing the decision makers, if you want a ‘correct’ answer, you should be ‘on their side’ in the way you present them with the situation. You might not even need to generate theories as to why people choose what they do or, to use another paradox as an example, why people invariably choose the ‘wrong’ outcome in Newcomb’s Paradox. Potentially a questioner could invent any number of scenarios that ‘force’ people to act counter to their own interests. This is hardly surprising to the author and is a bit like asking “what would you do with a million dollars?” in that it is neither real nor likely. Certainly, this is the case with situations like Newcomb’s and ‘Death in Damascus’.

One further point on this subject: the author believes mathematics, including logic statements, to be a language with which it is often easier to describe situations than, say English and that graphs, a visualization of mathematical functions, are simply another ‘language’ used to describe something. Perhaps too much attention is given to various logic or other expressions of a problem when emphasis should be given to whatever language gets the point across easiest. If we want to know the trajectory of a cannonball for example, describing it as “going up from here and coming down over there” (English), is less accurate than “following a parabolic trajectory from here to a point 315m away” (English with a mathematical concept) which is still less accurate than height = 100sin(x), distance = {0...315} (see Figure 1 for a graph of this function).

However, this last expression, albeit very accurate, is useless to someone who doesn’t know anything about trigonometry. As a fundamental design concept, a DSS should use whatever language is most conducive to getting the point across and therefore resulting in the best decision outcome for the decider, whether that be probability, Bayesian statistics, colour graphs or English sentences.

For a final point in regard to this matter of appropriate and accurate language usage, the author notes that the logic philosopher Ludwig Wittgenstein believed that confused language use was the cause of most philosophical problems (Wikipedia 2019) and this view now carries much support since its publication in his book Philosophical
Investigations in 1953, a couple of decades after logic philosophers such as Bertrand Russell produced many paradoxes.

Figure 1: A graphical version of height = 100sin(x), distance = {0...315}

1.2.4. Outcomes from Decision Theory investigations

- The shift of emphasis away from a single outcome as the marker of utility is very important for irrigation decisions where the outcome of a decision may not just be measured in terms of a single parameter such as water use efficiency;
- A design principle that may be learned from decision theory is that it may simply be better for a questioner, or DSS designer, to present as much information as possible to the decision maker and leave estimations of expected utility to the decision maker. With this in mind, two types of DSS can be classified, they are: 1) directive and 2) facilitative. A 'directive' DSS leads the user to a decision whereas a 'facilitative' DSS just presents data, or information derived from data, to the user. The author acknowledges that it may not always be possible to pigeon-hole DSS in this way but the concept he feels to be useful;
- Decision support for any decision in which the decision maker already makes unassisted decisions that lead to outcomes of some utility, which is most decision making, will need to take into account the decision makers’ preferred outcomes, on an individual user basis if possible, and quantify and assign weight to those preferred outcomes, in order to optimise decision support utility. This suggests that decision support in an environment where unsupported decisions are already made would need to follow through an iterative approximation process where perhaps first best 'scientific' outcome were calculated, then an individual's preferences added and then best 'individual' outcomes were calculated;
- Decision paradoxes and seemingly erroneous decisions made by decision makers can probably be avoided if the questioner – DSS designer – is 'on the side' of the decision maker and therefore attempting to support them in making the best decision for them.
Note that some of this investigation into decision theory is reiterated in the introduction to the paper in Chapter 6 with more links to computer representations of decision making.

1.2.5. Decision Support Systems

1.2.5.1. DSS generally

Decision Support Systems (DSS), simply put, are systems that aid people in decision making. This thesis focussed on computer programs, as opposed to DSS presented in other media, such as printed material. This was due to the great rise in Internet-based systems in Australia at the start of this PhD work, see Section 1.3.1, and the desire, on behalf of the irrigation research community, to realise some of those benefits within irrigation decision making.

There is a very large international community of DSS researchers and designers with publications such as:

- Decision Support Systems\(^{12}\)
- International Journal of Decision Support System Technology\(^{13}\)
- International Journal of Decision Support Systems\(^{14}\)
- Journal of Decision Systems\(^{15}\)
- Journal of Soft Computing and Decision Support Systems\(^{16}\)

This plethora of academic attention indicates both that there is significant interest in DSS from society – the motivation for the pursuit – and that there is much work that can be done – the reason for the volume of work. Topics listed by these journals as being relevant to DSS in 2006, such as Data Mining, Fuzzy Systems and Neural Networks, Multi-Criteria Optimization and Knowledge Management Systems acted as starting points for some of the early investigations for this thesis.

1.2.5.2. DSS network paradigm

Initially, the way in which DSS connected to data and users, rather than techniques for processing and presenting data, were considered to be of prime importance to the irrigation sector’s use of DSS. Due to this, in 2006, an attempt was made by this author to classify examples of DSS according to a “network paradigm” (Car et al. 2007) (also Appendix D). This was due to the connectedness of DSS to data sources being seen to be the key determinant of a DSS’ capability to both handle many users over large areas – something desirable for tool affordability and for aggregated use benefits like benchmarking – while still remaining relevant to individual users – by being able to connect to local, or locally-relevant data sources to personalise the support given. Also, DSS, whatever data they connected to and whatever ways in which they processed it, needed to be maximally accessible to target users. This latter issue is addressed in

\(^{12}\)https://www.journals.elsevier.com/decision-support-systems/
\(^{13}\)https://www.igi-global.com/journal/international-journal-decision-support-system/1120
\(^{14}\)https://www.inderscience.com/ihome.php?jcode=ijdss
\(^{15}\)https://tandfonline.com/toc/tjds20/current
\(^{16}\)https://jscdss.com
Section **Error! Reference source not found.**. Table 1 lists the categories as originally conceived in Car *et al.* (2007) and as presented in a 2009 draft of this thesis’ Literature Review (see Appendix B).

**Table 1**: Author’s categories of DSS, based on the DSS’s networking function, as per Literature Review draft, 2009 (see Appendix B)

<table>
<thead>
<tr>
<th>Cat</th>
<th>Name</th>
<th>Description</th>
<th>Irrigation Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>none</td>
<td>standalone desktop application</td>
<td>Water balance spreadsheet</td>
</tr>
<tr>
<td>2</td>
<td>none, single link</td>
<td>desktop application that collects data from a single machine or sensor</td>
<td>G-lg probe</td>
</tr>
<tr>
<td>3</td>
<td>local computer network</td>
<td>desktop/intranet-based DSS with information only from intranet computers local network databases and local network sensors</td>
<td>a farm with an automatic weather station, soil moisture probes and historical weather database</td>
</tr>
<tr>
<td>4</td>
<td>private (enterprise) network</td>
<td>desktop/intranet-based DSS that uses intranet resources as well as non-intranet resources connected to the network via a proprietary network</td>
<td>a company farm with soil moisture probes, databases and a network of weather stations</td>
</tr>
<tr>
<td>5</td>
<td>internet</td>
<td>internet-based DSS that sources information from, and sends information to IP-based servers and clients</td>
<td>WaterSense (34)</td>
</tr>
<tr>
<td>6</td>
<td>2nd gen. internet</td>
<td>DSS that uses recent internet technologies and design concepts, such as Web 2.0 to allow DSS users for more flexibility than has previously been possible. This concept is fully detailed in Section 2.3.2</td>
<td>no examples exist</td>
</tr>
</tbody>
</table>

The establishment of this classification system was useful for this thesis in that it outlined a potential DSS future which helped formulate the research question. Also, the classification system was adopted in several places, such as undergraduate teachings on DSS (Dowling *et al.* 2016).

By considering where DSS retrieved data from and how they delivered it, it was speculated that users could define their own datasets for a DSS that was designed without knowing about them to use, data sources could be placed on the Internet in such a way that DSS could “mix ‘n match” between them and Web 2.0 techniques could be used to provide personalised views of DSS results.

The main conclusions reached by this author in 2007 from this network paradigm work was that systems then didn’t exhibit the characteristics that would lead to user-defined data sources, data source mixing ‘n matching and so on. For example, one conclusion was “The ability of a DSS to allow a user to choose which data sources to use, of all those available to the DSS, at it’s design time, has not been exhibited in any DSS in the literature but this functionality may be able to be included in future DSS designs in a way similar to a user turning on or off features on a service provider website” (Appendix B, page 25). It is clear, in retrospect, that much of the difficulty in affording DSS these possibilities is not directly related to its network connectivity: while that is a necessity – you can’t mix ‘n match remote data sources if you can’t connect to them – it’s in connectivity design that the majority of engineering or scientific effort is needed. Rather, the semantic understanding of and thus assessing the commensurability of data sources are now understood to be the key issues of enabling the data source-related
goals of the Car et al. (2007) paper. The goals related to DSS visualisation would now be termed User Experience (UX) design issued. Some of these are addressed in Car et al. (2008).

Some other conclusions from this work led to whole areas of investigation for this thesis, such as “DSS design must be wary of the lessons learned from decision theory paradoxes and attempt to be ‘intuitive’ for decision makers to use. This will probably be a very hard goal to accomplish based on the author’s viewing of unintuitive DSS front ends” (Appendix B, page 25). For DSS/Decision Theory-based thought, see Section 1.2.2.

1.2.5.3. DSS back-ends

As indicated in the previous section’s “network paradigms”, DSS in 2007 connected to other systems in many different ways and thus range in the data sources that they use considerably. Some of those data sources were proprietary real-time sensor systems (Vafaei & Cecere 2004), non-real-time satellite imagery (Barros 2005), web pages that can be scraped for text (CSIRO 2007), local or remote weather stations (Ligetvari 2005), user input values and GPS coordinates in quasi real-time (Hornbuckle et al. 2005) and so on. Around 2007, there was a surge in the agricultural and environmental sectors in the availability of wireless sensor networks\(^{17}\) to act as data source providers with one such contemporary example being the NICTOR system (National ITC Centre 2007).

Industrial process and operations DSSs, one of the oldest and probably the most commonly seen type of large-scale DSSs, tend to have biophysical sensors connected to private SCADA or other networks that transmit data back to a central location. One large-scale example of this is that which is used to monitor real-time power line loading by the power company EnergyAustralia in their Sydney control room. Another example of a system that uses data sources in this way, also from a private network and also in real-time, is that which is used to assist in traffic congestion easing given in (Hegyi et al. 2001). Unlike the EnergyAustralia system, the traffic control system also uses historical information stored in databases, as well as current traffic situation measurements taken from its sensor network to provide decision support. This combination of current data and historical data is the norm for DSS dealing with complex situations where a single course of action, such as reducing the load on a power cable by single instance rerouting, is not usually possible.

Many papers available in 2007 detailed DSS that dealt, in real-time, with flood forecasting and warnings ([Gunderson & Krzysztofowicz 1999], (Quinn & Hanna 2003) and (Shao-zhong et al. 2002)). These systems all used networked biophysical sensors to collect information at a single point for quick decision making. These sorts of DSS are

\(^{17}\) From the author’s own organisation, CSIRO, a multitude of publications around the use of wireless sensors in networks appears around 2007, such as


usually expensive because of the cost of extensive sensor and networking hardware. If the DSS is to operate in real-time, then issues such as responses to peak information loads and the reliability of data acquisition greatly increase the costs. The CORMS-IA DSS (Vafaie et al. 2004) and the Water Resources Observation Network (WRON) (CSIRO 2007) were then multi-million-dollar projects partially because of these factors.

Since perhaps 2009 (see Section 1.3.1 for discussion of dates like this) the major way in which data transfer over the Internet has occurred, with relevance to Australian agriculture, has been via Application Programming Interfaces (APIs). With APIs, the data provisioning system makes available machine-readable functions that can be called remotely to provide data. This is opposed to web page scraping and custom, non-Internet, network data transfer methods. With this concept of an API, and with the pervasiveness of the Internet and fast connections to many DSS accessing data now do so via Internet-based APIs, even when the data sources are established by the DSS designers. This is the case for IrriSatSM & IrriSat, described in Chapters 2 – 4 and the use of Internet APIs has reduced the cost of DSS that draw from large numbers of remote data sources, like CORMS-AI did. Recent papers such as Sommer et al. (2018) indicate that DSS designers are able to source data from in-situ biophysical sensors, such as weather stations and soil moisture probes, via Internet-based APIs.

Part of this outcome was foreseen by this author in 2009 with a conclusion then that “For physical connectivity, TCP/IP networks are becoming all pervasive. This suggests future DSS back-ends should use TCP/IP” (see Appendix B, page 47). However, as per thoughts on the utility of classifying DSS by “network paradigm” (provisos section), Sommer et al. (2018) indicates that the work to make data from disparate sources intelligible and to ensure commensurability of information far outweights efforts to connect to them. This effort gap has grown due to Internet APIs making connectivity easier.

DSS that utilise data from one or a few local biophysical sensors are at the opposite end of the “network paradigms” classification used in the previous section and exist in contrast in complexity and cost to those large systems mentioned above. An example of such a system is the irriMAX computer desktop display program for data from SENTEK's EnviroScan probes (SENTEK 2007). Such a system is used over a small, private network to assist an irrigation manager decide when the crop needs irrigating, based on the soil moisture probe readings alone. This sort of system still costs the individual user a reasonable sum as the probes are often not cheap. Previously too, before pervasive Internet connectivity and mobile phone coverage (see Section 1.3.1) networking of probes to the DSS was expensive also. In the case of the SENTEK system mentioned above, the front end system for displaying probe data (not quite a DSS, just a visualisation interface) is supplied with the probes for free, due to the vendor's profits being derived from probe sales.

DSS that use only databases or user entry data for their data sources were once very common but decreasingly so due to Internet pervasiveness. Examples of this sort of database are common and universally used, if one considers that a simple spreadsheet computer program, such as Excel, is a type of DSS. A more standard DSS example of a user input-only system is WaterTrack™, which is a farm-level water balance calculator. It models water distribution, inflow, outflow and storage on a farm (Scolari Software

2005). It is a stand-alone desktop application. This is an example of a modern category 1 DSS, as per the previous section’s classifications. A desktop program, similar to WaterTrack in terms of data input but with additional data collection features, is MaizeMan (White et al. 2002). This program:

“works by allowing the user to input information for each paddock, starting with the conditions the crop will be grown under its location, weather, rainfall and soil type … The user inputs information about planting conditions planting time, seed type and variety, row spacing, rate, sowing depth and soil conditions (CSIRO ICT Centre 2004)”

This program allows the user to import variable weather data, in quasi real-time, from certain organisation’s web pages that present information from those organisation’s private weather stations and weather information service pages that are publicly available or available for public purchase. This is an example of a category 5 DSS in that it utilises the internet to access information from more than one 3rd party private network. In the case of MaizeMan, the private weather station information is from the CSIRO’s Land and Water website (http://www.clw.csiro.au/services/weather/) and the publicly purchased information from SILO (Qld Govt. 2007).

Access to data, other than weather data, over the internet is also becoming increasingly available. Non-biophysical data, such as economic data like water or livestock prices are increasingly being placed under Internet access and websites such as http://www.watertradingaustralia.com (now https://www.ruralcowater.com.au) started listing water prices for irrigation areas in Australia from about 2005 onwards.

Data Storage

Originally the author intended to devote a large section of his Literature Review about DSSs back ends to data storage but then felt that the topic needed only to be addressed briefly. The reasons for this, as given in 2009 (see Appendix B, page 29) were that:

- virtually all computerised data are currently stored in relational databases. This leaves little room for discussion with regard to how data is logically stored;
- the concepts of relational databases are widely known and need not be discussed here;
- virtually all computerised data are currently physically stored on computer hard drives, leaving little need for discussion;
- the exact ‘where’ and ‘how’ of data storage is unimportant when looking at a DSS in an other-than computer engineering mode;
- the systems likely to be used by the author are unlikely to be radically different from those used by most other IT users due to the high cost and availability of unusual systems.

Following on from those reasons, this author reached the conclusion that “Data storage is not thought to present many challenges or opportunities for research for irrigation DSS”. Experience since then has shown that, for DSS designers, data storage concerns indeed aren’t a major issue or focus of design however the reasons now are not entirely those given above. It is not true, and wasn’t when stated in 2009 as listed above, that “virtually all computerised data are currently stored in relational databases”; much imagery and other forms of data are stored as files or within non-relational data cubes. Also, “virtually all computerised data are currently physically stored on computer hard drives...” may be true in one sense – they are – but not in the sense this author intended
in 2009. Much data now is stored remotely to users in ‘clouds’ (Internet storage and compute platforms located in data centres).

A recent irrigation DSS that accesses large stores of data, is IrriSat (Vleeshouwer et al. 2015) (and Chapter 4) which uses 10+ years’ data from the entire historical LANDSAT satellite data archive collected within the Google Earth engine Internet platform19 and 10+ years’ worth of weatherstation weather data from a CSIRO network. That DSS has not experienced difficulties due to data storage since both the data it accesses and that which it generated are stored on very large public clouds.

Despite issues with the reasons listed above, the overall conclusion that “Data storage is not thought to present many challenges or opportunities for research for irrigation DSS” is still thought to be true.

1.2.5.4. DSS front-ends

User interface (UI) or front-end subsystems are an essential part of DSS design as all DSS must eventually present their calculations or predictions to a human user to assist with their decision making. This section addresses front-end subsystems of DSS, or those systems that interface DSS with users. This section will consider the existing and possible future ranges of physical interface mediums, problems and techniques for presenting information from a DSS to users, ways in which users may feed information back to a DSS as well as interfaces for different user groups. The ability of a future DSS to be used by multiple user groups exhibiting multiple use cases and to use different delivery mechanisms was mentioned in the “network paradigm” work from Car (2007) described in Section 1.2.5.2 but little detail was given. The interface components of this will be explored here.

Interface theory has necessarily been a part of computing theory since computing theory’s inception in the 1940s due to the fact that computers must eventually interface with humans if they are to be used. Early computing systems used punch cards, paper printouts, lights, segmented displays, command line consoles (such as Microsoft DOS and Unix) and text-based graphics as some of the more common interface types. Since the advent of the VGA personal computer monitor in the 1980s, graphical user interface (GUI) systems have been used extensively to the point where most computer users only use GUIs like Microsoft Windows. Initially with the advent of mobile phones, there was a re-visitation of text-based, console, simple GUI (such as menu-driven interfaces) and other interface types due to the limited computing capacity of early mobile phones and their capacity to physically display many pixels and colours. A 2007 search of the cheapest mobile phones, such as the Nokia 3330, found them to contain a basic, menu-driven, GUI. More recently (2009+ in Australia, see Section 1.3.1), mobile phones have exhibited high-resolution colour screens and their GUIs now mirror computer display.

Many computer-based DSS use graphical means to convey information. Custom software in the DSS sector can be generated via software packages, used for calculation that includes graphical packages. Such an interface for fisheries management is described in Truong et al. (2005). It uses the program MATLAB’s graphical modules to produce spatial maps and graphs. In this particular example, very little detail of the GUI mechanism or its construction process were given which suggests that the authors felt

it was not a significant part of the overall DSS. The images supplied in the paper show an interface that only someone familiar with the system could use as it is quite complex. This is also the case in Quinn & Hanna (2003) which supplies the DSS user with numerous graphs tracking wetland salt loads that only have meaning for skilled staff (environmental scientists) and require some specific training to interpret effectively.

In a paper entitled “Interface for displaying probabilistic river forecasts to decision makers” (Gunderson & Krzysztofowicz 1999) the authors present a custom-built interface to display river height data which highlights, with alarm boxes, rivers that have reached certain levels chosen by the users. The various graphs and maps for this interface were presented to the user via a web browser but were generated by specialised software, the details of which were not given. As the ‘interface’ is the main work of the paper, more detail is given of it than in the previous paper mentioned but still little analysis of the design of the individual graphs is given. The author believes that this is a result of the designers of the interface believing that the order of the graphs and which graphs to show when, which is covered in the paper, is the only task of interest to the interface designer as the individual graphs are of a type that is well understood by the target user groups of the system. This final point in the thinking, that the graphs will be understood by the target audience, must be a result of the DSS being either designed for an already trained audience, or one that will be trained, or that the concepts to display are simple enough to be understood by anyone.

Since this original DSS front-end research, much effort has been put by IT developers into web-based graphing of information, as evidenced by the now plethora of technical tools available for creating charts in Internet browsers20 however, still proportionally little effort is put into explaining GUI elements to users: perhaps it is taken for granted that system users are familiar with computer-based display elements.

In 2009, this author asked whether there may be value in research looking at users’ understanding of individual graphs? This would be separate to looking at which factors in a multi-factor decision making environment would be most useful to display graphically. DSS outputs that, in the past were displayed as numbers, basic graphs and simple tables may be better displayed in a way that is more intuitive for untrained individuals to. This could potentially avoid encountering non-rational decisions that paradoxes in decision theory bring out due to decision makers misinterpreting statistics and probabilities or by placing undue emphasis on certain parts of basic front-ends.

An irrigation DSS project named irriGATE was proposed in 2006 by members of the CRC-IF. It attempted to assist irrigators answering question “when to irrigate, based on evapotranspiration?” with interfaces delivering graphs similar to that shown in Figure 2. One part of the proposal was focussed on attempts to communicate uncertainty in DSS advice, as evidenced by the green probability bounds of future ET0 as shown in Figure 2. Aspects of this attempt to communicate uncertainly were incorporated into future DSS such as IrriSat (Vleeshouwer 2015).

The author believes the display element in Figure 2 does not lend itself to ease of use due to its complexity. The graph is easily understood by agricultural scientists but perhaps not by those without scientific or mathematical training or domain expertise in

---

20 See the many articles listing JavaScript charting/graphing libraries, such as "Compare the Best Javascript Chart Libraries" (https://blog.sicara.com/compare-best-javascript-chart-libraries-2017-89be8cb112d) & "24 JavaScript Libraries for Creating Beautiful Charts" (https://www.sitepoint.com/15-best-javascript-charting-libraries/) etc.
irrigation. If the target users of a DSS that employed graphs such as these were able to be trained, then they may be able to gain considerable benefit from such graphs but if not, then perhaps the graph would be bewildering.

Figure 2: The irriGATE project’s evapotranspiration graph

Another example of irrigation DSS graphing is given in Figure 3. This image, from the APSIM-based sugar cane front end WaterSense (Inman-Bamber & Attard 2007) shows soil water deficit at different depths on the same timeline as irrigation events, ET0, rain and tolerance boundaries set by the irrigator for his or her reference. This graph has a zero level at its centre line with positive water events (rain and irrigations) above the line and negative events (crop water use) below.

Figure 3: WaterSense’s more intuitive evapotranspiration graph from (Inman-Bamber & Attard 2007)

The author believes that the interface in Figure 3 is more intuitive (easier to use by untrained people), despite having more display options, than that given in Figure 2 due to its positive/ negative treatment of water balance, rather than a cumulative graph just referenced to ET0. However, the author feels that Figure 3 gives a scientific-, rather than irrigator-focussed perspective on crop water balance and therefore may also not be as easily used as it could be. The authors of WaterSense state that it is a research tool, rather than a marketable product, which might explain design choices in this instance.

The author believes that the example of a user’s interface page given in Figure 3 represented the state-of-the-art in irrigation information and data visualisation as of about 2007. Then, the author could not find papers that specifically researched the usability of DSS front-ends and information presentation or the cognitive aspects of information interpretation in the agricultural or irrigation fields. The author believed that little work had then been done in utilising the large amount of research in the cognitive informatics area for irrigation DSS. Cognitive informatics is a part of the informatics research in a number of organisations including the University of
Edinburgh's School of Informatics, the Institute for Computing, Information, and Cognitive Systems at the University of British Columbia and many others.

The author posed the following questions: What improvements to irrigation DSS graphs and front-ends generally can be made that will yield increased ease of use by untrained persons? What fields of study need to be considered to help with this task and will increased ease of use and understanding make a significant difference to DSS uptake and DSS results? How are these concepts tested?

The author's partial answers to these questions in 2007 – 2009 was that there was then significant work emerging in the field of technical information presentation for both desktop and Internet front-ends by companies such as Microsoft that had not been used in irrigation DSS and it could be. One example of this was the Windows Presentation Foundation, initially released in 2006\footnote{https://en.wikipedia.org/wiki/Windows_Presentation_Foundation}, which was a graphical UI generation package starting to be then built into computers running the Windows operating system. It allowed for 3D graphing and will potentially enable data to be displayed in novel ways.

In 2009, the author also believed that there may have been work in the fields such as cybernetics, cognition and psychology that could have been of use in this area, but these were currently both outside his experience to that point and beyond the scope of this Literature Review. Certainly, no reference to these areas of study was made in irrigation DSS literature to that point.

With growth in the use of web browsers as GUIs for all kinds of systems due to increased Internet access and growth in SmartPhone use among Australian irrigators (see Section 1.3.1), the ability for irrigation DSS designers to leverage powerful display technologies increased rapidly during the first few years of this PhD. With this new capacity in mind, the author put much effort into relating the drastic paradigm shift in Internet technologies generally known as Web 2.0 and the possibilities it brought to irrigation DSS. This is related fully, as originally presented in 2009, in Appendix B, page 50 – 52. Following from this, the author investigated mobile phone-based UIs in depth too, as related in Appendix B, pages 54 – 59. This latter work directly informed the choice of experiments undertaken in this PhD (see Chapter 2).

1.2.5.5. DSS for irrigation in Australia

The adoption of scientific irrigation scheduling tools in Australia in 2006 (at the start of this PhD) was relatively low (Montagu et al. 2006) despite the relatively advanced scientific state of Australian farming. At least some of the barriers to the increased use of scientific irrigation scheduling tools were related to perceived notions of the use of such tools being overly complex or requiring too much of a learning curve to begin with, for example WaterSense (SRDC 2005) which was a sophisticated web interface for an APSIM-based sugar cane model to assist with irrigation scheduling which allowed irrigators to use the APSIM-based model by simply entering in certain variables into web pages and receiving graphical and textual outputs. The utility of such a system, in terms of water use efficiency (WUE) was proved in a number of papers (e.g. Inman-Bamber & Attard 2007) and yet uptake remained low.

A review of DSS and other software built for irrigation in Australia was undertaken to examine aspects of the tools and their projects, such as uptake. The majority of the tools
listed here were taken directly from a technical report entitled “Inventory of Australian Software Tools for On Farm water Management” (Inman-Bamber & Attard 2005). Tools not listed in the report but also considered here were:

- **APSIM**
- **FIDO v.2** (McClymont 1999)
- **Pocket SIRMOD** (Hornbuckle et al. 2005)
- **WRON** (CSIRO 2007)

Tool aspects considered were derived from the previous four sections’ word and were:

- whether or not the software caters specifically for irrigators, as opposed to consultants or researchers
- whether or not the software caters for different use cases (or more generally, different levels of use)
- whether or not it had a basic, cut down version
- whether or not the software caters for benchmarking
- whether or not it is currently still in use
- whether or not it used Case-Based Reasoning
  - no tools were found to use CBR therefore this point is not tabulated²³

<table>
<thead>
<tr>
<th>Name</th>
<th>for irrig.</th>
<th>use cases</th>
<th>basic</th>
<th>bench</th>
<th>in use</th>
</tr>
</thead>
<tbody>
<tr>
<td>APSIM</td>
<td>y</td>
<td>n</td>
<td>n</td>
<td>n</td>
<td>y</td>
</tr>
<tr>
<td>DamEas$y</td>
<td>y</td>
<td>n</td>
<td>n</td>
<td>n</td>
<td>n</td>
</tr>
<tr>
<td>FIDO v.2</td>
<td>y</td>
<td>n</td>
<td>n</td>
<td>n</td>
<td>n</td>
</tr>
<tr>
<td>FlowCast</td>
<td>n</td>
<td>n</td>
<td>n</td>
<td>n</td>
<td>n</td>
</tr>
<tr>
<td>HydroLOGIC</td>
<td>y</td>
<td>n</td>
<td>n</td>
<td>n</td>
<td>&lt;240</td>
</tr>
<tr>
<td>IRES</td>
<td>y</td>
<td>n</td>
<td>n</td>
<td>n</td>
<td>trial</td>
</tr>
<tr>
<td>IrriMAX</td>
<td>n</td>
<td>n</td>
<td>n/a</td>
<td>n</td>
<td>y</td>
</tr>
<tr>
<td>Magpie</td>
<td>n</td>
<td>n</td>
<td>n/a</td>
<td>n</td>
<td>&lt;350</td>
</tr>
<tr>
<td>MaizeMan</td>
<td>y</td>
<td>n</td>
<td>n</td>
<td>n</td>
<td>v.small</td>
</tr>
<tr>
<td>Micronet</td>
<td>n</td>
<td>n</td>
<td>n</td>
<td>n</td>
<td>some</td>
</tr>
<tr>
<td>Multi-Log</td>
<td>y</td>
<td>n</td>
<td>n</td>
<td>n</td>
<td>y</td>
</tr>
<tr>
<td>Probe</td>
<td>y</td>
<td>n</td>
<td>n</td>
<td>y</td>
<td>y</td>
</tr>
</tbody>
</table>

Table 2 & Table 3 list the DSS investigated and the four assessment points. Results from the review are as follows:

1. there are many software packages that cater specifically for irrigators and irrigation scheduling
2. no software caters for multiple use cases

---

²² [http://www.apsim.info](http://www.apsim.info)

²³ This aspect may seem out of context however it was investigated to see if any then current irrigation DSSs in Australia were using that AI paradigm due to the author’s interest in investigating such systems to provide empirical decision support advice, see Car & Moore (2011) as listed in the Preface of this PhD.
3. only the WRON 'introductory' software caters for very basic use that does not require some training or some sort of understanding process
4. only one package, the 'Probe for Windows', tabled as 'Probe', reports to have benchmarking applications currently under way

<table>
<thead>
<tr>
<th>Name</th>
<th>for irrig.</th>
<th>use cases</th>
<th>basic</th>
<th>bench</th>
<th>in use</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pocket SIRMOD</td>
<td>y</td>
<td>n</td>
<td>n</td>
<td>n</td>
<td>n</td>
</tr>
<tr>
<td>SWAGMAN</td>
<td>n</td>
<td>n</td>
<td>n</td>
<td>n</td>
<td>small</td>
</tr>
<tr>
<td>VineLOGIC</td>
<td>y</td>
<td>n</td>
<td>n</td>
<td>n</td>
<td>100</td>
</tr>
<tr>
<td>WaterBalance</td>
<td>y</td>
<td>n</td>
<td>n</td>
<td>n</td>
<td>y</td>
</tr>
<tr>
<td>WaterSense</td>
<td>y</td>
<td>n</td>
<td>n</td>
<td>n</td>
<td>small</td>
</tr>
<tr>
<td>WaterSupply</td>
<td>n</td>
<td>n</td>
<td>n</td>
<td>n</td>
<td>y</td>
</tr>
<tr>
<td>WaterTrack</td>
<td>y</td>
<td>n</td>
<td>n</td>
<td>n</td>
<td>y</td>
</tr>
<tr>
<td>WeatherFace</td>
<td>y</td>
<td>n</td>
<td>n</td>
<td>n</td>
<td>n</td>
</tr>
<tr>
<td>WRON</td>
<td>n</td>
<td>n</td>
<td>y</td>
<td>n</td>
<td>n</td>
</tr>
</tbody>
</table>

The impression one received from the results reported above is that there were a plethora of irrigation DSS and modelling software available in Australia in 2009 but few were for irrigators to use for scheduling in an other-than research fashion. None of the systems really allowed for a spectrum of usage levels, apart from those that just present information. In addition to this, none report multiple front-end delivery formats; all are either desktop or web browser based with just one example of a PocketPC backed application.

1.2.6. Conclusions

The key hypothesis upon which this literature review was based is that:

Informatics science can be applied to irrigation management to deliver better water use efficiency and productivity

This hypothesis is as originally posed by the PhD initialisation processes which also provided some additional research domain guidance:

- software models and most importantly integrated software engines tying together other technologies ... [could] ... provide a powerful irrigation water management tool for informing and predicting irrigation behaviour and radically increasing water use efficiency from paddock to region scales

This review of literature shows that there are large gaps in knowledge within current irrigation DSS research in Australia relating to:

1. Use of decision theory DSS
   - The sophisticated use of previous decisions in advising current decision making
2. DSS that allow users to customise the support they receive
DSS back-ends that are flexible and thus adaptable to multiple data source usage

How data, other than biophysical soil-plant-water data, may be captured and used to assist in decision support

3. Different physical and conceptual DSS human interfaces, especially those allowing for use outside of offices

Conclusions 3 is a variant of Conclusion 2. Currently irrigators cannot have input into DSS, either about what data is fed into the system to generate that support or where and when they receive support. Improving the design of current DSS by allowing users to be able to have such input would require not only the development of tools that would allow such customisation to take place but also, if it is to be broad reaching, the development of a new DSS conceptual framework that is broader in scope than those commonly seen today that are decision scenario-specific.

To allow individuals to include different non-biophysical data in decision support would be the ultimate goal of a flexible and interactive DSS. Rendering decision support using data other than classical biophysical soil-plant-water data, could partly be achieved in a piecemeal fashion with the addition of individual rules concerning specific non-biophysical data (such as factoring in water price versus the cost of not watering in terms of reduced yield) but this approach would not allow for individual customisation unless a comprehensive set of specific non-biophysical data rules were created that quantified every non-biophysical data set that one could suppose an irrigator might wish to employ in decision making. Such a set would not only be vast but may never be comprehensive as surely the non-biophysical data which irrigators include in their decision making varies with time thus catering for it may be unachievable with current DSS structures and systems which are always designed for specific scenarios.

1.3. Literature Review extensions

Earlier work done in this PhD, including the Literature Review in Section 1.2 above, had the perspective of irrigation & DSS advances until about 2009. Due to the length of time taken for the completion of this PhD, further perspectives on this thesis’ topic were gathered and are now reported.

1.3.1. Tech changes in Australian irrigation, 2007+

In 2007, Australian irrigation farmers experienced the start of a series of technology revolutions, most noticeably the uptake of mobile phones and farm home and small business Internet connectivity. These technology revolutions resulted in an information revolution for Australian farmers that forever altered their capability to easily connect to many things that could assist them with objective decision making. These included electronic irrigation equipment like pumps, sources of remote data such as satellite imagery, computational power in the form of online servers and expert systems at research agencies and service providers’ offices.
1.3.1.1. Mobile phones and irrigation

The first of these revolutions, uptake of mobile phones, gave irrigators tools that, while not as capable as the SmartPhones of today with their high-resolution displays and high-bandwidth connectivity, nevertheless gave irrigators the transformative ability to be able to send instant electronic messages (SMS: Short Messaging Service messages) from a large proportion of Australian farms to other people and also systems at any time of convenience for them. This allowed them to report irrigation decisions to centralised systems as they made them, such as how much they decided to water while in the pump shed, rather than recording them for later transmission. It also allowed them to receive messages anywhere and at any time, rather than in dedicated computer use sessions.

The conference paper (Car 2007), also Appendix A, outlined work undertaken by the author to implement an irrigation Decision Support System (DSS) which used SMS as a user interface. Chapter 2 is the paper Using a mobile phone Short Messaging Service (SMS) for irrigation scheduling in Australia – Farmers’ participation and utility evaluation (Car et al. 2012) published in 2012 which relates several season’s field trials of the IrriSatSMS system which was outlined (Car 2007). This latter paper has been cited 19 times since its publication indicating that the work has been of continued interest to academia.

In addition to its academic interest the system and demonstrated utility to the trial users related in the paper, IrriSatSMS, was the subject of an Australian irrigation industry award (Irrigation Australia Limited’s Best New Irrigation Product, 2010) and its scalability and low cost of the suggested that it could be of interest to industry as not just a research system but also as a product. As a result, some efforts were made to communicate the general approach of the system’s development through workshops with potential commercialising partners and also through the publication of technical reports (Hornbuckle et al. 2009).

1.3.1.2. The Internet and irrigation

During the testing of IrriSatSMS, and shortly after it, the proportion of mobile phones in Australia that were SmartPhones grew enormously, as did the proportion of rural households with Internet connectivity, as indicated in Figure 4.

![Figure 4: Proportion of Australian mobile phones that were SmartPhones (left) after (Mackay 2014). Proportion of Australian rural households with internet access (right)](image-url)

To capitalise on these trends with respect to irrigation decision support, an updated versions of the IrriSatSMS back-end system with new front-end delivery (not SMS but instead Smartphone and web-based applications) was developed (Montgomery et al. 2015; Vleeshouwer, Car, and Hornbuckle 2015). This system, just called IrriSat, is still in experimental use by Deakin University researchers, as of December 2017. Details of this system are related in Chapters 3 & 4.

1.3.1.3. DSS Connectivity

With the growth in both mobile phone and internet availability for irrigators, Decision Support System (DSS) builders were able to implement not just new user interfaces for their irrigation DSS but also new data sources. In (Car et al. 2007), also Section 1.2.5.2, we outlined a series of DSS categories that used the method of their connectivity to data sources or computation as the classifier. This classification has been used in a series of university engineering textbooks since 2009 and most recently in 2016 (Dowling et al. 2016) as a sensible way of contextualising DSS with respect to networking technologies and the data they may use in their decision support as a result of types of connectivity.

In addition to DSS classification, (Car et al. 2007) also framed part of the investigations of this PhD by proposing requirements for a back-end architecture that allowed for changes in data sources and data source selection after DSS construction. This would mean that a DSS could be built and the particular data sources used by it chosen later.

The paper partly addressed the great difficulty of data source integration with an appeal for the use of the Open Geospatial Consortium’s Sensor Web Enablement set of protocols (Open Geospatial Consortium 2017) which was quite new in 2007. The paper did not reference the more broadly scoped set of Semantic Web technologies that, in 2017, look to be crucial to data source integration, given the range of possible data sources that a DSS might use, well beyond sensor data handled by SWE.

The paper suggested a need for, but proposed no mechanism to, make data sources discoverable. Given the inability of general web search engines to sensibly index data sources, other than by indexing descriptive web pages of their content as of 2017\(^{24}\), and the lack of a non-search engine tool or protocol to use, this problem remains unsolved. This indicates that there are a set of problems for DSS that are impervious to even tremendous increases in technology.

Chapter 5 relates some investigations into data brokering by this author.

\(^{24}\) In communication with Google Australia, 2015, a colleague of the author put the question to them: “why does Google not index standardised data sources such as Web Services delivered according to the Open Geospatial Consortium’s W*S set of standards (Web Map Service, Web Feature Service & Web Coverage Service) to which the answer was that they weren’t aware of any to index (in Australia). The interlocutor indicated that there were very many such services publicly available, but Google Australia ultimately seemed to have no interest in the specialised indexing of them that would be required to make a usable data index.
1.3.2. The Semantic Web

In 2009, it was not recognised by the author that the core problems associated with DSS’ use of disparate data sources were ones of data semantics: what was meant by individual data sources’ data. As a result, early work in this PhD was focussed on DSS structure (such as network connectivity) and data formats, however papers such as “Irrigation modelling language for decision support” (Car et al. (2009), listed in Preface) and “Towards standardising irrigation DSS inputs data formats through adaptation of the WDTF/WaterML” (Car & Moore (2011), also listed in Preface) did touch on some data semantics. The latter paper investigated the representations of evapotranspiration needed to ensure commensurability, given the different methods for its calculation.

From about 2011 onwards, the author became acquainted with the concept of the Semantic Web (Berners-Lee et al. 2001) and, recognising that its promise of making an Internet of machine-readable data would greatly assist irrigation DSS designers achieve data source flexibility, he oriented later research towards Semantic Web ideas.

The core principle that was taken to be relevant to irrigation DSS from the Semantic Web was that:

The Semantic Web provides a common framework that allows data to be shared and reused across application, enterprise, and community boundaries.  

The relevance is that if communities with information relevant to irrigation – national authorities, water supply companies, irrigator community groups, equipment manufacturers etc. – could adopt Semantic Web principles, there would be a greater capacity for DSS to access that information and become more useful for decision makers.

The Semantic Web framework now extends beyond data structures such as the Resource Description Framework (RDF) and data access mechanics such as Linked Data to general-purpose conceptual modelling languages such as the Web Ontology Language (OWL) (W3C OWL Working Group 2012). This set of technologies allows different communities to present and share their data but also to model it using specialised domain models – ontologies – that make sense to them. If constructed well, these ontologies also make sense to others.

Abstracted ontologies have been created to allow communities making specialised domain ontologies to do so with reference to a common set of concepts that then allow connection points between multiple domain ontologies to be made and thus information represented according to them to be interoperable. One such abstracted ontology for “all things” (the authors’ assertion) is the Basic Formal Ontology (BFO) (Arp et al 2016) which separates objects modelled according to Aristotelian logic. Another, less general, model is the Provenance Data Model (PROV-DM) (Moreau & Missier 2013) which is narrower in scope – a process model – but which nevertheless provides an abstracted model for many domain models’ integration due to its general concepts of things and events.

26 https://www.w3.org/RDF/  
27 https://www.w3.org/standards/semanticweb/data
After learning about Semantic Web concepts after about 2009, the author endeavoured to incorporate some of their principles into the scope of this thesis. The major work following from this is Chapter 6, the modelling of decision making.

### 1.3.3. Possibilities for representing decisions

With early PhD interest in decision theory, investigations into representing decision making were considered from perhaps 2007 onwards. Initially the author had no specific direction for the technical representation of decisions so the philosophical/logic representations of decision making in general were explored (see Section 1.2). After encountering the concept of the Semantic Web, its conceptual mechanisms were explored.

A fulsome investigation into different ways of representing decision making in computer systems is given in Chapter 6. In that chapter, both Semantic Web and non-Semantic Web systems are considered.

### 1.4. Research Question

Considerable change to the research question and topics explored for this thesis have occurred since its initiation in 2007. Appendix A outlines in detail these changes, and Appendix B contains a complete thesis Literature Review written in 2009 which contains an early version of this listing of research questions, not all of which were pursued. See Appendix A for the transition from those questions to these below.

The final research question for this PhD is:

> Can decision support systems with new approaches to irrigator interaction, external systems connectivity, and new mechanisms for representing decision making, enhance irrigation decision support system uptake?

The research topics investigated to test this question within this PhD were:

1. Better Decision Theory Use
2. Customisation (of DSS)
3. New DSS Tools
4. Empirical DSS
5. DSS Adoption Analysis

Appendix A gives a description, by topic, as to why it was chosen and how it evolved from earlier topic selection.

The next section outlines the mapping of these research topics to sections of this thesis.

### 1.5. Thesis outline

This thesis is described in two ways: firstly, by parts and secondly with reference to the research topics listed in the previous section.
1.5.1. Thesis parts

Chapter 1 is this Introduction which includes a Research Scope, Literature Review, Research Question and Thesis Outline.

Chapter 2 presents the journal article “Using a mobile phone Short Messaging Service (SMS) for irrigation scheduling in Australia – Farmers’ participation and utility evaluation” (Car et al. 2012) which summarises work on SMS-based irrigation DSS built and tested between 2008 and 2012.

Chapter 3 presents accounts of extensions to the IrriSatSMS tool that were made to assist the irrigation of crops other than grapes (as per IrriSatSMS) and in other regions (beyond the Murrumbidgee Irrigation Area).

Chapter 4 presents the journal article “Tracking the value of an innovation through the new product development process: The IrriSat family of agricultural decision support system tools” (N.J. Car 2017a) which reflects on the attempted commercialisation of IrriSatSMS and relates the design of its successor system to reduce the chances of future repeated commercialisation failure. The paper also describes the changing nature of the use of technology relevant to irrigation DSS over the period 2007 - 2015.

Chapter 5 presents some experiments, starting in 2008 and continuing until the present, in irrigation DSS data source integration. This chapter includes references to conference papers that relate work done, and reflections on them in 2017. These pave the way for Chapter 6.

Chapter 6 presents the journal paper “Modelling decisions and using decision models to enable better irrigation Decision Support Systems” (Car 2018), the follow-on conference paper “Modelling causes for actions with the Decision and PROV ontologies” (N.J. Car 2017b) and lists online resources. These review existing decision modelling systems for use with irrigation DSS, produce a new decision modelling system informed by the review and give access to some other resources also developed respectively.

Chapter 7 is a Discussion presenting a brief description of the total irrigation informatics story told by this thesis.

Chapter 8 is this PhD’s Conclusion. It sums up major lessons learned from this more than a decade long, multi-experiment, investigation into irrigation informatics.

Appendixes A – H

(A) notes on the progression of this thesis’ research question
(B) an early draft of a literature review
(C) the conference paper “Tools for Improving Water Use Efficiency: Irrigation Informatics implemented via SMS” (Car et al. 2007b)
(D) conference paper “Towards a new generation of Irrigation Decision Support Systems – Irrigation Informatics?” (Car et al. 2007)
(E) extensions of the IrriSatSMS system in South East Queensland
(F) experiments involving crop factor generation from mobile phone imagery
(G) current decision modelling work presented at a provenance conference
(H) source code for example decisions modelled
1.5.2. Mapping research topics to sections

Table 4 is a mapping between research topics from Section 1.4 and thesis sections.

<table>
<thead>
<tr>
<th>Research Topic</th>
<th>Relevant Thesis Sections</th>
</tr>
</thead>
<tbody>
<tr>
<td>Better Decision Theory Use</td>
<td>Chapter 6</td>
</tr>
<tr>
<td>Customisation (of DSS)</td>
<td>Chapters 2, 3 &amp; 5</td>
</tr>
<tr>
<td>New DSS Tools</td>
<td>Chapters 2 &amp; 3</td>
</tr>
<tr>
<td>Empirical DSS</td>
<td>Chapter 7</td>
</tr>
<tr>
<td>DSS Adoption Analysis</td>
<td>Chapter 4</td>
</tr>
</tbody>
</table>

Each topic has conclusions presented in the sections relevant to it.

1.6. References


..., 2011 Household Use of Information Technology, Australia, 2010-11.


2. IrriSatSMS

This chapter is the journal article Using a mobile phone Short Messaging Service (SMS) for irrigation scheduling in Australia – Farmers’ participation and utility evaluation represented as published in the following pages.

The citation for this paper is:

Nicholas J. Car, Evan W. Christen, John W. Hornbuckle, Graham A. Moore, Using a mobile phone Short Messaging Service (SMS) for irrigation scheduling in Australia – Farmers’ participation and utility evaluation. In Computers and Electronics in Agriculture, June 2012, 84, 132-143, ISSN 0168-1699, DOI: 10.1016/j.compag.2012.03.003

This paper reports the work conducted in 2008 – 2011 to build a decisions support system for irrigators that used new forms of advice generation and communication. It is within this thesis’ research topics New DSS tools and Customisation.
Using a mobile phone Short Messaging Service (SMS) for irrigation scheduling in Australia – Farmers’ participation and utility evaluation

Nicholas J. Car\textsuperscript{a,b,⇑}, Evan W. Christen\textsuperscript{a,c}, John W. Hornbuckle\textsuperscript{a}, Graham A. Moore\textsuperscript{b}

\textsuperscript{a}CSIRO Land & Water, Griffith Laboratories, Griffith, NSW 2680, Australia
\textsuperscript{b}University of Melbourne, Melbourne, Victoria 3010, Australia
\textsuperscript{c}University of New England, Armidale, NSW 2351, Australia

\section*{A R T I C L E   I N F O}

Article history:
Received 8 July 2011
Received in revised form 29 February 2012
Accepted 5 March 2012

Keywords:
Irrigation
Decision Support System
Utility
Short Messaging Service
Australia
Winegrape

\section*{A B S T R A C T}

Irrigation scheduling Decision Support Systems (DSS) have seen poor uptake despite proved usage benefits. The failures of some previous systems with proven model accuracy and water savings ability have been attributed to interface difficulties and inappropriate information for end users. Use of the mobile phone Short Messaging Service (SMS) text messages was trialed as an interface to overcome these difficulties. Irrigation system dripper run time scheduling advice was sent daily to 72 Australian irrigators’ mobile phones from a water balance system called IrriSatSMS. Irrigators sent back information on irrigations and rainfall, also via SMS, to update the water balance. This trial showed that a complex, water balance-based, DSS could rely on SMS as the sole interface.

All 72 irrigators involved were content to receive messages daily for the entire growing season (200 days). A measure of engagement and utility of the system was determined by those who returned their irrigation and rainfall data; 45 sent in their data all season, 13 for half the season and 14 never sent in any data. Thus we infer that 45 users (63\%) found the SMS system of enough utility to use for the whole season. Also, at end of season, 6 of the 13 who had stopped half way through said that in retrospect they wished they had not. Thus overall 80\% of irrigators found the system useful.

User interview data showed the simplicity of use, advice and the prompting effects of intrusive delivery (phone ringing) were key features in the resultant strong engagement of irrigators. Success also relied on appreciating that irrigators will only use objective decision support advice as one element in a set of decision making tools that include subjective and unquantifiable elements, such as plant appearance.

This strong uptake reverses the trend in irrigation decision support which has seen poor uptake of sophisticated systems that produce comprehensive scheduling support but which are, or are perceived to be, complex and time consuming to use. Additionally, high participation rates show that much model input data may be collected from irrigators via SMS so it can be used as a very cheap bi-directional communication channel.

Crown Copyright © 2012 Published by Elsevier B.V. All rights reserved.

\section*{1. Introduction}

Irrigation scheduling Decision Support Systems (DSS) have experienced poor uptake amongst irrigators in Australia despite much investment and well publicised objective evidence that they can increase water use efficiency. For example, WaterSense, a sugarcane irrigation scheduling DSS with proven ability to increase water use efficiency (\textcite{Inman-Bamber2005}) has fewer than a hundred users among a potential pool of thousands. Australia-wide, a 2005 survey of irrigation DSS found that 21 are in operation but most have only a dozen or fewer users (\textcite{Inman-Bamber2005}).

A survey by Olivier and Singels (2004) of the reasons for not adopting scientific irrigation scheduling techniques by South African irrigators identified two main barriers to adoption. The first was the complexity of use and hence the difficulty of applying them to farm practice and the second was whether their use would actually translate into benefits. Much work in Australia on barriers to DSS adoption (\textcite{Carberry2001, McCown2006}) notes a ‘gap’ between scientific and industry approaches to scheduling that supports the South Africa experience.

Analysing a typical irrigation scheduling tool such as ‘WaterSense’ (\textcite{Inman-Bamber2005}) we see that it requires Internet access, the entry of much data and a long wait time (40 min) for results to be generated. WaterTrack Rapid (\textcite{Watertrack2006}), another recent Australian DSS, designed to minimise user effort through limiting data reporting,\textsuperscript{1} still requires irrigators to access...
the Internet from a computer, enter in data and then run the system for a response. Given the known rate of adoption of all DSS in Australia, is extremely low (Inman-Bamber and Attard, 2005) and knowing the well understood barriers to adoption, it is clear that in the case of WaterSense, the benefits to its use do not outweigh its difficulty of use. In the case of WaterTrack Rapid, its use is still not sufficiently easy to result in widespread adoption.

Irrigators who do use some form of decision support do so only in conjunction with many other data sources. Pre-trial interviews with our test group of irrigators showed that all irrigators, including those who had soil moisture probes installed, used vine observations, weather observations and experience to help them irrigate. Many used soil wetting pattern observations and shovel soil testing and a few individuals also used less common data sources, such as infra-red crop images or weather warnings from other locations as a forecast to help them. For many of these data sources there is no objective, deterministic method that can be used to generate irrigation volumes. Invariably, irrigators must rely on heuristics and past experience to do this. Thus DSS such as SOAK (Costigan, 2008), a farm water management software package that tries to coalesce all the irrigation data sources that its designers believe are relevant to irrigation scheduling, must inevitably fall short of providing ‘all the answers’ for many irrigators both for their inability to include the whole, wide, range of data sources currently in use and their inability to measure the non-quantifiable sources. Since the announcement of the development of SOAK and a prize it won in 2008, there have been no further references to it publicly in the irrigation industry.

The cellular mobile Short Messaging Service (SMS) is increasingly used in many contexts to simply and quickly deliver and gather data from people with mobile phones. One recent example of its use is by diabetic clinics in managing remote patients’ blood sugar levels (Hanauer et al., 2009). In irrigation, SMS has been used to promote the understanding of how a flexible, water budget-based, irrigation schedule can save water and increase productivity over a fixed schedule (Singels and Smith, 2006). In this South African trial messages were sent weekly to 5 irrigators telling them to “stop-”, “start-”, “continue-“ or “do not” irrigate based on a crop growth model using estimated irrigations and measured rainfall. The study concluded that by communicating the model outputs to irrigators in the simple SMS form, water use efficiency was increased by 48%. The authors found that weekly communication was required to assure the participating irrigators that the system was still functioning. They also concluded that it could be advantageous to obtain measured irrigation volumes from the participating irrigators to improve model accuracy (Singels and Smith, 2006).

In addition to its ease of use, the low deployment cost of SMS and its ubiquity of use by many people, even the poorest, in both the developed and developing world means it is a technology that helps bridge the digital divide rather than widen it, thus modified forms of systems used in the developed world may be more easily modified for use in the developing world than Smart Phone or personal computer-based technologies.

This paper describes the use of high-end IT systems for irrigation scheduling and the response of irrigators to an SMS-based, textual, DSS interface for irrigation scheduling. It also reports the utility of such an approach for irrigation scheduling as determined both by system data and interviews with end users. Methods detailing technical aspects of the system, the trial location and participants, interactions with participants, the DSS calculations and the DSS operation in general are given. Results of system performance, irrigation applications and system cost to irrigators are provided from measured data. Results of irrigators’ understanding of the system and perceived system utility are provided from interview data. Discussion of user participation and user utility follows and finally a conclusion is given.

The motivation for this research is the desire to improve the utility of DSS to irrigators in order to increase the proportion of them using such tools. Increased use will not only lead to benefits to the individual but also to the irrigation industry as a whole as such DSS also easily function as data collectors for aggregate water use statistics.

2. Methods

2.1. System description, calculations and operation

The DSS used in this trial was called IrrisatSMS, which used satellite derived crop coefficients in a daily water balance approach. The mechanisms of the water balance calculations and crop coefficient generation and use are described in detail in Hornbuckle et al. (2009). An overview of the Decision Support System’s (DSS) architecture, as well as the methods of communication used between components, is shown below in Fig. 1.

The central server hosted the DSS calculation code (Microsoft C#.NET), the web page presentation (ASP.NET) and the database (MySQL). The cellular gateway service was a SOAP-based web service that accepted text generated by the DSS and passed it to commercial cellular networks in Australia that distributed the text via the Short Messaging Service (SMS) to users’ mobile phones.

The DSS used four streams of data to generate decision support. They were:

1. Weather data: Obtained daily from a weather station web server which in turn collected the data from a weather station network using dial-up radio modems.
2. Irrigation Management Unit (IMU) measurements: An IMU being a crop area of one or more fields but under a single irrigation management regime – measurements were taken of geographical location, area and the irrigation system application rate. This information was manually entered into the DSS database.
3. Satellite image data of land surface reflectance values: Used to estimate crop ground coverage. These data were collected from satellite images and processed to produce one average $K_r$ reading per IMU per satellite pass (as per Hornbuckle et al. (2009)). These were then also stored in the DSS database.
4. Irrigation application and rainfall data: These data were sent in by irrigators for their specific IMU to the DSS via SMS. The SMS messages were required to be formatted for ease of processing and, once received, values were automatically time-stamped and stored in the DSS database to contribute to individuals’ water balances. Thus the number and frequency of irrigator responses to the system were able to be tracked. Incoming irrigation and rainfall messages were processed automatically with custom software to feed information directly into the water balance model. Messages that were incorrectly formatted and were unable to be processed were automatically responded to via SMS asking the irrigator to re-enter the data and a log of the incorrect message was kept. In cases where the irrigator did not re-enter the data correctly, the irrigator...
was contacted via telephone in order for their entry to be understood and manually entered into the DSS.

DSS output messages were sent to irrigators at 7:30 am as most claimed to check their crop and irrigation systems in the morning. Only 3 irrigators would have preferred an alternative time, early evening (about 6 pm) as they thought it would allow them to better cater for evaporation from the current day. Fig. 2 shows an example of an irrigator’s phone displaying a typical daily message.

As all of the irrigators in this trial were using drip irrigation systems, the DSS calculated a daily dripper run time (DRT), for each IMU, based on a cumulative calculated crop water deficit (CWD) and using measured system parameters, as in Eq. (1):

\[
DRT = \frac{A_e \cdot Apr}{60 \cdot CWD}
\]  

DRT: dripper run time (min), \(A_e\): area per emitter (m\(^2\)), Apr: emitter application rate (l/h), 60: constant, 60 minutes per hour (used for a result of dripper run time in min), CWD: crop water deficit as depth per unit area (mm).

Cumulative (from the start of the season to the current date) crop water deficit (CWD) was calculated from cumulative effective rainfall (\(\sum \text{R}\) in mm), irrigation (\(\sum \text{I}\), in mm) and crop evapotranspiration (\(\sum \text{ET}_c\)) as per Eq. (2):

\[
CWD = \sum \text{R} + \sum \text{I} - \sum \text{ET}_c
\]  

Irrigation values in millimetres were calculated from dripper run times and rainfall values were measured by farmers individually using their own rain gauge. Rain gauges were supplied to farmers who had insufficiently accurate gauges. Rainfall values supplied by farmers were modified to obtain an approximate ‘effective rainfall’ value using a reducing function developed through local observation of rainfall runoff on a local similar soil type – the laboratory field site which was in the centre of the trial area.

The crop evapotranspiration, \(\text{ET}_c\), was determined from reference evapotranspiration, \(\text{ET}_0\), and a crop coefficient, \(K_c\), using Eq. (3) following the method outlined in the Allen et al. (1998) but using a modified Penman equation with parameters for the Griffith area provided by Meyer (1999).

\[
\text{ET}_c = \text{ET}_0 K_c
\]  

\(\text{ET}_c\): crop evapotranspiration (mm), \(\text{ET}_0\): reference evapotranspiration (mm), \(K_c\): crop coefficient (dimensionless).

The daily \(\text{ET}_0\) readings were calculated from local automatic weather station data. Evaporation and transpiration were not separated but were accounted for using crop coefficient \(K_c\) values.
Monthly $K_c$ values were determined for each irrigator’s individual IMU from remote sensing using Normalised Difference Vegetation Index (NDVI) in a process similar to that described in Johnson et al. (2007). The monthly $K_c$ values were interpolated forward using previous season future month values and then updated when current season values were measured. $K_c$ values were manually uploaded to the DSS by researchers. The resolution of LANDSAT images used was $30 \times 30 \text{m}$ which was much smaller than any of the IMUs in this study.

Using the water balance calculations described above each irrigator was sent a daily SMS message which provided a dripper ‘run time to zero’ in minutes. This was the length of time, calculated from the CWD, that the irrigation system should be run in order to replace the CWD. Since CWD was cumulative (from the start of the season to the current date), irrigators were able to see the positive compound effects of irrigations and rain as well as the negative compound effect of cumulative daily evapotranspiration. In cases of a positive CWD, negative minutes were sent to the irrigator. Care was taken to start irrigators’ water balances at a ‘zero’ point or point in time at which their IMU had a zero CWD. This was realised practically by starting irrigators either after a reasonable rainfall event or after a large, early season, irrigation.

Messages sent by irrigators back to the system containing rainfall and irrigation information were able to be sent at any time. Since the system’s calculations ran at 6 am each day with the resultant messages sent at 7:30 am, messages sent after 6 am on a particular day were only included in the water balance calculations for the next day. Irrigators were free to choose the timing of their irrigation reporting (before or after events) and rainfall (at stages through long rain events or at their end) but were encouraged to be consistent, whatever their choice.

The approach in this DSS meant that irrigators were not told specifically when or how much to irrigate but rather just how much they would have to irrigate, on any given day, to return their CWD to zero. This meant irrigators retained full flexibility of when to irrigate and how much to apply and so were able to continue to adapt their irrigation regimes to suit their own particular conditions and preferences.

The only monetary cost to irrigators associated with participation in this trial was the cost of sending in text messages of irrigation and rainfall.

### 2.2. Trial participants

The trial was run in the Griffith region of New South Wales, Australia (34°17’24’S, 146°2’24”E). This is a semi-arid region where summers are hot and dry and the winters are mild. Mean annual rainfall is 418 mm, but is highly variable ranging from 140 to 700 mm occurring throughout the year and in isolated stormy bursts during summer. Mean annual potential evapotranspiration ($ET_0$) is 1800 mm.

The irrigation season for grapes in the Griffith region is from September to April/May. The trial was conducted from September 2008 to May 2009.

Participants were wine grape growers selected for their use of drip systems and mobile phone ownership. Mobile phone ownership was close to ubiquitous amongst all irrigators in the region so this was not a limitation. They were also selected to be representative of the range of drip irrigators in the region by the local grower association technical officers. The reason for this was to ensure as representative cross section of the local irrigation community as possible was used in the trial. The sizes of the grape farms of participants are given in Fig. 3A, their ages in Fig. 3B and their years of irrigation experience in Fig. 3C. All irrigators were fluent in English even though for seven it was not their first language.

During initial interviews irrigators were asked about their irrigation systems and their use of mobile phones, computers and the Internet. They were then asked about their irrigation scheduling techniques in general and their experience with evapotranspiration in more depth. Table 1 gives the number of irrigators who used various methods and tools to schedule before the trial. Igirators used more than one method or tool to determine when and how much to irrigate. Some common combinations were visual inspection of the vine and soil moisture determined by digging and soil moisture probe readings with weather forecasts.

Irrigator’s previous use of evapotranspiration based methods for scheduling was very limited, indeed only one irrigator could demonstrate use of a water balance approach. To ensure irrigators understood the information they received throughout the season, they had training in evapotranspiration basics (how a simple crop water balance is driven by water in (rainfall and irrigation) and water out (evapotranspiration and drainage) and that the evapotranspiration of particular crop fields can be estimated through measured reference evapotranspiration and satellite imagery of the crop). They also had the phone data they received explained to them on several occasions. It was intended that they should easily understand the information they received.

All irrigators spent 15 min or less, on average, deciding on their irrigation application volume for each IMU, each day, including the time taken to use scheduling tools listed in Table 1. Most irrigators had between one and several (2–5) IMUs meaning a total irrigation decision time of less than an hour was normal. Irrigators all approached the task of deciding how much to irrigate each day having an approximate figure in mind due to the time of the season, a pre-existing schedule, previous experience with that crop or system limitations.

![Fig. 3](image-url) (A) Participating irrigators’ farm sizes, (B) participating irrigators’ ages, and (C) participating irrigators’ irrigation experience.
Irrigators’ knowledge of irrigation concepts and other irrigation season questions focused on the collection of factual information, inactivity. Communicating with the system were called after 1–2 weeks of on, they were contacted only as required. Those found to be not after the initial meetings and the following 1–2 months. From then tors were held to choose a single irrigation management unit (IMU) as indicated in Dec–Feb) than at either the beginning or end of the season.

Researchers interactions with participants varied over the season as indicated in Table 2. Initial meetings with individual irrigators were held to choose a single irrigation management unit (IMU) – one specific block or field managed as a single entity – to be used in the trial. Measurements of their irrigation system parameters – crop type, size location and application rate were also taken during this visit. The system application rate was used to convert the dripper run times sent in by irrigators into millimetres for use in the water balance and to convert millimetres of evapotranspiration into minutes of dripper run time.

Most contact with irrigators during the season was 1–2 weeks after the initial meetings and the following 1–2 months. From then on, they were contacted only as required. Those found to be not communicating with the system were called after 1–2 weeks of inactivity.

Irrigators were asked questions during pre- and post-season interviews. All the interviews were structured interviews. Pre-season questions focused on the collection of factual information, irrigator’s knowledge of irrigation concepts and other irrigation DSS use. Post season questions assessed the outcomes of the irrigator’s season and the block involved with the trial in particular. They also assessed the irrigators’ experience with using the DSS at different points in the season and overall.

These questions were used to determine the utility of the DSS. Utility here was defined as the usefulness of the DSS to the irrigator:

1. Compared to other DSS and decision making tools the irrigator used.
2. By the effort of use.
3. In terms of useful behaviour changes.
4. The learning/educational benefits.

Irrigators were asked firstly for their perception of the DSS’s utility at various season stages (early, mid-year, late season, post-harvest) compared with other scheduling tools they use such as visual inspection of the vines, soil moisture probes and other tools. They were also asked for their perception of the overall impact that the DSS had on their scheduling regime and the effort required to use the DSS.

Finally, irrigators were also asked an open ended question as to what effect participation in the trial had on their irrigation practice. The resulting unstructured answers were subjected to qualitative thematic analysis.

3. Results
The service began in September 2008 with 23 irrigators and this number increased through the early part of the season. By November 2008, 54 irrigators were enrolled, by December 2008, 67 and from January 2009 onwards, 72.

3.1. SMS system performance
The automated delivery of SMS messages and processing of SMS messages from irrigators technically worked well. However, it was important to test the reliability of the Short Messaging Service as no guarantee of timely delivery of messages is made by cellular phone providers, messages may be delayed up to a week. In practice it was found that mobile phone service provider faults only prevented 1 day’s messages from being delivered and did not delay

\begin{table}[h]
\centering
\caption{Irrigation scheduling method and tool use by the 72 trial irrigators before trial commencement.}
\begin{tabular}{|l|l|}
\hline
Tool/method & No. of irrigators$^a$ \\
\hline
Experience & 66$^b$  \\
Moisture probes & 45  \\
Shovel – to inspect soil moisture & 28  \\
Visual plant inspection & 26$^c$  \\
Fixed schedule & 19  \\
Weather (irrigator’s estimations of crop water demand) & 18  \\
ET$_g$ – growers using reference ET from a local weather station & 5  \\
ET$_r$ – a grower using local weather station ET and his own estimates of crop coefficients & 1  \\
Infra-red probe – a grower using an IR camera to tell when vine leaves reach background temperature & 1  \\
\hline
\end{tabular}
\end{table}

$^a$ Numbers add to more than 72 as irrigators used multiple tools and methods.
$^b$ For some irrigators (9) this was their first year using drip systems and some of those claimed to have little or no experience with drip system schedules.
$^c$ All irrigators used plant inspections but not all listed it as a scheduling tool.

\begin{table}[h]
\centering
\caption{Interactions between trial participants and researchers.}
\begin{tabular}{|l|l|l|}
\hline
Time period & Interaction & Methods \\
\hline
Pre irrigation season Jul–Aug 08 & Initial contact with 72 irrigators & Telephone calls polling interest \\
Pre irrigation season Aug 08 & Initial project information session & Group meeting at grower association offices \\
Early irrigation season Sep–Oct 08 & Initial interview of 72 irrigators & Irrigator farm visit and interview by a single researcher. Irrigation system data gathering, trial sign-up and SMS message sending and receiving practice \\
Early irrigation season Oct–Nov 08 & SMS sending and interpretation assistance & Telephone calls and SMS as needed \\
Mid irrigation season Dec 08–Jan 09 & System check up & Telephone calls, one or two farm visits and phone conversation for irrigators as needed. System restart for some lapsed users wishing to re-engage \\
End irrigation season May 09 & System shut-down & SMS alert, some phone calls \\
Post irrigation season May–July 09 & Post season interview & Farm visit and interview for system usage and irrigator season results data gathering as well as an assessment of irrigators’ continued interest for future seasons \\
Post 08/09 irrigation season, early 09/10 irrigation season Sep 09 & Post project information session & Overall project results presented to participants, sign-up of past participants for the 09/10 season, information session for new participants \\
\hline
\end{tabular}
\end{table}

\begin{table}[h]
\centering
\caption{Days of faulty messages, shown in brackets, [], and days of messages not sent, shown without brackets, for the whole seasons.}
\begin{tabular}{|l|l|l|l|}
\hline
Mobile phone service fault & Database or software error & Weather station data error & Season days of messages \\
\hline
\hline
Total days of correct messages & & & 223 (94%) \\
\hline
\end{tabular}
\end{table}
any messages for more than a few seconds. Table 3 shows the faulty message counts.

3.2. Results from irrigator interviews

Results from interviews and the water balance database were used to try and understand if the irrigators understood the information they were receiving. They started receiving SMS messages after an initial phone call, a group meeting at grower association offices and a farm visit and interview by a researcher during which general principles of ET-based scheduling and the SMS system overall were explained. However, even after these three explanatory interactions, some of the irrigators apparently still did not understand either ET scheduling or the SMS system. All of these irrigators took some faith in the researchers and enrolled in the service.

Results of early season phone contact with irrigators found that once they saw the minutes of dripper ‘run time to zero’ increase each day over some time and then decrease when they sent in rainfall or irrigation events, they then understood how the water balance system worked even if they did not understand ET or crop coefficients.

Half the irrigators did not understand the relationship between dripper run times in minutes and depth of water applied in millimetres (see Eq. (1)) or the conversion of millimetres times hectare to Megalitres (100 mm ha = 1 ML), the unit that is used for irrigation metering. Of those that did not already have a working knowledge of these conversions, some came to understand approximate minute to mm conversions for their blocks through explanation and some realised it themselves from seeing daily messages with dripper run times in minutes and water balances in mm. This did not seem to affect their use or understanding of the SMS system as there was no lack of system use on behalf of those irrigators who did not understand the conversions.

Almost all irrigators immediately understood that the dripper run times derived from the water balance were to be used in the context of multiple scheduling data sources (such as soil moisture and weather forecasts) and thus used the dripper run times as a guide, rather than absolute figures to be followed exactly. Those few irrigators who wished to abandon all other scheduling tools and rely on the SMS messages were asked not to do so.

3.3. Irrigators interaction with DSS

Regarding the mechanics of SMS message use almost all growers had received a message (94%) and read it on their mobile phones before using this system, but much fewer (55%) had created and sent them. While almost no irrigators expressed concern in reading incoming messages, many had trouble initiating message creation and then message typing before sending. Most of the phones used by irrigators in this study allow the user more than one way to create a message and this caused confusion. Typing out the messages presented problems for about 20 irrigators especially regarding phone predictive text functions. In all but 4 cases, sending method and typing problems were overcome by repeated demonstration and advice provided over the telephone in the first month of service. Of the 4 irrigators who could not overcome these problems, 2 delivered their irrigation and rainfall events to the researchers by telephone and 2 stopped using the system. It is possible that the inability to send messages may have stopped another two from using the system but was not given as the reason by them.

None of the message-sending irrigators found sending in the messages to be an inconvenience. The time taken by irrigators to send in messages was considered negligible after their first few messages had been sent. All irrigators understood the format and were able to send in correctly formatted messages. Three irrigators who took holidays away from their farm during the season needed reminding of the message formats upon return but then resumed message sending with little effort.

Only three irrigators wanted the messages delivered at a time other than 7:30 am and only two irrigators were annoyed by the daily receipt of messages.

The daily receipt of the message had a ‘prompting effect’ for 66% of growers in actively thinking about irrigation scheduling. All the irrigators checked the message daily, either upon receipt of the message or shortly thereafter. A few moved to checking the messages only every second day or so later in the season, especially post-harvest.

Three quarters of the irrigators referred to each message only once and then either ignored it afterwards or deleted it from their phones. The other quarter either saved all the messages on their phones or recorded the values delivered on paper or using a computer spreadsheet.

Irrigators stated that they sent in data using SMS most regularly after receiving the daily morning water balance message and this is backed up with inbound message arrival timestamps showing that 60% of all inbound messages arrived between the morning message (7:30 am) and midday with 7:30–8:30 am being the time most used; 24% of all messages. This was because irrigators found that the incoming message reminded them to communicate back to the system.

The number of messages sent in to the system varied greatly between growers and over the course of the season. Some growers were sending in 15 or more messages per month but the average was 6 messages per month. Fig. 4 shows the number of messages sent in by growers per month over the main irrigation period from November to February.

Irrigators growing wine grapes in this region with drip systems have an approximate maximum interval between watering events, rain or irrigation, of 10 days due to drip system output capacity. This means that for an irrigator to fully participate with the DSS, they must send in at least 3 or 4 messages a month. Thus Fig. 4 shows that during the peak irrigation season, December and January, about 50 irrigators (76%) were participating fully. In February this number reduced as some growers harvested in early February, after which they irrigate less and treat irrigation as less critical. Fig. 4 also shows an increase in the number of irrigators sending in many messages per month in December and January. This indicates irrigators sent more messages during the period of intense irrigation.

![Fig. 4. Count of irrigators sending in certain numbers of messages per month.](image-url)
Rainfall can also be used to help estimate the number of irrigators actively engaging with the system. The system requires that messages be sent in regarding rainfall, thus comparing the number of messages reporting rainfall with known rainfall events can provide an estimate of user engagement with the system. Two significant rainfall events occurred in the peak irrigation season; the 7th–8th November and the 12th–14th December. Table 4 shows the number of irrigators who reported rainfall in or just after these periods. Due to summer rainfall variability, not all irrigators would have received rain however these periods were chosen due to the magnitude of the rainfall events (27 and 41 mm, respectively) indicating widespread rain from a large storm front and thus is can be expected that a large proportion of farms would have received significant rain.

Table 4 indicates a very high response rate after rain of 75–95%. This is a similar estimate to the 76% of growers being fully engaged by assessment of irrigation input SMS.

Only two requests were received to turn off the service; one in early February 08 following an early harvest and the other only 2 weeks before the planned shut-off date. As such it appears that the receipt of daily SMS messages is acceptable to growers; even those growers who at the final interviews stated that they did

Table 4
Number of irrigators who sent in messages for two major rainfall events.

<table>
<thead>
<tr>
<th>Rainfall period</th>
<th>No. of irrigators using the system</th>
<th>No. of irrigators who sent messages</th>
</tr>
</thead>
<tbody>
<tr>
<td>7th–8th Nov 08</td>
<td>54</td>
<td>51</td>
</tr>
<tr>
<td>12th–14th Dec 08</td>
<td>72</td>
<td>54</td>
</tr>
</tbody>
</table>

![Fig. 5](image_url) Partial cumulative CWD graphs for 2 irrigators showing differing responses to the system. The top graph (A) shows the irrigator kept a positive CWD using both rainfall and irrigation while the bottom graph (B) shows the irrigator returning the CWD to zero when it reached a fixed approximate value of about −35 mm.
not use the information much were content to continue to receive the messages.

3.4. Irrigator response to DSS recommendations

The utility of the system can also be quantified by assessing the irrigator behaviour with respect to the graph of daily CWD and constituent components provided by the DSS. Fig. 5 shows two such daily CWD graphs from different irrigators over approximately the same 2 month period: late October/early November to late December. Graph A shows the irrigator maintaining a positive water balance for almost the entire period and irrigating when that water balance reduced below zero on the 10/12/2008 and the 23/12/2008. Graph B shows the irrigator maintaining a negative water balance that reduces until approximately 12/2008. Graph A shows the irrigator maintaining a positive water balance reduced below zero on the 10/12/2008 and the 23/12/2008. Graph B shows the irrigator maintaining a negative water balance that reduces until approximately 12/2008. All three irrigators represented in Fig. 6 continued to interact with the tool until the end of the season despite their differing cumulative CWDs and associated suggested dripper run times. All three also chose to participate with using the tool again for the 2009/2010 season which confirms their perceived utility of it.

Fig. 6 shows that irrigators with different objectives and under differing conditions can gain benefit from using the SMS tool. The utility of the tool ranges from a direct to an indirect scheduling guide and that irrigators are able to interpret and use the information received from the tool in ways that benefits their situation.

From the DSS daily CWD graphs, usage can be placed into six named groups:

1. **Close throughout** – those who followed the DSS recommended dripper run time closely, irrigating within 10% of the recommended minutes throughout the season.
2. **Closely mostly** – those who followed the DSS recommended run-time closely until late December 08, then applied less than recommended for 3–4 weeks, then applied as recommended for rest of season.
3. **More throughout** – those who applied more than recommended consistently across the season. Note this was only up to 15% more than the total recommended number of minutes.
4. **Stopped sending** – those who followed the recommended amounts until late December 08/mid January 09 and then stopped sending input data.
5. **Less throughout** – those who those who applied much less than suggested.
6. **Never sent** – irrigators who never sent in any data.

The number of irrigators in each of group is given in Table 5. Irrigators in Groups 1–3 (68%) can be collectively grouped as full system participants. They engaged with the system until the end of the season, albeit with different watering preferences.

Irrigators in Group 5, less throughout, did not ascribe to a single motive for doing so in end-of-season interviews. Two said that they deliberately reduced their watering to stress their crops for quality reasons, two said they did not think it was economically viable to fully water their crop and two said their crop was ‘happy’ with the amount of water that they had given it.

Group 4, stopped sending, all ceased sending messages as they saw their suggested dripper run times increase beyond what they
perceived to be reasonable and so did not persevere. It would seem they lost confidence in the system. Half the irrigators in this group (7/12) stopped after some usage as they felt the system offered them no benefits. Three stopped for reasons unrelated to irrigation such as personal and other family issues.

Some irrigators who were generally happy with their involvement in the trial and whose actual dripper run times differed from the suggested dripper run times and but did not change their behaviour explained that the differences between the volumes were necessarily so due to their blocks being non-standard. One irrigator who watered much more than the suggested dripper run times said that his block was steeply sloped and had unusually sandy soils. Others who watered much less said it was due to their heavy clay soils.

The important message from these responses is that no irrigators that followed the advice were unhappy with having done so and no irrigators that received the advice and monitored it through to the end of the season were unhappy about having done so, indeed many wished they and followed the advice, not just monitored it. This realisation was corroborated both at the post irrigation season group meetings and also through the numbers of irrigators signing up for the service for the 2009/2010 season.

When asked how irrigators translated the DSS advice into action, 25% answered that they allowed minutes to accumulate for a number of days and then irrigated for a time close to that number. The majority (79%) said they took advice from several data sources (such as weather forecast, auger results, vine inspection) and would then come to a final decision based on a fusion of this type of information with the DSS information. None of these irrigators could quantify the implicit weight they ascribed to each data type with the DSS information. None of these irrigators focusing less on irrigation in the post-harvest period and would then come to a final decision based on a fusion of this information.

Table 5

<table>
<thead>
<tr>
<th>Group</th>
<th>Number</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Close</td>
<td>26</td>
<td>36</td>
</tr>
<tr>
<td>2. Close with step</td>
<td>7</td>
<td>9.7</td>
</tr>
<tr>
<td>3. More</td>
<td>3</td>
<td>4.2</td>
</tr>
<tr>
<td>4. Much less</td>
<td>6</td>
<td>8.3</td>
</tr>
<tr>
<td>5. Stopped</td>
<td>15</td>
<td>21</td>
</tr>
<tr>
<td>6. Never sent</td>
<td>17</td>
<td>24</td>
</tr>
<tr>
<td>Total</td>
<td>72</td>
<td>100</td>
</tr>
</tbody>
</table>

Of the irrigators who took note of the irrigation and rainfall totals, 8% viewed their water balance graphs online also. These irrigators all responded that the graph better represented the individual effects on CWD due to rainfall and irrigation than did the cumulative totals in the messages.

When asked about how the cumulative rainfall affected their irrigations, no irrigator was able to give a quantitative answer, not even three who specifically recorded cumulative rainfall at different stages in the season. Answers to other questions about the use of rain by irrigators referred to watching the changing dripper run times. Irrigators had been reminded on multiple occasions that both rainfall and irrigation affected their dripper run times.

Fig. 7 shows the utility scores that irrigators gave to the system overall at four points in the season: early (Oct/Nov), mid (Dec/Jan), late (Feb/Mar) and post-harvest (Mar +). Note the lack of ‘1’ responses (not useful) in the first two time periods as well as the downward trend of ‘5’ with time and the increasing ‘2’ and ‘1’ responses with time.

The average utility scores from each of the 4 time periods in Fig. 7 are given in Fig. 8. Note the early increase in utility (the high ‘4’ response for the ‘mid’ season group) and then the decline over time (increasing ‘2’ and ‘1’ results in ‘late’ and ‘post-harvest’).

When irrigators who reported an initial increase in utility were asked why, all responded that their confidence grew with time as they understood how the tool worked and they had seen it in action, however all irrigators also noted the heightened importance of scheduling tools, including this tool, during the peak irrigation period so the increased utility score must be due to a combination of both reasons. The later decline in utility is, in general, due to irrigators focusing less on irrigation in the post-harvest period and more on other farm issues and, in 20% of cases, due to disenchantment with the tool. For the season as a whole, 66% said that this system was not as useful as their primary scheduling tool but that it was still seen as useful.

Of the irrigators actively involved in this trial, 40 irrigators answered the open ended question “How did the tool affect your scheduling behaviour over all?”. Their answers followed 5 similar behaviour themes which were:

- **Small**: The dripper run times were close to what they were doing so observing and/or following it resulted in little change in their former practice. The tool was still seen as valuable in that it gave them confidence in their current practice and acted as a back-up. The DSS did thus influence their practice by positive reinforcement and resulted in a small change in behaviour whereby a new irrigation data source was considered in coming to a decision.

- **Large**: They perceived the dripper run times to be higher than they believed necessary but they still followed them on trust, resulting in a large change to their practice.

- **Significant**: They used the DSS as their main irrigation scheduling tool but could not describe how this may have changed their practice. Thus the tool can be deemed to have influenced their practice, but the behaviour change is unknown.

- **Marginal**: The DSS information was interesting and educative to them but and changed their understanding but did not change their behaviour.

- **None**: They perceived the dripper run times to be too high and did not follow them, resulting in no change to their behaviour.

Table 6 gives the percentage of respondents who subscribed to the various views described above.

When assessing the total effect of the system on irrigator behaviour we can sum Group 2 and 3 giving 46% as those for whom the DSS has significantly affected their behaviour. We can then add Group 1 to this as those for whom there was some influence on
behaviour (73%). Beyond that we can add Group 5 to this to give 80% as those who were educated by the DSS. However, further discussion with Group 4 irrigators found that their post season analysis told them that they could have benefited from following the DSS, and would do so in future seasons. As such we can see that the DSS had a remarkable level of influence on behaviour and educative capability.

After discussing the effect of the DSS on their behaviour the irrigators were asked to reflect upon this. Group 1 all said that using the tool gave them confidence in their current method of operation. No one from Group 2 who had followed the advice of the tool and put on more water than they initially thought necessary was unhappy with having done so in retrospect. Three example responses from Group 2 are given in Listing 1 below:

Group 3 irrigators could not articulate their behaviour change due to:

- inter-seasonal changes in circumstance (e.g. crop age, financial returns, weather variations);
- lack of previous seasons’ experience with that particular block;
- this being their first year using drip irrigation;
- insufficient records from previous years.

This confounded efforts to draw comparisons between the trial season and previous seasons however, all of these irrigators thought the DSS useful and did not regret having followed it.

Several Group 4 irrigators said that in retrospect they wished that they had followed the suggested irrigation times that they originally did not follow due to thinking them too high. Listing 2 gives some example responses from this group.

Group 2 and Group 4’s responses related to the heat wave experienced by all growers in late January 2009. Due to the system’s recommendations for irrigation being higher than average practice from mid December 2008 to January 2009, those that followed the advice had well watered crops that suffered less in the heat wave than the crops of other growers. Some of those that lost crop due to the heat wave thought that they may have avoided doing so had they had followed the DSS’s advice.

Group 5 irrigators stated they found the involvement with the DSS interesting for a number of reasons such as learning about crop/weather interactions and quantitatively comparing rainfall and irrigation. All of these irrigators asked to trial the system again in the 2009/2010 season.

3.6. Cost and effort

Trial involvement was free so the total cost for an irrigator was due to messages sent. Eighty seven messages at $0.25 each (largest number sent at a high per-SMS cost) costs $21.75. No irrigators mentioned cost as an issue in interviews. The commercial system costs are reported in Hornbuckle et al. (2009).

Irrigators said they took “just a few seconds” or “no time” or “a minute” to form and send messages to the system. Observations of irrigators sending messages after some coaching suggest a sending time of about 30 s. Cumulative season sending time was therefore...
less than 43 min. If the time taken to read the daily messages was also 30 s, total season message time would be 116 min or about 40 s per day.

4. Discussion and conclusions

4.1. Participation

For participants in this trial, cost was virtually non-existent and the labour effort required was very low. Receiving and sending messages throughout the season did not perturb the majority of irrigators, no irrigators that discontinued with the trial cited reading or sending messages as the only issue leading to them not using the DSS. It appears that the effort required for effective participation did not exclude any irrigators from using the DSS. The complexities of ET-based water balance cycles were able to be hidden from end users with only knowledge of water in and water out being required for use.

This trial also indicates that SMS is a cheap, effective and minimally intrusive way of collecting water use data from a large number of irrigators. Such data is close to real-time and was able to be used for regional water use benchmarking. This trial showed that a complex water balance-based DSS could rely on SMS as the sole interface medium.

4.2. Utility

Use of this DSS did not prevent irrigators from using their other information sources and using these conjunctively. Also, as it did not stipulate when to irrigate or the amount, irrigators could be flexible in timing of irrigation to suit their farm management and the DSS could cope with varying management styles. The scheduling DSS advice was used by irrigators in conjunction with their current tools and heuristics. Growers had many and varied approaches to determining when and how much to irrigate and no irrigators reported relying solely on one data source to help them schedule. The simplicity of use of this DSS allowed it to sit comfortably with irrigators’ current approaches, adding information to their decision making in a complimentary manner. This we believe was the key to its acceptance.

For this reason and for the overall non-intrusive nature of the support, it appears that most irrigators felt that using the system offered a “can’t lose” proposition unlike other, complex/expensive scheduling tools.

Interview results confirmed the authors’ initial assumption that scheduling is a small, routine task of a few minutes a day on which irrigators do not wish to expend much energy. This DSS seems to have been able to be used without forcing irrigators to radically change their decision making behaviour. Irrigators who used offices regularly were able to use the web interface easily but for the majority who were more accustomed to spending time outdoors, the portability of access to the DSS via the mobile phone (SMS) interface was regarded as critical and greatly impacted on uptake.

More than 50% of the irrigators interviews showed that the ‘prompting effect’ of having their phone, which they always carried with them, beeping each morning upon receipt of the DSS-generated SMS message prompted them into thinking more about irrigation matters than they otherwise would have. Many also commented that the somewhat intrusive nature of the DSS’ daily interaction also contributed to keeping them involved with the trial.

The presentation of water balance data as drip run times hid the underlying science from the irrigators and indeed although some had trouble understanding rain and irrigation’s influences on water balances as well as some of the ET concepts they were still able to use the system (communicate with it and understand it’s suggestions) after seeing the water balance results over 1 or 2 weeks.

The utility scores that the irrigators assigned to the tool compare well with their other statements of use, their descriptions of changing use over the season and also their DSS use compared to their other scheduling tool use. As understanding the DSS’ function

Irrigator 1:
- put on a few extra hours
- happy with this
- could/should have put on more
- very useful for learning
- very simple - good

Irrigator 2:
- was sceptical of high readings
- followed anyway
- excellent results
- will continue to follow
- surprised how ETc related to min

Irrigator 3:
- used early on and until pre harvest
- didn’t use post harvest (saving water)
- yield was ok
- happy to have used it
- happy to use again next year

Listing 1. Example Group 2 responses.

Irrigator 4:
- reduced watering this season due to grape price
- lost lots of yield in heatwave
- retrospect should have kept closer to evap
  [as given by this DSS]

Irrigator 5:
- SMS said too much water
- fell behind due to other crop requirements
- wishes he kept up with it
- will follow closely next season

Listing 2. Example Group 4 responses.
early was related to continued use and as average perceived utility increased with familiarity, it is thought that a return farm visit to all irrigators one week after their initial farm visit and sign up to further clarify the system would enhance DSS utility and irrigator participation.

Some of the irrigators who discontinued using the tool never really engaged with it and did not understand how it worked. However, some of those who were initially skeptical of it (identified through low starting perceptions of utility and interview comments, see Fig. 7) came to appreciate it more with time. The trend seems to be that familiarity bred trust. It would seem then that if better initial engagement with potential users is undertaken resulting in better explanations of ET-based scheduling and the system’s functionality, the rate of discontinuation may be reduced. Since this tool has already achieved a high rate of user engagement in its first year of deployment, if the discontinuation rate is reduced, the system will be able to claim unusually high potential user penetration among irrigation DSS tools compared with those listed by Inman-Bamber and Attard (2005).

The DSS was offered for use by growers in the following season (2009/1200). Of the original 72 irrigators contacted at the start of the 2008/2009 season 69% wished to use it again. Of those irrigators who used the DSS for the full 2008/2009 season 83% wished to use it again. This occurred without any ‘marketing’ merely a single phone call to find out if they were interested to use the DSS again. This is a remarkable retention rate compared to the usage seen in other irrigation DSS.

Although this system worked only with irrigators using drip systems, if modification can be made for use with other irrigation systems, the approach of intrusive but minimal effort usage via the simple, mobile, interface may be relevant to large numbers of irrigators internationally as its reasons for success seem to be based on aspects of human nature. Such modification would also allow use by irrigators in developing countries with minimal high technology infrastructure.

References

3. **IrriSatSMS Extensions**

Chapter 2 related an experiment in which daily waterbalance-based irrigation scheduling advice was delivered to winegrape irrigators in Australia’s Murrumbidgee Irrigation Area via the mobile phone (cellular network)’s Short Messaging Service (SMS). It was apparent to this author and the co-authors of the previous chapter’s paper that many extensions to it could be perhaps be usefully made. Here related are a series of extensions – delivery of scheduling advice to farmers of other crops, farmers in other irrigation regions, advice made using different sources of waterbalance data – that were made with differing levels of success. The extensions were:

1. Farmers of crops other than winegrapes in the Murrumbidgee Irrigation Area;
2. Turf farms in Australia’s South-East Queensland;
3. Vegetable farmers in Australia’s Sydney Basin;
4. IrriSatSMS values calculated using mobile phone crop-coefficient estimates (rather than estimates from satellite imagery);

The regions these extensions were in are indicated in Figure 1 below.

![Figure 3.1: The areas of IrriSatSMS extensions indicated on a map of Australia showing proportion of irrigated land cover. The blue numbers 1 – 5 indicate the areas in which extensions, as per the number list above, were conducted.](image)
These extensions were selected due to each being a little different from the original trial: each would test a new or extended aspect of the original system. The first extension – other crops in the MIA – would be conducted in the same area as the original IrriSatSMS trial on different crops using similar irrigation systems, water ordering regimes and, in some cases, the same farmers. This would test applicability of IrriSatSMS to crops other than winegrapes. The second extension – turf farms in South-East Queensland – would test a new crop in a new area. The Sydney basin vegetable extension would test a new crop in a new area and also ones under a very different water regimes – urban water as opposed to rural water. IrriSat for cotton was conducted much later than the earlier trials and considered a different crop in a similar area but under a very different irrigation regime: flood irrigation as opposed to drip or sprinkler.

The mobile phone crop coefficient trials used the same regions and crops as other extensions but tried to test different sources of waterbalance input data.

3.1. Other crops in the MIA

In 2007/2008/2009, the most common perennial irrigated crops near Griffith, NSW, were winegrapes, oranges and prunes and it was common to see fields of the three crop types interspersed.

After the first full season’s trial of IrriSatSMS with winegrape growers, an attempt was made to introduce prune and orange farmers around Griffith to IrriSatSMS. The same system used by the winegrape growers was used with additional crop factor prediction curves introduced into the waterbalance calculations database which, right from the establishment of IrriSatSMS, was built to cater for such extensions given the expectation that the system could work for different crop types or varieties.

While a number (10 – 15) orange and prune farmers were recruited for a season’s trial, no summary report or scientific paper of the work was recorded. This was due to the trial farmers not seeing benefit in the system in the same way as winegrape farmers did (see previous chapter).

The main reason for this, as observed by this author, is that with perennial fruit trees such as oranges and prunes, irrigators tended to water conservatively, that is ensuring the trees were not stresses at all. This meant there was little room to aim for watering that was close to requirements or even below requirements, as trialled by winegrape farmers, especially red grape farmers who believed that their crops could tolerate, or even benefit from, water scarcity-induced stress.

3.2. Turf farms in SE Queensland

In 2008/2009, an attempt was made to bring irrigation decision advice similar to that delivered to winegrape growers in the Murrumbidgee irrigation area to horticulturalists and turf growers in South-East Queensland. The aim was to expand the geographic area and crop types of farmers involved in the use of IrriSatSMS to test its boarder applicability. Unlike IrriSatSMS used in the Murrumbidgee irrigation area, no local weatherstation-measured reference evapotranspiration,
ET₀ was available for South-East Queensland at the time of the trial so instead ET₀ taken from a nationally-calculated, interpolated, weather grid, known as SILO¹ was used.

An initial report from one season’s work with SE Qld irrigators, written in 2009, is presented in Appendix E.

The results from the report, which came before results from IrriSatSMS’ complete write-up, presented in Chapter 3, are broadly consistent with them. Indicated here, and proved in Chapter 3, was the utility to irrigators of being prompted to consider/better consider irrigation scheduling by intrusive text messages, regardless of the message content.

Another outcome from the SE Qld work not indicated in the report was deficiency of the LANDSAT imagery used to produce a crop coefficient value used to convert reference crop evaporation, ET₀, to usable crop-specific evaporation, ETc. This process, outlined in Chapter 3, depends on regular images of crops in order to know their growth stage and thus required water use. For SE Qld, some of the properties visited in field trials where poorly serviced by LANDSAT in 2009/10 imagery due to the cessation of LANDSAT 5, damaged sensors on LANDSAT 7 (parts of its images are unusable) and the sporadic nature of usable images due to frequent cloud cover. This prompted an investigation into other methods for determining crop coefficients by determining crop canopy cover which are related in Section 1.3 below.

### 3.3. Sydney Basin vegetable farmers

Starting at approximately the same time as the SEQ trail but lasting two seasons, vegetable farmers in the Sydney basin were also delivered IrriSatSMS messages. The intention of this trial was, as per the SEQ trial, to test the extended use of the IrriSatSMS approach over different areas, crops and irrigation methods.

One particular reason for this trial was motivation, on behalf of the NSW State Government, to reduce the amount of fertiliser ending up in Sydney rivers (the Hawksbury-Nepean system) as a result of over application of both fertiliser and water. There was thought that, if vegetable farmers in the target river catchment could use an objective irrigation DSS, they would reduce water applied.

The NSW Department of Primary Industries (DPI) engaged the CSIRO to test the utility of delivering mobile phone messages of irrigation decision support with a trial group of mixed vegetable and fruit farmers similar in establishment to the group used for IrriSatSMS trials but including a range of crop and irrigation system types. This work saw the installation of four weatherstations in the Sydney basin, marking of a few 10s of farms blocks and the establishment of the IrriSatSMS satellite image processing system for these blocks. A press release introducing the trial can be seen on an archived NSW DPI web page².

---

¹ SILO is still delivering ET₀ data online, see

Delivery of SMS messages to Sydney farmers took place over a trial period of many months and the farmers responses to those messages were recorded.

For the purposes of this PhD, no technical or DSS-related social findings resulted from this work. This was due, in this author’s opinions, to the complexity of the farming systems, crop variety, small scale of the farms (mixed in with other industry and thus hard to image and calculate waterbalances for), different motivations for styles of irrigation management and even language barriers.

3.4. Mobile phone crop coefficients

The work from SE Qld related above and also observations in the Riverina indicated that in certain areas and at certain times of the year, LANDSAT imagery was not sufficient for effective crop coefficient estimation. This was due to several reasons, the first being that some images were obscured by clouds or altered by haze. Missing one image at a critical crop growth stage, given the time between images was approximately 14 days, meant that useful crop coefficient curves could not be determined. A second reason was while some areas around Griffith NSW were well supplied with images from both the LANDSAT5 and LANDSAT7 satellites, other areas received only one or the other satellite’s images and, in addition, LANDSAT7 developed a fault with its main camera meaning that the edges of its images were unusable. Areas to the South West of Griffith were in this LANDSAT7-only zone with clipped usable images meaning that some farms of interest received practically no useful LANDSAT satellite imagery. Initially for this reason, an investigation into other methods of determining crop coefficient was made. A further reason for undertaking an investigation was one of spatial resolution: for small fields, the 30m resolution of both LANDSAT 5 & 7 was not fine enough to assist irrigators address crop variability which they were able to do via drip system adjustments3.

Due to the increasing occurrence of irrigators with mobile phones capable of taking photos around 2008 and irrigators’ successful use of phones in the IrriSatSMS trials, it was hypothesised that images of crops taken with a mobile phone might be able to be used to provide an estimate of crop canopy cover and, from that, a crop coefficient using established relationships per crop type (such as those given in (Trout & Gartung, 2006); work the author was lead to by his supervisors). If mobile phone-based imagery of crops was useful, it would be an even lower cost method crop imaging than other low cost methods published around this time, such as imagery by remotely controlled aircraft, as per (Stombaugh, Simpson, Jacobs, & Mueller, 2003) and could perhaps be implemented for very frequent and spatially dense crop coefficient measurements.

3 Experiments into dripper system uniformity were conducted by this author’s research labs at the same time as this mobile phone crop coefficient work, see (John Hornbuckle et al., 2009, 2012) for a contemporary conference paper on the subject and a later one more easily accessible.
3.4.1. Experiments

A campaign of field trials was undertaken from November 2007 until January 2009 to determine whether it was possible, and practical, for irrigators to take photos with their mobile phones that were then able to be used to determine crop coefficients.

The intention was, once crop coefficients were able to be generated from mobile phone images, a study of the relative utility of crop coefficients from multiple sources (satellite, mobile phone, growth curve prediction and farmer estimation) would be made.

Extensive field and IT work was undertaken to implement systems perceived to be required for these experiments and they are detailed in Appendix D.

3.4.2. Outcome

The outcome from this work was, unfortunately, inconclusive. The author did not obtain sufficient data to prove that crop canopy cover, and thus crop coefficients, determined by image processing was more or less accurate than human estimates of it. The author was then not able to progress to the next task of comparing the utility of crop coefficients from multiple sources.

3.4.3. Reflections

In retrospect, it can be seen that the work undertaken here was far too large in scope to have been successfully undertaken as just one experiment, within a set of four.

Significant knowledge of image processing is needed for accurate image processing results from semi unconstrained images like field photographs and this was only partly acquired in 2007 – 2009.

Far too much effort was spent on implementing a semi production messaging system to communicate between image acquirers and the image processing server in order for it to ultimately be integrated with IrriSatSMS than should have been before accurate image processing results were obtained.

More than two crop fields/types would have needed to have been imaged though their full life cycles over two seasons at least for defensible image processing results to have been obtained and yet not enough time allocated for this or crops sought. One of the two crop fields studied was abandoned part-way in to the season.

Significance of crop-specific knowledge – growth stages, canopy development, irrigation method etc. – was required for tuning of the system per crop type. Low crops (tomatoes & faba beans) were selected for ease of image acquisition at body height (as opposed to winegrapes whose height introduces issues due to obscuration of image) and it was thought that these would be straightforward to create scheduling advice for, given the development that had taken place for winegrapes but irrigation system differences – flood for low crops, drip for winegrapes – proved to require much adjustment that sapped project resources.

If posed in 2017, this experiment might seem to make far less prima facie sense due to the prevalence now of cheap and easy-to-handle drones that can obtain high quality aerial images of crops, thus the need to rectify ground-based images may be perceived to be less. However, using drones, managing imagery from them and so on are tasks that are either out of reach of, or
would impose a great learning and effort cost on, farmers and their use is still outside the target use of this work. The volume, coverage and resolution of satellite imagery has increased too, as have the tools available to convert their data into crop coefficients (IrriSat for cotton – see next section – uses Google Earth Engine’s freely available LANDSAT-derived NDVI dataset, see https://earthengine.google.com/datasets/) and this development has definitely removed some of the drivers for mobile phone imagery.

3.5. IrriSat for cotton

3.5.1. New system

Mindful of the established DSS for cotton in Australia CottonLOGIC but noting its mixed success in uptake (Hearn & Bange, 2002; Mackrell, Kerr, & Von Hellens, 2009), a variant or IrriSat was created for cotton irrigation decision support. Unlike CottonLOGIC, this was to be a simpler tool that didn’t extensively model the cotton crop but, like IrriSatSMS, simplistically modelled waterbalances of, in this case, cotton fields, and drew data from local weatherstations, satellite imagery and the irrigators themselves.

A radical change in the back-end systems and user interfaces of the DSS was implemented for IrriSat for cotton. Since the construction of the original IrriSatSMS system, public compute cloud providers had become available as well as publicly accessible sources of data relevant to the IrriSat systems, such as a large LANDSAT imagery of Australia archive. This allowed the 2014/2015 construction of IrriSat for cotton to rebuild its calculation database such that it could auto-scale from a few test fields and a few LANDSAT images to essentially all cotton fields in Australia and multiple years of LANDSAT archive, should such scaling become necessary. This scaling was due to Google’s Earth Engine product’s serverless architecture which recruits compute power on demand (with the user being billed proportionately).

A full description of the IrriSat system for cotton is given in (Vleeshouwer, Car, & Hornbuckle, 2015), also Appendix E.

The tool has continued to be developed until the present with presentation at cotton conferences (viz. J Hornbuckle, Montgomery, Vleeshouwer, Hoogers, & Ballaster, 2016; Montgomery, Hornbuckle, Hume, & Vleeshouwer, 2015) and other cotton sector events. It is currently operating as a desktop browser cotton DSS (without any mobile interface) online at https://irrisat-cloud.appspot.com.

3.5.2. Reflections

With the data sources and calculations of IrriSat for cotton basically the same as those for IrriSatSMS with the exception that weather forecasts have been incorporated (this was attempted but never achieved for IrriSatSMS), the type decision support offered isn’t radically different from that of the original application, except that it can provide advanced warning so the need to irrigate ahead of a heatwave. The desktop computer, web browser-based, user

In trials of IrriSatSMS, this author observed irrigators catering for heatwaves in advance by following weather forecasts. They did not have recommended pump rine times from IrriSatSMS
interface allows for a much richer set of interactions by the decision maker than the original SMS-based user interface did and the cloud-based, auto-scaling back-end allows for far more users than the original.

It is unknown at this stage, due to a lack of published statistics, what the user uptake of IrriSat for cotton is. At the end of the 2015/2016 cotton growing season which, was the last season with which this author was involved, despite much system capacity and publicity about the it, IrriSat for cotton had essentially zero irrigator users. This was gauged by the lack of any full-season crop waterbalance calculation statistics which would have been needed in order to deliver decision support. IrriSat for cotton was used for laboratory-based crop field comparisons using the satellite imagery archive of multiple fields and may have been used in an educational manner but it appears not to have been used for providing tactical watering advice directly to irrigators or for supplying benchmarking data to them.

3.6. IrriSat evaluation

The following chapter contains a paper titled “Tracking the value of an innovation through the new product development process: The IrriSat family of agricultural decision support system tools”. Written in 2016/2017 and accepted for publication in Australasian Agribusiness Perspectives, this paper assesses “attempt to improve the uptake of a new agricultural Decision Support System”, IrriSat.

This chapter summarises the work undertaken after the completion of the initial IrriSatSMS system for grape and orange growers, including work on a variant system for vegetable farmers, work for turf growers in Queensland, work for vegetable growers around Sydney and a system for cotton growers in New South Wales and Queensland known only as IrriSat.

3.7. References

https://doi.org/https://doi.org/10.1016/S0308-521X(02)00019-7


Hornbuckle, J., Christen, E., Car, N., Smith, D. J., Goodwin, I., McClymont, L., & Kerridge, B. (2012). Drip irrigation distribution uniformity in vineyards of Australia: Tools for mapping and

5
http://www.agrifood.info/perspectives/

for this but were able to apply water in advance and then later on, after the heatwave, use IrriSatSMS to calculate the post-event water balance and adjust further irrigation accordingly.


4. DSS case study

This chapter is the journal article *Tracking the value of an innovation through the new product development process: The IrriSat family of agricultural decision support system tools* represented as published in the following pages.

The citation for this paper is:


The work for this paper was undertaken after the establishment of the IrriSat system which followed on from IrriSatSMS in 2014+. It falls within the research topic DSS Adoption Analysis.
Tracking the value of an innovation through the new product development process: The IrriSat family of agricultural decision support system tools

N.J. Car
CSIRO Land & Water
EcoSciences Precinct, Dutton Park QLD 4102, Australia
nicholas.car@csiro.au

Abstract

In this paper, an attempt to improve the uptake of a new agricultural Decision Support System (aDSS) is described. The approach was to design it with an understanding of the successes and failures of predecessors and of the changes in patterns of relevant technology use over time, the “usage context”. Even though its predecessor, IrriSatSMS, showed great potential in pilot seasons, that system failed to be commercialised successfully. An investigation into whether this failure can be attributed to “technicentric design” – an aDSS problem lamented by many authors of papers on aDSS in the 2000s – is undertaken. Some relevant aspects of the design of the new aDSS system, IrriSat, are related and why a belief in its ability to overcome some of the commercialisation issues faced by IrriSatSMS and other aDSS is held. For this, the changing landscape of IT, digital agricultural data and farming lifestyle since the 2000s is considered and how it has and will affect aDSS usage and uptake. Finally, issues still faced by systems like IrriSat, both old and new, are considered.

These considerations indicate that while commercialising a new aDSS is always going to be risky for an organisation, particular aDSS design choices that are now available, such as the use of cloud computing, can reduce running costs and staffing effort significantly, thus substantially reducing that risk for certain aDSS types. Also evident is that a step change in IT use in farming since the first trials of IrriSatSMS, in Australia at least, has seen many issues that once plagued aDSS use regarding farmers’ interactions with IT systems evaporate. However, new issues, such as data deluge, have surfaced.

With these technology and technology usage changes, I conclude that the pessimism shown in aDSS papers in the 2000s was based on factors that are no longer dominant in the aDSS landscape; a paradigm shift has occurred. However, the new paradigm has its own issues. Experience in other areas indicates that this paradigm’s issues can also be overcome and thus the future for aDSS in general, and perhaps IrriSat specifically, looks bright.

A number of challenges are revealed that can face the development and commercialisation of new products. Central among them is the importance of tracking changes in usage context of, and therefore competition for, the new product, and related impacts on product value to the user.

Keywords: Decision Support System, agriculture, irrigation, cloud computing, informatics, Australia
1. Introduction

In the first decade of the 21st century, a pessimism about the broader adoption of objective agricultural Decision Support Systems (aDSS), based on their advancing technical capability alone, took hold. Papers such as “Changing systems for supporting farmers’ decisions: problems, paradigms, and prospects” (McCown, 2002) and “The FARMSCAPE approach to decision support” (Carberry et al., 2002) lamented an aDSS past where sophisticated models were built by scientists and engineers that were ultimately unsuitable for farmers to use due to cost, complexity or their inability to work in the “real world”. Those papers, perhaps best summarised by Matthews (2008) looked to leverage participatory action research (PAR) to better engage potential users in system development, thus making tools more appropriate for them. They aimed to find the most appropriate places for their aDSS deployment, based on “institutional and socio-political” considerations, not just the “technical or theoretical aspects of the tools themselves”.

In the Australian irrigation DSS context, a subset of aDSS, survey results of system use from the middle of last decade bear out Matthews: Inman-Bamber & Attard (2005) show that there were then only 21 systems in operation Australia-wide but most with only a dozen or fewer users.

A few years later, at the start of this decade, Car et al. (2012) showed that a technical DSS (IrriSatSMS), created without PAR but with an eye to simplicity of use, could deliver utility and see good uptake, at least in trial phases where cost of use was low or zero. Survey results of participants in that aDSS trial, also related in that paper, showed many of them also used very costly soil moisture probes – a form of aDSS – which indicates that cost alone was then not necessarily a barrier for use. While IrriSatSMS was much hyped at the time, including winning an industry design award¹ it has, nevertheless, joined the ranks of many other aDSS that showed promise but, a few years on, are no longer in operation.

In this paper three main tasks are undertaken: first, a follow-up on the fate of the IrriSatSMS DSS described in Car et al. (2012) and an analysis of the attempt to commercialise it. (aDSS history must be reviewed in order for it not to be repeated!) Second, a discussion about a new aDSS derived from IrriSatSMS, simply titled IrriSat, that is now under trial. While many of the technical details of IrriSat have been published previously (Vleeshouwer, Car and Hornbuckle, 2015), here an analysis of its raison d’être – how it attempts to overcome some of the issues faced by IrriSatSMS, particularly those relating to attempts at commercialisation – and its usage to date are given. The new light this sheds on points of view taken by McCown (2002), Carberry et al. (2002) and Matthews (2008) are specifically indicated. Third, the changing landscape of IT, digital agricultural data and farming lifestyle since the 2000s in Australia is considered, as is the effect it has already had on aDSS since then and what effects it seems likely to have in the future.

---

¹Irrigation Australia Ltd. award for “Best New Irrigation Product”, 2010.
2. The Fate of IrriSatSMS

![System diagram of IrriSatSMS](image)

**Figure 1**: A system diagram outline of the IrriSatSMS system (after Car et al., 2012).

IrriSatSMS was an aDSS that used semi-automated processed satellite data and automatically collected local weather data to estimate daily crop water use. That information, along with farmer-collected rainfall and irrigation values, was then used to run a daily timestep waterbalance model, results from which were sent to the farmer daily via mobile phone Short Messaging Service (SMS) (Car et al., 2012) (see Figure 2). The results gave an objective indicator of tactical (day-to-day) potential crop water requirements at the field scale. They indicated to the farmer how long to run his pumps for in order to return his crop’s waterbalance to zero; advice he could choose to follow directly or to use as a potential only, varying applied water around the figures as desired to achieve a particular crop outcome, such as stressed red grapes for wine quality. The system was able to be operated by farmers entirely via SMS and was scalable to many hundreds of farms with low increasing system cost.

2.1. Attempted Commercialisation of IrriSatSMS

After IrriSatSMS showed good uptake results in trial seasons (Car et al., 2012), the research body mostly responsible for its development, the Commonwealth Scientific and Industrial Research Organisation (CSIRO), attempted to commercialise it in order that it might be run continuously for Australian irrigators. This is a normal process both for the CSIRO and for the multi-agency collaboration that funded its development, the Cooperative Research Centre for Irrigation Futures (CRC-IF), both of which have been involved with numerous commercialisation projects in different research areas.² This author was involved in the following steps that were undertaken to assist with

---

²It is recognised that commercialisation of an aDSS is not the only possible outcome criterion for success. For example, influencing behaviour at a large scale could be achieved through user learning, not necessarily by selling licences for, or subscriptions to, systems; however, for a system that provides watering advice on current atmospheric and crop conditions, continuous operation is required for users to gain benefits, thus the system must outlive limited research funding time-spans
commercialisation:

- Generation of a report aiming to detail “the conceptual framework and the practical elements that need to be assembled to make such a service operational” (Hornbuckle et al., 2009);
- Personnel resource allocation to improve the IrriSatSMS software for continued operation;
- Open Source licencing of the IrriSatSMS software;
- Engagement (travel to and conversation with) agricultural service providers to transfer knowledge of IrriSatSMS operations;
- Arranging a small start-up grant to be given to assist the targeted commercialising partner with taking on new skills.

The report lists three main elements to be assembled in order to operationalise IrriSatSMS (Hornbuckle et al., 2009) and they are:

1. **Data sourcing** – getting the satellite, weather and farmer data the system needs to run;
2. **Server systems** – writing the IT systems that do the work;
3. **Verification** – of IrriSatSMS results by comparison with measured on-ground results.

In 2010, the CRC-IF entered into negotiations with a single community agricultural information service provider to commercialise IrriSatSMS (CRC for Irrigation Futures, 2010). Care was taken to describe the CRC-IF’s processes for operating the system in order to inform the would-be commercialiser of the tasks they were likely to encounter.

### 2.2. Reasons for failed commercialisation

Despite the steps listed above, commercialisation of IrriSatSMS failed with the single commercialising partner not replicating the research agency infrastructure or offering the service. This was due to a range of factors which I list from memory, having been involved, and from recent interviews conducted for this paper with the researchers and the commercialising partner.

As seen from the research agency side\(^3\), the issues were:

- investment required versus uncertain income;
- availability of commercial partner’s skilled (GIS) personnel for data processing.

The first issue is a risk borne by any organisation adopting a new research product and is somewhat unavoidable but certainly may be reduced by cheap deployment costs (which became a design goal of the successor aDSS, IrriSat (see Section 3.2)). The second may be somewhat commercialisation-partner specific but, from this author’s interactions with agricultural bodies, it is widespread: agricultural service providers, such as agronomists and consulting companies, do not typically have either spatial informatics or high-level IT training. Dependence of service providers on these skills for DSS delivery may be greatly reduced via system design. Figure 5 lists design choices that specifically cater for GIS skill-dependence. Additionally, IT skills in the agricultural service and thus commercialisation, of some form, is necessary for the success of this aDSS.

---

\(^3\)Personal Communication with Christen, E.W. in March 2016. He was the senior CSIRO research scientist involved in the project.
provision sector are growing alongside such skills in the agricultural sector as a whole (see Section 4).

It is also somewhat avoidable through technological means where the specialist knowledge required to train others in a tool’s use may be reduced. This is also a design goal of the new IrriSat (see Section 3.2) and for the specific issue faced by this commercialising partner relating to GIS data processing, see Figure 5.

From the point of view of the commercialising partner⁴, the major issues were:

- IrriSatSMS’s lack of a polished user interface;
  - the research project’s interface was sufficient for use but not perfect;
- availability of staff for user engagement and usage training;
- lack of support for the commercialising partner from government farmer liaison bodies.

The first issue above is not uncommon for any new technological system. Subsection 3.2 describes how the User Interface (UI) of IrriSat has evolved from that of IrriSatSMS to address this issue and Subsections 4.1 & 4.2 discuss changes in users’ acceptance of IT, especially their familiarity with the sorts of UIs relevant to aDSS such as IrriSat. The second may appear not to be avoidable being, seemingly, a factor of the specific commercialising partner’s business but in fact it too, as above, can at least be partially addressed by a DSS design that reduces required training for use (as per Subsection 3.2) and through growing users’ familiarity with relevant technologies (Section 4). The third issue from the commercialising partner’s point of view can only be addressed by government and industry collaboration and this has happened with IrriSatSMS’s successor (see Subsection 3.4).

This author, who was the research agency’s DSS engineer for IrriSatSMS, recalls the reasons for the failed commercialisation being:

- investment required versus uncertain income;
  - uncertainty around system costs and operations due to lack of experience in the required IT fields;
- unwillingness on the behalf of commercialising partners’ staff to adopt new tools (mostly IT) and approaches towards service delivery to farmers.

Of the two issues identified by this author, the first has been dealt with above and the second, once again, is at least partly addressed by changing use of technology in the agriculture sector (see Section 4) which affects service providers in it as well as farmers.

3. A New aDSS Design

3.1. Successor Design Goals

In mid-2014, a successor project to IrriSatSMS was initiated by the Cotton Research and Development Corporation (CRDC)⁵. The publicly stated design goals of this successor system (Hornbuckle & Car, 2013) were to:

---

⁴Personal Communication with Argus, S. in March 2016, formerly principal of the commercialising partner.

⁵http://www.crdc.com.au
• implement a prototype operational aDSS for cotton growers in Australia based on the IrriSatSMS system with 100+ farmers;
• use scalable technologies for the aDSS to allow for easy growth from small numbers of users to large, potentially all the cotton growers of Australia;
• use the latest satellite remote sensing products to generate field-specific crop waterbalance advice;
• provide forecast waterbalance advice;
• deliver advice to farmers via SmartPhone applications.

Most of these goals were simple updates to the design goals of the IrriSatSMS system from which IrriSat was derived; however, from the start of the IrriSat system build, technological attempts were also made to address some of the issues with IrriSatSMS commercialisation. Those issues, and the IrriSat system design features that attempt to address them are related in Figure 5. In addition, the institutional arrangements delivering the new IrriSat aDSS also attempted to overcome some of the other issues with IrriSatSMS commercialisation. They are related in Figure 6.

In building the aDSS for a specific industry group, the CRDC, its scope was limited to a particular crop type, cotton; however, only the last of several equations applied to satellite imagery to generate crop coefficients are cotton-specific; all other elements would remain the same for other crops. The cotton-specific equation details are given in Montgomery (2015).

3.2. IrriSat System design

![Diagram of IrriSat system](image)

**Figure 2:** A system diagram outline of the IrriSat system (after Vleeshouwer, Car and Hornbuckle, 2015).

At the highest level, Figure 2 represents architecture of IrriSat as well as IrriSatSMS with just a single change needed: the block “SMS Messages” needs to read “SmartPhone Messages” which indicates that, from the data flow perspective, the systems are functionally equivalent albeit with the update in mobile phone delivery from SMS to SmartPhone application. However, they operate very differently from a systems manager’s point of view. As listed in the previous Section, public cloud-based infrastructure is used for almost the entire IrriSat system. Specifically, Google Earth
Engine (GEE)\(^6\) is used for fully automated satellite data acquisition and processing, in place of the previous IrriSatSMS semi-manual, semi-desktop processing workflow, and the Google App Engine (GAE)\(^7\) is used for waterbalance calculations and waterbalance data storage (Vleeshouwer, Car and Hornbuckle, 2015) in place of IrriSatSMS’s use of a virtual server. Both the GEE and GAE applications are fully automated and auto-scale with only service fees changing, meaning that non-erroneous processing one or one thousand users’ waterbalances and the requisite satellite data requires precisely the same DSS management effort: none. The use of GEE also allows IrriSat to access and blend imagery from several LANDSAT satellites (currently 7 & 8 and historically 5 & 7), not all of which were available at the time of IrriSatSMS’ creation (8). Also, GEE will acquire new imagery over time, meaning that IrriSat’s base data will grow in volume, acquisition frequency and likely precision over time with continual minor changes required in order to leverage new data.

The use of public cloud infrastructure for the entire IrriSat system also means that there is no need for commercialising partners to implement a clone of the infrastructure used by the aDSS developers to operationalise the aDSS: they can either take direct ownership of the aDSS and its infrastructure, since it’s not within a private organisation and the “keys can be handed over”, or they can duplicate the system and infrastructure within the same public cloud almost with the “click of a button” as many public clouds, including the relevant ones for IrriSat, offer this sort of replication.

3.3. User Interface

IrriSat presents a web-based interface to managers and users that is conceptually similar to that presented by IrriSatSMS but with the addition of new features provided by the IrriSat architecture and support for a much greater range of user actions. It allows users to enter rainfall and irrigation values for multiple crop fields and displays waterbalance traces for them generated using those inputs and crop evaporation as per Vleeshouwer, Car and Hornbuckle (2015). Figure 3 shows some images of its web UI. Some similar images of IrriSatSMS’s web interface are show in in Figure 4 for comparison.

\(^6\)https://earthengine.google.com/
\(^7\)https://appengine.google.com/start
Figure 3: Three parts of the IrriSat web UI. Clockwise from top left: a user’s new test field marked out ready for analysis; the waterbalance graph of the test field; the test field’s crop coefficient taken from blended satellite imagery (LANDSATs 7 & 8).

Figure 4: Three parts of the IrriSatSMS web UI. Clockwise from top left: a crop’s waterbalance trace; past irrigation (i) and rainfall (r) data inputs for block ’a’; the web form for adding irrigations and rainfall via the web UI.
Compared with IrriSatSMS’ web UI, IrriSat’s is ‘polished’ with better aesthetics and operations such as input validation within form elements that catch incorrect values, such as impossibly large rainfall readings. It also provides for a wider range of user actions; for example, due to the use of map drawing toolkits, the IrriSat UI allows users to mark out their own fields and save them for analysis, as shown in Figure 3. For IrriSatSMS previously, field marking had to be done by a project staff member and loaded into the system for the irrigator. This did not allow irrigators to change their fields easily. Due to IrriSat’s cloud-based infrastructure, it is able to recalculate the waterbalances for a user’s fields after each data entry of rainfall or irrigation on the fly, meaning the user sees instantaneous changes in the waterbalance graph.

Changes like these do not affect the core operations of the crop waterbalance but greatly enhance the user experience of interacting with the system. The SMS interface present in IrriSatSMS was not continued in IrriSat due to the familiarity of intended users with SmartPhones and Internet interfaces (see Section 4). Initially, IrriSat planned on releasing a Smartphone mobile UI (Vleeshouwer, Car and Hornbuckle, 2015) for irrigators to use in the field; however, the web UI is thought to be sufficient for the first few season’s testing thus a Smartphone UI may be developed at a later date.

The improvements in IrriSat’s web UI over IrriSatSMS’ are due to three things:

1. continued investment in aDSS research and development;
2. growing experience by the developers with particular aDSS UI issues;
3. changing web technologies allowing for more UI options at lower cost.

IrriSat UI improvements, where relevant to issues with IrriSatSMS’ failed commercialisation, are given in Figure 5.

**Figure 5:** Issues with IrriSatSMS commercialisation and the corresponding IrriSat design feature attempting to deal with it.

<table>
<thead>
<tr>
<th>Issue</th>
<th>Design Feature</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Investment uncertainty versus uncertain income</td>
<td>Auto-scaling infrastructure</td>
<td>Reduces installation costs to zero; reduces running costs to as-needed</td>
</tr>
<tr>
<td>Availability of commercial partner’s skilled (GIS) personnel for data processing</td>
<td>Fully automated satellite data processing</td>
<td>IrriSat automatically blends imagery from a range of satellites (^8), removing cloud effects and calculates crop coefficients from NDVI (^9). No manual processing is required thus no staff investment required</td>
</tr>
<tr>
<td>Lack of a polished user interface</td>
<td>Polished UI</td>
<td>Compared with the web &amp; SMS UIs of IrriSatSMS, IrriSat UI allows for better input validation, more functions and</td>
</tr>
</tbody>
</table>

---

\(^8\)All these as available in Google Earth Engine: LANDSAT 5, 7 & 8 as well as MODIS Terra

\(^9\)Normalised Difference Vegetation Index, see (Vleeshouwer, Car and Hornbuckle, 2015)
3.4. IrriSat institutional arrangements

Unlike IrriSatSMS, which was built by a research agency as a proto-operational, proof-of-concept system (with high expectations of an easy transition to operations that were never realised), IrriSat was built for an agricultural community agency, the CRDC\textsuperscript{10} that intended, right from the start of the project, to operationalise it (Hornbuckle & Car, 2013). This meant that IrriSat would not need to go through a quasi-tender process for commercialisation upon project completion. However, it would/may still have to go through a technological transition (see below). In working with an industry association that is able to commercialise tools right from project inception, IrriSat is following the commercialisation option Case Study presented in Hornbuckle et al. (2009) named By Government Institution. Figure 6 lists the major institutional arrangements around IrriSat – both those related to the commercialising partner and others – and the IrriSatSMS commercialisation issues they address.

**Figure 6:** Issues with IrriSatSMS commercialisation and the corresponding IrriSat institutional arrangement attempting to deal with it.

<table>
<thead>
<tr>
<th>Issue</th>
<th>Organisational Arrangement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability of commercial partner’s staff for user engagement and usage training</td>
<td>Partnership with a commercialising partner over many years of product development enabling skills build-up if necessary</td>
</tr>
<tr>
<td>Lack of support for the commercialising partner from government farmer liaison bodies</td>
<td>Partnership with a government-assisted commercialising partner agency</td>
</tr>
<tr>
<td>Unwillingness on the behalf of commercialising partners’ staff to adopt new tools (mostly IT) and approaches towards service delivery to farmers</td>
<td>As per first point above. Additionally, the CRDC is an agency that, due to government support, has a longer horizon than a small enterprise</td>
</tr>
</tbody>
</table>

\textsuperscript{10}http://www.crdc.com.au
4. The Changing Australian Farming Technology Landscape

High levels of Internet access and SmartPhone ownership, detailed in the following sections, mean that virtually all Australians are now familiar with the Internet, web pages and the other IT tools used for the user interfaces of aDSS like IrriSatSMS & IrriSat.

4.1. Changing Internet access

Figure 7: Proportion of Australian rural households with internet access (Australian Bureau of Statistics 2009, 2011, 2016 & 2016).

Since the first field trials of IrriSatSMS in 2009, the percentage of Australian rural households with Internet access has increased from 65 to 82.2 in 2016, see Figure 7. While these figures for rural households are approximately 10% lower than for urban households, Internet access all over Australia is high and appears to have reached saturation levels with virtually no growth from 2012 to 2014. These high penetration rates mean that designers of aDSS for the Australian market can rely on the Internet as a delivery mechanism for most of their potential users.

4.2. Changing mobile technology ownership

Figure 8: Proportion of Australian mobile phones that were SmartPhones (Mackay, 2014).

Australia has some of the highest SmartPhone ownership rates in the world with 74% of adults
owning at least one in 2015 (Australian Communications and Media Authority, 2015). Additionally, the percentage of mobile phones in Australia that are SmartPhones is now greater than 89% (Mackay, 2014, p.9). Importantly for IrriSatSMS & IrriSat, Smartphone penetration as a percentage of total mobile phones in Australia grew rapidly from around 25% when IrriSatSMS was first field-trialled in 2008 to around 60% at its last trial in late 2009 (see Figure 8). The SMS interface for IrriSatSMS was built specifically to cater for irrigators who had mobile phones but not SmartPhones.

4.3. Changing data availability

While not listed in sections above as a primary issue for IrriSatSMS’ commercialisation, the ability to access quality data required for its waterbalance calculations in particular geographical areas was a real one noted by this author. The trial seasons for IrriSatSMS were not limited by satellite data access as the particular datasets used – LANDSAT 5 & 7 – covered all of Australia, but were confined to zones of uniform weather types around individual weatherstations. In 2007 at IrriSatSMS’ inception, there were three major sources of weatherstation data in Australia: 1. the Australian Bureau of Meteorology’s (BoM) national observations network; 2. a few dedicated agricultural weatherstations maintained by state departments for agriculture and primary industries; and 3. weatherstations supplied by CSIRO, the IrriSatSMS developing agency, for IrriSatSMS. Data from the BoM weatherstations was unsuitable for the waterbalance calculations due to differences in evaporation calculation techniques and data from all of the departmental stations was impossibly to access in a timely fashion due to restrictive IT policies preventing its automated release. This meant that all IrriSatSMS trials relied on CSIRO-supplied weatherstations.

The range of weatherstation data sources in the cotton-growing areas of New South Wales, Australia, in 2014 able to be used in the trialling of IrriSat was much increased from those available to IrriSatSMS in 2007. In 2014, data could be automatically collected from: 1. expanded state departments of agriculture weatherstation networks; 2. multiple private weatherstation networks; and 3. IrriSat-dedicated CSIRO stations. Additionally, the BoM is looking to produce a nation-wide gridded data evaporation product that will likely be available as IrriSat trials conclude.

Data from these weatherstations can be automatically and reliably collected due to a change in general web development which emphasises the delivery of data and function delivery via Application Programming Interfaces (APIs) that then power human-readable web UIs over web UI-only delivery. Web pages made in this way make the data they display available for access independently of the particular web UI they deliver. The IrriSat system is able to pull in data from many stations not managed by CSIRO using direct API access. This expands the possible IrriSat application areas at very low (almost zero) cost as opposed to expansion of IrriSatSMS which required the placement of a custom-built weatherstation. Figures 9 and 10 show human-readable and machine-readable versions of the same weatherstation’s web page respectively.


12See this from 2011 for an in-depth discussion of the issue as seen during the rise of APIs: https://code.tutsplus.com/articles/the-increasing-importance-of-apis-in-web-development--net-223
The number of weatherstations which publish data to the web in any form has grown dramatically with a decrease in the cost of mobile phone 3G and 4G technologies that allow connected devices to use the HyperText Transfer Protocol (HTTP), and in the cost of the physical weatherstation electronic components. In 2007, each CSIRO station used a fixed landline phone connection for data transfers that cost more than $AU300 per month. Telecommunications costs reduced to about 10% of that in 2008 with the arrival of a widespread 3G carrier\textsuperscript{13}. Current monthly mobile phone charges for CSIRO’s weatherstations are approximately $AU15 per month; 5% of the 2007 charges.

The cost of automated weatherstation’s hardware, although hard to quantify exactly, has clearly reduced significantly from 2007 to the present. While ultra-cheap ‘hobby’ units now exist for around $AU100, stations rugged enough and containing components of sufficient quality to be useful for farm applications, and which may publish information to the web, are generally available now for a

\textsuperscript{13}Telstra’s NextG network. The current NextG / 4G network coverage for Australia can be found at https://www.telstra.com.au/coverage-networks/our-coverage
few thousand dollars which appears to be a reduction of perhaps 80% since 2007. Typically, weatherstations are sold as part of a ‘solution’ which includes data management with information stored on servers and accessed via web pages.

The commensurability of evaporation data from weatherstations with Irrisat’s equations is still an issue, however, as only calculations made using the ASCE’s standardised reference equations (Walter, 2001) are usable and these are not implemented in all systems; many still use those equations’ precursors such as the FAO56 equation (Allen et al., 1998).

![Data from the machine-readable API for the Lyrup Flats weatherstation corresponding to Figure 3](http://aws.naturalresources.sa.gov.au/api/data/?timestep=minutes&station_ids=RMPW05&start_date=2015-04-17&end_date=2015-04-17)

**Figure 10:** Data from the machine-readable API for the Lyrup Flats weatherstation corresponding to Figure 3 from http://aws.naturalresources.sa.gov.au/api/data/?timestep=minutes&station_ids=RMPW05&start_date=2015-04-17&end_date=2015-04-17.

The centralisation of a large LANDSAT archive in Google Earth Engine’s data repository means the entire archive of such imagery for all of Australia, and updates close to when they are acquired, are available for use via the GEE API. This simple access drastically reduces effort, cost and system complexity for the DSS designer. Competitor systems to the GEE, such as Geoscience Australia’s Australian Geospatial Data Cube promise to deliver a greater variety of satellite imagery via a single API. This would enhance the spatial and temporal resolution of systems such as IrriSat.

A growth in the range of types of data sources available to farmers not directly relevant to IrriSat has also occurred in line with cheapening electronics and information technology, too. For example, many companies now offer Australian farmers soil moisture sensor networks that are affordable by small family enterprises where once such offerings were very expensive and affordable for large corporate farms only.

---


15Measurement Engineering Australia’s ‘Plexus’ (http://mea.com.au/soil-plants-climate/soil-moisture-monitoring/plexus) is a low cost soil moisture sensor system
4.4. Changing information systems use in agriculture

The high levels of internet connectivity across Australia and the high availability of web and Internet technologies, as well as cost reductions in electronics and computer systems more generally, have led to a rapid rise in the use of information systems for agriculture. However, there has also been a substantial rise in the use of non-agricultural information systems in the Australian populous generally, within which Australian farmers must operate. For example, until recently, all major Australian banks offered online banking to supplement in-person banking. Now, online banking is the norm with banks offering either higher fees for in-person banking or not offering it at all. Along with banking, many other financial operations are now required to be carried out online, such as insurance, superannuation, investments etc. and farmers must use these tools just like everyone else. This means they are used to a wide range of information systems use in the non-agricultural parts of their lives. There is much crossover between agricultural and non-agricultural information systems with government and many companies offering online tools for agricultural commodity markets and similar\textsuperscript{16}.

5. Performance of IrriSat

IrriSat’s first version was completed in time for use over the 2015/16 Australian cotton season which is approximately September 2015 to May 2016. Workshops with target users (cotton irrigators and consultants to them) were held in 2015 to promote the system and system sign-ons were accepted from about September, 2015.

5.1. Usability

Once a field is marked out via the IrriSat user interface, the satellite imagery for it going back one year from the present is accessed and processed generating a timeseries of crop coefficients “on the fly” (as the user waits). If irrigation and rainfall events are then added, a crop waterbalance graph can be drawn. These steps are shown in Figure 3. It takes less than a minute for the field marking and satellite data access and processing.

In farmers workshops in 2015, no major issues were discovered with farmers’ use of the user interface. It appeared that the relative simplicity and responsiveness of the UI, key design goals (see Figure 5), mean some problems encountered with IrriSatSMS use did not appear to arise. For example, errors in irrigation and rainfall data entry – the main source of IrriSatSMS’ waterbalances errors – can be seen on the nearly instantaneously-generated waterbalance chart and quickly corrected, all within a single usage session with no need for a support request cycle. With IrriSatSMS, erroneous entries were often only seen after an overnight waterbalance calculation run and then a support request from user to administrator was required to resolve it. This new mode of operations both reduces user frustration and reduces the requirements for system support.

By mid cotton season 2015/16, over 300 users were signed up to use IrriSat and had registering between one and several hundred fields each. Due to IrriSat’s scalable cloud-based architecture, it performs with the same service level regardless of the number of users so that those with hundreds of fields are able to click on them and generate waterbalance charts in the manner described above

just as easily as those with one field. No issues relating to high loads on the website due to multiple simultaneous users have been observed, either.

The effort required for irrigators, or consultants on behalf of irrigators, to input rainfall and irrigation records for each field was a problem for IrriSatSMS. It was partially solved with a very simple SMS-based input method (see Car et al., 2012) and limiting irrigators to one field per person. For IrriSat, the SMS input option has been removed and the limit of one field per person too. Over the course of the 2015/16 season, several methods for easing the effort of data input to IrriSat have been tried. While automated rainfall capture from regional weather stations is not appropriate due to Australian summer rainfall patterns being very patchy, readings from in-field automated rain gauges are able to be used. For irrigations, many cotton growers’ irrigation schedules are planned long in advance due to irrigation water ability and thus per-field dates and volumes exist in spreadsheets that are able to be bulk-loaded into the system. Finally, the IrriSat team has worked with third party agricultural service providers delivering scheduling advice based on methods other than evaporation, such as soil moisture probes, that already collect irrigation and rainfall information in order to feed data captures into IrriSat without further manual effort from the farmer. So far, this approach has proved to be popular with farmers who claim to be “overloaded with apps, websites & information”^{17}.

5.2. System autonomy

After initial system release for full user access in September 2015, some refinements were made to assist users with data capture. Examples are the ability for the system to accept uploads of Google Earth KML files^{18} marking fields’ boundaries, rather than requiring users to make them within the IrriSat UI only, and to offer weather station data choices, rather than fixing weather data for fields to the closest station only.

Apart from these changes, little technical engineering work has needed to be undertaken on the system and certainly none regarding day-to-day operations or system scaling. This is attributed to mature system design over many years’ worth of iterations and comprehensive laboratory testing over the 2014/15 cotton season by experienced aDSS staff and state departmental extension officers.

5.3. System cost

At the time of writing, GEE is free to use for “trusted users” and thus the project incurs no cost for using it. This is likely to change as GEE matures. Google App Engine (GAE) costs are free for small use and priced as per https://cloud.google.com/appengine/pricing for larger use. At this stage, IrriSat has mostly operated within the free quota but has just, at peak usage for several months, incurred a small cost of less than $AU100 per month to run.

---

^{17}Personal Communication with Montgomery, J. in April 2016. She is a New South Wales Department of Primary Industries extension officer conducting workshops on IrriSat usage with farmers.

^{18}https://developers.google.com/kml/
6. Discussion

6.1. Addressing the inherited issues

When testing IrriSatSMS, the author was conscious of it being another technical aDSS that could easily fail to achieve widespread implementation due to reasons such as the inapplicability of scientific models to farming practice noted in papers such as McCown (2006) or the narrowness of ‘technicentric’ solutions to farm system management (Matthews, 2008). This was nevertheless believed to be worthwhile since the tool trial was not just attempting to leverage new technologies that enhanced the accuracy of the scientific models, such as field-scale crop coefficient readings from satellite imagery, but also technology that fundamentally changed the way the farmers could interact with the aDSS. The SMS messages used by IrriSatSMS really did work well with farmers’ work patterns (Car et al., 2012) and we saw real promise in the system’s future.

With the failure of IrriSatSMS’s commercialisation, described in Section 2 of this paper, it is seen that it was ultimately issues other than the efficacy of IrriSatSMS as an aDSS that contributed to its demise. When designing IrriSat this knowledge was retained and thus IrriSat work didn’t just look to improve aDSS technical capacity but also looked to directly address some of those reasons for its commercialisation failure. The technical and institutional arrangements listed in Figures 5 & 6 respectively make some good inroads into addressing those failure reasons.

6.2. New challenges

Based on the good trial performance of IrriSatSMS and the deliberate attempt to overcome its commercialisation problems with IrriSat’s design, the author feels that some amount of success is likely for the newer system. However, it is now clear from an understanding of the recent rate of farm usage change of technology relevant to IrriSatSMS & IrriSat, as described in Section 4, that what were once problems for farmers and agricultural service providers are no longer problems and that there are now new problems. Where the reason that lead us to drop support for SMS in the new IrriSat – that of farmers having smartphones and good Internet connectivity and familiarity with using them – has proved a boon regarding the reliance that can be placed on farmers in Australia being able to use advanced web-based UIs, it has also meant that they have become swamped with many competing decision support offerings; farmers, like people in other sectors of society, are facing a data deluge. Farmers’ recent greatly enhanced connectivity, and the rapidly increasing array of interoperable local and remote data sources available to them (see Subsection 4.3), both add possibilities for IrriSat but also threaten its viability as users become overwhelmed with multiple information products competing for their attention.

One attempt that the IrriSat team have recently undertaken to address this issue is to integrate some part of IrriSat’s decision advice into another information tool, a commercially-provided weatherstation and soil moisture probe system, already used by cotton growers. This is to reduce the number of systems they need to interact with daily for, while responses to IrriSat’s interface in the workshop sessions mentioned in Section 5 were positive, workshop participants also indicated that they were “overloaded with apps, websites and information”\(^{19}\). This was not a complaint the

\(^{19}\)Personal Communication with McIntosh, J. in March 2016, a New South Wales Department of Primary Industries extension officer working with IrriSat.
This author recalls a time, around 2007, when social network users were bombarded with calls from friends to join multiple social network sites such as Bebo, MySpace Friendster and Facebook that each offered slightly different features to users. The sudden rise in popularity of these sites was due to increased internet connectivity among users (teenagers and young adults) and some technology changes making interactive web pages more usable. To use all of those different but competing social networks in parallel was not possible for most people and now, in 2016, of Bebo, MySpace Friendster and Facebook, only Facebook retains large numbers of users. It is possible that potential users of IrriSat will see it competing directly with other systems for their attention, even if these other systems and IrriSat do not offer exactly the same features and that, ultimately, they may choose to use IrriSat or another system, but not both, due to the time commitments for effective use involved. This could mean that IrriSat is pitched against, perhaps, soil moisture probe systems, even though IrriSat’s advice and that of the probes does not conceptually overlap exactly.

One aspect of Facebook’s emergence as the winner of the social network competition that is worth noting here is that, since its dominance, Facebook has added many new features and is able in 2016, for instance, to support fan pages for music artists and home page customisations, once main boasts of MySpace over its rivals. If IrriSat is out-competed for farmers’ attention by an application not directly offering IrriSat’s utility (that of locally customised, evapotranspiration-based waterbalance modelling that can enhance water use efficiency), it is possible that IrriSat’s value offering may eventually be accredited by that other system. This is akin to the integration of IrriSat’s advice into another tool, as was mentioned as being trialled above. This mode of commercialisation was not explicitly indicated in Hornbuckle et al. (2009). The system design of IrriSat is such that it would require very little effort on the IrriSat side for it to be integrated into another system’s UI. This is because IrriSat makes use of APIs between system components, as described in relation to weather information in Figure 10.

6.3. Alternative pathways to on-going operations

In this paper I have already noted that, as part of the attempts to commercialise IrriSatSMS, a report was written aiming to detail “the conceptual framework and the practical elements that need to be assembled to make such a service operational” (Hornbuckle et al., 2009) and that it contained several potential commercialisation “Case Studies” outlining several different ways IrriSatSMS could be provided to farmers in an on-going fashion. Several of those Case Studies considered pathways to ongoing IrriSatSMS use that did not rely on users of the system perceiving enough utility in the system for their own water-scheduling needs for them to pay subscription costs for its use. Several of the alternate pathways included on-going sponsorship by government, water supply institution or farmers’ organisation for IrriSatSMS operational establishment and on-going costs, thus enabling free tool use for farmers. It was thought that such pathways would be viable given the obvious potential for those organisations to benefit from collecting field-scale farmer watering data in near real-time. This commercialisation pathway now being pursued for IrriSat via its Cotton Research and Development Corporation (CRDC) sponsorship is perhaps not new in the aDSS world but it is very much in line with the business models of many Internet ‘apps’ made freely available to end users under the strategy of harvesting their usage data for on-selling as the true product of the system.
This strategy achieved some notoriety from around 2010 onwards and has been associated with the catch-cry of “If you are not paying for it, you’re not the customer; you’re the product being sold.”²⁰ By following this strategy for commercialisation, then, IrriSat’s future custodians may need to work hard to convince end users that tool use is truly in their interests.

7. Conclusions

In this paper I have reflected on the failure of one aDSS commercialisation and related the design of its successor system to reduce the chances of future repeated commercialisation failure. I have also described the changing nature of the use of technology relevant to these aDSS on farms and the availability of relevant data.

Design choices were able to be made with the availability of new IT tools that have substantially reduced the risks involved for commercialisation partners in commercialising certain types of aDSS. Also, changes in technology use on farms substantially alter the environment in which aDSS of IrriSat’s sort now operate and this means that some of the challenges faced by systems in the past are no longer relevant while new challenges are.

Noting that some of the new challenges we believe face aDSS like IrriSat, such as similar systems competing for users’ attention, have been encountered before in other non-agricultural fields, the author believes that IrriSat’s ability to overcome some of those new challenges is as unknown to us now as the eventual success of Facebook was to everyone in 2008. As much flexibility, as possible, has been built into the IrriSat system in order to enable a number of deployment options which, it is hoped, positions it well for these new challenges.

The experience I report here points to a further challenge to overcoming “technicentric” approaches to commercialising innovations: significant and rapid changes in usage context imply that identifying the usage context, by using focus groups or representative farmers, is only a starting point. It is also necessary to track changes in the usage context as they occur and, further, to project changes to the value of characteristics of innovations (and possibilities such as bundling of characteristics with other products) in ways that farmers probably cannot. The more substantial or prompt are related innovations, the more difficult will this tracking of, and optimisation of, innovation value be, with implications for effective management of commercialisation.

Acknowledgements

Cotton Research and development Corporation and the NSW Department of Primary Industries: for coordinating and running interactions with the trial users of this system.

Cotton farmers of northern NSW and southern Queensland: for their willingness to participate in trials, even when the return on their time spent wasn’t evident.

Finally, Jamie Vleeshouwer: for his excellent front and back-end software engineering that actually implemented the recent IrriSat system.

²⁰Some of the first Internet chat to include this phrase is archived at http://www.metafilter.com/95152/Userdriven-discontent#3256046
References


Hornbuckle J W, Car N J, Christen E W, Stein T M and Williamson B. IrriSatSMS - Irrigation water
management by satellite and SMS - A utilisation frame- work. Science Report 04/09, CSIRO Land & Water, CSIRO Land and Water, Griffith, NSW, Australia, Apr 2009. URL (archived copy)
This chapter relates a series of investigations undertaken into irrigation DSS enhancement through data source integration. Unlike the previous and following chapters, this chapter doesn’t have a corresponding major journal publication however its work has been published in a series of six conference papers:


This chapter is of importance to this thesis since it relates part of the journey of this PhD to questions initially posed for it. It then shows how work for those questions was instrumental in developing the subsequent PhD focus on decision modelling. The work on User-Defined Data (See 6.4) lead to investigations into flexible user interfaces (Car et. al 2008) and finally Case-Based Reasoning (Car and G. A. Moore 2011) for which the decision modelling in Chapter 7 was conducted.
Work in this Chapter falls within this thesis's research topic *Customisation (of DSS)*.

### 5.1. Motivation

One of many issues noted by authors reflecting on the poor uptake of agricultural DSS in Australia in the early years of the 21st century was the poor relevance to real-world practice of the data they used. Authors such as McCown (McCown 2001), and at the start of this PhD, Nguyen (Nguyen, Wegener, and Russell 2007), noted that DSS need to be situation-specific to in order to be accurate and be trusted by users yet many DSS use generic data such as crop variety rather than specific crop metrics or area average water applications rather than individual fields’ values.

While McCown primarily addressed research and DSS development methodologies to try and design DSS that better fitted within farming practice and also to lead farmers on a DSS development journey that they would own, Nguyen noted “...intergenerational change that is occurring in the management of Australian farms is a positive factor that may encourage more widespread use of these tools [DSS]”. By this he meant irrigators fearful of computerised systems like DSS were being replaced by those not fearful.

Regardless of how accepting new irrigators might be of computerised DSS, the DSS must still supply accurate and locally-relevant results to be of use. As per Chapter 1, Section 1.3.5, the Internet and mobile phones, when coupled with irrigators not fearful of computerised systems have made many more tailor able DSS configurations possible: we can now use ubiquitous data transfer mechanics (the Internet) with remote and local data sources and send customised results to irrigators who have the tech savvy in order to use them.

Despite these advances, there remains an on-going challenge to make DSS results more locally specific, more relevant. We see this in the survey results within the paper represented in Chapter 2 where some irrigators place a lot of weight on particular, local, systems that, ideally, would integrate with any DSS like IrriSatSMS.

We need to also acknowledge that the irrigation DSS experiments carried out in Chapters 2 & 3 and that likely the wider the variety of situations any particular DSS attempts to address, the more local customisation it may come to rely on in order to be locally accurate.

### 5.2. Initial work and topic changes

Early work in this PhD sought to classify data sources by “network paradigm”, the way they were *network technically*\(^1\) able to be accessed by a DSS, given that network/Internet connectivity was growing rapidly at the time. This lead to the publication of *Towards a new generation of Irrigation Decision Support Systems – Irrigation Informatics?* (Car et al. 2007) (Also

---

\(^1\) This awkward phrase means by physical communications channel access, i.e. some sort of wires or wireless data transfer network. This reflects the author’s initial electrical engineering training.
Appendix B). This work, while apparently of interest to educators of undergraduate engineers\textsuperscript{2} did not lead to any major ways to improve irrigation DSS data source use.

Next, a research question was formulated in 2008 around the idea of measuring the utility to irrigators of using different evapotranspiration data sources within a waterbalance-based irrigation DSS. The evapotranspiration sources were to be the national, interpolated grids (e.g. SILO, see below), large weatherstation networks (e.g. Lower Murray Water's AWS network, also below) and local weatherstations, set up for the occasion as additions to CSIRO's weatherstation network (also below). The waterbalance system was to be IrriSatSMS or a derivative system.

This work was never started for a number of reasons, the first of which was the time spent on, and ultimate failure of, the MMS-based crop imagery experiment (see Chapter 3, Section 3.4). This derailed work on new data source integration generally as, after it, IrriSatSMS was focussed purely on area and crop expansion (to the Sydney basin and vegetable growers after the initial Griffith NSW area and wine grape growers) given that this seemed to be more likely to succeed. The second was the difficulty in catering for different methods of measuring evapotranspiration. Multiple algorithms may be used to calculate evapotranspiration (see (Meyer 1999) for the algorithm used by IrriSatSMS and (Barton and Meyer 2007) for a comparison of multiple methods) and not all systems reporting evapotranspiration figures make explicitly which calculation they are using. This realisation lead to two outcomes: the first was a conference paper presented that was critical of the Australian AWRIS national water data system (see below)'s efforts to represent aspects of water data crucial for its agricultural use using evapotranspiration as the primary example (Car and G.A. Moore 2011). The second was that this author gained a general understanding of the large breadth of information (metadata) required for effective data use within a DSS where data sources and DSS are not directly designed to be used together. Phrased in hindsight, a holistic observations model is needed to ensure measurement commensurability. Simple things such as time and place and units of measure are not sufficient to indicate exactly what phenomena are observed, of what natural world feature and whether the observation methods are relevant to know (as is the case for \textit{standard} evapotranspiration, given there are multiple standards!).

The initial thesis topic described for work in the area of data sources at the time of this thesis’ conversion from Masters to PhD was \textit{Determine the extent to which XML specifications may be adopted by irrigation data source for their interoperability and ‘discoverability’ over the internet}. Due to the reason of measurement commensurability complexity mentioned above and the expectation that an assessment of “...the extent to which...” would be very difficult (see Appendix B for full details) this topic was changed to being about \textit{User-Defined Data} (see Appendix B). The next section relates investigations into this topic.

\textsuperscript{2} The classification system from this paper has been used in repeated versions of the undergraduate engineering textbook \textit{Engineering your future} (Dowling et al. 2016).
5.3. Newly available data

In addition to new research & development methodologies and generational change among irrigators mentioned above in Section 6.1, the sources of data available for DSS designers and DSS users changed rapidly just before the commencement of this PhD.

Some examples of Internet-based irrigation-relevant data sources newly available online in Australia in 2007 were:

- **SILO**
  - "...an enhanced climate database...[which]...contains Australian climate data from 1889 (current to yesterday), in a number of ready-to-use formats, suitable for research and climate applications." (Department of Science, Information Technology and Innovation 2011)

- **Queryable soil maps**
  - District maps for Murrumbidgee, Coleambally & Murray irrigation areas
    - An online system to deliver these maps was built by this author in 2007 (Hornbuckle 2008) and remained online until 2016
  - National maps
    - “ASRIS provides online access to the best publicly available information on soil and land resources in a consistent format across Australia”

- **Water trading**
  - Online water exchanges were new in 2007. There are now a plethora of these, such as WaterFind, Ruralco Water & H2OX but in 2007 there were few, but at least one exchange was available to irrigators in the Griffith, NSW, area who partook in IrriSatSMS trials.

- **Weatherstation data**
  - CSIRO Land & Water’s weatherstation network
    - A network re-implemented by this author to supply IrriSatSMS with hourly weatherstation data
  - Lower Murray Water’s AWS network
    - A network of stations owned and operated by Lower Murray Water, a regional Australian water supplier partially built by this author
    - There are many weatherstation networks like this in Australia

- **Sensor-measured soil moisture data**

---

3 http://www.asris.csiro.au
4 https://www.waterfind.com.au
5 http://www.ruralcowater.com.au
6 http://h2ox.com
7 http://weather.csiro.au
8 http://www.lmwweatherstations.vic.gov.au
SENTEK and MEA\(^9\), among many soil moisture sensor vendors in Australia, offer the ability for clients to access data from their networks online.

**Crop imagery**

- LANDSAT and other satellite imagery available online, the use of which for IrriSatSMS is described in Chapter 2.
- Mobile-phone-based crop imagery, experiments about the acquisition of which were conducted as part of this PhD. See Chapter 3 & Appendix D.

Note that some of these data sources are remote sources, such as satellite imagery, and some require local infrastructure, such as soil moisture probes. What they all have in common is the use of the Internet to transfer or publish data. Some companies in 2007, and still now, do use non-Internet-based networks for data delivery, such as wired LANs, however the Internet is used very widely and Internet protocols are the only viable option when trying to access irrigation DSS-relevant data from a range of data source makers who never built their tools to be used within a DSS context. This is due to equipment providers and data publishers being able to design for Internet protocols with a high degree of certainty that eventual users will be able to work with them, again due to the Internet's pervasiveness, whatever their interests, DSS or otherwise.

In addition to the sources above, a very large data integration exercise was started then in Australia to centralise water storage, observations and usage data, as well as some weather data, within the Bureau of Meteorology in response to the Australian Millennium Drought of the late 1990s and first decade of the 21st century. The Australian Water Act 2007 (Commonwealth of Australia 2007) and resultant regulation required that Bureau of Meteorology continuously acquire water data from over 200 state and non-government organisations and that they manage it, for which they developed a system known as the Australian Water Resources Information System (AWRIS)\(^10\). This promised to make water data from 200+ water control agencies and companies available in once place in a standard format, the Water Data Transfer Format (WDTF) (Walker et al. 2009), but, as of even 2018, water data such as supply availability is not able to be retrieved from this system (it's not really for this purpose), instead most irrigators deal with water trading organisations for water availability.

While soil moisture sensors were listed above, many more sensors with Internet connectivity became available around 2007 onwards and more appear still: irrigation pumps themselves, piping systems and their statuses, webcams of crops, etc.

### 5.4. User Defined Data

So, given the growing range of Internet-accessible data sources – whether made for DSS or not – how could DSS designers make more of them available as inputs for DSS?

---


Three main avenues of investigation have been undertaken over the course of this PhD to address this:

1. Networks, UIs and other direct technical methods
2. Commensurability methods
3. Decision representation

The first of these is discussed in Subsections 5.4.1, 5.4.2 & 5.4.3, the second in Subsections 5.4.4 and the last in Subsection 5.4.5.

Overall, and despite the considerable work put into this area over the course of the last decade, User-Defined Data itself is not considered a final research topic to have yielded substantial results. Work started within this topic has gone on to yield useful outputs (see Chapter 6 for decision modelling) or are on-going (Subsection 6.4.5 below). Nevertheless, here follows a reiteration of work done within this PhD that is mentioned elsewhere but is summarised to overview the User-Defined Data topic work.

5.4.1. Network paradigms

As already stated in Section 5.2 above, initial work done in this PhD to assist DSS designers cater for a wider range of data sources involved a categorisation of DSS, in present in 2007, according to a “network paradigm” or the way they then connected to sources of data. This work resulted in the publication of a conference paper (Car et al. 2007) which was included in an initial literature review draft of this thesis (see Appendix A, page 16). While that work was never carried further than that within this thesis, it has been included in a series of textbooks for undergraduate engineers to assist them in the way they design systems starting in 2013 and right up until the latest edition in 2016 (Dowling et al. 2016).

The positive elements of that 2007 paper, looking back from 2018, are that it did correctly predict/wish for a so-called User-Defined Data Set that would able to be “…added to the pool of data sources used by the DSS by the unassisted DSS user…” and this is something that is still aimed for today (see Section 6.7 below), so the idea is not incorrect or perhaps unattainable but it is somewhat obvious and it was naïve to suggest it was likely or even possible that a single PhD could bring such a vision into reality.

Another element from that paper that stands the test of time was the idea to use a range of Open Geospatial Consortium standards for sensor description and data transfer, very new at the time, for describing data sources such that a DSS could cater for a series of protocols and then be able to use data sources formulated using them.

What was completely missing from this work was an appreciation of the magnitude of data commensurability issues that need solving before multiple data sources could be used by a DSS, even when the physical network and data format issues are solved. This issue, raised in Section 6.2 is discussed further below in Section 6.4.4.

The author, following advice in this paper, between 2007 and 2009, re-implemented the CSIRO Land & Water weatherstation network of 10+ stations. Although there were cost savings and other practical reasons for doing so (a change to using cellular 3G technology for weatherstation-to-base data communication over land line use saved money and increased weatherstation deployment flexibility enabling the development of IrriSatSMS detailed in
Chapter 2 and newer Internet protocols for base-to-client communications (the use of HTTP, rather than FTP) greatly improved the data access 11), the ultimate goal in network reimplementation was to allow the weatherstation data to be communicated using the then new Sensor Observation Service (SOS) standard (Open Geospatial Consortium 2007). This was in order to establish a set of irrigation DSS input data sources (proved to be such through IrrisatSMS work) in Australia, available freely on the web, using a standardised, machine readable, data access protocol.

This task was ultimately achieved only in 2015 with the development by the author of the aws_pywebsite code base that read from the weatherstations’ database 12. The reason for this considerable delay was the lack of interest by his funding body or anyone else in seeing such work conducted.

The set of data and network standards and the domains they address that were implemented through this work were the protocols TPC & HTTP to pass messages via the Internet (as per all websites), XML for the generic data format and SOS for the sensing system description. By the 2015 implementation, WaterML2.0, a timeseries data exchange format created partly as a result of the WDTF work mentioned above in Section 6.3 (Open Geospatial Consortium 2014), was used within the SOS system for timeseries representation. Inherent in the 2015 SOS installation implemented was also a representation of the observations’ methods, measurements and related objects using the Observations & Measurement (O&M) standard which became available in 2011 (ISO 2011). For further thoughts on the use of O&M, see Section 6.4.4 below.

Further on the stalled implementation of SOS: The use of OGC protocols to standardise online data sources, for DSS or other uses, was attempted around 2007 by at least one irrigation-related equipment manufacturer in Australia, Measurement Engineering Australia Ltd 13. They too, in around 2008, tried to use SOS for representing data streams from soil moisture sensors that they manufacture but also found the work valueless given that no other equipment manufacturers known to them were also attempting this and given that this would only be beneficial if multiple suppliers implemented similar protocols 14.

In 2017, engineers within CSIRO Land & Water within a unit similar to that in which the author’s 2007+ IrrisatSMS work was undertaken, implemented a SOS-based system to present data from a range of sensors from different agricultural equipment manufacturers in a standardised manner. This work is currently the subject of publication (Sommer, Stenson, and Searle 2018) and represents the latest methods employed for agricultural/irrigation heterogeneous data source integration in Australia. This work uses different computer

11 The weatherstation network is still visible with 5 stations, online using web code written partly in 2007, at http://weather.csiro.au.

12 This Python code repository is available online at https://bitbucket.csiro.au/projects/EIS/repos/aws_pywebsite/.

13 http://www.mea.com.au

14 Personal communication with Andrew Skinner, founder of Measurement Engineering Australia, approx. 2009.
programming paradigms to any considered by this author\textsuperscript{15} but is, in essence, still in line with the intentions expressed in the author's early conference paper (Car et al. 2007).

Two thoughts relevant to the thoughts here about this 2017 system implementation were communicated to this author in early 2018:

1. SOS is still not widely used and is perhaps unlikely to ever be used. The author of the 2017 work anticipates implementing the newer OGC SensorThings (Open Geospatial Consortium 2016) standard in its place in the future. This is believed to be due to a number of reasons, the main one being that SensorThings better uses of data representation to make data transfer more efficient which is important for recent data sources producing large amounts of data. Methods other than SOS, like SensorThings, are also seemingly easier than SOS for most web developers to implement.

2. The 2017 work suffered a great deal due to the lack of metadata from vendors describing the properties and features of the environment that their systems observed/measured. Despite the existence of standards for this purpose, such as O&M, none were implemented and much back-end work was needed by the 2017 author to try to make sensor readings commensurate (e.g. soil moisture observed by different manufacturers soil moisture probes doesn't always indicate measurement depth, units used, absolute moisture or % moisture).

The first point indicates that SOS has failed to gain traction and, if an irrigation data source-relevant standard is to be implemented in the future, it will likely not be SOS, however, the use of a standard is still seen as necessary. In line with this point, this author implemented a custom web API for data access to a large South Australian weatherstation network in 2012. Both a SOS-based system (Car and Cutting 2011) and a non-SOS system were implemented. The reason for this was that target users of this system's data found SOS difficult to work with so current (2012) Internet programming paradigms were used\textsuperscript{16} alongside the SOS implementation. The non-SOS implementation has since been used to power mobile phone weather data applications whereas the SOS implementation has languished unused except for demonstration.

The second point from the 2017 work author indicates that, as unperceived by the author in 2007 but perceived later, O&M-style modelling of data sources’ data is very much required for the tasks originally outlined in 2007. The inclusion of O&M in both SOS and SensorThings has seems to be leading to wider adoption of O&M and recognition of the benefits of using such a sensor situation description system, as opposed to simply a data access API.

\textsuperscript{15} The code base makes heavy use of asynchronous Internet web service functions to efficiently transfer data while rendering user interfaces. This has been necessary due to the large amounts of data available from sensors and thus long times taken to transfer it via the Internet. The code for this work is stored at https://bitbucket.csiro.au/projects/EIS/repos/sos-service/browse but is not directly accessible to the public. Contact the author or CSIRO for access.

\textsuperscript{16} This system, documented at https://github.com/nicholascar/aws_api, uses JSON as a data representation format and uses a SOS-like structure for navigating between elements in the API.
5.4.2. Customisable UI

From the earliest days of this PhD work, effort was put into considering how new user interfaces (UIs), particularly web page-based URLs, could assist with user data source definition and use. The thought was that it may be possible, due to standardised Internet display protocols, to create UIs components that any user could display via a web browser. These components could be assembled by developers with little effort, and perhaps even non-technical system users, to capture custom data source configuration such that data from that data source could be fed in to a DSS. The visual and non-technical assembly of UI elements required for data source characterisation seemed a reasonable prospect in 2007 due to the emergence a few years before of visual web form generation tools, such as Microsoft’s Visual Studio¹⁷ which allowed ‘drag and drop’ web form creation without the requirement to write programming code.

Initial PhD work on UIs is described in an 2008 paper (Car, N.J. et al. 2008). There, a system reliant on standardised Web Services to describe data which is then integrated for display via a calendar was demonstrated. The use of a visual calendar was made due to the thought that irrigation scheduling decisions are inherently calendar-based and that a calendar could provide the decision framework within which to integrate data. The calendar was reliant on a waterbalance as a numerical data integration paradigm which was appropriate given the scope of the DSS discussed – irrigation scheduling.

The approach to data integration via a decision-dependent UI, here a calendar, informed later PhD work both to consider more generalised UIs capable of representing things other than waterbalances, for example irrigation economics incorporating water and crop pricing, and also decision categorisation. Both these considerations were addressed in a 2011 paper by the author (Car and G. A. Moore 2011) which introduced the use of an Artificial Intelligence method, Case-Based Reasoning (CBR) to attempt to sidestep difficult data source characterisation and integration by referring to data sources relevant to a decision only as an input variable and then (in words not present in the paper) effectively performing a sensitivity analysis on which variables were important in leading to successful decisions (something CBR records) to indicate how future decisions should be made. In order to allow the input variables of an irrigation situation to be characterised in CBR terms, a demonstration web-based UI was created. A simple forms interface allowed a user to characterise an input variable (termed a “feature” in the paper) by Name (observed property) and Value (observation result including units). The CBR aspects of this paper and other CBR work in this PhD are described below in Section 6.4.5.

This UI half implemented the original intention of a drag and drop-style UI that users could customise to represent data sources. As presented in the 2011 paper, this UI did not actually characterise a data source – it only related a data source value present within a particular decision case – but it did allow for open-ended irrigation situation characterisation via case input description.

Other non-forms-based UI work was undertaken too, for example, the use of Google Earth, the use of which was new in the agricultural sector in Australia around 2007, to spatially integrate data. Early work took datasets from a collection, such as government-collected soil maps and

¹⁷ Still a current Microsoft product: [https://www.visualstudio.com/](https://www.visualstudio.com/)
presented them for inspection and machine access (Hornbuckle 2008). Later work aimed at allowing users to add their own spatial datasets to Google Earth or to describe them spatially via GPS and upload them to a processing service to provide spatial irrigation decision support, such as where to improve pump pressures to aim for higher irrigation uniformity (Hornbuckle et al. 2009; Hornbuckle et al. 2012).

5.4.3. Data Brokering Layer

Alongside work describing data sources according to particular standards, work was done from early in the PhD (2008) to create something which has latterly been termed a Data Brokering Layer. This work focussed on establishing an intermediary layer between data sources and data consumers (originally thought of as a DSS in particular). The brokering layer’s roles were to standardise the values from, add metadata to and centralise access for addition to data sources.

For the years 2012 – 2015, this author was involved in environmental informatics projects, such as eReefs18 which aimed to, among other things, improve the interoperability of data from multiple data source about the Great Barrier Reef and to assist with decision making about the reef. This resulted in a nodal information architecture described in 2013 (Car 2013) and (Car et al. 2015) which, while reef-specific in its content, is the same structural as one that was imagined and could be produced for irrigation.

The current and continuously operating eReefs Data Brokering Layer19 uses a web-based spatial UI to represent different data sources, as aimed for in the Google Earth and other web spatial work described in the section above, but unlike the work above implements a data semantics layer behind the interface within the Data Brokering Layer that establishes data commensurability. The semantics of a Data Brokering Layer, data sets, sources and web services as well as the measurements of properties and their related features are described in the summarising journal paper (Yu et al. 2016).

The design and establishment of a published and operating DBL are tendered here as secondary outputs of this PhD. They were initiated by PhD requirements but completed within the context of other work and with much assistance from other staff, nevertheless the general considerations, design and implementation are perfectly in line with the task of using informatics to inform irrigation DSS.

5.4.4. Observation models

As has been mentioned in several sections above, the issue of data incommensurability dawned on the author over the course of this PhD work and several attempts to address the commensurability of data from heterogeneous sources were undertaken. In 2009 the author

18 [http://ereefs.org.au](http://ereefs.org.au) – project website

proposed an “Irrigation Modelling Language” (IML) for “not only must the technical standardisation of data source outputs occur but a conceptual informatics framework that describes the data required, and the techniques used, to generate irrigation decision support advice is also required” (Car et al. 2009). This work did present a list of Requirements that are still sensible for any data integration domain model and the paper also indicated an O&M-like observation/method separation with particular elicitation regarding evapotranspiration values and the various methods used to calculate it. It also recognised the different domains data sources for irrigation DSS would need to be integrated against: data access, structure and semantics. The paper did not present an in-depth solution thus this IML work is much more an explanation of a problem domain than it is a system to be used.

Some mechanics from that paper still relevant to observation model work in 2018 are the use of online registries of system’s descriptions for system characterisation within an IML document. The use of Universal Resource Identifiers (URIs) as indicated in the paper is still Linked Data modelling20 best practice.

Following the work above, the Australian Water Data Transfer Format was published (Walker et al. 2009) and the work was extended to address irrigation needs within the more general framework of WDTF. This work dealing with the representation of evapotranspiration and other water and weather measurements is contained within the author’s 2011 paper (Car and G.A. Moore 2011).

The seminal international work on frameworks for natural world observations, the ISO standard Observations & Measurement (ISO 2011) which has been mentioned in sections above was introduced in 2011 too and later versions of WDTF which were merged with WaterML to create WaterML2 incorporate O&M. This work edited by a now colleague of the authors was unknown to the author in 2011 and until about 2014.

The most recent (2016) major implementation of O&M practices by a standards body is the Sensor, Observation, Sample, and Actuator (SOSA) ontology within the Semantic Sensor Network (SSN) ontology published by the World Wide Web Consortium (Haller et al. 2017). Within this work, an explicit mapping between the O&M-style representation of observed data’s surrounding concerns and an ontological form of a standardised process flow model, PROV-O (Lebo, Sahoo, and McGuinness 2013) is made21. This mapping was made to facilitate standardised representation of information transformations using a general-purpose model (not sensor- or even observation-specific) to help with understanding results. This is therefore an extension to the domains of knowledge that may be considered when assessing data commensurability. Alignment of modelling systems with standardised representations of provenance like this was one of the motivations for the work in Chapter 6, Section 2: if the processing of data from sources and the information flow into decision scenarios relevant for irrigation could be represented in standardised ways, data commensurability could be more easily assessed.

---

20 [https://www.w3.org/wiki/LinkedData](https://www.w3.org/wiki/LinkedData)

21 The specific section in the ontology documentation is [https://www.w3.org/TR/vocab-ssn/#x6-5-prov-alignment-module](https://www.w3.org/TR/vocab-ssn/#x6-5-prov-alignment-module)
5.4.5. Case-Based Reasoning

As noted in Section 5.4.2 above, some efforts were undertaken in this PhD to sidestep the perceived necessity to characterise data sources and structure data in detail in order to get commensurate information. The Case-Based Reasoning methodology was investigated due to the obvious CBR-like nature of irrigation scheduling decisions where decision situations (a current "case") can be related to one’s previous "cases" and also perhaps the cases of others for which outcomes are known in order to convey norms or best practice. Literature review work into decision theory for this PhD revealed CBR to this author (see Appendix A, page 42) and a PhD topic was introduced with the wording "Determine whether CBR can provide a rigorous model for irrigation scheduling and provide decision support" was introduced at conversion review time in 2009 (see Appendix B).

In addition to notions that CBR might assist with irrigation DSS, it, being an Artificial Intelligence methodology, was seen as serious informatics and thus it should be brought to bear on a PhD titled Irrigation Informatics. No work other than literature review was done on CBR before the conversion review in 2009 so that by then, it was posed as a future work topic.

CBR work in earnest was undertaken in 2010 with a conference paper describing work until that point delivered in 2011 (Car and G. A. Moore 2011). There the sidestepping of detailed data source characterisation and the basic mechanics of a CBR for use in irrigation, including a case-capturing UI, are presented. While this work showed promise, the complexity of such system establishment and testing prevented it making it through to a full chapter of this thesis (see topic progression in Appendix B).

Further work undertaken in this PhD to explore the possibilities for CBR to assist with irrigation DSS looked at recording data sources influencing irrigation decisions not normally considered by irrigation DSS, such as an irrigators’ family life preferences. This work was partially represented in extensions to the calendar UI work (Car, N.J. et al. 2008) where a mask to prevent work at certain times could be applied to a calendar and then the costs of adhering to that mask would be calculated by the DSS for the user’s consideration. This work was not independently published.

The CBR-informed thoughts in this area lead to the attempt to characterise all the data sources influencing an irrigation decision, including traditional and non-traditional ones such as water availability and perhaps children’s school timetables respectively. While this work has also not been independently published, it led into the decision to better characterise decision making itself which is addressed in the next section and in Chapter 6. The reasoning here was that a decision’s relevant data sources couldn’t be effectively characterised unless better knowledge of the decision scenario and how people encounter decisions was determined.

Additionally, current work of the authors looks to leverage CBR for irrigation decision making within the context of his more recent understanding of Semantic Web tooling and the decision modelling from Chapter 6. An accepted poster for a conference in July 2018 on this topic is given in Appendix G titled "Case Base Reasoning decision support using the DecPROV ontology for decision modelling".
5.5. Decision modelling

Resulting from several strands of enquiry into User-Defined Data was the notion that better framing of decision making itself is needed to better enable data integration and thus better decision support. It has been described in introductory text to this PhD and various papers within it that irrigation decisions usually model a biophysical scenario (IrriSat in all forms models a waterbalance) and sometimes a biophysical scenario plus economics (water/crop price) and that this sort of modelling is inevitably narrower than the "real world" scenarios that irrigators face when making the decisions the DSS claim to support. One way tried to overcome this was the above-mentioned CBR methodology which could reduce data source characterisation effort. Another way was broader decision modelling.

Another way is to say that if DSS are always too narrow in their scope, then efforts to change this could look to mapping the total influences on a decision and then dealing with the results. Social science investigations into influences on irrigators were conducted by the CRC for Irrigation Futures, the body that funded this PhD, and they resulted in some situation description publications (Montagu et al. 2006; Whittenbury and Davidson 2009). Work to represent irrigation decision situations holistically and in a manner that would be suitable for computer systems to use motivated the decision modelling work related in Chapter 6.

The characterising of decision making processes was started early in this PhD, around 2007, with literature reviews into decision theory (see Appendix A) but no decision theory experiments were undertaken until much later, 2015. The reasons for this time delay are given in Chapter 1, the Introduction. In addition to those reasons, it can be seen, with hindsight, that the author did not possess the modelling skill in order to characterise decision making as effectively as now done in 2017 previously.

Decision modelling work produced in this PhD is currently being used in non-irrigation decision scenarios such as spatial decision making (Ivánová et al. 2018) and is informing a new round of work on Case-Based Reasoning that is dependent on having a standardised decision representation system (see Appendix G).

5.6. Future vision of a data marketplace

The irrigation decisions support research community in Australia, in either 2007 or 2018, was not a very large one. For this reason, work in the field tends to acquaint one with all of the data sources used by existing irrigation DSS fairly quickly and some of the newer data sources available are listed in Section 6.3 above. For this reason, it has seemed possible for some time to produce a data marketplace for irrigation data in Australia that establishes standardised data sources for use by multiple possible DSS. Such a marketplace could not only provide the mappings of legacy data sources to standardised formats but could also actually process the data – applying the mappings – as needed too.

Two forms of a data marketplaces informed by this PhD's early work that could, in turn, inform a future irrigation data marketplace, have recently been established; the soil moisture related
The data sources integrated using SOS characterisation (Sommer 2018) in Section 6.4.1 and the eReefs DBL described in 6.4.3 above.

The first provides semantic and network mechanisms to integrate data from a wide range of agricultural sensors is able, technically, to provide legacy data in standardised formats as needed. The second integrates observations and model results within a domain and continuously operates to provide access to standardised data sources.

A future irrigation data marketplace would be able to follow the SOS-based systems mechanics and implement the semantic resources mapped out in the reef domain by eReefs for irrigation. This would require research work into the specifics of irrigation decision making scenarios which could be characterised by a decision templating system such as that described in Chapter 6, Section 3.

5.7. Reflections

As of 2018, there has been no major revolution in the data sources available for irrigation DSS in Australia. The number of Internet-based sources has grown since 2007 making a longer list than given in Section 6.2 however there is no catalogue or even a general method available to discover and use irrigation data sources.

Current work within CSIRO Land & Water (Sommer et al. 2018) is looking at translating data source's data in Internet-available data in proprietary formats – as implemented by most field equipment manufacturers – to standardised ones. While the general idea of standardising agricultural equipment data formats is not new, the latest approaches do new things:

1. Standardise both the structure of the data and its semantics using a sophisticated observations model;
2. Use newly available Internet data manipulation mechanics.

Most previous agricultural data standardisation attempts focussed entirely on the structure of the data (e.g. how to encode time/value pairs) or use a very simple observations model, perhaps only relating a value and with a type label and a unit of measure. In addition to using the Sensor Observation Service (Open Geospatial Consortium 2007) for data structure, Sommer et al. (2018) uses the Observations & Measurements standard (ISO 2011) to associate measurement results with observations of phenomena and their associated features of interest. Only recently libraries of references of properties and features have become available (Cox, Simons, and Yu 2014; Yu et al. 2015).

The system uses an on-the-fly translation of data from proprietary formats to standardised ones. This has only been technically possible for distributed (non-warehoused), large data volumes recently due to faster national (in Australia) Internet speeds with which to transmit data and faster data processing tools, such as asynchronous Python web data toolkits22.

---

22 The Sommer system uses Python's Sanic library (https://sanic.readthedocs.io/en/latest/) to read and translate data from multiple sources at once while generating graphs of related information for users, something Internet servers have only been capable of recently.
These approaches address issues of data commensurability, referenced above, and technical access issues given current data loads respectively and while they have already been a large amount of work, they are clearly nowhere near solving these issues, despite building on now almost a decade of work. So, in addition to the prerequisites for taking these two approaches not being available for work carried out in 2007 – 2010, the recent work shows that the original scope to try and substantially enhance just the commensurability issue alone was too optimistic.

5.8. References


http://academicworks.cuny.edu/cgi/viewcontent.cgi?article=1178&context=cc_conf_hic.

Department of Science, Information Technology and Innovation 2011 SILO Climate Data. 
Government department. The Long Paddock Website. 


https://www.w3.org/TR/vocab-ssn/.

https://doi.org/10.4225/08/585ac48e90da3.


https://www.w3.org/TR/prov-o/, accessed September 17, 2016.


5. **Data sources**

This chapter relates a series of investigations undertaken into irrigation DSS enhancement through data source integration. Unlike the previous and following chapters, this chapter doesn’t have a corresponding major journal publication however its work has been published in a series of six conference papers:


This chapter is of importance to this thesis since it relates part of the journey of this PhD to questions initially posed for it. It then shows how work for those questions was instrumental in developing the subsequent PhD focus on decision modelling. The work on User-Defined Data (See 6.4) lead to investigations into flexible user interfaces (Car *et al*. 2008) and finally Case-Based Reasoning (Car and G. A. Moore 2011) for which the decision modelling in Chapter 7 was conducted.

Work in this Chapter falls within this thesis’s research topic *Customisation (of DSS)*.
5.1. Motivation

One of many issues noted by authors reflecting on the poor uptake of agricultural DSS in Australia in the early years of the 21st century was the poor relevance to real-world practice of the data they used. Authors such as McCown (McCown 2001), and at the start of this PhD Nguyen (Nguyen, Wegener, and Russell 2007), noted that DSS need to be situation-specific to in order to be accurate and be trusted by users yet many DSS use generic data such as crop variety rather than specific crop metrics or area average water applications rather than individual fields' values.

While McCown primarily addressed research and DSS development methodologies to try and design DSS that better fitted within farming practice and also to lead farmers on a DSS development journey that they would own, Nguyen noted “...intergenerational change that is occurring in the management of Australian farms is a positive factor that may encourage more widespread use of these tools [DSS]”. By this he meant irrigators fearful of computerised systems like DSS were being replaced by those not fearful.

Regardless of how accepting new irrigators might be of computerised DSS, the DSS must still supply accurate and locally-relevant results to be of use. As per Chapter 1, Section 1.3.5, the Internet and mobile phones, when coupled with irrigators not fearful of computerised systems have made many more tailorable DSS configurations possible: we can now use ubiquitous data transfer mechanics (the Internet) with remote and local data sources and send customised results to irrigators who have the tech savvy in order to use them.

Despite these advances, there remains an on-going challenge to make DSS results more locally specific, more relevant. We see this in the survey results within the paper represented in Chapter 2 where some irrigators place a lot of weight on particular, local, systems that, ideally, would integrate with any DSS like IrriSatSMS.

We need to also acknowledge that the irrigation DSS experiments carried out in Chapters 2 & 3 and that likely the wider the variety of situations any particular DSS attempts to address, the more local customisation it may come to rely on in order to be locally accurate.

5.2. Initial work and topic changes

Early work in this PhD sought to classify data sources by “network paradigm”, the way they were network technically able to be accessed by a DSS, given that network/Internet connectivity was growing rapidly at the time. This lead to the publication of Towards a new generation of Irrigation Decision Support Systems – Irrigation Informatics? (Car et al. 2007) (Also Appendix B). This work, while apparently

---

1 This awkward phrase means by physical communications channel access, i.e. some sort of wires or wireless data transfer network. This reflects the author's initial electrical engineering training.
of interest to educators of undergraduate engineers\textsuperscript{2} did not lead to any major ways to improve irrigation DSS data source use.

Next, a research question was formulated in 2008 around the idea of measuring the utility to irrigators of using different evapotranspiration data sources within a waterbalance-based irrigation DSS. The evapotranspiration sources were to be the national, interpolated grids (e.g. SILO, see below), large weatherstation networks (e.g. Lower Murray Water’s AWS network, also below) and local weatherstations, set up for the occasion as additions to CSIRO’s weatherstation network (also below). The waterbalance system was to be IrriSatSMS or a derivative system.

This work was never started for a number of reasons, the first of which was the time spent on, and ultimate failure of, the MMS-based crop imagery experiment (see Chapter 3, Section 3.4). This derailed work on new data source integration generally as, after it, IrriSatSMS was focussed purely on area and crop expansion (to the Sydney basin and vegetable growers after the initial Griffith NSW area and wine grape growers) given that this seemed to be more likely to succeed. The second was the difficulty in catering for different methods of measuring evapotranspiration. Multiple algorithms may be used to calculate evapotranspiration (see (Meyer 1999) for the algorithm used by IrriSatSMS and (Barton and Meyer 2007) for a comparison of multiple methods) and not all systems reporting evapotranspiration figures make explicitly which calculation they are using. This realisation lead to two outcomes: the first was a conference paper presented that was critical of the Australian AWRIS national water data system (see below)’s efforts to represent aspects of water data crucial for its agricultural use using evapotranspiration as the primary example (Car and G.A. Moore 2011). The second was that this author gained a general understanding of the large breadth of information (metadata) required for effective data use within a DSS where data sources and DSS are not directly designed to be used together. Phrased in hindsight, a holistic observations model is needed to ensure measurement commensurability. Simple things such as time and place and units of measure are not sufficient to indicate exactly what phenomena are observed, of what natural world feature and whether the observation methods are relevant to know (as is the case for standard evapotranspiration, given there are multiple standards!).

The initial thesis topic described for work in the area of data sources at the time of this thesis’ conversion from Masters to PhD was Determine the extent to which XML specifications may be adopted by irrigation data source for their interoperability and ‘discoverability’ over the internet. Due to the reason of measurement commensurability complexity mentioned above and the expectation that an assessment of “...the extent to which...” would be very difficult (see Appendix B for full details) this topic was changed to being about User-Defined Data (see Appendix B). The next section relates investigations into this topic.

\textsuperscript{2} The classification system from this paper has been used in repeated versions of the undergraduate engineering textbook \textit{Engineering your future} (Dowling et al. 2016).
5.3. Newly available data

In addition to new research & development methodologies and generational change among irrigators mentioned above in Section 6.1, the sources of data available for DSS designers and DSS users changed rapidly just before the commencement of this PhD.

Some examples of Internet-based irrigation-relevant data sources newly available online in Australia in 2007 were:

- **SILO**
  - “…an enhanced climate database…[which]…contains Australian climate data from 1889 (current to yesterday), in a number of ready-to-use formats, suitable for research and climate applications.” (Department of Science, Information Technology and Innovation 2011)

- **Queryable soil maps**
  - District maps for Murrumbidgee, Coleambally & Murray irrigation areas
    - An online system to deliver these maps was built by this author in 2007 (Hornbuckle 2008) and remained online until 2016
  - National maps
    - “ASRIS provides online access to the best publicly available information on soil and land resources in a consistent format across Australia”³

- **Water trading**
  - Online water exchanges were new in 2007. There are now a plethora of these, such as WaterFind⁴, Ruralco Water⁵ & H2OX⁶ but in 2007 there were few, but at least one exchange was available to irrigators in the Griffith, NSW, area who partook in IrriSatSMS trials.

- **Weatherstation data**
  - CSIRO Land & Water’s weatherstation network
    - A network re-implemented by this author to supply IrriSatSMS with hourly weatherstation data⁷
  - Lower Murray Water’s AWS network
    - A network of stations owned and operated by Lower Murray Water, a regional Australian water supplier⁸ partially built by this author
    - There are many weatherstation networks like this in Australia

- **Sensor-measured soil moisture data**

---

³ [http://www.asris.csiro.au](http://www.asris.csiro.au)
⁶ [http://h2ox.com](http://h2ox.com)
⁷ [http://weather.csiro.au](http://weather.csiro.au)
o SENTEK and MEA\(^9\), among many soil moisture sensor vendors in Australia, offer the ability for clients to access data from their networks online

- **Crop imagery**
  o LANDSAT and other satellite imagery available online, the use of which for IrriSatSMS is described in Chapter 2.
  o Mobile-phone-based crop imagery, experiments about the acquisition of which were conducted as part of this PhD. See Chapter 3 & Appendix D.

Note that some of these data sources are remote sources, such as satellite imagery, and some require local infrastructure, such as soil moisture probes. What they all have in common is the use of the Internet to transfer or publish data. Some companies in 2007, and still now, do use non-Internet-based networks for data delivery, such as wired LANs, however the Internet is used very widely and Internet protocols are the only viable option when trying to access irrigation DSS-relevant data from a range of data source makers who never built their tools to be used within a DSS context. This is due to equipment providers and data publishers being able to design for Internet protocols with a high degree of certainty that eventual users will be able to work with them, again due to the Internet’s pervasiveness, whatever their interests, DSS or otherwise.

In addition to the sources above, a very large data integration exercise was started then in Australia to centralise water storage, observations and usage data, as well as some weather data, within the Bureau of Meteorology in response to the Australian Millennium Drought of the late 1990s and first decade of the 21st century. The Australian Water Act 2007 (Commonwealth of Australia 2007) and resultant regulation required that Bureau of Meteorology continuously acquire water data from over 200 state and non-government organisations and that they manage it, for which they developed a system known as the Australian Water Resources Information System (AWRIS)\(^10\). This promised to make water data from 200+ water control agencies and companies available in once place in a standard format, the Water Data Transfer Format (WDTF) (Walker et al. 2009), but, as of even 2018, water data such as supply availability is not able to be retrieved from this system (it’s not really for this purpose), instead most irrigators deal with water trading organisations for water availability.

While soil moisture sensors were listed above, many more sensors with Internet connectivity became available around 2007 onwards and more appear still: irrigation pumps themselves, piping systems and their statuses, webcams of crops, etc.

### 5.4. User Defined Data

So, given the growing range of Internet-accessible data sources – whether made for DSS or not – how could DSS designers make more of them available as inputs for DSS?


Three main avenues of investigation have been undertaken over the course of this PhD to address this:

1. Networks, UIs and other direct technical methods
2. Commensurability methods
3. Decision representation

The first of these is discussed in Subsections 5.4.1, 5.4.2 & 5.4.3, the second in Subsections 5.4.4 and the last in Subsection 5.4.5.

Overall, and despite the considerable work put into this area over the course of the last decade, User-Defined Data itself is not considered a final research topic to have yielded substantial results. Work started within this topic has gone on to yield useful outputs (see Chapter 6 for decision modelling) or are on-going (Subsection 6.4.5 below). Nevertheless, here follows a reiteration of work done within this PhD that is mentioned elsewhere but is summarised to overview the User-Defined Data topic work.

5.4.1. Network paradigms

As already stated in Section 5.2 above, initial work done in this PhD to assist DSS designers cater for a wider range of data sources involved a categorisation of DSS, in present in 2007, according to a “network paradigm” or the way they then connected to sources of data. This work resulted in the publication of a conference paper (Car et al. 2007) which was included in an initial literature review draft of this thesis (see Appendix A, page 16). While that work was never carried further than that within this thesis, it has been included in a series of textbooks for undergraduate engineers to assist them in the way they design systems starting in 2013 and right up until the latest edition in 2016 (Dowling et al. 2016).

The positive elements of that 2007 paper, looking back from 2018, are that it did correctly predict/wish for a so-called User-Defined Data Set that would able to be “…added to the pool of data sources used by the DSS by the unassisted DSS user…” and this is something that is still aimed for today (see Section 6.7 below), so the idea is not incorrect or perhaps unattainable but it is somewhat obvious and it was naïve to suggest it was likely or even possible that a single PhD could bring such a vision into reality.

Another element from that paper that stands the test of time was the idea to use a range of Open Geospatial Consortium standards for sensor description and data transfer, very new at the time, for describing data sources such that a DSS could cater for a series of protocols and then be able to use data sources formulated using them.

What was completely missing from this work was an appreciation of the magnitude of data commensurability issues that need solving before multiple data sources could be used by a DSS, even when the physical network and data format issues are solved. This issue, raised in Section 6.2 is discussed further below in Section 6.4.4.

The author, following advice in this paper, between 2007 and 2009, re-implemented the CSIRO Land & Water weatherstation network of 10+ stations. Although there were cost savings and other practical reasons for doing so (a change to using cellular 3G technology for weatherstation-to-base data communication over land line use saved
money and increased weatherstation deployment flexibility enabling the development of IrriSatSMS detailed in Chapter 2 and newer Internet protocols for base-to-client communications (the use of HTTP, rather than FTP) greatly improved the data access, the ultimate goal in network reimplementation was to allow the weatherstation data to be communicated using the then new Sensor Observation Service (SOS) standard (Open Geospatial Consortium 2007). This was in order to establish a set of irrigation DSS input data sources (proved to be such through IrriSatSMS work) in Australia, available freely on the web, using a standardised, machine readable, data access protocol.

This task was ultimately achieved only in 2015 with the development by the author of the aws_pywebsite code base that read from the weatherstations’ database. The reason for this considerable delay was the lack of interest by his funding body or anyone else in seeing such work conducted.

The set of data and network standards and the domains they address that were implemented through this work were the protocols TPC & HTTP to pass messages via the Internet (as per all websites), XML for the generic data format and SOS for the sensing system description. By the 2015 implementation, WaterML2.0, a timeseries data exchange format created partly as a result of the WDTF work mentioned above in Section 6.3 (Open Geospatial Consortium 2014), was used within the SOS system for timeseries representation. Inherent in the 2015 SOS installation implemented was also a representation of the observations’ methods, measurements and related objects using the Observations & Measurement (O&M) standard which became available in 2011 (ISO 2011). For further thoughts on the use of O&M, see Section 6.4.4 below.

Further on the stalled implementat

In 2017, engineers within CSIRO Land & Water within a unit similar to that in which the author’s 2007+ IrriSatSMS work was undertaken, implemented a SOS-based system to present data from a range of sensors from different agricultural equipment manufacturers in a standardised manner. This work is currently the subject of publication (Sommer, Stenson, and Searle 2018) and represents the latest methods employed for agricultural/irrigation heterogeneous data source integration in Australia.

11 The weatherstation network is still visible with 5 stations, online using web code written partly in 2007, at http://weather.csiro.au.
12 This Python code repository is available online at https://bitbucket.csiro.au/projects/EIS/repos/aws_pywebsite/.
13 http://www.mea.com.au
14 Personal communication with Andrew Skinner, founder of Measurement Engineering Australia, approx. 2009.
This work uses different computer programming paradigms to any considered by this author\(^\text{15}\) but is, in essence, still in line with the intentions expressed in the author’s early conference paper (Car et al. 2007).

Two thoughts relevant to the thoughts here about this 2017 system implementation were communicated to this author in early 2018:

1. SOS is still not widely used and is perhaps unlikely to ever be used. The author of the 2017 work anticipates implementing the newer OGC SensorThings (Open Geospatial Consortium 2016) standard in its place in the future. This is believed to be due to a number of reasons, the main one being that SensorThings better uses of data representation to make data transfer more efficient which is important for recent data sources producing large amounts of data. Methods other than SOS, like SensorThings, are also seemingly easier than SOS for most web developers to implement.

2. The 2017 work suffered a great deal due to the lack of metadata from vendors describing the properties and features of the environment that their systems observed/measured. Despite the existence of standards for this purpose, such as O&M, none were implemented and much back-end work was needed by the 2017 author to try to make sensor readings commensurate (e.g. soil moisture observed by different manufacturers soil moisture probes doesn’t always indicate measurement depth, units used, absolute moisture or % moisture).

The first point indicates that SOS has failed to gain traction and, if an irrigation data source-relevant standard is to be implemented in the future, it will likely not be SOS, however, the use of a standard is still seen as necessary. In line with this point, this author implemented a custom web API for data access to a large South Australian weatherstation network in 2012. Both a SOS-based system (Car and Cutting 2011) and a non-SOS system were implemented. The reason for this was that target users of this system’s data found SOS difficult to work with so current (2012) Internet programming paradigms were used\(^\text{16}\) alongside the SOS implementation. The non-SOS implementation has since been used to power mobile phone weather data applications whereas the sOS implementation has languished unused except for demonstration.

The second point from the 2017 work author indicates that, as unperceived by the author in 2007 but perceived later, O&M-style modelling of data sources’ data is very much required for the tasks originally outlined in 2007. The inclusion of O&M in both SOS and SensorThings has seems to be leading to wider adoption of O&M and recognition of the benefits of using such a sensor situation description system, as opposed to simply a data access API.

\(^{15}\) The code base makes heavy use of asynchronous Internet web service functions to efficiently transfer data while rendering user interfaces. This has been necessary due to the large amounts of data available from sensors and thus long times taken to transfer it via the Internet. The code for this work is stored at https://bitbucket.csiro.au/projects/EIS/repos/sos-service/browse but is not directly accessible to the public. Contact the author or CSIRO for access.

\(^{16}\) This system, documented at https://github.com/nicholascar/aws_api, uses JSON as a data representation format and uses a SOS-like structure for navigating between elements in the API.
5.4.2. Customisable UI

From the earliest days of this PhD work, effort was put into considering how new user interfaces (UIs), particularly web page-based URIs, could assist with user data source definition and use. The thought was that it may be possible, due to standardised Internet display protocols, to create UI components that any user could display via a web browser. These components could be assembled by developers with little effort, and perhaps even non-technical system users, to capture custom data source configuration such that data from that data source could be fed in to a DSS. The visual and non-technical assembly of UI elements required for data source characterisation seemed a reasonable prospect in 2007 due to the emergence a few years before of visual web form generation tools, such as Microsoft’s Visual Studio\(^\text{17}\) which allowed ‘drag and drop’ web form creation without the requirement to write programming code.

Initial PhD work on UIs is described in an 2008 paper (Car, N.J. \textit{et al.} 2008). There, a system reliant on standardised Web Services to describe data which is then integrated for display via a calendar was demonstrated. The use of a visual calendar was made due to the thought that irrigation scheduling decisions are inherently calendar-based and that a calendar could provide the decision framework within which to integrate data. The calendar was reliant on a waterbalance as a numerical data integration paradigm which was appropriate given the scope of the DSS discussed – irrigation scheduling.

The approach to data integration via a decision-dependent UI, here a calendar, informed later PhD work both to consider more generalised UIs capable of representing things other than waterbalances, for example irrigation economics incorporating water and crop pricing, and also decision categorisation. Both these considerations were addressed in a 2011 paper by the author (Car and G. A. Moore 2011) which introduced the use of an Artificial Intelligence method, Case-Based Reasoning (CBR) to attempt to sidestep difficult data source characterisation and integration by referring to data sources relevant to a decision only as an input variable and then (in words not present in the paper) effectively performing a sensitivity analysis on which variables were important in leading to successful decisions (something CBR records) to indicate how future decisions should be made. In order to allow the input variables of an irrigation situation to be characterised in CBR terms, a demonstration web-based UI was created. A simple forms interface allowed a user to characterise an input variable (termed a “feature” in the paper) by Name (observed property) and Value (observation result including units). The CBR aspects of this paper and other CBR work in this PhD are described below in Section 6.4.5.

This UI half implemented the original intention of a drag and drop-style UI that users could customise to represent data sources. As presented in the 2011 paper, this UI did not actually characterise a data source – it only related a data source value present within a particular decision case – but it did allow for open-ended irrigation situation characterisation via case input description.

Other non-forms-based UI work was undertaken too, for example, the use of Google Earth, the use of which was new in the agricultural sector in Australia around 2007, to

\(^{17}\) Still a current Microsoft product: [https://www.visualstudio.com/](https://www.visualstudio.com/)
spatially integrate data. Early work took datasets from a collection, such as government-collected soil maps and presented them for inspection and machine access (Hornbuckle 2008). Later work aimed at allowing users to add their own spatial datasets to Google Earth or to describe them spatially via GPS and upload them to a processing service to provide spatial irrigation decision support, such as where to improve pump pressures to aim for higher irrigation uniformity (Hornbuckle et al. 2009; Hornbuckle et al. 2012).

5.4.3. Data Brokering Layer

Alongside work describing data sources according to particular standards, work was done from early in the PhD (2008) to create something which has latterly been termed a Data Brokering Layer. This work focussed on establishing an intermediary layer between data sources and data consumers (originally thought of as a DSS in particular). The brokering layer’s roles were to standardise the values from, add metadata to and centralise access for addition to data sources.

For the years 2012 – 2015, this author was involved in environmental informatics projects, such as eReefs\(^{18}\) which aimed to improve the interoperability of data from multiple data source about the Great Barrier Reef and to assist with decision making about the reef. This resulted in a nodal information architecture described in 2013 (Car 2013) and (Car et al. 2015) which, while reef-specific in its content, is the same structural as one that was imagined and could be produced for irrigation.

The current and continuously operating eReefs Data Brokering Layer\(^{19}\) uses a web-based spatial UI to represent different data sources, as aimed for in the Google Earth and other web spatial work described in the section above, but unlike the work above implements a data semantics layer behind the interface within the Data Brokering Layer that establishes data commensurability. The semantics of a Data Brokering Layer, data sets, sources and web services as well as the measurements of properties and their related features are described in the summarising journal paper (Yu et al. 2016).

The design and establishment of a published and operating DBL are tendered here as secondary outputs of this PhD. They were initiated by PhD requirements but completed within the context of other work and with much assistance from other staff, nevertheless the general considerations, design and implementation are perfectly in line with the task of using informatics to inform irrigation DSS.

5.4.4. Observation models

As has been mentioned in several sections above, the issue of data incommensurability dawned on the author over the course of this PhD work and several attempts to address

\(^{18}\text{http://ereefs.org.au} – \text{project website}\n
\(^{19}\text{https://research.csiro.au/ereefs/ereefs-data/information-architecture/data-brokering-layer/}\n
82
the commensurability of data from heterogeneous sources were undertaken. In 2009 the author proposed an "Irrigation Modelling Language" (IML) for “not only must the technical standardisation of data source outputs occur but a conceptual informatics framework that describes the data required, and the techniques used, to generate irrigation decision support advice is also required” (Car et al. 2009). This work did present a list of Requirements that are still sensible for any data integration domain model and the paper also indicated an O&M-like observation/method separation with particular elicitation regarding evapotranspiration values and the various methods used to calculate it. It also recognised the different domains data sources for irrigation DSS would need to be integrated against: data access, structure and semantics. The paper did not present an in-depth solution thus this IML work is much more an explanation of a problem domain than it is a system to be used.

Some mechanics from that paper still relevant to observation model work in 2018 are the use of online registries of system’s descriptions for system characterisation within an IML document. The use of Universal Resource Identifiers (URIs) as indicated in the paper is still Linked Data modelling best practice.

Following the work above, the Australian Water Data Transfer Format was published (Walker et al. 2009) and the work was extended to address irrigation needs within the more general framework of WDTF. This work dealing with the representation of evapotranspiration and other water and weather measurements is contained within the author’s 2011 paper (Car and G.A. Moore 2011).

The seminal international work on frameworks for natural world observations, the ISO standard Observations & Measurement (ISO 2011) which has been mentioned in sections above was introduced in 2011 too and later versions of WDTF which were merged with WaterML to create WaterML2 incorporate O&M. This work edited by a now colleague of the authors was unknown to the author in 2011 and until about 2014.

The most recent (2016) major implementation of O&M practices by a standards body is the Sensor, Observation, Sample, and Actuator (SOSA) ontology within the Semantic Sensor Network (SSN) ontology published by the World Wide Web Consortium (Haller et al. 2017). Within this work, an explicit mapping between the O&M-style representation of observed data’s surrounding concerns and an ontological form of a standardised process flow model, PROV-O (Lebo, Sahoo, and McGuinness 2013) is made. This mapping was made to facilitate standardised representation of information transformations using a general-purpose model (not sensor- or even observation-specific) to help with understanding results. This is therefore an extension to the domains of knowledge that may be considered when assessing data commensurability. Alignment of modelling systems with standardised representations of provenance like this was one of the motivations for the work in Chapter 6, Section 2: if the processing of data from sources and the information flow into decision scenarios relevant for irrigation could be represented in standardised ways, data commensurability could be more easily assessed.

20 https://www.w3.org/wiki/LinkedData
21 The specific section in the ontology documentation is https://www.w3.org/TR/vocab-ssn/#x6-5-prov-alignment-module
5.4.5. Case-Based Reasoning

As noted in Section 5.4.2 above, some efforts were undertaken in this PhD to sidestep the perceived necessity to characterise data sources and structure data in detail in order to get commensurate information. The Case-Based Reasoning methodology was investigated due to the obvious CBR-like nature of irrigation scheduling decisions where decision situations (a current “case”) can be related to one’s previous “cases” and also perhaps the cases of others for which outcomes are known in order to convey norms or best practice. Literature review work into decision theory for this PhD revealed CBR to this author (see Appendix A, page 42) and a PhD topic was introduced with the wording “Determine whether CBR can provide a rigorous model for irrigation scheduling and provide decision support” was introduced at conversion review time in 2009 (see Appendix B).

In addition to notions that CBR might assist with irrigation DSS, it, being an Artificial Intelligence methodology, was seen as serious informatics and thus it should be brought to bear on a PhD titled *Irrigation Informatics*. No work other than literature review was done on CBR before the conversion review in 2009 so that by then, it was posed as a future work topic.

CBR work in earnest was undertaken in 2010 with a conference paper describing work until that point delivered in 2011 (Car and G. A. Moore 2011). There the sidestepping of detailed data source characterisation and the basic mechanics of a CBR for use in irrigation, including a case-capturing UI, are presented. While this work showed promise, the complexity of such system establishment and testing prevented it making it through to a full chapter of this thesis (see topic progression in Appendix B).

Further work undertaken in this PhD to explore the possibilities for CBR to assist with irrigation DSS looked at recording data sources influencing irrigation decisions not normally considered by irrigation DSS, such as an irrigators’ family life preferences. This work was partially represented in extensions to the calendar UI work (Car, N.J. et al. 2008) where a mask to prevent work at certain times could be applied to a calendar and then the costs of adhering to that mask would be calculated by the DSS for the user’s consideration. This work was not independently published.

The CBR-informed thoughts in this area lead to the attempt to characterise *all* the data sources influencing an irrigation decision, including traditional and non-traditional ones such as water availability and perhaps children’s school timetables respectively. While this work has also not been independently published, it led into the decision to better characterise decision making itself which is addressed in the next section and in Chapter 6. The reasoning here was that a decision’s relevant data sources couldn’t be effectively characterised unless better knowledge of the decision scenario and how people encounter decisions was determined.

Additionally, current work of the authors looks to leverage CBR for irrigation decision making within the context of his more recent understanding of Semantic Web tooling and the decision modelling from Chapter 6. An accepted poster for a conference in July 2018 on this topic is given in Appendix G titled “Case Base Reasoning decision support using the DecPROV ontology for decision modelling”.

84
5.5. Decision modelling

Resulting from several strands of enquiry into User-Defined Data was the notion that better framing of decision making itself is needed to better enable data integration and thus better decision support. It has been described in introductory text to this PhD and various papers within it that irrigation decisions usually model a biophysical scenario (IrriSat in all forms models a waterbalance) and sometimes a biophysical scenario plus economics (water/crop price) and that this sort of modelling is inevitably narrower than the “real world” scenarios that irrigators face when making the decisions the DSS claim to support. One way tried to overcome this was the above-mentioned CBR methodology which could reduce data source characterisation effort. Another way was broader decision modelling.

Another way is to say that if DSS are always too narrow in their scope, then efforts to change this could look to mapping the total influences on a decision and then dealing with the results. Social science investigations into influences on irrigators were conducted by the CRC for Irrigation Futures, the body that funded this PhD, and they resulted in some situation description publications (Montagu et al. 2006; Whittenbury and Davidson 2009). Work to represent irrigation decision situations holistically and in a manner that would be suitable for computer systems to use motivated the decision modelling work related in Chapter 6.

The characterising of decision making processes was started early in this PhD, around 2007, with literature reviews into decision theory (see Appendix A) but no decision theory experiments were undertaken until much later, 2015. The reasons for this time delay are given in Chapter 1, the Introduction. In addition to those reasons, it can be seen, with hindsight, that the author did not possess the modelling skill in order to characterise decision making as effectively as now done in 2017 previously.

Decision modelling work produced in this PhD is currently being used in non-irrigation decision scenarios such as spatial decision making (Ivánová et al. 2018) and is informing a new round of work on Case-Based Reasoning that is dependent on having a standardised decision representation system (see Appendix G).

5.6. Future vision of a data marketplace

The irrigation decisions support research community in Australia, in either 2007 or 2018, was not a very large one. For this reason, work in the field tends to acquaint one with all of the data sources used by existing irrigation DSS fairly quickly and some of the newer data sources available are listed in Section 6.3 above. For this reason, it has seemed possible for some time to produce a data marketplace for irrigation data in Australia that establishes standardised data sources for use by multiple possible DSS. Such a marketplace could not only provide the mappings of legacy data sources to
standardised formats but could also actually process the data – applying the mappings – as needed too.

Two forms of a data marketplaces informed by this PhD's early work that could, in turn, inform a future irrigation data marketplace, have recently been established; the soil moisture related data sources integrated using SOS characterisation (Sommer 2018) in Section 6.4.1 and the eReefs DBL described in 6.4.3 above.

The first provides semantic and network mechanisms to integrate data from a wide range of agricultural sensors is able, technically, to provide legacy data in standardised formats as needed. The second integrates observations and model results within a domain and continuously operates to provide access to standardised data sources.

A future irrigation data marketplace would be able to follow the SOS-based systems mechanics and implement the semantic resources mapped out in the reef domain by eReefs for irrigation. This would require research work into the specifics of irrigation decision making scenarios which could be characterised by a decision templating system such as that described in Chapter 6, Section 3.

5.7. Reflections

As of 2018, there has been no major revolution in the data sources available for irrigation DSS in Australia. The number of Internet-based sources has grown since 2007 making a longer list than given in Section 5.3 however there is no catalogue or even a general method available to discover and use irrigation data sources.

Current work within CSIRO Land & Water (Sommer et al. 2018) is looking at translating data source's data in Internet-available data in proprietary formats – as implemented by most field equipment manufacturers – to standardised ones. While the general idea of standardising agricultural equipment data formats is not new, the latest approaches do new things:

1. Standardise both the structure of the data and its semantics using a sophisticated observations model;
2. Use newly available Internet data manipulation mechanics.

Most previous agricultural data standardisation attempts focussed entirely on the structure of the data (e.g. how to encode time/value pairs) or use a very simple observations model, perhaps only relating a value and with a type label and a unit of measure. In addition to using the Sensor Observation Service (Open Geospatial Consortium 2007) for data structure, Sommer et al. (2018) uses the Observations & Measurements standard (ISO 2011) to associate measurement results with observations of phenomena and their associated features of interest. Only recently libraries of references of properties and features have become available (Cox, Simons, and Yu 2014; Yu et al. 2015).

The system uses an on-the-fly translation of data from proprietary formats to standardised ones. This has only been technically possible for distributed (non-warehoused), large data volumes recently due to faster national (in Australia) Internet
speeds with which to transmit data and faster data processing tools, such as asynchronous Python web data toolkits.

These approaches address issues of data commensurability, referenced above, and technical access issues given current data loads respectively and while they have already been a large amount of work, they are clearly nowhere near solving these issues, despite building on now almost a decade of work. So, in addition to the prerequisites for taking these two approaches not being available for work carried out in 2007 – 2010, the recent work shows that the original scope to try and substantially enhance just the commensurability issue alone was too optimistic.

5.8. References


---

22 The Sommer system uses Python’s Sanic library (https://sanic.readthedocs.io/en/latest/) to read and translate data from multiple sources at once while generating graphs of related information for users, something Internet servers have only been capable of recently.


6. Decision Modelling

Early in this PhD it was recognised that serious considerations of decision making and decision theory were likely to play a role in addressing some aspects of DSSs lack of uptake. Within this thesis’ Literature Review, investigations into decision theory were presented and some thought was put into how decision theory might specifically impact decision making in the irrigation context (Chapter 1, Section 1.2.2). Some outcomes from these early investigations into decision theory are also listed (Section 1.2.2.2).

Work on decision theory was put aside during the establishment of systems such as IrriSatSMS (Chapter 2) since the focus of study at that time was on different forms of advice and different methods of advice delivery.

In 2011, work on methods to include a wider range of decision input data sources into DSS was undertaken with results presented in the conference paper:


The work to incorporate a broader range of decision sources into irrigation DSS was ceased after the paper above was written. This was done in favour of the author taking a step back from DSS mechanics and instead focusing on the commensurability of decision inputs, part of the consideration of which was the framing of decision inputs within a decision scenario itself, modelled using decision theory. The reason for this change was the author’s reflections on data source integration issues in other work, particularly in Semantic Web applications (see Chapter 5, Section 5.4 & 5.5).

Referring to the Literature Review conclusion that “there are large gaps in knowledge within current irrigation DSS research in Australia relating to...[the] use of decision theory DSS” (Section 1.2.4), and the research topics “Better decision Theory Use” (Section 1.4), the work in this chapter was undertaken to bring decision theory into focus within irrigation decision making.

One early realisation was that there have been no publications specifically addressing formal decision modelling in the irrigation context or, for that matter, in the agricultural context. Formal decision modelling, of the sort missing in literature and now presented in this chapter, is not synonymous with decision theory; the latter addresses many more aspects of decision making than just the representation of processes or information objects, such as decision maker psychology. Nevertheless, formal process or information models of decision making are needed for the serious analysis of decision making. For example, comparing multiple decisions made across space and time and different contexts (different crop/irrigators etc.) requires formal models of decision making for the comparison of instances which, once obtained, can be used in decision theory analysis to discern decision making patterns. Formal decision modelling is then a vehicle for the ultimate inclusion of decision theory principles into DSS if decision theory analysis, on top of modelled decisions, can be folded back into DSS design.
The papers and resource making the rest of this chapter address decision modelling in the irrigation decision making and DSS context.

6.1. **Paper 1**

This first part of chapter is the original submission for the journal article *Using decision models to enable better irrigation Decision Support Systems*. The final paper, which is essentially the same but omits detailed system descriptions to conserve journal length, is published as:


6.2. **Paper 2**

The second part of this chapter is the conference paper *Modelling causes for actions with the Decision and PROV ontologies*, as published.

This paper was written to progress satisfaction of the requirements for a future decision modelling system for irrigation DSS designers, as outlined in part 8 of the paper in Section 8.1. Specifically, this work builds a "Technically up-to-date" (Req. 1) decision modelling system which is "Model-based, format enabled" (Req. 2), is "Well documented, including links to decision theory" (Req. 6) and is "Interoperable with other modelling systems" (Req. 7). Future work is to have this ontology "Backed by a well-respected standards organisation" (Req. 8).

The citation for this paper is:


6.3. **Online resources**

The third part of this chapter is a listing of online resources built to further address the requirements listed in the paper in Chapter 6.1. The purpose of each item listed (the requirements each addresses) is given.

6.4. **DecPROV irrigation example**

The fourth part of this chapter is a modelling of the simple irrigation question "How much should I water today?" using the DecPROV ontology. This follows on from the work in Section
6.1 where that question was modelled using Decision Modelling Notation, the Decision Ontology and the Decision-Making Ontology.

This modelling is included in its own small section as no irrigation example is given in Section 6.2, the conference paper that introduces DecPROV, due to the need for that publication to build on medical decision scenario examples.

This modelling example is also available online in the online repository containing DecPROV which is listed in Section 6.3 and referenced in Section 6.4.
Review

USING decision models to enable better irrigation Decision Support Systems

Nicholas J. Car
CSIRO Land & Water, Dutton Park, Queensland, Australia

ABSTRACT

Many attempts have been made to enhance irrigation decisions using Decision Support Systems (DSS). These have met with limited success for many reasons, one of which is well known: that DSS encode decision rules (waterbalances, financial models) narrower in scope than the criteria farmers really use to make decisions, thus their advice is of limited value or perhaps entirely irrelevant.

To assist irrigation DSS designers build more flexible systems, we suggest they heed decision theory and decision modelling, separately from domain-specific DSS tasks. They may then find better ways of modelling real-world decisions which might allow for wider ranging sets of decision rules than previously.

To facilitate this, we review three different decision modelling systems and with each model the seemingly straightforward irrigation decision “How much should I water today?”. In doing this we show how they can assist with wide-ranging rule integration. The systems we chose are: Decision Modelling Notation (DMN) from the business analysis community; the Decision Ontology (DO), a Semantic Web modelling system; and Decision Modelling Ontology (DMO) a formal ontology from Information Systems Engineering.

We have determined that each of these modelling systems have useful aspects for irrigation DSS designers, which we list, but that they are not equally useful. Also, none of the systems provide designers with both the best modelling system and best technology tools. We complete our work with a list of requirements for a future decision modelling system based on the intersection of the strengths of the systems investigated and our perceptions of irrigation DSS need. We believe a future system is possible to make and could serve irrigation DSS designers better than any current system.

In future work, we indicate what steps might be taken with existing systems to evolve them in line with our future system requirements. Finally, we conclude with a summary of our findings.

1. Introduction

Designers of Decision Support Systems (DSS) for agriculture and the more specialised field of irrigation have long been able to prove that they can improve decision outcomes for users (Car et al. 2012), this author and more recently (Giusti and Marsili-Libelli, 2015, among many others). Despite this, (Mackrell et al., 2009) present a long list of publications by DSS designers lamenting uptake in Australia, especially amongst small business irrigators.

Deep analysis of this seeming paradox has been undertaken within Australia (McCown et al., 2006) and, also internationally (Matthews et al., 2008) since this problem is identified world-wide. Technical and social researchers within Australia have made multiple efforts to understand irrigation decision making there, specifically (Jakku and Thorburn, 2010), (Montagu et al., 2006), (Whittenbury and Davidson, 2009). Reasons given for this paradox include a perceived research practice/farming practice gap, socio-technical perceptions of trust and limited on-farm technology access.

One aspect of many of these reasons that is easily understood is the narrowness of DSS’ decision logic compared with ‘real world’ decisions which forms part of the practice gap. DSS, such as that described in (N J Car et al., 2012), base their support on biophysical models only, not the much wider set of influences on a farmers’ irrigation decision making that are well known, thus even when the DSS assist with better bio-physical outcomes, they may be of no or negligible benefit to an irrigator overall. Possible technical DSS design methods to address this narrowness, specifically for irrigation DSS within Australia, have been proposed previously (Car et al., 2009; Car and Moore 2011b, 2011a) but few built. The methods proposed in these system designs focus on standardising data source representation for easier DSS data consumption, thus enabling the building of better capacity within the DSS's internal systems or interface to cater for a wider range of decision influences.

In this paper, we step away from explicit DSS design and irrigation DSS-relevant data management to review the state-of-the-art of formalised systems for modelling decisions themselves. We do this because...
we believe that many irrigation DSS that have been built, including some by this author, have not leveraged good knowledge of how decisions are actually made which is decision theory.

In Section 2, we give a few notes on the history of decision theory; how its use presents us with the opportunity of exploring new ways to assist with irrigation DSS adoption and our criteria for choosing the systems we reviewed.

In Sections 3, 4 and 5 we present reviews of the three decision modelling systems followed by a general discussion about the systems with direct comparisons in Section 7. We also present what we believe to be sensible requirements for a future decision modelling system. Finally, in Section 9, we conclude with a summary of our review findings and our reasons for proposing requirements for a future system for irrigation DSS designers’ use.

2. Decision theory and computerised decision making

Decision theory – theory about decision making – as a philosophical or academic discipline has some of its roots at least as far back Aristotle’s systematic investigations in Prior Analytics (Aristotle 350AD). The more statistical elements of decision theory stem from at least the 18th and possibly the 13th centuries CE (Wallis, 2014).

2.1. Decision logic

Logic and reasoning have received continued interest since Aristotle with many philosophers contributing including Averroes in the 12th C with extensive comments on Aristotle’s work (Averroes 12th C). Averroes defined decision logic against his contemporaries who sought to rely on non-logical decision making, such as direct instruction from religious texts (Bole, 2016). In the 17th C term logic was introduced (Arnauld and Nicole, 1662) using rigorous semantics for logical propositions and requires differentiation between instances of things and concept classes of possible instances (Buroker, 2014). Such structured semantics and class/instance differentiation are particularly important for modern decision modelling systems. Philosophical logic has since become more suitable for automated reasoning with works by mathematical and early computational luminaries, including Leibniz, Boole (of “Boolean” logic) and Russell yielding the reasoning mechanism used currently by computers. This history of codifying logic has moved from ambiguity to certainty in the expression of decision concepts and from qualitative to quantitative processes for calculating decision outcomes.

Sets of logical propositions expressed for decision making – “decision rules” – can be used to calculate a result given a particular set of inputs; a “decision scenario”. Collections of such rules were introduced to computer processing relatively early in the field of computer science, 1960s at least, with programming language extensions such as FORTAB (Armerding, 1962) using “decision tables” and enabling calculations with them. Many modern programming languages still implement FORTAB-style tables, for example, current versions of the popular Python programming language contain a module called decisionTable (Uroš, 2015).

2.2. Decision processes

Theories for procedural decision making and scenario representation – not the formalised logic of weighing choice elements – are often reported as having been started by the Marquis de Condorcet drafting the French constitution of 1793 (Hansson, 1994), specifically his essay on voting systems (Condorcet, 1785), however some sources indicate a 13th century origin (Wallis, 2014).

General, systematic, procedural methods to assist individual decision making were proposed in the early 20th century by Dewey (1910) who suggested segmenting deciding into “five logically distinct steps”:

(i) a felt difficulty;
(ii) its location and definition;
(iii) suggestion of possible solution;
(iv) development by reasoning of the bearings of the suggestion;
(v) further observation and experiment leading to its acceptance or rejection; that is, the conclusion of belief or disbelief.

Computer-based systems designed to systematically progress through scenarios (“states”) have existed from the 1940s with models such as finite state machines (Aziz et al., 2014). Modern programming environments that are effectively finite state machines, such as Microsoft’s Windows Workflow Foundation (Chappell, 2009), have been used in Australian irrigation DSS (Inman-Bamber and Attard, 2007) and likely many others DSS use of similar tools.

While state progression, decision logic and process flow are fundamental to modern computer operations, systems designed to model human decision making specifically, with steps similar to Dewey’s, have come to exist only more recently. The three systems represent some of the earliest decision modelling-specific computational systems.

2.3. Recognising where decision theory might assist

One of the reasons for irrigation DSS’s narrowness is the difficulty DSS designers have in “bolting together” seemingly incommensurate units of logic. For example, waterbalance and personal effort considerations relating to the decision “How much should I water today?” do not easily lend themselves to being optimised in standard equations. By considering both the logic and process aspects of decision theory, as outlined above, we hope to assist with this “bolting together” problem and others related to it.

Another well-recognised “gap” problem is that of on-farm DSS access (timeliness and ease of DSS use) that does not match laboratory conditions. While DSS have been built to cater for outdoor farming life and ease of use (Hornbuckle et al., 2006) and close-to-ubiquitous support for computerised DSS has been realised in some locations (viz. Internet & mobile technology use in Australian households 2008–2015 in (Australi an Bureau of Statistics, 2009) and (Australian Bureau of Statistics, 2016)) decision theory may indicate better ways of staging DSS use or incorporating it into daily farming life.

Systematic, system-independent, decision representation using decision theory may also allow an irrigation knowledge base to be created which may assist in establishing decision best-practice.

2.4. Decision modelling system selection

Here we look to computerised expressions of decision theory only so to directly incorporate its lessons into future DSS. Multiple computerised decision theory-based systems exist and we explore three in this paper:

1. Decision Modelling Notation (DMN)
2. The Decision Ontology (DO)
3. The Decision-Making Ontology (DMO)

These systems represent decision logic and processes in system-independent and standardised ways. The first system, DMN (Object Management Group, 2016), is the most widely adopted decision modelling system we could find. It has a large, professional, user base and support network, mainly within the business analytics community. The second, DO (Nowara, 2017), while still in development and with a negligible user base, has been developed under the auspices of a rigorous and renowned information standards body, the World Wide Web Consortium. The third, DMO (Kornyshova and Deneckère, 2010), has had many years of rigorous academic development within a non-
agricultural computer science-related community – that of Information Systems Engineering – and references much work in the Operations Research and management science fields which are well known with respect to decision modelling.

With each of the three systems selected, we have modelled a demonstration irrigation decision answering “How much should I water today?” with qualifications, where appropriate, to indicate aspects of the systems’ use.

3. Decision modelling Notation

Decision Modelling Notation (DMN) (Object Management Group, 2016) was designed specifically for decision modelling within other business processes and thus integrates with process modelling tools, particular the widely known Business Process Model and Notation (BPMN) (Object Management Group, Inc. 2011). It is supported by standards bodies and also multiple companies, including tool vendors. Some companies support dedicated DMN tooling, such as "DecisionsFirst Modeler" and others support DMN within multi-paradigm modelling tools².

3.1. Using DMN

DMN allows for the modelling of the “logic” used in decision making as well as processes and object surrounding decisions which it terms “requirements”. This split in DMN’s capabilities, which reflects the split in Decision Theory’s sub-disciplines as indicated in Section 2, is promoted as a strength allowing for the compartmentalised representation and consideration of how decision makers weigh options and the broader context in which they do so.

DMN models requirements in a workflow-like manner using a standardised set of components, the diagrammatic forms of which are used in Fig. 1 in which a simple, but likely not very useful, requirements model for the irrigation question “How much should I water today?” is given. This figure implements Dewey’s first and second process; the first, identifying a “felt difficulty” (the decision), and its location and definition (in relation to available information). The second process, “suggestion of possible solution”, requires decision tables, as related below.

Despite Fig. 1 not utilising all DMN elements, some useful modelling has already occurred: placing process modelling before logic modelling has allowed incommensurate factors affecting the decision (i.e. factors unlikely to be able to be directly weighed against one another) to be represented in a qualitative way, such as Crop stage (biophysical) and Expected harvest price (economic). Also, decisions modelled as per Fig. 1 have a standardised, machine-readable, serialisation format which allows DMN models to be stored in repositories, referenced, reused, and validated.

Decision logic in DMN is presented in decision tables as per Section 2.1 and DMN implements an expression language, known as FEEL (Friendly Enough Expression Language) for automated calculations (Object Management Group, 2016). Extensions/replacements to FEEL are allowed. A possible decision table for the question modelled in Fig. 1 using FEEL is given in Table 1.

A textual description of the logic presented in Table 1 would be something like:

- Do not water if:
  - there is zero water availability
  - the water trade price is greater than the expected crop price
  - if the likelihood of rain is > 75%
- else, do water, in with varying proportions of the waterbalance calculation value up to a maximum of daily pump capacity, dependent on rain likelihood and crop stage

DMN can easily be placed within an organisation’s or a person’s larger BPMN business process model thus, it may be placed within modelled wider farming practice which may help to promote the value of a particular style of DSS use, such as daily use. It may also indicate where/when within farming practice DSS should be used which could reduce the ‘gap’ between laboratory and real-world use.

3.2. DMN used for irrigation decisions

One of the purposes of using DMN is to attempt to simplify, or at least make explicit, all the elements within complex decision scenarios. Given the different types of answers arrived at in Table 2: A Possible Decision Table for the decision “How much should I water today?”, as represented in Table 1, the binary do/do not water for some results and then the graduated numerical value if watering for others, the decision represented in Fig. 1 appears to be suitable for decomposition into multiple, dependent, decisions. Decision decomposition is indeed recommended in DMN modelling guides to make the logic underlying decisions more explicit (Silver, 2016a). Fig. 2 splits the single decision represented in Fig. 1 into three separate decisions. This allows the binary question of “Should I water today?” to be asked and answered before “How much should I water today?” is asked, if at all. A further split is added asking “Should I abandon the crop?” to separate large-scale farm decisions from smaller decisions. Though not further discussed in this paper, it would be perhaps sensible to model this last decision in a separate BPMN process from the first two for consideration at entirely different times within a farming operation.

Decomposition of complex scenarios into simpler ones is a common engineering and IT task. The fact that DMN provides the expressive power to do this for decisions, and guidance in manuals and tutorials on how to do it, is of potentially great utility to irrigation DSS designers in considering how to present complex decisions to users.

Fig. 2 uses different input data for separate decisions and also shares some. DMN requirements modelling initially helps determine what goes where and may also allow for data & knowledge elements, once defined, to be reused not just within a scenario but across different models for different DSS.

Using Fig. 2, clearer decision logic can be represented for “How much should I water today?” than was given in Table 1, partially due to the separation of the How much… decision from the other decisions such as Should I water and partly due to the extraction of other decision components from the single logic table, such as the waterbalance calculations. Table 2 shows simplified logic modelling. Some of the results obtainable in this table’s precursor, Table 1, would be present now in Decision Tables for the other decisions shown in Fig. 2, not this single Decision Table.

In Table 2, note that where the input data of “Should I water today”, which is the output of a prior decision, is “No”, this decision table results in “0”. This could be represented in modelling systems other than DMN such as workflow-like BPMN which, given the output of “No” from “Should I water today” bypass the decision of “How much should I water today” altogether and terminate the decision-making exercise with a 0 result from that. In this table, the results from that prior decision are entirely “fed in” to this decision, thus it’s the outcomes from this decision alone that result in a final decision about how much to water today.

² This tool is a product of the company Decision Management Solutions, see http://www.decisionmanagementsolutions.com/about-decision-management-solutions/

³ Enterprise Architect is an Australian-developed, multi-modelling system development tool with a specific toolkit for DMN, as of version 11. Version 12 (current) DMN documentation: http://www.sparsystems.com/enterprise_architect_user_guide/12/domain_based_models/decision_models.html
3.3. Automated use of DMN within the irrigation decision context

It is possible for anyone to implement tools that use DMN representations in automated ways and vendors have already done so, such as the previously mentioned DecisionsFirst Modeler. In addition, there are a large number of tools that support BPMN and it can be expected that such tooling will increase its support of DMN now that DMN is standardised. There is, however, some recognised lack of uptake and standardisation of FEEL (Silver, 2016b) which likely prevents automated execution of decision logic once coded in that form.

At the time of writing, we are not aware of any open source code libraries or system-independent data models that implement DMN objects that could be incorporated into custom build irrigation DSS but implementing perhaps a database schema to store DMN objects and their relationships would not be particularly hard, given the limited set of elements.

3.4. Potential value for irrigation decision modelling in DMN

By using DMN for irrigation decision modelling, the irrigation community will be entering a modelling community using a sophisticated, mature in many aspects, and used by very many large and small entities. This brings many benefits used to promote the utility of DMN generally and include a better understanding of decision modelling; a better appreciation of the components within certain modelled decisions; standardised ways of describing irrigation decisions; potential reuse of decisions that have been modelled; and access to tools that can work with DMN models.

The inclusion of DMN modelling within broader BPMN modelling must also be considered a benefit if indeed DSS designers need to better place the use of agricultural DSS within real farm practice (Matthews et al., 2008).

With references to decision theory, though DMN doesn’t specifically reference it, it clearly informs DMN design, thus a user of DMN is adopting at least some thought from decision theory. Additionally, due to DMN’s process/logic split, it is possible for DSS designers to re-use decision requirements models for multiple DSS scenarios, or even multiple DSS, even if the specifics of the decision logic change. This may allow DSS designers to build a generic DSS that implements a certain class of Decision Requirements Graph, including all the relevant sources of business knowledge but then adjust only the decision logic, say per crop or per irrigation business.

Separating waterbalance calculations from economic and other calculations likely appears natural to many irrigation DSS designers as, traditionally, these knowledge domains and systems made to work within them have been separate. Having said that, mapping all of the biophysical, economic and other decision components together in the decision requirements graph for the “How much should I water today?” with similar tools to represent their logic – DMN – is new and really does assist with the “bolting together” problem described in Section 1. Due to the fact that DMN models decisions generally, it is entirely appropriate to model biophysical, economic and social decisions altogether.

Table 1
A possible DMN Decision Table for the decision “How much should I water today?”, as represented in Fig. 1. Input Data (items in blue) are taken directly from the figure. FEEL expressions are used for table cell values’ notation.

<table>
<thead>
<tr>
<th>How much should I water today</th>
<th>Table</th>
</tr>
</thead>
<tbody>
<tr>
<td>U</td>
<td>Water availability</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>&gt; 75</td>
</tr>
<tr>
<td>3</td>
<td>[50..75]</td>
</tr>
<tr>
<td>4</td>
<td>[0..10]</td>
</tr>
<tr>
<td>5</td>
<td>[10..50]</td>
</tr>
<tr>
<td>6</td>
<td>[10..50]</td>
</tr>
<tr>
<td>7</td>
<td>[10..50]</td>
</tr>
<tr>
<td>8</td>
<td>[10..50]</td>
</tr>
<tr>
<td>9</td>
<td>[10..50]</td>
</tr>
<tr>
<td>10</td>
<td>[10..50]</td>
</tr>
</tbody>
</table>

Fig. 1. A DMN Decision Requirements Graph for “How much should I water today?”.

Table 2
A possible Decision Table for the decision “How much should I water today?” as represented in Fig. 2.

<table>
<thead>
<tr>
<th>How much should I water today</th>
<th>Table</th>
</tr>
</thead>
<tbody>
<tr>
<td>U</td>
<td>Should I water today</td>
</tr>
<tr>
<td>1</td>
<td>No</td>
</tr>
<tr>
<td>2</td>
<td>–</td>
</tr>
<tr>
<td>4</td>
<td>–</td>
</tr>
<tr>
<td>6</td>
<td>–</td>
</tr>
</tbody>
</table>

Silver, 2016b
4. Decision Ontology

The Decision Ontology (DO) (Nowara, 2017) is a Web Ontology Language (OWL) (W3C OWL Working Group, 2012) model which was presented by members of a W3C group conducting investigations into decision modelling. This ontology, from 2011, has been slightly updated and re-presented online as a website and code repository for the purposes of this paper and for continued development. Just as DMN is interoperable with other business modelling systems, so too is DO by virtue of its base being OWL, however, OWL is a general modelling language and thus interoperable with other business modelling systems. DO is by the author, DO users could implement decision logic using the standardised SPARQL query language (W3C SPARQL Working Group, 2013) for OWL data which is easily able to represent FEEL-like logic.

The DO claims to be designed for two types of use:

1. data-driven: archiving current or historical decision-making processes and their outputs
2. normative: outlining decision-making scenarios

Both uses of the DO can cater for decision requirements and logic. Normative use indicated decision patterns (best practice) that should be followed. Data-driven use can be used to record the results of past decisions made which then, in common with other OWL-based systems, can be stored in a graph database. Stored DO instances can also be combined with instance data modelled in non-DO OWL models to store any other information relevant to the decisions modelled. An example of a decision modelled in the normative use scenario is given in the ontology’s documentation.

4.2. DO used for irrigation decisions

Using DO the irrigation decision “How much should I water today?” (HMSIWT?), split into parts as per Fig. 2, is represented in Fig. 4, initially in a normative use scenario with additions (coloured) to create a data-drive use scenario (as if the template had been followed and a decision made). The “Apply Water” classes are not part of the DO itself but are instances of an action modelled just for this example that satisfies the requirements of choosing particular Options. Note that the result of choosing an option is either the selection of a particular instance of the Apply Water, which the main question “WMSIWT?” indicates (question for indicating), or it initiates another question which will

---

4. Decision Ontology

The Decision Ontology (DO) (Nowara, 2017) is a Web Ontology Language (OWL) (W3C OWL Working Group, 2012) model which was presented by members of a W3C group conducting investigations into decision modelling. This ontology, from 2011, has been slightly updated and re-presented online as a website and code repository for the purposes of this paper and for continued development. Just as DMN is interoperable with other business modelling systems, so too is DO by virtue of its base being OWL, however, OWL is a general modelling language and thus interoperable with other business modelling systems. DO is by the author, DO users could implement decision logic using the standardised SPARQL query language (W3C SPARQL Working Group, 2013) for OWL data which is easily able to represent FEEL-like logic.

The DO claims to be designed for two types of use:

1. data-driven: archiving current or historical decision-making processes and their outputs
2. normative: outlining decision-making scenarios

Both uses of the DO can cater for decision requirements and logic. Normative use indicated decision patterns (best practice) that should be followed. Data-driven use can be used to record the results of past decisions made which then, in common with other OWL-based systems, can be stored in a graph database. Stored DO instances can also be combined with instance data modelled in non-DO OWL models to store any other information relevant to the decisions modelled. An example of a complex decision modelled in the normative use scenario is given in the ontology’s documentation.

4.2. DO used for irrigation decisions

Using DO the irrigation decision “How much should I water today?” (HMSIWT?), split into parts as per Fig. 2, is represented in Fig. 4, initially in a normative use scenario with additions (coloured) to create a data-drive use scenario (as if the template had been followed and a decision made). The “Apply Water” classes are not part of the DO itself but are instances of an action modelled just for this example that satisfies the requirements of choosing particular Options. Note that the result of choosing an option is either the selection of a particular instance of the Apply Water, which the main question “WMSIWT?” indicates (is_question_for_indicating), or it initiates another question which will

---

Fig. 2. A more detailed Decision Requirements Graph model of the decision “How much should I water today?” and other decisions and business knowledge that it depends on. Shown also are Knowledge sources.

Fig. 3. A more detailed Decision Requirements Graph model of the decision “How much should I water today?” and other decisions and business knowledge that it depends on. Shown also are Knowledge sources.

Fig. 4. A more detailed Decision Requirements Graph model of the decision “How much should I water today?” and other decisions and business knowledge that it depends on. Shown also are Knowledge sources.
result in such, as the “Abandon: no” option does. This is always true, in this example, as long as the act of applying no water is also modelled as an Apply Water class instance (“Apply Water volume 0”). In this way, outcomes from all sub-questions of the main question ultimately result in the selection of an object of the appropriate class. In the data-driven use, Apply Water 10 mm is chosen, Apply Water 0 mm is not.

In its current form, while the DO allows for Question classes to have Answers of a particular type that indicate specific subclasses of their type (which could be instances), it doesn’t contain axioms that would allow for instances of decisions, to be validated as having achieved a legitimate result. To validate the data-driven use in Fig. 4, an axiom such as the following would have to be included in the DO:

Questions for indicating must specify the class of their Answer and for every instance of a Decision class containing that Question, the Answer must indicate a Named Individual which is also a subclass of the Answer class.

General methods of specialising ontologies in OWL could easily lead to the creation of specialised ontologies for particular decision scenarios. An example could be an irrigation volume ontology that further constrained the DO to require certain irrigation-specific modelling to be present. Such specialised normative use could be implemented to ensure best practice for certain decisions takes place either by DSS or people. Specialised normative use templates could also be used to assemble
knowledge bases of *data-driven use* instances as a database schema is used to assemble individuals that adhere to that schema.

### 4.3. Automated use of the DO within the irrigation decision context

Since, like DMN, DO is an openly available and standardised modelling system it is also possible for anyone to implement tools for or using DO. Unlike DMN, the OWL set of tools have very many tools, also open, available for use. One can manually implement instances of DO using free ontology editors such as Protégé (Stanford Center for Biomedical Informatics Research, 2017), store instances of decisions modelled in DO directly into graph databases such as Apache Jena (Soroka et al., 2017) and operate over them using the SPARQL query language and reasoning engines such as Pellet (Clark & Parsia, LLC 2011).

In order to incorporate either DO instance collection capabilities or DO data display within custom irrigation DSS, any one of a number of OWL graph data manipulation frameworks, such as Python’s *rdflib*\(^7\) or dotNetRDF\(^8\) for Microsoft systems can be used freely and quite easily. Such frameworks exist for most major programming languages. They also make it possible to store DO (or any OWL model) data in relational or other databases, not just graph databases.

The set of tooling that works with DO allows it to be operated somewhat like a regular database schema that all irrigation DSS designers would be familiar with. Values could be input into a system built adhering to the DO and outputs calculated according to the normative use mode.

OWL and its tooling can be used for sophisticated instance validation, far beyond that which is available to XML Schema-based specifications and even relational database models. This is due to OWL ontologies being able to encode axioms that can be chained and validated with OWL machine reasoning tools such as Pellet. OWL is also growing its validation engines, e.g. *Shapes* (World Wide Web Consortium (W3C), 2017) DMN itself does not appear to have similar sophisticated validators available, only the XML schema available in the DMN specification. BPMN validation, as implemented in some recent tools (Geiger et al., 2017) extends pure XML schema validation with the addition of reference checking and some semantic validation.

### 4.4. Potential value for irrigation decision modelling in DO

Irrigation DSS designers would benefit from using OWL and the associated tooling, as discussed above. Also, if instances of decisions are stored in DO, they can be exported from graph databases in RDF exchange formats such as turtle (Prud’hommeaux and Carothers, 2014) and ingested directly into any others; such is the power of graph-based ontologies that contains their own. This would allow irrigation DSS designers to change with technology and easily persist decision data.

DO is also able to be automatically displayed as graphs with general OWL visualisation tools such as *WebVOWL* (Link et al., 2017), basic script visualisers such as PROV SHOW (Gar, 2017) as well as tools integrated with graph databases (Neo4U (Neo Technology, Inc., 2017)). It needs to be noted that filtering complex ontology instances for particular elements to visualise is very straightforward using a combination of the visualisation tools just listed and SPARQL queries.

As per DMN, some elements of decision theory are incorporated within the DO and using it conveys benefit in that way too. While, on the surface, the DO seems to make less of a distinction between decision process and logic than DMN does, there are separate objects for each in the ontology and filtered views of objects in a graph database are easy to create.

DO users may reuse decision logic in multiple scenarios, as DMN users can, however the Semantic Web principles implemented by the DO also allow any collection of model objects (technically a sub-graph) to be published on the Internet allowing for universal, direct, reuse. This would allow irrigation DSS designers to perhaps create instances of irrigation decisions using the DO in its normative use mode and then for anyone to be able to reuse them, including specialising them as well as for collections of *data-driven use* instances to be used as a knowledge base.

### 5. Decision-Making Ontology

The Decision-Making Ontology (Kornyshova and Deneckère, 2010) has been developed to provide “a representation, shared between researchers and practitioners, of DM [decision-making] concepts and their relations with DM problems in ISE [Information Systems Engineering]”. The model, a formal ontology, aims to represent the three concepts that recent decision theorists identify as playing fundamental roles in structuring decisions. The ontology paper cites (Roy, 2005):

1. *Problematic* – the decision problem formulation, including scenario classification
2. *Alternatives* – potential actions to take
3. *Criteria* – methods used to assess the suitability of Alternatives

The authors of the ontology also stress the existence of a larger body of detailed academic knowledge about decision making extending on the work of (Roy, 2005) and base the particular classes and relationships of their ontology on their summary of that work. This is different from both DMN and DO which clearly exhibit the influence of decision theory but nowhere make direct attribution to elements of it.

The DMO contains a DM situation concept which contains the main DM elements the ontology represents and indicated a process which is undertaken which then uses a CriteriaSet to rank alternatives. Detailed decision logic is able to be represented within the application of criteria to alternatives through the use of weight, preference rule, and threshold information: all forms of criteria discussed in decision theory that the DMO is based on. When quantified, these species of criteria are analogous to forms of DMN’s FEEL Expressions within decision tables. Fig. 5 shows some of the DMO elements used to detail the structured decision-making concepts related above.

Fig. 5 models the problematic (DSSituation), alternatives (AlternativeSet & Alternative) and criteria (CriteriaSet & Criteria) concepts from (Roy, 2005) as well as further elements used to encode decision logic such as PreferenceRule, Weight and Threshold. Adapted from Fig. 1 in Kornyshova and Deneckère (2010).

The DMO organises its elements into six general classes: component; actor; intention; activity and context, and defines relationships between elements in these classes in order to indicate a general decision solving methodology. This is intended to assist with the logical flow of the creation of the more specialised DMO elements and would also allow for high-level ontology axioms to be stated. Some of these classes, such as actor, have no direct representation in either DMN or DO however, given their possible integration with BPMN and other OWL ontologies respectively, those systems can adopt classes for actors and other elements modelled elsewhere. DMO has to include within itself classes for elements such as actors, if they are to be used at all, since it doesn’t integrate with any other modelling system.

### 5.1. Using DMO

With its multiple classes and detailed properties, the DMO is promoted as being able to represent current and well-known decision making approaches, such as the Analytic Hierarchy Processes (Saaty, 1980), an example of which is given in (Kornyshova and Deneckère, 2010). In the example given, a pair-wise comparison of candidate alternatives for a particular problematic – the decision of which candidate
to choose – are given and the candidates ranked according to a set of criteria.

Kornyshova and Deneckère (2010) also give another example of using their ontology, this time for a software system choice for a particular project. This example has the problematic of having to choose one or more relevant products from a set of alternatives and criteria within a set of criteria for this example are given: purchase cost and maintenance cost. Some description is given as to how preference rules apply to the criteria and how they are to be given relative weight. In this case the preference rule is minimum since the desired outcome from the problematic is the choice of a system that incurs the least cost. Since the way of calculating cost with two criteria expressed in currency is a simple sum, the weight used is equal. Finally, the example concludes with some description of the roles that agents related to decisions play, for example, in the establishment of preference rules and also in validation of a final decision.

The DMO’s PreferenceRule, Weight and Threshold classes gives specific and somewhat fixed ways for a user to evaluate options. This is unlike the DO which leaves evaluation mechanisms unspecified (thus powerful but complex) and DMN which uses FEEL expressions which may be extended.

5.2. DMO used for irrigation decisions

DMO modelling of “How much should I water today?” is given in Fig. 6. It includes fewer decision factors than Fig. 1 to show DMO-specific elements: Weight, PreferenceRule and Threshold.

It is apparent from Fig. 6 that DMO is not set up for questions answered with a numerical result. While it may be possible to model Alternatives as a number within a range differently to the single value Alternatives shown in Fig. 6, the DMO focusses on weighing distinct outcomes, as modelled here. It also focusses on weighing distinct Criterion to determine Alternative choice. The combination of Weights (all of value 1, indicating equal weight), Threshold and PreferenceRule shown here equates to a formula of:

\[ \text{water vol} = \min(\text{water availability}, (\text{(crop stage requirement}) - (\text{scaling} \times \text{rain likelihood volume})) \]

To model “How much should I water today?” as a series of dependent decisions, as per Fig. 2, modelling in Fig. 7 could be used. Here the dependency between questions is indicated with a Consequences class instance that results from choosing the “Yes” Alternative to the first question. In a similar way, the 3rd question in Fig. 2 “Should I abandon the crop?” could also be linked.

Classes in DMO contain a series of properties seemingly taken directly from the literature that the authors of the ontology reference in their publication of the model. For example, Criterion may be annotated with a property informationType with possible values deterministic, probabilistic, fuzzy or mixed. Properties such as these indicate that the DMO designers were catering for a range of Criterion types which would imply a wide range of possible PreferenceRules and other model mechanics able to cater for a range of decision logic.
6. Discussion of decision modelling systems reviewed

While there are likely general benefits for irrigation DSS designers using one or other of the systems investigated here, it’s clear that none of them cover all the aspects of the others. This means it would be impossible to realise all possible benefits by using just one of them. It would, of course, be virtually impossible to use more than one system together due to the very incompatible classes of objects they contain, to say nothing of their technical incompatibility.

5.3. Automated use of DMO within the irrigation decision context

The DMO is presented only as an ontology formalised in UML class diagrams and such formulation does not lead directly to executable tooling. Unlike DMN, where an XML schema implementation of the model is given, this model presents none, making it one step further removed from being able to be used in an automated way, although still ultimately possible.

5.4. Potential value for irrigation decision modelling in DMO

This ontology, or at least the publication that describes it, presents concepts from the field of decision theory and its sub-fields such as multi-criteria analysis. While it seems that such theory also informs both DMN and DO, due to some of their introductory text and decision object formulation, no references to such are made in the DMN standard document or the DO code repository ((Object Management Group, 2016) & (Nowara, 2017), respectively), thus it’s harder to find deep reasoning for DMN and DO modelling choices than DMO.

The decision logic classes of DMO – Weight, Threshold etc. – present fixed yet still powerful way to represent some sorts of decision logic. These mechanisms could be expanded to be more wide ranging, more like DMN’s FEEL expressions or even DO’s open-ended logic representation, however having only a few, quite fixed, decision logic classes within an ontology seems useful for its simplicity of use in some scenarios.

Value to the irrigation DSS community might also be gleaned from a deeper analysis of decision theory and DMO does provide a useful point to lead into the theory, starting from a modern information model potentially usable by DSS, rather than a general introductory textbook which might be harder to relate to.

6.1. General lessons for irrigation DSS from modelling decisions

Using any decision model would force an irrigation DSS designer to grapple with decision modelling systems designers’ methods and could perhaps lead to better decision framing through techniques such as decision decomposition (splitting decisions into sub-decisions, as per Fig. 2).

Using decision models, DSS designers may also implement components and perhaps even calculating systems (databases) that can be reused in future systems. For example, after building a DSS for the “How much should I water today?” question, irrigation DSS designers could perhaps re-use crop economic models for strategic planning DSS. This sort of modularised work is currently undertaken in irrigation modelling by systems such as APSIM (Holzworth et al., 2014) but currently not from a decision-centric point of view, instead a systems point of view only. Decision-centred perspectives might be useful in building DSS that focus on answering questions, rather than detailed systems modelling.

An obvious shortcoming of all the systems investigated here is their difficulty in representing graduated decision results: all systems either expect, or are best able to represent, distinct Alternatives (using the DMO term). The “single-“ and “multi-hit” tables of the DMN used to derive an outcome necessarily result in a row or rows being selected to generate an output. Neither the fixed type class-based approach to defining outcomes (in DMO) or the less tightly specified and much more powerful open-ended class-based approach (in DO) is able to easily handle a range result such as X mm of water or Y pumping hours. In the case of the DMN, this is due to calculating logic of the waterbalance type usually being contained within Business Knowledge components, not general requirements modelling.

6.2. Comparing decision modelling systems’ utility for irrigation DSS

DMN is clearly well thought out, broad in scope, widely used and able to represent decision requirements generally and then decision logic also down to a fine level of detail, if required. Regarding requirements: it’s able to represent potentially anything irrigation DSS designers would wish to represent, such as farmer personal preferences and not just waterbalance models, however it’s standardised expressive power is severely limited compared to the DO and DMO due to only a few types of requirements objects (diagram elements) being specified. The few types mean every decision requirement can be represented, but in limited detail. Use of DMN for requirements by irrigation DSS designers would likely see them needing to generate many more classes of
object or perhaps take them from other, related modelling systems like BPMN. Regarding decision logic: the DMN logic tables are easy to use and well understood but very simple compared with the class-based approaches taken by DO & DMO, especially DO since it’s open-ended, given the integration with OWL. DMO’s FEEL expressions are really limited to what, in ontologies such as DO, would be regarded as simple types (data classifications and numerical operators, not complex objects with complex comparison operators). Despite even the simple type level of detail a in DMN’s FEEL language, it is apparently known to suffer from a lack of detail such that implementers use their own literal expressions questioning FEEL’s claims to be a standard (Silver, 2016b). So, due to the lack of specificity in DMN, while it can be, and is, used in automated decision-making systems, it forces implementers to “innovate”, with respect to logic representation, which is a bad thing in the use of standards. The option, granted by DMN, of implementing one’s own expression language, while powerful, is also “innovative” in the negative sense from a standardisation point of view. Additionally, DMN doesn’t use the latest or best modelling standards (no Semantic Web use; no use of data exchange formats more recent than XML) so that it’s not able to be directly implemented in powerful calculating technology such as a relational or graph databases. It can be, and has been implemented in such by proprietary tools such as DecisionsFirst Modeler but is not able to be done so easily or cheaply and certainly not in an open, standardised way.

The DO clearly has great, but unrealised, potential. It uses modelling techniques that are incredibly expressive due to their extensibility, that are easily integrated with a huge set of other models (OWL ontologies) and directly implementable in automated systems due to the Semantic Web set of standards. It also has from a well-respected standards organisation, the W3C. It does, however, have problems which make it unlikely to be able to be used in its current form. Firstly, it is not already integrated with mainstream OWL models, such as PROV, and OWL ontology users, such as this author, now take that integration for granted since integration of models is part of the Semantic Web promise. Secondly, it does not easily allow for the input or the generation of something like decision tables which, while less expressive than its class-based approach to logic modelling are well understood and cater for many decisions. Thirdly, it is long-winded; in order to model simple decisions, a series of framing classes, such as Decision, Decision making, Process and Situation all need to be implemented. It’s not clear, from the very limited documentation, what exactly the role of each is and which are strictly required, and when. Fourthly, the limited documentation is a big problem for general use, however this would be relatively easy to solve. To some extent, papers like this do that. Fifthly, DO is close to providing mechanisms for decision verification but doesn’t actually do so. As indicated in the discussion following Fig. 7, some small additions to DO could do this and provide a much sought-after function for DO not well handled by DMN or DMO.

Despite the claims of the DMO authors that Information Systems "engineers could use this [the DMO] ontology in every case of DM" (Kornyshova and Denekere, 2010), the DMO is extremely lightly documented, shows no evidence of actual use and likely not developed since the 2010 or perhaps a 2011. It has some detailed modelling elements which would likely be able to be used to represent decision logic effectively, especially modern, named decision situations, such as Analytic Hierarchy Processes, as described in the paper, but many of the properties specified for the model’s classes lack illustrative examples to indicate expected use of formal, semantic, property descriptions to constrain use. It also lacks a formal model document, other than the pictographic UML model and Appendix containing class and property textual descriptions meaning some aspects of the model are not specified. Its self-contained nature prevents DMO from allowing open-ended decision logic like DO while its mechanisms are more expressive than those of DMN without BPMN. The self-contained nature also limits, or does not make explicit, the relations DMO objects would have to non-DMO objects in other systems.

Neither DMN or the DO reference academic or other work to back their modelling approaches. While the DMO references recent academic work on decision theory, it doesn’t reference basic decision logic or other computer models. This means the three systems don’t nicely sum up decision theory for potential users in text while they clearly do in their constitution.

Table 3 summarises some of the aspects of the three decision modelling systems just related.

7. Requirements for a better decision modelling system

While there is much benefit for irrigation DSS designers to be had in adopting any one of the systems investigated here, none of them is seen as ideal or being able to full replace one of the others. This indicates that a future system could, and probably profitably should, be built for irrigation DSS designers or an existing system evolved. Such a system would likely be beneficial to many other sorts of DSS designers and those interested in decision modelling, not DSS’ per se too.

The requirements for a new or improved system are to display the total set of all three considered system’s advantages and they are that a better system be:

1. Technically up-to-date
   - To allow for greater concept representation and automated reasoning available in recent systems.

2. Model-based, format enabled
   - Particular syntaxes/formats never suit all users. A future system should be model-based and allow for many formats as the PROV Data Model does (Moreau and Mislier, 2013): it is expressed in UML and presents implementations in RDF and other formats.

3. Precise with its ontological relationships
   - Precise definitions and relationships between model elements so there is no ambiguity in class relations, property values or cardinalities is important to ensure consistent model use which is required for good standardisation.

4. Powerful regarding logic representation
   - A future system could improve on the logic representation present in the three systems investigated here.

5. Able to verify instances
   - A future system should provide a mechanism, within the model, not an implementing system, to verify modelled past decisions.

6. Well documented, including its derivation from accepted decision theory
   - It is very hard to understand how to use conceptual models such as ontologies by inspection alone.

7. Interoperable with other modelling systems
   - For integrating into business, scientific or other models within DSS

8. Backed by a well-respected standards organisation
   - For wide acceptance.

8. Future work

Requirements for a future decision modelling systems could be met by evolving existing systems rather than creating a new one. Clearly DMN has large scale adoption and thus it is likely to be attractive for future use, however it will also be hard to evolve. DO could be evolved and the willingness of its authors to update its online presentation recently suggest this is a possibility. DMO shows little signs of life and evolution of it would likely have to be done without the original authors.

---

9 This tool is a product of the company Decision Management Solutions, see http://www.decisionmanagementsolutions.com/about-decision-management-solutions/
Regarding DMN, it could:

- Be made model-based, format enabled (Req. 2) with mapping to non-XML formats
- Be made more technically up-to-date (Req. 1) by the same method as above
- Have its FEEL expressions tightened up (Req. 3) (this may already be planned) and expression language(s) implemented in line with the mechanisms used in DMO (Req 4)

In its current form, it would be very difficult for DMN to adopt complex object logic like DO or to implement verifiability (Req. 5).

Regarding DO, it could:

- Be made verifiable (Req. 5), as per discussions around its data-driven use
- Be better documented (Req. 6) with more examples. Partly done for this paper
- Be made more interoperable (Req. 7), with recent OWL ontologies, by mapping to them
- Be backed by the W3C (Req. 8) but possible a Working Group reconsideration that would require the above point and implementation testing
- Include optional decision scenario constructs from DMO (Req. 9) to enable simple decision logic modelling

Regarding DMO, it could:

- Be made technically up-to-date and model-based, format-enabled (Reqs 1 & 2), as per DMN
- Have its ontological relationships tightened (Req. 3) by issuing the model in a semantic format, not just as diagrams
- Be better documented (Req. 6) with many more examples
- Be mapped to other models (Req. 7) but particular models would have to be chosen
- Backed by a standards body (Req. 8), but one would have to be sought

9. Conclusion

Decision theory has lessons to teach irrigation DSS designers about how to best represent decisions and place them within experiential context, regardless of the specific domain. Lessons, if they are to be found, are most usefully conveyed through the use of a decision modelling system since DSS designers are system implementers and may incorporate system methods or components in their work. Such decision modelling systems do exist, are current and available for use.

DMN is useful to disentangling multiple, dependent, decisions, to determine multiple, relevant, sources of decision logic for questions and to add rigour to decision specification generally by the use of its well-considered formalisms. We encourage irrigation DSS designers to use DMN to model decisions and to also consider the use of BPMN to model the wider context of their DSS use.

While the DO is much less mature or supported than DMN, it offers by far the best data structures for encoding complex decisions and, with additions, mechanisms for normative use. It also offers the best, as yet unrealised, integration with a very large body of other models (Semantic Web ontologies). Even its current form offers by far the most direct and only open source data management as well as the best exchange tools. It is also the system investigated here most easily implemented in free, open source off-the-shelf or custom-made systems allowing for automated use.

We believe that DMO has little to offer irrigation DSS designers in terms of an implementable system however the extensible class-based yet fixed approach it takes to modelling decision logic is worthy of consideration for custom DSS builds, or for use in future decision modelling systems. Lessons from decision theory can be learned by its use and its documented examples.

None of the decision modelling systems reviewed are perfect for irrigation DSS use and a better system that draws from the three system’s strengths can easily be imagined. Requirements for such a system have been given to guide future development.

References


DecisionTable (version 0.0.3). Python. https://pypi.python.org/pypi/decisionTable.


Modelling causes for actions with the Decision and PROV ontologies

N.J. Car

Geoscience Australia, Symonston ACT 2609
Email: nicholas.car@ga.gov.au

Abstract: Provenance modelling systems such as PROV and its precursors, PML and OPM, do not have specific mechanics to deal with causes for actions. They can represent what information was used within processes and who did what, when, but not why. However, modelling human decision making to represent why has long been of interest to philosophers, politicians, mathematicians and, more recently, computer scientists and multiple decision modelling systems have been implemented over time. Aristotle in ancient Greece, codified representations of decision logic. The Marquis de Condorcet in 18th century France, proposed fair voting systems for elections. John Dewey in the early 20th century described logical step-wise decision-making processes. Computer models for aspects of decisions have existed as long as computers and recently, several standards groups have implementing specifications about how to record decisions such that knowledge of them may be shared and understood by others.

In this paper, we first demonstrate conceptually mapping part of a decision representation system, the Decision Ontology (DO), to the ontology version of PROV, PROV-O. We indicate that a complete mapping is not possible due to the DO modelling templates for possible future decision making and PROV-O only dealing with past actions such as decisions already made.

We deliver our DO/PROV-O mapping in a step-wise manner by first modelling a decision using DO, then extending the model to include extra concepts for agency and then re-modelling it as PROV-O. This exercise shows that some decisions modelled in depth using DO can be well understood using general PROV-O provenance terms.

Next, introduce a stand-alone ontology, DecPROV (http://promsns.org/def/decprov) which is a specialisation of PROV-O that captures the elements of DO we can map to it. We have chosen to implement a new ontology rather than creating a new version of DO due to the incompleteness of the mapping. We indicate why we have made certain modelling choices where they might be controversial. We also model our example decision in DecPROV.

In addition to showing that certain detailed decisions can be modelled in DO and PROV-O and thus DecPROV, we describe several common provenance scenarios and indicate causes for the generation of elements within them using a few DecPROV elements.

We conclude with a few thoughts about some of the aspects of decision modelling that the exercises related here have unveiled.

Keywords: decision theory, decision ontology, provenance, provenance ontology, ontology alignment
1. INTRODUCTION

PROV-O (Lebo, Sahoo, and McGuinness 2013) is the Web Ontology Language (OWL) (W3C OWL Working Group 2012) version of the PROV Data Model and is the only provenance representation system recognised by an international standards body – the World Wide Web Consortium. It is a generic process model that, in its most basic implementation, uses three classes, Activity, Agent and Entity to represent steps in processes, the causative actors for them (human or machine) and the inputs and outputs (real or conceptual) of those processes. PROV-O does not contain specific classes to describe motivations for a particular course of action although this may be inferred by a representation of a process that includes instructions in Entities that were consumed by Agents who performed it. PROV-O can also only represent things that have been: it has no way of creating templates for prospective actions like many workflow models do, thus it cannot validate a provenance record according to a desired plan. A recent addition to PROV attempts to allow for “provenance templating” (L. Moreau et al. 2017).

The Decision Ontology (DO) (Nowara 2012), also formulated using OWL, was created “for describing [human] decisions and decision-making”. It was published by a W3C Incubator Group in 2012 in order to be considered as an input to a process which may have resulted in a Recommendations (a standard) but this process was never completed. The DO contains classes and relationships that it claims can represent decisions made – data-driven use – and can outline decision-making scenarios – normative use. Combining the two uses, allows validating actual decisions against predicted decisions. In data-driven use, Question, Requirement and Option classes, among others, can indicate why a particular decision outcome was selected. In normative use, those classes can be used to make explicit why some decision should be made or at least what outcome from a decision should be expected. The DO presents elements for describing decision requirements – the information flows and processes relevant to a decision-making scenario and some elements for describing decision logic – the rules used to select one option over another, however it leaves logic description deliberately open-ended which means decision logic descriptions can be powerful – anything you like – but difficult to implement and compare. The DO is dependent on a Requirements ontology (RO) module for some logic modelling but this and the DO are treated as one in this paper due to their interwoven use and the RO’s small size. Figure 1 gives class models for PROV-O and DO and introduces symbols used throughout figures in this paper.

Other decision modelling systems exist, including the Decision Modelling Notation (DMN) (Object Management Group 2016) and the Decision-Making Ontology (DMO) (Kornyshova and Deneckère 2010). DMN is UML-based, simple, mature and very widely used within business process modelling. DMO is also UML-based and uses recent decision theory to justify its element. It is made for the information systems engineering domain but it shows no evidence of real use and lacks proper documentation.

We have chosen to work with DO here as it is the only OWL-based system we could find and by staying within the OWL modelling universe, we can formalise DO to PROV alignment. It would be possible to conceptually map DMN or others to PROV-DM and even map serialisations of DMN in XML to PROV-XML (the XML serialisation of PROV-DM).

2. OUTLINE OF THE DECISION ONTOLOGY

The DO models decision-making as a process within which situations are encountered and data elements considered. DO Options are encountered and indicate, perhaps based on Requirements, the selection of an OWL
Modelling causes for actions with the Decision and PROV ontologies

Thing; something requiring more specification on a per-decision basis. It could be an object or the starting of a new process, including another decision-making process.

As the DO was designed before PROV-O, its general process model was not considered and no PROV-O-like conceptualisation in it exists.

3. SCENARIO – DO DOCUMENTATION EXAMPLE MAPPED TO PROV-O

The DO documentation supplies a single example of use and it demonstrates normative use. Here we convert that example to a data-driven use scenario by adding decision outcome and phrasing it as if it had taken place. Figure 2 shows a diagram of that DO example, a reading of which is that a decision maker (unrepresented) enters a Process to decide about a therapy which is initiated by a Question for indicating. The outcome of this Process is a specific Recommendation to do something. The Options considered by the Decision Maker (only two of the original example’s three are shown) are Choosing penicillin and Choosing amoxicillin. These Options have various Requirements, some of which multiple Options share. Requirements are satisfied by the presence of objects with certain properties. In this example, these objects required are classes of types of Patient.

Figure 3 shows the core of Figure 2 with additions, using DO constructs, to arrive at a data-driven use scenario. Various elements in Figure 2 are not presented in Figure 3 such as descriptive superclasses of core classes and the Questions for confirming that the Deciding about a therapy process initiates. The Question classes are removed because the DO example does not link them to Option’s actions, despite obvious associations seen on inspection due to labels.

Added in Figure 3 in blue is Answer Y indicating an instance of the Therapy class, Amoxicillin. The Question for indicating a therapy required that an instance of Therapy ultimately be indicated by an Answer instance, so this outcome is legal. Another Requirement for the Option, Choosing amoxicillin, is also added (not present in the original example) to demonstrate Requirements requiring satisfaction by objects in disjoint classes, in this case Patients who are either allergic or not allergic to penicillin. The instance Patient X is added to the diagram which satisfies the Requirements for Choosing amoxicillin but not those for Choosing penicillin and the wording of the main Process changed to Deciding about a particular therapy for Patient X to indicate a specific situation is being considered, not a generic one as per normative use. To indicate which Option has resulted in an outcome, the involves_choosing of the Choosing penicillin Option (given in the ontology example but not shown in Figure 2) is removed but the Option choosing Amoxicillin is added. Finally, in red, a Decision Maker Z has been added to indicate agency with relationships to the Decision Process and the Answer Y.

The general outcome from the analysis for Figures 2 & 3 is that data-driven use instances of the DO can be generated from normative use instances of DO if Named Individuals are added but that potential outcomes must also be removed to indicate a final outcome.
Car, N.J., Modelling causes for actions with the Decision and PROV ontologies

Figure 3: Core elements of Figure 2 with Question/Answer linking (not in original example), an addition of Patient X & rewording of the Deciding shown in blue and a new Requirement and requirement satisfaction object addition shown in orange. The non-DO Decision Maker object and relationships added are indicated in red.

Figure 4 maps the data-driven aspects of Figure 3 to PROV-O using the graphical conventions of Figure 1 and laid out generally as per Figure 3 to indicate alignment. A complete mapping of DO classes to PROV-O is given in Table 1. Property mapping between DO & PROV-O can be estimated from Figures 3 & 4.

Figure 4: A PROV-O class and property-only representation of Figure 3

One outcome from the analysis leading to Figures 3 & 4 is that data-driven use instances of DO can be mapped to PROV-O. Another, not shown but derivable from Figure 4, is that instances of DO’s data-driven use can be unambiguously represented using PROV-O properties with no need for DO properties if DO classes are used. A third outcome is that the PROV-O process model cannot represent outcomes that were not selected (the objects from Options indicated via involves_choosing) even though DO’s data-driven use can.

Indicating un-selected outcomes using a variant of PROV-O is shown in Section 5.

PROV-O cannot indicate potential relationships. It’s properties for Activity/Entity relationships, for example, stem from either Activities having used or generated Entities which, in turn, stem from an influence between objects of those classes. Potential relationships have no influence. Possible PROV-O specialisation couldHaveUsed or couldHaveGenerated are difficult to imagine given the way PROV-O is phrased and intended.
4. SPECIALISING PROV-O FOR DECPROV

Figures 4 probably contains few modelling surprises for those familiar with PROV-O. The use of basic PROV-O classes with Plan being the only one outside the 3 simplest and the use of unqualified PROV-O class relations. For those familiar with DO, probably the only surprise is the interpretation of the Option class as a PROV-O Activity, implying a process with temporality.

DecPROV adopts the classes of DO listed in Table 1 and their associations with PROV-O with the single change of renaming Option to Option Selection to reinforce its temporal, process, nature. Such an interpretation allows Options to be thought of as Option Selection act of some kind (a PROV-O Activity) which consume Requirements and may or may not produce results, the process which is termed by DO as involves, choosing. DO Questions (PROV-O Entities) can then trigger multiple Option Selections only one of which, when in data-driven use, will be able to produce a result which either triggers further Questions or concludes the Decision Making process with an object of the class indicated by the Answer to the Question for indicating that initiated the Decision Making. This requirement that only one of multiple Option Selection process triggered by a Question produce a result and the requirement for two or more Option Selection processes for any given triggering Question are axioms of DecPROV. These axioms could be used by a future version of the DO for strong normative use validation of data-driven use instances if circumstances describing how instances of the classes of object indicated by the Requirements of Option Selections are realised or not realised (in DO terms, satisfied).

Table 1 Classes of DO and their PROV-O classes to which they are associate by being subclasses of them

<table>
<thead>
<tr>
<th>DO Class</th>
<th>PROV-O Superclass</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>Answer</td>
<td>Entity</td>
<td>Answer is more naturally a thing (imaginary thing), thus an Entity, rather than a subclass of Situation, as originally portrayed in DO.</td>
</tr>
<tr>
<td>Context</td>
<td>-</td>
<td>Not used: all scenarios here can be represented without Context</td>
</tr>
<tr>
<td>Decision</td>
<td>-</td>
<td>Not used: functions of Decision are subsumed by question, Answer and Decision Making.</td>
</tr>
<tr>
<td>Decision, Making</td>
<td>Activity</td>
<td>Decision Making is a time-bound process, naturally an Activity. Renamed DecisionMaking</td>
</tr>
<tr>
<td>Option</td>
<td>Activity</td>
<td>Option is more naturally a process than a subclass of Situation – something to be statically encountered – as originally portrayed in DO. Renamed Option Selection</td>
</tr>
<tr>
<td>Normative Value</td>
<td>Entity</td>
<td>NVs as documented in the DO are meant to indicate norms but the DO only demonstrates the use of Requirements, a specialised form of NVs. Requirements are used by Option processes to determine decision outcomes thus are either an Entity or perhaps a Plan</td>
</tr>
<tr>
<td>Process</td>
<td>Activity</td>
<td>A generic Process is a generic Activity. DecPROV uses only Activity.</td>
</tr>
<tr>
<td>Question</td>
<td>Entity</td>
<td>Like Answer, more naturally an imaginary thing.</td>
</tr>
<tr>
<td>Requirement</td>
<td>Plan</td>
<td>An instruction to an Option process thus a PROV-O Plan</td>
</tr>
<tr>
<td>Situation</td>
<td>-</td>
<td>Not used: PROV-O does not represent situations as it has no state machine-like functionality: everything represented in PROV-O instances occurred in the past.</td>
</tr>
</tbody>
</table>

DecPROV does not use any DO properties for such like has_requirement associating an Option with a Requirement or is_consideration_of associating a Question with Options when PROV-O is used with DecPROV classes. This is because PROV-O basic properties, used with the PROV-O superclasses of the DecPROV classes, deterministically model relationships between DecPROV classes thus it is redundant to specialise PROV-O properties further for DecPROV Specialisation by necessity also diminishes general understanding so it is to be avoided where redundant. Figure 5 shows the DecPROV class diagram and Figure 6 a DecPROV version of Figure 3.

5. PROV-O SCENARIOS AND CAUSATION ANSWERED WITH DO

Figure 7 shows three pure PROV-O scenarios, A B & C where causes for actions are posed. In scenario A, why (as opposed to how) particular Output Data is generated by a Processed Data Activity which consumed Input Data and Software Code is posed. This is answered by adding a DecPROV Question instance, Question X, which influenced the existence of Output Data and was the trigger for starting the Activity that generated it. Detailed modelling of how Question X was resolved to result in Output Data could then be modelled as per the scenario in Section 3. In scenario B, establishing why Input Data, as used in scenario A, was selected for use as opposed to Potential Input Data Unused requires a set of DecPROV Option Selection Activities to have taken place, one of which must have resulted in the generation of Input Data and none of which generated Potential Input Data Unused. The logic used to determine Input Data’s generation and no other object’s is contained within the Requirement instances used by the Option Selections. While the set of objects unused by Process Data is infinite and thus a list of instances of them cannot be related in a provenance scenario, a particular object, such as Potential Input Data Unused, may be easily assessed to have been a possible outcome of a parallel Option Selection to the one that ultimately generated the Input Data. The corollary of this is that the total set of Requirement instance must have excluded it or preferred another to its use else it would have been used.
Scenario C of Figure 7 shows that an instance of an Agent, Nick, undertook an Activity, Took out the rubbish, and why did he did so is asked (perhaps in relation to him possibly doing something else. Here a DecPROV Question is naively linked to the Activity and indicated as having started it, as this may seem appropriate: a question triggers an action. However, on reflection, this doesn’t make sense: Nick must have chosen to undertake the Activity Took out the rubbish in a decision-making process before undertaking it. Thus, the answer is modelled as a set of Option Selection processes, one of which resulted in the PROV-O Plan to undertake Took out the rubbish.

The lack of outcome from the Choosing whether to do modelling Option Selection indicates that Activity was a possibility for Nick but that the total set of Requirements for the set of Option Selection processes eliminated it.

6. CONCLUSION

All elements of DO’s data-driven use can be modelled in PROV-O and some DO classes have been mapped to PROV-O and presented in the DecPROV ontology but no properties have been as they are semantically redundant. DecPROV can both model past decisions in detail and add information about causes for some decisions modelled in PROV to provenance information.

In detailed decision modelling, describing what decision was made is easy enough, but describing what options were presented to a decision maker and then not chosen is hard. The best approach with PROV-O and DecPROV seems to be to model the sets of possibilities a decision maker had before them – the Requirements
Car, N.J., Modelling causes for actions with the Decision and PROV ontologies

classes – and then to model what they were ultimately faced – the instances of objects within the Requirements classes.

Where initial questions lead to sub-questions the construct of DecisionMaking could be applied to each sub-question. This would allow surfacing sub-questions as whole decisions in themselves which might be useful in collections of decisions for the same sub-questions might be present in multiple, different, questions and knowing this could be useful to establish decision similarity.

In the scenarios here, the decisions (Questions for confirming and accompanying Option Selections) always result in one outcome only. Using DMN terms, this would be called a one-hit scenario where decision logic. DMN and other decision modelling tools also allow for multi-hit scenarios and additions to DecPROV could allow that within it too.

One aspect of DecPROV is its ability to model decisions simply (Question, DecisionMaking & Answer) or more complexity (the classes listed before plus Option Selections, Requirements and sub-questions). This allows for stated but unexplained causes or finely detailed causes for actions to be modelled giving users great choice.

Figure 7: Why questions answered using DecPROV (green outline)

REFERENCES
6.3. Online resources

This section lists online resources built after writing the paper in Section 7.1 to progress in implementing the Requirements listed in that paper’s Section 8.

1. Decision Ontology - [https://promsns.org/def/do](https://promsns.org/def/do)
   - Updated version of the 2012 W3C incubator group’s Decision Ontology (W3C Decisions and Decision-Making Incubator Group 2012)
   - Public code repository: [https://github.com/nicholascar/do-ont](https://github.com/nicholascar/do-ont)
     - Includes original machine-readable ontology definition, original usage notes & original example
     - Includes update human-readable ontology definition, ontology diagrams, machine-readable formats and resolvable URIs (namespace URI)

2. DMN Ontology - [https://promsns.org/def/dmn](https://promsns.org/def/dmn)
   - An OWL version of the DMN 1.1 specification (Object Management Group 2016)
   - Public code repository: [https://github.com/nicholascar/dmn-ont](https://github.com/nicholascar/dmn-ont)
     - Includes both human- and machine-readable formats of the DMN specification translated into a Web Ontology Language ontology, templates and a script for converting the DMN specification’s XML expression into an RDF document for the ontology version

3. Decision-Making Ontology - [https://promsns.org/def/dmo](https://promsns.org/def/dmo)
   - An OWL version of the Decision Making Ontology (Kornyshova and Deneckère 2012)

4. DecPROV Ontology - [https://promsns.org/def/decprov](https://promsns.org/def/decprov)
   - Public code repository: [https://github.com/nicholascar/decprov-ont](https://github.com/nicholascar/decprov-ont)
     - Model definition, documentation, examples, link to publication (Ch 7.2)
     - Will include links to published use over time

6.3.1 DO, DMN & DOM ontologies

The purpose of presenting the Decision, DMN and Decision-Making Ontologies online as OWL ontologies is to enable the more direct comparison of their concepts than would be possible comparing against their legacy formats. In addition, the Decision Ontology was rescued from disappearing off the web as its original online locations ceased to be published and a machine-readable version of the DMO was made available for the first time.

6.3.3 References


6.4. DecPROV irrigation example

The DecPROV ontology has had development since its presentation in Section 6.2 paper and changes have been made to its modelling and documentation have been made to the repository listed in 4. above.

Of particular importance is that the DecPROV ontology, as presented in Section 6.2, did not include an irrigation-specific example of its use. This was due to space limitations; the conference modelling example has to build on the precursor DO’s example of a medical decision (see 1. above for the original). Now (2018), the DecPROV repository contains an irrigation example following on from that presented in the paper in Section 6.1 and reproduced below in Figure 1. This example also includes notes on methods of use and extension of relevance to irrigation DSS.

![Diagram](https://promsns.org/def/decprov)  
*Figure 1:* A reproduction of Figure 8 from the DecPROV repository’s examples, see [https://promsns.org/def/decprov](https://promsns.org/def/decprov). Here the decision “How much should I water today?” (HMSIWT), as originally modelled in Section 6.1 using multiple systems is modelled using DecPROV.

6.4.1. Normative & instance use

Figure 1 shows the decision “How much should I water today?” (HMSIWT?) modelled as a `decprov:Question` which then triggers sub-questions: “Should I abandon crop?”, “Should I water at all?” & “How much should I water?”. The guidance to perform this breakdown could be supplied by a normative decision-making template which would be the elements from Figure 1 without the instance data, that is, without the `decprov:Requirement` class objects (3 yellow
ellipses on the right) and without the \texttt{decprov:Answer} (orange ellipse, top right). Fundamentally unanswered by DecPROV or work in this thesis is how a decision maker—whether human or machine—would discover a normative decision template of relevance to them. We can imagine this may occur through searching a database of decision templates, perhaps within a particular sector.

When an instance of decision making using this template has occurred, the facts of the \texttt{decprov:Requirement} class objects and the resultant \texttt{decprov:Answer}, the latter determined by implementing the logic contained within the template (ontology axiom instances), will have been recorded and thus the resultant instance may be inspected for logical inconsistencies and also to inform future decisions.

In order to derive real value from a decision instance for future decisions, some utility of the decision’s answer will be necessary to record. Such recording is a requirement of decisions modelled for use in a Case-Based Reasoning systems (See Section 5.4.5 and Appendix G). Such a result, while currently outside the scope of DecPROV would be a useful addition.

### 6.4.2. Decision requirements & logic

DecPROV does not indicate detail for decision elements such as \textit{where} requirements or other facts (instances of the template classes, present at a specific decision-making event) do or could come from, unlike DMN (see Section 6.1) however such attributions, if needed can be made using regular PROV provenance modelling (Lebo \textit{et al}, 2013) that DecPROV is based on. Such modelling would allow after-the-fact use of decisions modelled in DecPROV to indicate how information was obtained, for example, the “Waterbal indicated 10mm” fact in Figure 1 could come from a particular waterbalance model system, Waterbal X, and thus the PROV attribution \texttt{:Waterbal_indicated_10mm prov:wasAttributedTo :Waterbal_X} could be added to the DecPROV data. Such attribution would require the preservation of system identity, something a DSS that accessed information from multiple sub-systems, like models, could implement.

Decision logic is implied in Figure 1 but not explicitly diagrammatically. One instance of implied logic is that only one of the pair of \texttt{decprov:OptionSelection} classes “Abandon” and “Do not abandon” may generate a result (select an option). This logic could be explicitly coded within ontology axioms and, in this case, it would take form of constraints on \texttt{decprov:OptionSelection} activities, started by the “should I abandon crop?” \texttt{decprov:Question} instance such that a count of results generated by them (options selected) must be equal to one. This could be usefully implemented by creating a particular type of \texttt{decprov:Question} (a subclass of it), perhaps called something like “Question with binary options” whose definition was liked to that axiom and thus existence of instance of it could be validated against decision instance data.

From the previous two paragraphs, DecPROV can be seen to be able to implement both forms of modelling implemented in DMN (see Section 6.1) but using a single modelling system – OWL – unlike DMN which uses DMN & FEEL or other logic languages. This is a strength of DecPROV modelling and OWL modelling generally although it must be admitted, OWL modelling of decision logic is unconstrained and thus less intuitive than FEEL expressions. A FEEL-like ontology could be created as an adjunct to DecPROV with similar FEEL ease of use but works natively with DecPROV (and OWL systems generally). Additionally, expressions of decision logic in tabular form, as per Figure 1, Section 6.1, could be utilised either directly with DecPROV or with a FEEL-like ontology and from there with DecPROV if a mapping from the tabular form to OWL ontology statements was made. There are examples of this sort of mapping, for example the RIF in RDF (Hawke & Polleres, 2013).
6.4.3. Validation of data

Data claiming adherence to DecPROV could be tested by applying reasoning to that data and checking that reasoning conclusions match expectations – this has always been the case with RDFS and OWL ontology data (Allemang & Hendler, 2011) – however new constraint languages made for Semantic Web/RDF applications, such as the Shapes Constraint Language (SHACL) (Knublauch & Kontokostas, 2017) could be usefully employed in creating validating templates both for decision requirements data and also decision logic.

For decision requirements, you could, for instance, validate that a decprov:Question was answered with a decprov:Answer and that the decprov:DecisionMaking triggered at least one decprov:OptionSelection. You could provide specialised templates for specialised kinds of decision, for example a watering decision that would, perhaps, necessarily include a decprov:OptionSelection pairing of types ToWater and ToNotWater.

For decision logic, you could, for instance, ensure that instance data for a decision in which a decprov:OptionSelection, when faced with a decprov:Requirement to select its option (produce a result) when a value of \( x > 5 \), did in fact select that option.

The inclusion of SHACL or other RDF constraint language assets within the DecPROV codebase is considered future work.

6.4.4. Ontology specialisation

With PROV's un-specialised prov:Entity things and prov:Activity events, one cannot easily ask specific decision-related questions like: "What is the Answer for this Question?". Without specialising prov:Entity and prov:Activity classes, such questions could only be answered via attribution matching of instances, e.g. looking for a prov:Entity with a label literal value of "answer" which is unsound OWL/RDFS modelling practice.

In the same way that DecPROV specialises PROV, tailored versions of DecPROV could be made to better represent irrigation decisions. In Figure 1 above, no specialisation of DecPROV is made and thus the irrigation-specific scenario information is captured only in object annotations, for example the decprov:DecisionMaking activity (Figure 1, top left) is labelled “Deciding HMSIWT” (How much should I water today?) but it is just an un-specialised decprov:DecisionMaking activity. A theoretical IrrigDecProv might specialise decprov:DecisionMaking to create an irrigdecprov:IrrigationDecisionMaking class or perhaps even sub-specialise multiple times to make a hierarchy of irrigation-specific decision classes, perhaps:

- irrigdecprov:DailyWateringDecisionMaking
- irrigdecprov:DripSystemIrrigationDecisionMaking
- etc.

Such a hierarchy would lend itself to scenario-specific question answering when particular attributes of the specialised decisions were also modelled, more specialised decision instance data validation (through the use of specialised validation templates) and also decision classification of the sort necessary for Case-Based reasoning systems.

6.4.5. References


7. Discussion

This thesis, created over a long time, addressed the research question:

**Can decision support systems with new approaches to irrigator interaction, external systems connectivity, and new mechanisms for representing decision making, enhance irrigation decision support system uptake?**

Individual parts of this research question were addressed in a series of research topics, investigations within which are related in this thesis in accordance with the topic/chapters mapping given in Section 1.5.2.

There is no doubt this is built out of many parts, some old that have had other lives and some new. Nevertheless, the work done does address a series of issues with irrigation DSS and has contributed to growing skill in systems’ design over time with several test irrigation DSS built and the majority of this work having been published in conference and journal articles which have then informed other work.

In the substantial time since the creation and testing of the IrriSatSMS system, the paper documenting it, presented in Chapter 2, has been referenced as an example of both mobile decision support and of utility assessment of DSSs (Rinaldi and He 2014) and now has more than 20 citations. Some of the early work of data source classification (Chapter 5) has been referenced in general Australian engineering works (Dowling et al. 2016). Chapter 6’s main paper (Section 6.3) has now also just started to be cited.

Less tangibly but perhaps equally importantly in terms of influence, the paper of Chapter 2 has been the subject of discussion over many years, including 2018, between the author and irrigation researchers or extension officers wishing to emulate the system internationally in India, Pakistan & Argentina – places where, the author assumes, local irrigators are displaying the uptake of technology that Australia displayed in 2010. IrriSat (follow-on work from IrriSatSMS) is still alive and the author still contributes to its continued development. This means IrriSat has been used/tested for almost a decade in 2018.

Despite all of this, it is still very hard to assess irrigation industry use except via the very narrow metric of (tool) “uptake”, as proposed in this research question. Chapter 4 does present a review of uptake for the IrriSat family of tools as per 2017 but there has been no systematic survey of this PhD’s early DSS tools and there is no direct pathway to DSS uptake from the latter tools (the decision modelling) without the further construction and testing of new DSS based on their principles, timing for which is outside this PhD.

Next are a series of brief discussions about aspects of this PhD that do not follow the previous chapters exactly but are aligned; their aim being to draw the generalisable lessons from this thesis’, and its related paper’s, work.

116
7.1 Scope of investigations

The process of establishing the topics for this thesis, described in Chapter 1, the Introduction, and Appendix B, was a long and painful one. On reflection, it was caused by deviation from the initial target of investigation – wireless sensor networks – and then the overwhelming of the student and supervisors by both the challenges and opportunities in the broad space of irrigation informatics.

If time could be repeated, even the topics after review (see Chapter 1, Figure 3) are far too broad for a PhD in which substantial technical work and evaluation of it is expected: fewer, deeper topics should have been undertaken.

It is possible that success in completing a whole thesis could have come off the back of the initial IrriSatSMS work (Chapter 2) if follow-on work deliberately extended the informatics used there, perhaps with other crops and areas, attempts of which are described in Chapter 3, however a couple of things prevented this:

1. The technical difficulty of some extensions like crop coefficient estimation from mobile phone images (Appendix F)
2. Time allocation issues for the student. Both the success of IrriSatSMS as a tool, CSIRO/CRC for Irrigation Futures’ desire to commercialise it and family circumstances meant the student increasingly spent time off topic 2010+.

The first issue was perhaps unavoidable – the legitimate result of technical investigation – and indicated to this author the very real limits of technical skill that a single person can apply to a problem within a limited time, given the learning a person must undertake for complex scenarios such as image processing. The second issue was instructive in how not, from a PhD’s point of view to benefit from the success of early work. Since the focus of PhDs is on scientific investigation, not product development, the outcome of early success should have been follow-on investigations. Thankfully, the author was surrounded by scientific investigations even while product development was undertaken and investigations initially earmarked for undertaking during the Literature Review phase of this work – decision modelling in particular – were still of interest to the irrigation DSS field after several years of attention elsewhere.

Ultimately, this thesis has addressed, with tangible results, a series of topics within irrigation informatics, some of which were originally proposed after the change from sensor networks, some of which were developed later as a result of interim investigations.

7.2 DSS Adoption crisis

The motivating issue for this work – to increase uptake of irrigation Decision Support Systems (DSS) by irrigators through the use of informatics – has been meaningfully impacted by this PhD. Chapter 2 shows that tangible differences to adoption can be made by incorporating DSS better into the farming lives of irrigations and Chapter 4, while dissecting the failure of commercialisation of IrriSatSMS, relates how such incorporation is enabled by shifts in backgrounding technology adoption – Internet and digital mobile phones. These together mean that we are neither doomed to accept a particular, low, rate of DSS uptake nor that non-technical means only can impact
adoption, as lamented by DSS developers (see Chapter 1), since both technical improvements and the natural process of computer-based systems adoption by farmers is seeing increased DSS use.

If we understand that technical DSS development does positively affect adoption and if we believe that decision modelling can assist us in undertaking better technical development (as per Chapter 6), then it can be realistically claimed that this PhD lays the groundwork for further irrigation DSS adoption through new development in two main ways:

1. By improving aspects of data source integration (Chapter 5), mostly through the development of architectures for brokering data and making sense of it (data semantics).
   a. As noted in Chapter 5, this work was motivated by early PhD topics was partly conducted as environmental informatics, not specifically related to irrigation, but the outcomes are relevant.

2. By opening up the area of decision modelling within irrigation DSS (Chapter 6)

Where work in these two areas might have been imagined prior to this thesis, now there is a tangible starting point for research into each area.

Continued work by this author and his research group will use both these ways with projects looking to further extend irrigation-relevant data source integration such building on (Sommer, Stenson, and Searle 2018) build and test an irrigation decisions Case Base (following on from work outlined in Appendix G).

7.3. **Industry learning**

Some reflections on systemic issues within the irrigation DSS development community affecting DSS uptake were given in Chapter 4, some of which were, or still are:

- Investment uncertainty versus uncertain income
  - There is not a well-established revenue pattern from agricultural DSS operation
- Lack of computing skills by agricultural support businesses
  - Potential commercial partners for research organisations lack GIS and data processing skills
- Lack of support for potential commercialising partners from government farmer liaison bodies
  - In Australia, grower cooperatives and government-run farmer groups are important in providing access to farmers and yet these bodies don’t always work well with small businesses attempting to commercialise DSS

While industry may and certainly is learning some ways to overcome some of these issues, such as a skilling-up in data processing by agricultural businesses, attempts to communicate these reflections highlighted other issues. The paper making Chapter 4 was submitted to *Computers and Electronics in Agriculture*, the original publisher of IrriSatSMS work and two other, similar journals but was rejected by all three not,
according to the reviews, due to a lack of quality in the manuscript, but due to the work being off-topic. Given that the work squarely addresses a previously published irrigation DSS and the journals’ claim ultimate interest in DSS adoption, the author believes the journals are generally uninterested in negative stories about DSS development and analysis of where things have gone wrong. This is detrimental to DSS adoption: if the industry is prevented from hearing about the fate of systems originally published with great promise, how can it learn? Due to this issue, there are far more systems published in irrigation DSS-relevant journals with great promise than there are actual, operating systems. To highlight this, of the more than a dozen (13) irrigation DSS designed for use at the field/farm scale made since the late 1990s (including IrriSatSMS and 3 other Australian systems) listed by Rinaldi & He in 2014, only one or two appear to still be in operation 5 years later (not IrriSatSMS, perhaps only APSIM) and yet no papers reflecting long-term system fates (except APSIM\(^1\)) are published.

### 7.4. DSS & Decision theory

Decision theory is a deep topic and clearly of relevance to irrigation DSS. A whole PhD could have been conducted just on the nexus of the two with some of the areas perhaps most worth pursuing being:

- Decision-theoretic explanations for decision paradoxes present in the irrigation context
  - It is known that paradoxes occur in decision making – where people seem to make decisions that are counter to their own goals – (see Chapter 1, Section 1.2.2) and some irrigators seem to make paradoxical decisions, such as under- or over-watering crops in the face of good evidence not to, so what can decision theory tell us about this?
- How decision utility can be measured effectively other than focusing on a single parameter of a decision scenario, such as water use
  - While this is an age-old problem, decision theory might provide new/better tools for weighing decision parameters that leads to a better appreciation of irrigation decision making
- How DSS might present choices to users in either ‘facilitative’ or ‘directive’ forms to achieve better outcomes
  - This area of potential investigation was posed for this PhD but ultimately not pursued as a research topic due to the perceived effort required for collecting evidence for rating better outcomes (see Appendix A)
- How risk, usually described in likelihood statistics, can be better communicated to decision makers to support them in better understanding the impacts of decisions they are making
  - While communicating economic and crop risks to irrigators isn’t decision theory-specific, decision theory may offer standard mechanisms, based on decision making psychology, to communicate difficult optimal decision making concepts relevant to farmers, such as economic real options analysis (see McClintock (2009) for a relation of this field to irrigation)

---

\(^1\) Holsworth et al. (2014) describe changes to the APSIM model – not entirely a DSS but part of one – over 20 years.
This PhD has not been able to grapple with many potentially interesting issues arising from this nexus but it has been able to deal with several in small ways (see conference papers referenced in Chapter 5) and one in a major way: that of representing decision making processes in a technical manner of relevance to current DSS development as related in Chapter 6.

The work to build on three decision modelling systems and to create a new system has yielded something simple enough to be used within this PhD for describing decision process flow (see figures in Chapter 1) but also one that is also capable of being easily incorporated into current technology (see Appendix G). With this modelling as the foundation and its technical implementation as the enabler, both new DSS design and further academic work in the area of irrigation DSS might be conducted that could otherwise not have been before, thus either will have benefited from this work.

7.5. Australian irrigation DSS community coordination

Early work in this PhD did anticipate, or hope for, a community of data providers and some form of coordination within the domain of irrigation DSS resource presentation (see the very early (Car et al. 2007) and summaries of total work in Chapter 5) to allow for the effective utilisation of the increasingly available data sources relevant to irrigation to be by DSS.

Having worked now in irrigation and environmental informatics for a decade, the author understands the great complexity of data and system standards development as called for in (Car et al. 2009) (Chapter 5 expands here) and the inertia present in data providers and DSS developers preventing them from aligning systems that takes time and effort to overcome. On reflection, if this PhD was to have measurably affected data providers or DSS developers, it should have perhaps focussed only on modelling irrigation data sources and worked with the community on that modelling and its adoption. It could not and ultimately did not work on that while also building specific irrigation DSS IrriSatSMS, reflect on DSS adoption trends and also delve into decision modelling.

Despite this, this PhD does note a pathway for better community data and DSS development coordination and work from it, environmental informatics work and work continuing (see Chapter 5) do seek to improve the areas. The long time taken for this PhD work has meant that avenues of data and systems modelling, and community coordination attempted by this author and others, tested, reflected upon and the newer generation methods suggested as a result are able to be overviewed here (Chapter 5 again) with a useful summarisation of all the undertakings for the reader.

7.6. Cumulative impacts over time

Unusual for PhDs, this work has not focussed on a single topic and yielded a single result in an area of study. Nevertheless, it has produced a large number of publications and lead to others not within the core topic area (see the Preface), produced several
technical products and influenced irrigation DSS research and development in Australia, if not adoption, to some extent over its long life.

The long life has certainly enabled reflection and changes over time not usually afforded the usual 3 – 5 year PhD scholar and this has resulted in non-technical work which is both difficult to communicate (see Section 7.3 above) but, this author believes, important for the original task of increasing irrigation DSS update in the long term.

The long life has also allowed some questions, such as data source integration, to be addressed differently using multiple paradigms, not all of which were initially available to the research community. This then tells a story that is more beneficial than straightforward success (or certainly failure) would have been within just one paradigm.

### 7.7. References


8. Conclusion

The broad spectrum of work within the field of irrigation informatics presented within this thesis has shed new insights into how changing technology availability, use and general acceptance on Australian farms has answered some questions that were vexing the industry in 2007 and raised others. Due to technology alone, there has been a shifting of the goalposts for Decision Support System (DSS) uptake since the first decade of the 21st century in Australia however this is not the complete story told by this thesis. An additional understanding gained from this work is that some areas of philosophical logic – certainly decision theory as discussed here but perhaps others also – that can and do inform artificial intelligence and other IT systems have yet to influence irrigation DSS. Compared with 2007, the effort to bring some handling of decision theory concepts to irrigation DSS is reduced due to new technical modelling systems, some of which have been presented here. A further aspect of irrigation DSS addressed by this thesis is that the irrigation DSS industry isn’t reflecting on, and communicating, the reasons for particular DSS systems’ failure to be adopted, particularly with respect to the commercialisation of research systems. Part of this is due to the reluctance of scientific journals to accept irrigation DSS papers to present negative case studies of systems’ adoption. Returning to the research question posed for this thesis:

**Can decision support systems with new approaches to irrigator interaction, external systems connectivity, and new mechanisms for representing decision making, enhance irrigation decision support system uptake?**

Regarding **new approaches to irrigator interaction** – perhaps by trial construction but, as investigated in this thesis, by improving technical interfacing with irrigators – yes, they can and have enhanced at least the prospect of DSS uptake. In the case of IrriSatSMS, with decade long hindsight, adoption of the tool seems to have been positively influenced by what was, at the time it was implemented, a new approach to irrigator interaction. Sadly, unrelated factors to the DSS technical implementation, such as the Australian irrigation consulting industry’s lack of ability to support the on-going delivery of such a tool, seem to have ultimately prevented the uptake of the tool. Regarding **external systems connectivity**: this thesis has explored methods for enhancing external systems connectivity, largely through frameworks for understanding irrigation-related data now available over the Internet. The systems trialled in this thesis’ work – IrriSatSMS and extensions of it – utilised external systems connectivity to provide their support in situations where previously it could not be provided and again, this at least enhances the prospect of DSS uptake since something is offered where previously it wasn’t. Despite the lack of a detailed analysis as to the impact of this growing ability to connect to external systems, this capability does seem to be transformative for DSS and positively so (leading to enhanced DSS uptake) since now DSS can access more and more relevant sources of data for irrigators. Of course, a new risk for irrigators is that of “information overload”, as discussed in Chapter 5, but...
this is a feature of a new, digital, connected information environment as a whole, not specifically irrigation DSSs. Moving forward, it seems that growing machine-to-machine Internet systems such as the Semantic Web present great possibilities for further external systems connectivity; however, regarding the core role that DSS undertake for irrigators – the synthesis and interpretation of data relevant to a decision scenario – Semantic Web possibilities here are to connect the DSS to more and more relevant sources of data, not (external) systems.

Regarding **new mechanisms for representing decision making**: some have been developed within this thesis – the extension of decision-making models and discussion of their specific relevance to irrigation DSS. It is not yet proved that this work can enhance irrigation decision support system uptake since there has not been the time available in the life of this thesis for the creation and testing of a new irrigation DSS that incorporate the decision-making model work, so this must be left to future work. Having a new decision-making model using technologies relevant to irrigation DSS and one indicated for use by them has been a necessary step for any future work in this area and a difficult one due to the crossing of multiple disciplines: irrigation, decision theory, computer science/IT. With this new model and the discussion of the requirements for its creation now present, implementation and testing work in this direction could be undertaken immediately.

Reflecting on the research question as a whole, we can conclude that indeed innovation in the three areas it named can enhance irrigation decision support system uptake. Some innovation in some of the areas has appeared to do so but there are systemic issues outside this PhD’s research area that have prevented measurable uptake over a decade. I can also conclude that new modelling theory and technical possibilities may enhance uptake.
Appendix A

PhD topic progression, 2009 - 2019

This appendix captures the progression of this PhD’s research questions from the first thesis draft in 2009 until this final document in 2019.

A.1. PhD Initiation

In 2003 in Australia, the Cooperative Research Centre for Irrigation Futures, a government, academic and private sector collaboration aimed at delivering “irrigation Research, Education and Training to give confidence to growers, industry, governments and the communities to invest in better irrigation, a better environment and a better future” was established (CRC-IF 2007). Part of its role was to fund PhD scholarships and initial discussions about this PhD determined its focus to be informatics for irrigation decision support. Figure 1 shows visual modelling of the decision to address the question What should this PhD be about? undertaken by PhD Supervisors, a group consisting of Evan Christen from the CSIRO and Ivan Mareels and Graham Moore from the University of Melbourne.

Figure A.1: The decision to Decide about PhD topic modelled as an OWL diagram containing DecPROV ontology classes (N J Car 2017) developed within this PhD and described in Chapter 7. The data file for this image is contained in Appendix H.
This modelling uses the DecPROV ontology (N J Car 2017, N J Car 2017c) was developed within this PhD and shows what was decided in the DecisionMaking event by giving an Answer to the Question *What should this PhD be about?* The specific answer value was *Informatics for irrigation DSS.* This Answer was chosen by the OptionSelection activity *Choose studying Decision Support Systems* which was determined to be both supervisable and implementable, as opposed to the alternative, *Choose studying Sensor Networks,* which did not determine that a Sensor Networks PhD was implementable. The core concern was that, in 2006, there was not the skills within staff at CSIRO’s Griffith Research Laboratories, where this PhD work was to be based, to assist with Sensor Network implementations.

The purpose of showing modelling of this decision in this way here is twofold: to make explicit the specific decision relating to this PhD’s direction and also to showcase one of the final outcomes of this PhD – a new form of decision modelling. The sort of decision modelling shown here will hopefully assist with the development of future irrigation DSS. This modelling was developed because current irrigation DSS only model the biophysical and sometimes economic circumstances of a decision, not decision elements, and this was thought to be an impediment to irrigation DSS adoption: by modelling decision elements themselves, past decisions can be characterised to establish current best practices, different decisions categorised and variations of a decision within a category made explicit, all of which could potentially assist DSS designers to better provide decision support.

In general – not just within the irrigation domain – if instances of decisions can be characterised using formal decision models, such as DecPROV used in Figure A.1, decision norms may be established that might guide best practice. This is because using formal models for multiple decisions allows for the correlation of parts between decisions and for patterns in parts and the relation of certain parts to outcomes to be seen. In this example case of choosing a PhD topic, it may be that future topic choosers could benefit from following the structured option expressions shown in Figure 1 and the explicit listing of Requirements against them.

### A.2. PhD Progression

The work of this thesis was conducted over an extremely long period of time for a PhD; from 2006 – 2017.

The initial phase, 2006 – 2009, was spent in applying software engineering, IT and ultimately informatics to a series of irrigation DSS scenarios. This resulted in a literature review which was accepted by PhD supervisors and an examination board and which is presented in Appendix A. It is given there and not in the main body of this document due to the changes in experimental direction that occurred after conducting the literature review meaning it is no longer relevant to the whole thesis. Conference papers on a range of irrigation informatics topics and the testing of multiple irrigation-related information systems, such as mobile-phone and Internet-based automatic weather station data and online mapping applications were also written during this time.

The next phase, 2009 – 2012, included only part-time PhD work due to personal circumstances. While experimentation and field work were limited, the write up of the operational deployment of the IrriSatSMS system and extensions to it for different crops and irrigation regions was completed. Also undertaken in this time were some preliminary investigations into general DSS techniques aimed at assisting irrigation DSS designers cater for a greater range of data sources in their systems – something thought by many researchers then, as now, to be useful.
From 2012 to 2015, very little PhD work was undertaken, again due to personal circumstances and also the breaking up of the Griffith-based CSIRO research group with which the field work for this PhD was conducted. Some general environmental informatics work was undertaken with relevance to irrigation informatics and irrigation DSS due to irrigation data being a subset of environmental data. In the latter part of this period, updated versions of the IrriSatSMS tool (just called IrriSat) were delivered and an assessment of the lifecycle of them made.

The final work for this PhD took place from 2016 – 2017. In this phase, a new set of informatics tools and methods, Semantic Web and Linked Data, learned by the author for other work, were applied to irrigation data flows and decision modelling. These tools and methods were not available generally in the early part of this PhD however they addressed issues that were raised early in the PhD, such as data source discovery, data fusion and decision modelling. As per the original aims of the PhD, these tools and methods do seem to be improving the lot of Decision Support Systems designers generally which includes those working in irrigation.

The experiments undertaken in this PhD returned mixed results; some were abandoned, some returned no useful outcome, and some resulted in award-winning systems. Figure A.2 documents the overall process flow of this PhD, following on from Figure A.1. As per Figure A.1, Figure A.2 uses DecProv.

![Diagram](image)

**Figure A.2:** General outline of the progress of this PhD’s experiment topics from initial ideas driven by the selected PhD topic (see Figure 1) to this document modelled as an OWL diagram containing PROV ontology (Lebo, Sahoo, and McGuinness 2013) & DecPROV classes. All relationships are defined in PROV-O. The data file for this image is contained in Appendix H.

In Figure A.2, the stages of work for this PhD are represented as PROV ontology (Lebo, Sahoo, and McGuinness 2013) Activities, each of which has a series of outputs (Collections) with the final Activity yielding this document. Experiment topics within each of the Collections listed in Figure A.2 were written and presented in various different ways to the PhD Supervisors and the funding body of this PhD, the CRC for Irrigation Futures.

### A.3 PhD Final Topics

This section explains how the topics finally addressed relate to the topics as ratified by conversion review in 2009.

This thesis document does not directly address the thesis topics as ratified by the Masters to PhD conversion review that took place in 2009. Those topics, as presented in the 2009 Literature Review in Appendix B were:
1. Better Decision Theory Use
2. Customisation
3. New DSS Tools
4. User-Defined Data
5. Empirical DSS
6. Better Irrigation Science Use

Instead, this thesis addresses the following final topics:

1. Better Decision Theory Use
2. Customisation
3. New DSS Tools
4. DSS Adoption Analysis
5. Empirical DSS

Four of the 6 topics were preserved however User-Defined Data and Better Irrigation Science Use were not. In their place, DSS Adoption Analysis was. The listing of Topics After Conversion and Final Experiments Topics and short reasons for the changes are shown in Figure A.3 below.

Figure A.3 lists the individual topics proposed and undertaken in the life of this PhD. It includes some high-level reasons for topic inclusion or rejection, modelled using DecPROV. The final topics selected influenced the writing of this document and indicated in Figure A.3 also are the final topic and thesis chapter relations.

**Figure A.3:** The progression of this PhD’s Experiment topics from the initial topics to this document’s Chapters modelled as an OWL diagram containing PROV-O & DecPROV classes. All relationships are defined in PROV-O, unidentified relationships are wasDerivedFrom. This diagram is an extension to Figure A.2. The data file for this image is contained in Appendix H.

Figure A.3 lists the topics by name only, however the next sub sections relate the specifics of how each research topic originally proposed progressed to the topics finally investigated. Both topics not finally investigated, such as User-Defined Data and new topics that were, DSS Adoption Analysis, are addressed.
A.3.1. Better Decision Theory Use

This topic evolved from the Initial Experiment Topic aiming to “determine the utility of minimally interactive ‘facilitative’ versus ‘directive’ DSS” in a 2009 thesis draft. It was initially proposed due to the author and supervisors’ observations of early prototypes for the IrrisatSMS system (see Chapter 3) that were used by farmers to facilitate decisions, rather than to direct them, that is the DSS results were taken as suggestions but were not followed exactly. Despite continued interest into facilitative/directive comparison, it was decided that investigations needed for this were not feasible given the surveying effort needed to establish rigorous comparison results and both the supervisors’ and student’s lack of skills in this area1. This is indicated in Figure A.3. by the decision Decided Gathering Responses Onerous.

The Better Decision Theory Use topic formulation looked to generalise the ‘facilitative’ versus ‘directive’ topic and to see what irrigations DSS design could learn from or, better put, what it may be missing out from by not knowing, the academic discipline of Decision Theory. Work was carried out into decision theory and the historical development of DSS and the decision indicating this in Figure A.1. is labelled Decided to Investigate DSS Origins.

This topic, as finally posed, is addressed in Chapter 7 where work in representing decision making is related. Such work was undertaken due to the discovery that there were limited decision-making modelling tools of the sort best suited for integration with irrigation DSS. The paper constituting the bulk of Chapter 8, Using decision models to enable better irrigation Decision Support Systems (Car 2018), delves into what decision modelling system would integrate well with DSS and the resulting work on DecPROV (Chapter 7.2 and Car 2017b) attempts to implement some of the recommendations made.

A.3.2. Customisation

In a 2011 thesis draft, this topic was described as aiming to allow “greater user customisation of DSS to facilitate the relevance of and ease of use of decision support systems by irrigators” which was a generalisation of the confirmation topics User-Defined Input and User-Defined Rules shown in Figure A.1. Several early attempts using then new technologies were made to enable better DSS customisation in different ways, for example, the methods outlined in (Car, N.J. et al. 2008) and then later (Car and G. A. Moore 2011) (second half of the paper about User Interfaces). These attempts suffered from far too great a scope: there are many ways in which customisation of DSS by users (as specified in thesis draft explanatory text) could be enhanced.

Relevant to both user-defined input and rules, a narrower direction that this author further pursued to improve DSS customisation was that of addressing data commensurability. This contrasted with focussing on user interfaces (UI) which are relevant to user-defined input but are not the major component preventing greater DSS flexibility there. While the effort required to access data that could to be used by DSS has been greatly reduced due to much data being placed in fairly standard forms on the Internet (see (Car 2017b) in Chapter 5 for a full discussion of this point) and it is possible for Internet users to define their own data sources

1 Note that substantial surveying of irrigators was ultimately carried out in this PhD – see Chapter 3 – but the intention, after thesis conversion review, was not to make surveying a major focus of the work carried out.
due to “Web 2.0”\(^2\), which is partly a UI issue, a vast amount of effort is required to ensure the commensurability of data if a DSS is to use it effectively. Just because we have data about soil moisture measured in percentage moisture, what is the particular geographic feature’s property measured (e.g. top 10cm of earth, top stratigraphic layer), what tooling was used to measure it (e.g. capacitance) and what are the methods used to arrive at an observation value (e.g. average of several readings over time, single instant)?

A paper on the standardisation of irrigation DSS inputs that pointed out the incommensurability of evapotranspiration readings represented in the Australian national water information holdings, AWRIS\(^3\), was written (Car and G.A. Moore 2011). This paper recommended additions to the Water Data Transfer Format (WDTF) used by AWRIS to make explicit the equation and other parameters such as wind measurement height used to calculate evapotranspiration. These sort of recommendation are now understood by the author to be in line with the intentions of standards such as Observations & Measurements (ISO 2011) that promote more holistic coverage of the features, properties and values of measurements of natural phenomena that were not available at the time of this work but are in widespread use today.

The author imagines that a future irrigation DSS might be able to access data from a vast range of Internet-publishing data sources if those sources meet a large number of criteria and not just some relating to commensurability assessment but others too, for example, data quality assessment. Data quality assessment of Internet data is the focus of current research by this author (see (Car 2016) for his initial foray into this field) but given that it is a large and current topic, it is not timely to be considered in great depth in this thesis.

This thesis does describe work on customisation via addressing data source commensurability in Chapter 6. Another way it also addresses customisation is via Better Decision Theory Use described above with work done to better represent decision making, as related in Chapter 7. The thought here being that better decision representation within DSS will lead to new opportunities to cater for those decisions’ data and other inputs.

A.3.3. New DSS Tools

A decision was made to Directly Test New Technology (Figure A.3.) as a result of some initial work with irrigators and SMS technology which, in 2006, was very new then. This was partly because new methods of decision support are often presented in literature through a product description, and partly, it has to be said, due to the CRC for Irrigation Future's desire to attempt to market a product. Ultimately it is the case that this thesis tested several new tools such as IrriSatSMS and spin-offs from it. The main vehicle then for testing “new methods of decision support” (from the earliest thesis outlines) was via tool implementation and results comprise Chapters 3, 4 and 6.

---

\(^2\) The notion of Web 2.0 was originally espoused by Tim O’Reilly and O'Reilly publishing provides this information clarifying the history of the idea: [http://www.oreilly.com/pub/a/web2/archive/what-is-web-20.html](http://www.oreilly.com/pub/a/web2/archive/what-is-web-20.html)

A.3.4. User-Defined Data

The subsection above, A.3.2, describes how the focus of customisation was narrowed to address issues of data commensurability. Attempts to enhance User-Defined Data for irrigation DSS necessarily require the user to ensure that the any data they supply is commensurate with the DSS' possible data inputs thus this topic is partly addressed by the final thesis topic Customisation in Chapter 6.

The specific wording given for the topic that this User-Defined Data subsumed, namely Interoperability & Discovery (see Figure A.3.) as present in the thesis draft presented at conversion review (see Appendix A) was:

Determine the extent to which XML specifications may be adopted by irrigation data sources for their interoperability and ‘discoverability’ over the internet

With hindsight, the intention of this topic- that of pursuing specification adoption by data suppliers – was laudable and has been, and still is, a major focus of this author’s research group’s work however it is overly ambitious for to test an adoption extent, long-term work with many data providers is likely to be required, which is out of the possibilities afforded to a PhD student. This thought, and some other attempts at handling User-Defined data are related in Chapter 6.

A.3.5 Better Irrigation Science Use

This topic, phrased in a 2011 thesis draft as aiming to generate “better user understanding of irrigation science and irrigation best practice”, was excluded almost entirely from this thesis due to a realisation that the author was not sufficiently skilled to be able to judge best irrigation practice. This is indicated in the decision Decided Candidate Not Skilled In Topic in Figure A.3. Instead of this topic, the topic DSS Adoption Analysis was introduced by Deciding To Relate Lessons Learned From IrriSatSMS and it is related in Chapter 5.

There is no direct causality going from Better Irrigation Science Use to DSS Adoption Analysis, rather this topic change must be understood in the context of what was feasible for the candidate to undertake, given his skills.

A.3.6. Empirical DSS

The 2009 PhD confirmation’s propose research question’s hypothesis was:

A decision support system with new approaches to irrigator interaction and external systems connectivity, in combination with an empirical reasoning framework, can be created to enhance objective irrigation decision making.

The inclusion of the phrase “…in combination with an empirical reasoning framework…” was due to the student and supervisor’s desire to test the utility of an Artificial Intelligence (AI) paradigm known as Case-Based Reasoning (CBR) for improving the relating of irrigator’s experiences to one-another.

Early work with IrriSatSMS prototypes indicated that such systems would collect multiple irrigators’ decisions and that such data collection could be used for benchmarking. However, the student also realised that if the individual decisions made by irrigators could be recorded with context, particularly the utility of the decision outcome, more than regular benchmarking could be achieved. The idea was to collect cases of irrigation decisions, that is decisions made with
context including decision outcome, and to characterise those cases in a way that would lead to sophisticated case comparison, the core of CBR.

The mechanics of CBR and possible implementations were investigated throughout the writing of this thesis and results published at several points, including (Car, N.J., and G. A. Moore 2011) & (Car, N J 2018b, reproduced in Appendix G).

Ultimately, no comprehensive handling of this topic was undertaken in this thesis – no operating CBR system was built – due to a focus on the representation of irrigation decisions – Chapter 7 – since powerful decision representation is necessary for any CBR system. It was, and still is thought, that the formal modelling of decisions as per Chapter 7 will lead to CBR or similar systems for irrigation case comparison and thus empirical DSS. Appendix G summarises this and suggests a specific technical way forward.

### A.3.7 DSS Adoption Analysis

With the extended time taken in the creation of this thesis, the lifecycles of the IrriSatSMS and the follow-on IrriSat product were able to be observed. Reflecting back on them in 2015, the author saw an opportunity to summarise these lifecycles and to relate on how changes in technology adoption within Australian irrigation affected them and could potentially affect future irrigation DSS. The results of this reflection are Chapter 5.

### A.3. References


https://www.w3.org/TR/prov-o/, accessed September 17, 2016.
Appendix B

Literature Review 2009

This appendix contains the Literature Review as submitted in 2009 to a university review panel to convert the thesis from a Masters to a PhD. It was accepted along with a thesis outline presentation and a Proposed Research Question, see page 85 in the Literature Review.

The Proposed Research Question given in this Literature Review was not directly investigated, instead the 5 parts of it were transformed, for a number of reasons, into topics that were ultimately investigated. Appendix A, Section A.3, details the transformation of the 5 topics into Topic After Literature Review and finally Final Experiment Topics.

Some parts of the Literature Review contained in this appendix have been preserved in Chapter 1 of this thesis.
Chapter 2

Literature Review

2.1 Introduction

Irrigation Informatics is a term that has been used to describe the use of informatics within the irrigation context, by the Cooperative Research Centre for Irrigation Futures (CRC-IF). Informatics, itself a recent word, the author defines to be the science of information use\(^1\) The Indiana Uni School of Informatics (48) states that

> "Informatics is a bridge connecting IT to a particular field of study such as biology, chemistry, fine arts, telecommunications, geography, business"

The CRC-IF(12) states that informatics is:

> “often, though not exclusively, studied as a branch of computer science and information technology and is related to database, ontology and software engineering. It focuses on the use of technology for improving access to, and utilization of, information.” (9)

\(^1\)By referring to the science of information use itself, rather than mathematics and science that facilitate information use the author differentiates Informatics from Information Theory. The Wikipedia, used for common definitions, describes Information Theory as “a discipline in applied mathematics involving the quantification of data with the goal of enabling as much data as possible to be reliably stored on a medium or communicated over a channel.” (68). The author believes this is a standard, uncontroversial, definition of Information Theory.
and, as a concept is about:

“developing the science of storage, retrieval, and optimal use of biophysical information, data, and knowledge for problem solving and decision making in irrigation management, e.g. Yield/$’s per irrigation decision” (9)

The aim of this research falls within the CRC-IF’s vision of:

“making better use of Australia’s water resources ...[by]... having state of the art, robust tools which can be adopted by water managers at all levels for more precisely managing and accounting for water use” (12)

These definitions focussed this literature review on decision theory, decision support systems and the past and current agricultural implementations of such systems.

2.1.1 Outline

This Chapter initially details the author’s investigations into the philosophy of decision theory in Section 2.2 and considers recent views on decision making generally and it’s relation to decisions made in the irrigation sector. Section 2.3 looks at Decision Support Systems (DSS) as described in the literature and investigates the usage of decision theory by DSS designers. Section 2.4 investigates the non-user interface aspects of DSS and Section 2.5 looks at the user interface aspects of DSS design as well as user interface design principles not related to DSS design but potentially of use. Section 2.6 evaluates past and current irrigation decision support systems used in Australia and Section 2.7 summarises the Section Summary presented in each of the previous sections. Finally Section 2.8 details the research questions that the author proposes in response to the findings of the literature review.


2.2 Decision Theory

Decision theory incorporates a range of disciplines that include philosophy, education, psychology, mathematics (probability, Bayesian, fuzzy logic) and computer science (artificial intelligence). Decision theory aims at an understanding of what decisions are to be made and also how and why people make the decisions they do. Firstly, general theories of decision making were examined and secondly, the decision theory relating to decision making, in the context of irrigation, was explored.

2.2.1 Decision theory philosophy

Much work has been done in the area of philosophical decision theory in the last three centuries. Early work on decision quantification, principles for rational decision making and decisions leading to 'fair' outcomes, as judged by egalitarian principles, was undertaken by French enlightenment philosophers, notably the Marquis de Condorcet (1743 - 1794) (13) in justification of aspects of the constitution of 1793. de Condorcet established processes and principles for decision making. An example: in his 'stage one' of the decision making process, the stage containing an individual’s decision process that leads into further stages of group decision making, an individual:

“discusses the principles that will serve as the basis for decision in a general issue; one examines various aspects of the issue and the consequences of different ways to make the decision”

Explicitly describing a process to be followed that leads to rational decision making seems obvious by today’s standards but was new at the time of de Condorcet’s writing.

2.2.1.1 Decision theory processes

According to the American education philosopher James Dewey (14), five consecutive stages of decision making can be followed, one after another, to assist an individual in the decision making process. These stages are:
1. a felt difficulty
2. the definition of the character of that difficulty
3. suggestion of possible solutions
4. evaluation of the suggestion
5. further observation and experiment leading to acceptance or rejection of the suggestion

(27) gives further, more recent, developments of this sequential decision making process and discusses criticisms of stepwise, sequential (termed 'linear' in decision theory literature) decision making processes. It is clear to this author that not much more can be helpfully said about linear decision making process other than what was said in (14) - the author feels that Dewey’s process is effectively up to date. The criticisms of linear models of the decision making process are recent and are neatly explained by (70).

“We believe that human beings cannot gather information without in some way simultaneously developing alternatives. They cannot avoid evaluating these alternatives immediately, and in doing this they are forced to a decision. This is a package of operations and the succession of these packages over time constitutes the total decision making process.”

The author feels that there is much merit in looking further into linear and non-linear decision making processes and how decision support system (DSS) design might benefit from work in this area by helping deciders to arrive at successful decisions by undertaking these processes. Both linear and non-linear DSS front end design is considered fully in Section 2.5.
2.2 Decision Theory

2.2.1.2 Quantification of decision parameters, unknowns and risk

Within (14), Dewey’s fourth (evaluation of the suggestion) and fifth (further observation and experiment leading to acceptance or rejection of the suggestion) stages: a decider is asked to choose an outcome, potentially from a range of possible alternatives, based on certain criteria. Decision theory has long tried to parameterize possible alternatives to enable a user to weigh them against one another, as rationally as possible, to prevent certain merits or demerits of those alternatives from being overlooked or under emphasized.

In this regard, histories of decision theory, such as (50), rapidly lead to mathematical probability\(^1\), then to Bayesian decision theory, game theory and other mathematical and scientific theories. Modern usage of probability is commonplace for decision making and Bayesian and other theories are incorporated into many DSS.

The extent to which we can quantify decision parameters, unknowns and risk leads into the philosophical field of philosophical logic and from there into mathematical logic. Using mathematical logic, the upper bound on establishing a complete and consistent set of mathematical axioms that can be used to describe problem parameters is established by limitations given in Godel’s incompleteness theorems (29). Additionally, this author has had experience with logic philosophers (John Howse of Melbourne University in particular (32)) who have hotly disputed the results of various philosophical logic problem expressions that leads him to believe that there is no consensus in logical philosophy either.\(^2\)

\(^1\) (50) states that “A decision theory based upon utility is intimately related to theories of probability, which are needed for the calculation of expected consequences.”

\(^2\)These rather esoteric mathematical and logical limits were investigated by the author to see whether their bounds may actually be the limiting factor in decision making. The author feels that he can conclusively say they are not, for the limitations of agricultural understanding and the uncertainty in weather and other environmental patterns place far greater bounds on effective decision making. Further to this, the author hopes that techniques in DSS design employed to circumvent problems posed in the Subsection 2.2.1.4 will also circumvent the bounds mentioned here.
2.2 Decision Theory

2.2.1.3 Analysis of decision outcomes

Necessarily related to the above theme are attempts to quantify what exactly 'good' and 'bad' outcomes are. This process, which the author will refer to as the gauging of utility, is fundamental to decision theory for without it one cannot interpret the results of decisions made. The process of gauging utility is a large topic and will not be addressed in this literature review. It will be a major component of methodology and design chapters.

2.2.1.4 Decision theory paradoxes

Decision theory has yielded many 'paradoxes' by describing scenarios where people seem to make decisions that are counter to their own goals. Recent undergraduate course texts such as (27), cover topics such as 'Decision making under uncertainty','Decision making under ignorance' and 'Decision instability' in which we can find instances of these 'paradoxes', such as Newcomb’s Paradox. Another paradox is known as 'Death in Damascus' which is a conundrum exploiting the concept of decision making under instability where the outcome of a decision is not known to the decider until after the decision is made, thus preventing good predictions of decision outcome until it is too late. These paradoxes raise questions about the validity of the decisions that individuals can make in that they challenge what we think of as 'rational' and perhaps point out that, in certain scenarios, it may not be possible to make a rational decision or that an individual will usually make a supposed irrational decision.

A recent example of a slightly different paradox is described at length in (16). This paradox or, more correctly, reasoning fallacy, is a result of lay people, in this case a jury, misunderstanding the results of Bayesian statistics. The lay people then are inclined to believe that people who are likely to be innocent are in fact likely to be guilty and therefore they convict them.

From these readings, the author can see that there is always a possibility that a new theory of statistics or probability will lead to a better understanding of a decision scenario. If a new theory does arise, how is it to be incorporated into an already established decision making process or DSS? In terms of paradoxes
and reasoning fallacies known, the author believes that they possess common attributes and that many of them can be avoided, for certain decision arenas (such as irrigation management) at least, with the correct decision situation establishment and good problem presentation. These thoughts will be expanded upon, with examples, in Section 2.2.2.1 and actual DSS designs that implement them in 2.4 and 2.5.

For new interpretations of data or for new ways to parameterise decision factors, we have to look to agricultural and other science, not philosophy or statistics, and that it outside the scope of this PhD.

2.2.2 Decision theory in the irrigation sector context

The above relation of decision theory is very broad. This section discusses the specific sorts of decision that we encounter in the irrigation management sector.

Irrigation management decisions are complex, multifaceted decisions that do not seem to be directly comparable to many of the examples given in philosophical papers on decision theory. One reason for this is undoubtably that, for the sake of making a philosophical point, only simple examples are used so that factors other than those under examination do not cloud the issues examined. This is not, however, the only reason: another is that the philosophical examples given are usually out of any context and certainly do not consider aspects of real decision making to do with a decision maker’s familiarity with the general area of the current decision to be made. Adding in ‘real life’ detail and placing a decision into a context where the decider has background knowledge should be sufficient to avoid all of the theory paradoxes given above. The next section elaborates on the differences between theoretical example decisions and irrigation management decisions.

2.2.2.1 Overcoming decision theory paradoxes

In all of the decision theory paradoxes that the author has seen, the paradoxes are ‘solved’ by establishing accurate expressions of the decision maker’s expected
2.2 Decision Theory

A: You receive $100 if you draw a red ball

B: You receive $100 if you draw a black ball

and

C: You receive $100 if you draw a red or yellow ball

D: You receive $100 if you draw a black or yellow ball

utility (EU) of decision outcome, the quantitative assessment of the decision outcome’s expected value (EV) and, critically, in making sure that these expressions are understood by the decision maker. The author believes that the ‘problems’ only exist when either the EU and EV of a decision are not well known or when the decision maker does not understand what the expressions of EU or EV are telling them. An example: it is due to a lack of understanding of what deciders are likely to see as a return, EV, that leads then to violate the expected utility hypothesis in the case of Ellsberg’s Paradox. To fully illustrate; Ellsberg’s Paradox (15) states that: suppose you have a box containing 30 red balls and 60 other balls that are either black or yellow. You don’t know what the ratio of black to yellow balls is, only that the total number of black plus yellow balls equals 60. The balls are evenly mixed so that each individual ball is as likely to be drawn out of the box as any other. You are now given two sets of two wagers:

The paradox occurs when one has to choose one wager from A or B and then another from C or D. The reason for this is that if you prefer A to B, based on the notion that you think drawing a red ball is more likely than a black ball, then you should prefer C to D for the same reason. Supposing you prefer B to A then, by the same measure, you should prefer D to C. However, when surveyed, people strictly prefer A to B and D to C. This violates expected utility theory (The idea that you have value in an an outcome, in this case that $100 is better than $0 for the choices are at odds with a single preference (red over black) either way).

Explanations as to why people strictly prefer A over B but then D over C usually focus on the fact that the probabilistic information available to the decision-
maker is incomplete and therefore people are exhibiting ambiguity aversion when they make the choices that they do. (22) Put another way, people are “more averse to uncertainty than they are to regular risks of known proportions” (27). A way to elicit a ’correct’ set of choices from a decider is simply to explain the above to them and make sure they understand what their natural reaction to the situation is and why it may not lead them to the best outcome. If you want people not to violate the expected utility theory the questioner might discuss with the decider the ins and outs of probability. Another way to look at this problem is to value the ambiguity aversion more highly than material returns of the decision and then say that the deciders are actually making the ’correct’ decision (ie one that most benefits them and does not violate expected utility) it is just that other factors have legitimate claims to holding utility for the decision maker: in this case benefit comes from psychological ease, not just from material gain.

For irrigation decision making either the EV of a decision is established by calculation (for example an irrigation at such-and-such time with so-much water results in a soil moisture content of such-and-such) and then the EU left to the individual irrigator to assess (they would certainly factor in many decision results, not just water use efficiency) or the EV is not easily quantifiable (for example should I irrigate tomorrow or go to the coast for a holiday) in which case the ’correct’ decision is a business decision on behalf of the irrigator and their particular circumstances that no mathematical or other system can effectively ’solve’. Further, ’decision instability’ that the author understands to be of particular interest to philosophers (this occurs when the final outcome of a decision is dependent on the decision made) is not of much relevance to irrigation management decisions due to the well-known nature of most irrigation management decision outcomes: we are not dealing with ’a’ or ’b’ atomic outcomes in the irrigation sector, as in point making examples in philosophy, but rather with incremental improvements to decision outcomes. Essentially the philosophy of unstable decisions deals with distinct decisions in isolation and this is fundamentally not the case with irrigation management decisions. We may expect that an irrigation decision maker is already making irrigation management decisions that, at least to some degree, bear them utility of outcome. A person using no decision support may, when
2.2 Decision Theory

A decision is judged by qualitative measures, achieve a utility of outcome 99% of that of a decision made with decision support. Qualitatively, a person may achieve 100% of the supported decision utility while not using decision support, particularly if they follow a course of self-fulfilling prophesy whereby the decision they make is the correct one for them, regardless of outcome, as judged by others. Lessons for irrigation decision making, based on these points are given in the 'Section Summary' subsection of this Section.

To generalize: the author believes that rather than testing the decision makers, if you want a 'correct' answer, you should be 'on their side' in the way you present them with the situation. You might not even need to generate theories as to why people choose what they do or, to use another paradox as an example, why people invariably choose the 'wrong' outcome in Newcomb's Paradox. Potentially a questioner could invent any number of scenarios that 'force' people to act counter to their own interests. This is hardly surprising to the author and is a bit like asking “what would you do with a million dollars?” in that it is neither real nor likely. Certainly this is the case with situations like Newcomb's and 'Death in Damascus'.

One further point on this subject: the author believes mathematics, including logic statements, to be a language with which it is often easier to describe situations than, say English and that graphs, a visualization of mathematical functions, are simply another 'language' used to describe something. Perhaps too much attention is given to various logic or other expressions of a problem when emphasis should be given to whatever expression language gets the point across easiest. If we want to know the trajectory of a cannonball for example, describing it as “going up from here and coming down over there” (English), is less accurate than “following a parabolic trajectory from here to a point 315m away”, (English with a mathematical concept) which is still less accurate than

\[ \text{height} = 100 \sin(x), \text{distance} = \{0...315, \}. \]

However, this last expression, albeit very accurate, is useless to someone who doesn’t know anything about trigonometry (see Fig. 2.1 for a graph of this function). As a fundamental design concept,

---

1There are many discussions of Newcomb's Paradox in the internet with [http://members.aol.com/kiekeben/newcomb.html](http://members.aol.com/kiekeben/newcomb.html) giving a complete discussion.
2.2 Decision Theory

Figure 2.1: A graphical version of $\text{height} = 100\sin(x), \text{distance} = \{0...315.\}$

a DSS should use *whatever* language is most conducive to getting the point across and therefore resulting in the best decision outcome for the decider, whether that be probability, Bayesian statistics, colour graphs or English sentences.

For a final point in regard to this matter of appropriate and accurate language usage, the author notes that the logic philosopher Ludwig Wittgenstein believed that confused language use was the cause of most philosophical problems(2) and this view now carries much support since it’s publication in his book *Philosophical Investigations* in 1953(71), a couple of decades after logic philosophers such as Bertrand Russell produced many paradoxes.

2.2.3 Section Summary

- The shift of emphasis away from a single outcome as the marker of utility is very important for irrigation decisions where the outcome of a decision may not just be measured in terms of a single parameter such as water use
efficiency. *(This concept is critical to studies in irrigation decision theory and will be discussed at length in further sections of this work)*

- A design principle that may be learned from decision theory is that it may simply be better for a questioner, or DSS designer, to present as much information as possible to the decision maker and leave estimations of expected utility to the decision maker. With this in mind, two types of DSS can be classified, they are: 1) directive and 2) facilitative. A 'directive' DSS leads the user to a decision whereas a 'facilitative' DSS just presents data, or information derived from data, to the user. The author acknowledges that it may not always be possible to pigeon-hole DSS in this way but the concept he feels to be useful.

- Decision support for any decision in which the decision maker already makes unassisted decisions that lead to outcomes of some utility, which is most decision making, will need to take into account the decision makers preferred outcomes, on an individual user basis if possible, and quantify and assign weight to those preferred outcomes, in order to optimise decision support utility. This suggests that decision support in an environment where unsupported decisions are already made would need to follow through an iterative approximation process where perhaps first best 'scientific' outcome were calculated, then an individual’s preferences added and then best ‘individual’ outcomes were calculated.

- Decision paradoxes and seemingly erroneous decisions made by decision makers when faced with them can probably be avoided if the questioner, or DSS designers, is 'on the side' of the decision maker and therefore not trying to catch them out

### 2.3 Decision Support Systems

#### 2.3.1 Introduction & definition

Decision Support Systems (DSS), simply put, are systems that aid people in decision making. The following discussion of DSS will focus only on computer
programs, as opposed to DSS presented in other mediums, such as printed material. The term is very broad and here follows DSS classification of currently deployed DSS in research and industry.

2.3.1.1 DSS classification

The author has seen many different classification schema for DSS based on what particular classifiers deem to be the most significant aspects of DSS. Classification can be based on architectural components as originally detailed in (61). Five typical components are used for classification by (19) are the:

1. database management system (DBMS)
2. model base management system (MBMS)
3. knowledge engine (KE)
4. user interface (UI)
5. user

Different DSS are then described according to how their architectures match, or don’t match, these typical parts, so for example we may look at DSS that use an internet browser as an interface as opposed to a Short Messaging Service (SMS) alert system. Classification in this way the author finds to be of limited use for people other than DSS designers because: 1. there are a vast range of DSS architectures due to the fact that DSS do not comprise of a single technology, specification of technologies or a single set of user requirements and attempts to classify DSS in this way must therefore have many exceptions that to be of little use, 2. apart from the inclusion of the ‘user’, little attention is paid to non technical aspects of a DSS and it is hard to determine the function (the end user experience) of the DSS in this way. Other authors prefer to use other criteria for classification and one that this author feels well addresses classification from a user’s point of view is that proposed by (51) uses the criteria of the DSS’s user assistance method. (51) proposes the following categories:

1. model-driven
2.3 Decision Support Systems

2. communication-driven
3. data-driven
4. document-driven
5. knowledge-driven

Model-driven DSS use mathematical, economic or other models, communication-driven DSS, such as Microsoft NetMeeting which “enables file, desktop and application sharing across networks for up to 10 people in the same room”\(^1\) and are designed for group decision support, data-driven DSS presents data from singular or multiple sources to a user to assist them in decision making, document-driven DSS use “a variety of storage and processing technologies to provide complete document retrieval and analysis” and knowledge-driven DSS are a form of data-driven DSS that derive information from data and present that ‘knowledge’ to users for use in rule-based, case-based or a combination of rule- and case-based reasoning, such as CORMS AI (64). There is no reason why a DSS could not be included in multiple categories.

The author believes that, within the context of irrigation management, model-driven, data-driven and knowledge-driven DSS are those most likely to be deployed. The reason for this is that decisions about irrigation management use many models, such as water balance models, biophysical, economic and other data and knowledge derived from them and little, if any, document retrieval. There is also no call for DSS of the communication-driven mode, at least for DSS relevant to individual irrigators.

A systems perspective may be used to classify DSS. The explicit aim of systems-perspective classification, as given by (24), is to quantify how useful a DSS’s output is and how well it integrates with other systems. This means it is a form of functionality rather than architecture classification but this system goes further than just noting the user assistance method in that it quantifies different aspects of the human-perceived DSS functionality. This author finds it useful when trying to assess DSS utility over a broad range of attributes.

\(^1\)from Microsoft NetMeeting’s Homepage at http://www.microsoft.com/windows/netmeeting/
The attributes that (24) use for DSS classification are:

1. Interactivity
2. Event and Change Detection
3. Representation Aiding
4. Error Detection and Recovery
5. Information out of Data
6. Predictive Capabilities

The author finds that these systems attributes can be used to conceptualise potential DSS software modules, for example ‘Information out of Data’ assessment may be able to provide guidance in what sort of reasoning engine to design or ‘Representation Aiding’ assessment may lead to interface design choices.

This sort of classification is indicative of the latest user utility-based DSS classification. The above example is one of many such classification systems that are potentially all useful in viewing past systems and designing new systems.

Power’s website, (51), provides a plethora of general information on DSS types but this author feels that the classifications given above adequately cover DSS classification in the literature in that they provide examples of a technical and ‘human use’ classification.

2.3.1.2 Classification by networking paradigm

The author finds it useful to combine some technical and some ‘human use’ classification and to classify DSS by their ‘networking paradigm’. These ‘networking paradigms’, constructed by the author are listed in Table 2.1 below.

The author believes there is much utility in classifying DSS by their networking paradigms for this classification allows one to see where/how the DSS gathers the information that underlies the decision, the time-step the DSS may operate on and also the age of the DSS as these paradigms have progressed through time. This classification method allows one to see how and from where data is
### 2.3 Decision Support Systems

Table 2.1: Author’s categories of DSS, based on the DSS’s networking function

<table>
<thead>
<tr>
<th>Cat</th>
<th>Name</th>
<th>Description</th>
<th>Irrigation Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>none</td>
<td>standalone desktop application</td>
<td>Water balance spreadsheet</td>
</tr>
<tr>
<td>2</td>
<td>none, single link</td>
<td>desktop application that collects data from a single machine or sensor</td>
<td>G-bug probe</td>
</tr>
<tr>
<td>3</td>
<td>local computer network</td>
<td>desktop/intranet-based DSS with information only from intranet computers local network databases and local network sensors</td>
<td>a farm with an automatic weather station, soil moisture probes and historical weather database</td>
</tr>
<tr>
<td>4</td>
<td>private (enterprise) network</td>
<td>desktop/intranet-based DSS that uses intranet resources as well as non-intranet resources connected to the network via a proprietary network</td>
<td>a company farm with soil moisture probes, databases and a network of weather stations</td>
</tr>
<tr>
<td>5</td>
<td>internet</td>
<td>internet-based DSS that sources information from, and sends information to IP-base servers and clients</td>
<td>WaterSense(34)</td>
</tr>
<tr>
<td>6</td>
<td>2nd gen. internet</td>
<td>DSS that uses recent internet technologies and design concepts, such as Web 2.0, to allow DSS users far more flexibility than has previously been possible. This concept is fully detailed in Section 2.3.2</td>
<td>no examples exist</td>
</tr>
</tbody>
</table>
collected and it tells you how data may be delivered. This is important when working with diverse data sets. It would need to be used in conjunction with systems-perspective classification if one wanted to gauge the utility of a system.

Early DSS were restricted to being of type 1, 2, 3 or 4 as the internet only came into being in the 1990s. In Figure 2.2, the author gives his view of the generic architecture diagram for category 1 - 4 DSS. Category 5 DSS have emerged since. Two recent examples of category 5 DSS are: a weather information system delivering national and international weather data to Hungarian farmers over the internet (41) and the Wateright website (17) that not only provides weather information and scheduling calculation but also educates the user in irrigation best practice methods. These DSS both collect information from a private network (in both cases that which connects weather stations to a bureau’s weather information database) and then return results from this network, via a collection point, to users’ client machines via the internet. Included in this category could be DSS that collect information from remote servers on the internet, which could in turn be gateways to other private networks. The author found that DSS that did collect data from one place on the internet and present it, perhaps with a new interface or after processing, such as (39) to a client user, still operated using client-server model simply with some remote databases. In Figure 2.3, the author gives his view of the generic architecture diagram for internet DSS.

A key feature of many of the DSS that the author reviewed that he places in category 1 - 5 was that they were all built with a certain, fixed, target user group in mind, usually by a single company, government department or authority. For example (64) describes a recent (2004), highly sophisticated, category 5 DSS which the authors call a “continuous operational real-time monitoring system”, this was built specifically for the US National Ocean Service for the National Ocean Service watch standing personnel. This system collects sea environmental data, from many field stations in real time, collates it and presents it to National
2.3 Decision Support Systems

Figure 2.2: Category 1-4 DSS Architecture
Figure 2.3: Category 5 DSS Architecture
Ocean Service personnel to enable them to monitor the environmental status of US ports.

Another form that the category 1 - 5 DSS might take is that of a software package to be bought and used by an individual or a company, for example Hornbuckle (30) describes the PocketPC version of the SIRMOD DSS that people can purchase to use for irrigation event management. Another software package, specifically designed for irrigation scheduling which the author classifies as a category 1 DSS, is WaterTrack\textsuperscript{TM}.

There is a gap that exists in the types of DSS described in the literature and that gap is the existence of a DSS that allows for much flexibility in terms of its connectivity to and user utilisation of data sources, be they field devices, such as sensors, or information resources, such as databases.

In addition to filling the gap mentioned above, the last few years have seen the expansion of such internet communications standards, such as eXtensible Markup Language (XML), Web Services Description Language (WSDL), the growth of machine-to-machine communications over internet protocol (IP) and the large number of cheaply available devices that can communicate over IP and this has created many new possibilities for DSS. The author believes that a category of DSS, which he has termed 2nd generation internet DSS, is able to be proposed and is done so in the Section 2.7 along with proposals for utilising decision theory in DSS design and back-end and front-end design.

### 2.3.2 2nd generation internet DSS - Category 6 DSS

There is the potential for a second generation of internet-based DSS. This second generation roughly corresponds to the notions of 'Web 2.0'. The key feature of the Web 2.0 concept is its Service-Oriented Architecture (SOA). (38) writes that:

> The SOA concept evolves from earlier component-based software frameworks. However, since Web services standards are based on readily and openly available Internet protocols, and are thus much cheaper and easier for companies to adopt, major computer and IT companies have quickly embraced SOA.

\footnote{Available from \url{http://www.watertrack.com.au}}
This results in non-technically trained end users of internet applications being able to:

participate in sharing information and enriching services. Users may offer their own contributions as open services to be composed into new components and services. In addition, the combined network effects of pervasive two-way participation are creating a phenomenal communal service architecture on the Web.

A definition of Web 2.0 using terms that are now familiar to most internet users:

Web 2.0 is characterized by blogs, wikis, mashups and hundreds of websites supporting social exchanges including tagged bookmarks, pictures and personal profiles. This supports universal simple approaches to publication, composition of services [...] (18)

Generally, Web 2.0 is a phrase referring to a perceived second generation of internet or web-based services that allow internet users who are not technically trained use the internet interactively where formerly they had only been able to use it passively. The author believes that a Category 6, 2nd gen. internet DSS (2Gi DSS) could use the concepts of Web 2.0, SOA and particular technologies such as Web Service, for their back-ends, and AJAX (see Section 2.5) for their front-ends to achieve 3 new major goals:

1. A variety of DSS interfaces that use the same back-end 'engine' and allow for both different user types and multiple access methods for users

2. DSS back-end architecture that allows the information sources it uses to be changed fairly easily by the DSS developers

3. DSS front-end architecture that allows users to easily integrate their own information sources, as opposed to simple relying on those utilised by the DSS makers
The author could not find examples of DSS that catered for many different use cases. The author believes that this is due to most DSS design being focused on a single user group with homogeneous skills but this is definitely not the case in irrigation. Unlike a company that deploys a DSS and mandates that its staff use it and trains them accordingly, irrigators who may use a DSS probably would not be able to be forced to do so and there is unlikely to be a centralised training program in place for irrigators. A DSS designer in the irrigation sector could see irrigators with different skill levels, motivation levels or connectivity requirements as different use cases. This requirement for irrigation DSS should lead the author into allowing options for different user experiences of the DSS, even by users utilising the same access medium.

2.3.3 Using Decision Theory in DSS

The author feels that there is significant work still to be done in linking concepts and lessons learned from Decision Theory, especially those that relate to how people react to risk and understand probabilistic concepts, to applications in DSS. An example of how people can favour the certainty of variables over uncertainty, even when this action would reduce their monetary returns due to a lower probably expected value, was given in Ellesberg’s Paradox in Subsection 2.2.2.1. If one wanted to help decision makers faced, with Ellsberg’s Paradox, to make a decision that would yield a higher monetary EV, one could potentially construct a DSS to do so, as long as the DSS designer was on the side of the decision maker.

The reason for this statement at the end of this section on DSS is that the author has seen many applications that purport to assist in decision making (thereby making them DSS) that have poorly designed front-ends or poorly designed processes that do not allow an untrained decision maker to easily gain an advantage in terms of decision making from using the system. This would be like making a DSS to help a person make a decision in an Ellsberg’s Paradox environment but not have the system adequately present the various variables so that only a person familiar with the concepts of decision making under uncertainty, uncertainty aversion and probability, would be able to make an effective decision. A well
designed DSS here would reduce the value, to the decision maker, of uncertainty avoidance and allow the decision maker to then focus his or her efforts at placing a bet that would maximise the probability of financial return or other return

An example of this type of poorly designed front end, that the author calls ‘unintuitive’, would be the stereotypical ‘screen of numbers’ that people see at stock exchanges. These screens, displaying ‘when/what to buy/sell’ are really only interpretable by highly trained stock traders: an untrained person would probably not find much utility in this system.

This sort of worst-case scenario of an unintuitive DSS can be seen to be ineffective, or at least not maximally effective, by a person with basic common sense and some simple interface design experience. What may not be readily apparent is that the DSS may actually be leading people into making decisions that are biased in ways that are not easily appreciated, for example, if the potential Ellesberg’ Paradox DSS were built and the paradox’s traps were now explained, decision makers could easily use the system without them, or the DSS designers, realising that the decision makers were biased against the Yellow/Black uncertain ratio choice. If neither group had an understanding of the decision theory explaining the inconsistencies in average people’s decisions in Ellesberg’ Paradox, the fact that the decision makers were not realising maximal financial returns might go unnoticed. Issues relating to DSS front ends and how to accommodate decision theory best-practices are examined, in detail, in Section 2.5.

1While in general ‘uncertainty avoidance’ may be a worthwhile pursuit on its own (it may for instance reduce the stress experienced by the decision maker) and therefore something that may be aimed for as a primary decision making goal, an intelligently designed DSS might be able to remove uncertainty from the decision making process altogether. This is uncertainty in knowledge or education and probability and not uncertainty in in the facts of the decision. In the example of Ellsberg’s Paradox, uncertainty in what probabilities mean and how to interpret them, rather than uncertainties in the data available, such as the total number of balls or the percentages of coloured balls, is tested and it is this that must be avoided.
2.3 Decision Support Systems

2.3.4 Section Summary

1. Few different physical interfaces mediums have been developed for DSS. The author feels that many interfaces can be created that allow users to access DSS functionality in a variety of ways, that there are few examples of these in the literature and that more and more interfaces are becoming relatively easy to deploy, due to products such as the Microsoft '.NET' Framework. This programming framework allows much of the same back-end and some of the same front-end code written for computer-based programs to be used on mobile devices very easily.

2. Different use cases for an irrigation DSS will need to be investigated and methods to educate the DSS users to remove some variability in their skill levels may need to be considered as the literature provided little guidance here.

3. The ability of a DSS to allow a user to choose which datasources of all those available to the DSS at it’s design time has not been exhibited in any DSS in the literature but that this functionality may be able to be included in future DSS designs in a way similar to a user turning on or off features on a service provider website.

4. The ability for a DSS to allow a user to create a user-defined data set (UDDS) does not exist in the literature but the advent of Web 2.0 technologies and technological standardisation along with ubiquitous computing resources may allow for this in future DSS design.

5. DSS design must be wary of the lessons learned from decision theory paradoxes and attempt to be ‘intuitive’ for decision makers to use. This will probably be a very hard goal to accomplish based on the author's viewing of unintuitive DSS front ends.
2.4 DSS Back-end systems

So far this document has discussed DSS at the highest architectural levels. This next section will delve into the lower level architecture of DSS back-end systems. By back-end systems the author means any subsystem of a DSS that does not interface with a decision maker using that DSS.

2.4.1 Data Sources

DSS range in the data sources that they use considerably. Some of these data sources are proprietary real-time sensor systems (64), non-real-time satellite imagery (6), web pages that can be scraped for text (39), local or remote weather stations (41), user input values and GPS coordinates in quasi real-time (30) and so on. There has been a recent surge in the agricultural and environmental context in the use of wireless sensor networks as data source providers.

2.4.1.1 Data Source connectivity

Industrial process and operations DSS, one of the oldest, and probably the most commonly seen type of large-scale DSS, tend to have biophysical sensors connected to private SCADA or other networks (increasingly internet protocol (IP) networks) that transmit data back to a central location. One large-scale example of this is that which is used to monitor real-time power line loading by the power company EnergyAustralia in their Sydney control room. Another example of a system that uses data sources in this way, also from a private network and also in real-time, is that which is used to assist in traffic congestion easing given in (28). Unlike the EnergyAustralia system, the traffic control system also uses historical information stored in databases, as well as current traffic situation measurements taken from it’s sensor network to provide decision support. This combination of current data and historical data is the norm for DSS dealing with complex situations where a single course of action, such as reducing the load on a power cable by single instance rerouting, is not usually possible.

Many papers [(25), (52) and (56)] detail DSS that deal, in real-time, with flood forecasting and warnings. These systems all use networked biophysical sensors to
2.4 DSS Back-end systems

collect information at a single point for quick decision making.

These sorts of DSS are usually the expensive, enterprise-scale DSS that governments and companies expect to pay large sums of money for as the sensor and networking hardware can be very expensive. If the DSS is to operate in real-time, then issues such as responses to peak information loads and the reliability of data acquisition make the costs of such as DSS very great. The CORMS-IA DSS (64) and the Water Resources Observation Network (WRON) (39) are multi-million dollar projects partially because of these factors.

As mentioned above, these sorts of systems are abundant and a DSS connected to a private network of biophysical sensors, category 4, or the internet and other networks, category 5, form the core of the author’s category 6 DSS, if that category 6 DSS is to be an enterprise-scale DSS.

DSS that utilise data from one or a few local biophysical sensors are the low-end versions of those mentioned above. An example of such a system is the irriMAX computer desktop display program for data from Sentek’s EnviroScan probes (44). Such a system is used, over a small, private, networks, to help him decide when his vineyard needs irrigating, based on soil moisture readings alone. This sort of system still costs the individual user a reasonable sum as the probes and their networking requirements are not cheap. In this example, the font end system for displaying probe data is supplied with the probes.

DSS that use only database or user entry data for their datasources are very common. Examples of this sort of database are common and universally used, if one considers that a simple spreadsheet computer program, such as Excel, is a type of DSS. A more standard DSS example of a user input-only system is WaterTrack™, which is a farm-level water balance calculator. It models water distribution, inflow outflow and storage on a farm (55). It is a stand-alone desktop application. This is an example of a modern category 1 DSS.

A desktop program, similar to WaterTrack in terms of data input but with additional data collection features, is MaizeMan (67). This program:
“works by allowing the user to input information for each paddock, starting with the conditions the crop will be grown under its location, weather, rainfall and soil type ... The user inputs information about planting conditions planting time, seed type and variety, row spacing, rate, sowing depth and soil conditions.\(^{(8)}\)”

This program allows the user to import variable weather data, in quasi real-time, from certain organisation’s web pages that present information from those organisation’s private weather stations and weather information service pages that are publicly available or available for public purchase. This is an example of a category 5 DSS in that it utilises the internet to access information from more than one 3rd party private network. In the case of MaizeMan, the private weather station information is from the CSIRO’s Land and Water website (http://www.clw.csiro.au/services/weather/) and the publicly purchased information from SILO (49).

Desktop applications that connect to the internet for some information are now very common with most office and home computers connected to the internet for at least some of the time. This means that many information providers, such as SILO\(^{1}\), choose to present their data using the internet protocol (IP). In Australia, the USA and other countries where much meteorological information is readily available, either freely or at low cost, over the internet, the author believes practically all future agricultural DSS, where weather information is a commonly required datasource, will be designed to access IP data. Further to this, practically all wireless and other devices, such as SmartPhones, that connect to the internet or large public network, such as mobile phone networks do so using IP. IP is by far the dominant networking format and becoming practically ubiquitous, therefore the author further believes that even hand held devices that offer decision support in agriculture will connect to weather information using IP.

Access to data, other than weather data, over the internet is also becoming increasingly available. Non-biophysical data, such as economic data like water or

\(^{1}\)SILO is an internet-based service that providers daily meteorological data for all of Australia. see http://www.bom.gov.au/silo/
livestock prices are increasingly being placed under IP access and websites such as http://www.watertradingaustralia.com/ list water prices for irrigation areas in Australia.

Finally, in this subsection of data source connectivity, the author feels it is worth reiterating that this dominance of a single networking standard and the advent of data formats and specifications, such as XML could lead to User-Defined Data Sources (UDDS) appearing in what the author would consider to be a category 6 DSS. This was originally stated in Subsection 2.3.2.

2.4.1.2 Data Storage

Originally the author intended to devote a large section to data storage but feels that this topic need only be addressed briefly in this literature review. The reasons for this are that:

- virtually all computerised data are currently stored in relational databases. This leaves little room for discussion with regard to how data is logically stored.
- the concepts of relational databases are widely known and need not be discussed here.
- virtually all computerised data are currently physically stored on computer hard drives, leaving little need for discussion.
- the exact 'where' and 'how' of data storage is unimportant when looking at a DSS in an other than computer engineering mode.
- the systems likely to be used by the author are unlikely to be radically different from those used by most other IT users due to the high cost and availability of unusual systems.

If data storage becomes a problem or is of significance in this PhD, it will be noted in other chapters.
2.4.1.3 Data Sources

Agricultural DSS obviously call on more data sources than just weather information as there are many factors that are pertinent to crop and livestock growth. A list to show the diversity of data sources found by the author, to be in use in existing agricultural DSS are:

- weather radar
- local weather station data
- soil type information (spatial and otherwise)
- water price
- water reserves in storage
- water availability
- soil moisture levels
- satellite imagery
- calendar information

Data sources are presented in different formats and at different time-steps as well as being available through different mediums. Network access to data sources was examined in the preceding section and data source time-step and format information for some of those data sources listed above are presented in Table 2.2. Also included in the table is the access method as seen from the user point of view, rather than from the networking point of view, for future reference.

This great range proves to be problematic to the DSS designer as it make the concurrent utilisation of different data sources very difficult. The spatial and temporal aspects of the above listed data sources prompt the author to consider what standards might need to be utilised/created to facilitate the use of a wide range of data source types in a DSS. Some data standards and exchange formats are investigated in Subsection 2.4.1.4 and the author has undertaken much investigation into how heterogeneous datasets may be used by a DSS and these are
### Table 2.2: Data Sources tabled with some of their attributes

<table>
<thead>
<tr>
<th>Data Source</th>
<th>Timestep</th>
<th>Data Format</th>
<th>Access Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>weather radar</td>
<td>5 minutes</td>
<td>spatial image</td>
<td>public internet downloadable</td>
</tr>
<tr>
<td>soil type maps</td>
<td>static</td>
<td>spatial image</td>
<td>private database</td>
</tr>
<tr>
<td>soil moisture</td>
<td>order of minutes</td>
<td>point data</td>
<td>direct connection to a probe</td>
</tr>
<tr>
<td>satellite NDVI</td>
<td>fortnightly/ daily</td>
<td>spatial image</td>
<td>privately downloadable</td>
</tr>
<tr>
<td>imagery</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>water price</td>
<td>daily</td>
<td>single number</td>
<td>public internet downloadable</td>
</tr>
<tr>
<td>decision maker’s</td>
<td>continuous</td>
<td>temporal values</td>
<td>privately stored</td>
</tr>
<tr>
<td>calendar</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
The author has noticed that calendar information is not often seen as a data source. Often, as is the case with irrigation water ordering websites like that which is used by Sunraysia (58), calendar information is presented as an operating framework and then information, such as water availability, are thought of as data sources. Water ordering websites need not be classified as DSS, even though they assist in water ordering choices, as they don’t usually provide support to an irrigator ordering water other than tell him when it is or isn’t available but the author believes it is obvious that irrigation scheduling DSS tools will naturally accompany those water ordering portals. To the author, the Waterright(17) website is exactly the kind of website that could be looking to combined its DSS functions with regional water suppliers to enable water ordering directly from the DSS. If cases such as this occur, then calendar information from the irrigator, water authority and other 3rd parties will need to be thought of as datasets so that they can be weighted alongside other information and used to provide decision support, rather than decision dictation.

Many programs now exist that allow people to share calendar information (the commonly used Microsoft Outlook and Microsoft Exchange server allows this) and some programs allow calendar sharing over the internet (Google’s Calendar allows for the sharing of public and private calendars) but this sort of calendar sharing is not entirely what the author has in mind. Calendar sharing will be addressed again in Subsection 2.4.2.1 along with other dataset fusion.

The author will need to undertake research into what other possible data sources may exist or can be created for use by agricultural DSS at a later stage in this PhD.

2.4.1.4 Data format standardisation

For data of different types to be used effectively by DSS, data standards and interchange formats will need to be examined in detail.

Already mentioned in Subsection 2.4.1.1 was that IP has become by far the most common networking standard for both private networks as well as its original area, the internet. This protocol, along with Transmission Control Protocol
(TCP), commonly referred to as TCP/IP, eliminates the need to look at data format standards below the 'session layer'. In terms for those unfamiliar with network infrastructure, this means one does not have to consider what hardware (servers, routers, cables) is used to transport data, how that data is directed over the network/internet or how DSS machines enable sessions to ensure authenticity of the data source. This leaves only the 'presentation layer' (how data is presented, such as in text format, XML or binary files) and the 'application layer' (user interface programs) to be considered.

Many people are familiar with at least some 'presentation layer' formats, such as text and XML, and it is at this level that discussion about data format standardisation is most useful. Discussion about the 'application layer' can be left to discussions about user interfaces (see Section 2.5).

The data formats most commonly used for situations where network bandwidth is at a premium (such as wireless applications) are invariably binary formats. Binary encoding can be translated into any other digital form, such as images, text, video etc but they are not 'human readable', that is they are not able to be read directly by people without a conversion process. If a single system is designed with fixed inputs and outputs (this is the case for the vast majority of systems including DSS), then binary formats are best used. Data sent from sensors or probes to a computer, from one computer to another, from a database to a decision engine etc have traditionally been sent in binary form. For a system where the end users are not using a single, set, interface (for example different internet browsers) or where data can come from a number of different sources that are not all created by the system designer, then XML is best used for data transmission. This is due to the interoperable nature of XML for is an open source standard which allows its free use by any organisation and defines formats that are predictably interpretable and also human readable. XML is usually network bandwidth intensive when compared with other formats.

\footnote{The term 'session layer' refers to the Open Systems Interconnection Basic Reference Model (OSI Reference Model) layer 5 in the 7 layer model, which is detailed in ISO 7498. A description of the OSI Reference model, other than that given in the paragraph above is not within the scope of this PhD, however the ISO standard can be downloaded from the ISO website.}
2.4 DSS Back-end systems

Literature dealing with recent, category 5 DSS assumes the use of XML if the DSS communicates via Web Services either with back-end data sources or front-end users for XML is the data format used by Web Services for machine-to-machine communication over the internet. It is used to allow devices with different, unknown programs and underlying operating systems to communicate.

Since the uptake of XML is ubiquitous and network bandwidth capacity is ever increasing (for wireless communication for example, 3G mobile networks are now seeing full implementation worldwide with Hutchison’s ‘3’, Telstra’s ‘NextG’ and Optus’ 3G networks across Australia (37)) the author assumes that future DSS will use XML almost exclusively for external communications.

XML and other internet standards derived from XML, such as XHTML, have been extensively promoted by all of the large players in the software market, such as Microsoft, Mozilla, Oracle etc which leads to the assumption that it is ‘here to stay’.

In an assessment of Australian on-farm water management tools, (33) recommends that programmers use XML to link software models and modules.

2.4.2 Data processing

All DSS offer some sort of data processing to value add to the data that is collected by the system. Data processing is wide ranging in its implementation with some systems simply aggregating data and presenting it in a single place, such as water height data viewed on the WRON website (39), and others running data through advanced statistical and mathematical processes, such as (56) which uses Bayesian Networks.

For ease of discussion, the author will break data processing into two stages. These stages are helpful when assessing all of the tasks a data processor might undertake. The stages defined here are:

- data fusion
- data analysis

Data fusion is primarily concerned with assembling and comparing data in a meaningful way, so it can be thought of as a preparatory stage, and data analysis
2.4 DSS Back-end systems

is primarily concerned with taking fused or un-fused data and using it to guide decisions and make predictions.

2.4.2.1 Data Fusion

Data Fusion is a large and complex topic. Early work in the area was undertaken by the US Department of Defence which commissioned large studies in the data fusion area with a focus on multi sensor and real-time data fusion. Presentations such as (43) describe how sensor data fusion can be used by military field units for friend/foe recognition and targeting where multiple, heterogeneous, sensors, such as radar sources, satellite imagery and laser rangefinders are used. These papers discuss data fusion at its most complex and time-critical end where battlefield equipment must be used for fast decision making under uncertainty and where lives are at stake. This typically means that the systems that use data fusion in this way are very costly and specialised so that the papers in which they are presented are of limited use in applications such as agricultural decision support which are quite different in many ways. They do, however, show that very different data sources can be effectively used in single systems, if the resources are there to facilitate it.

More recently papers have been produced on frameworks for data fusion from the sensor level right through to the information level, again originally for US military use but increasingly for civilian use. Some of these papers are enlightening in that they describe different approaches to data fusion that could be critical in designing a DSS that relies on data fusion.

(4) describes data fusion techniques in terms of when to fuse data: centralised, stored data fusion being one technique that fuses data from multiple sources at collection time and stores it, as fused data, in a centralised 'fusion' database. The opposite end of the fusion time technique spectrum is that which sees data left un-fused and stored separately until a user requests a process that needs fused data at which time it is then fused at 'run-time'.

Determining which of these fusion timing techniques to use will depend on the ability of the system to store information and make rapid calculations. DSS devices with small computing resources, such as SmartPhones, may not be able
to perform fusion at request time.

(26) asks where in the data flow, from source data to information as viewed by the user, one should actually fuse data. The paper’s author suggests that one may consider fusion at the following levels:

1. data level - fusion of raw input data
2. feature level - fusion of extracted data taken from the raw input data
3. decision level - fusion of data sets that have already had data analysis run on them

Noted by both (26) and (4) is that the highest level of data fusion takes place when humans manually view datasets and perform the fusion in their brains.

Data level fusion is commonly seen using pattern recognition and templating methods, such as neural networks and clustering algorithms. (26) tells us that in order for this level of data fusion, to be possible, the original data sources or sets from a source need to be homogeneous. An example would be using pixel-by-pixel analysis on images of a field taken at different wavelengths (this is done for NDVI imagery) or pixel-by-pixel analysis of two images of a field taken at the same wavelength but at different times.

Feature level fusion is undertaken when the data sources or sets to be fused are of a heterogeneous nature. Information vectors can be extracted from the raw input data and then fused. Hall gives no example of this in (26). This author imagines the following example: a field’s high yielding areas, as determined by a yield monitor with 1m x 1m pixels, could be extracted and compared with the general (extracted) trend in soil infiltration rates in a single direction for that field and fused to reveal the correlation of the two.

Decision level fusion takes place at the highest level of the three given here and could take the form of a Bayesian Network’s usage of many datasets to present a DSS user a single figure estimation of the utility of a decision that is affected by multiple criteria.
These papers on data fusion levels suggest that it is very likely that a future DSS that uses both multiple homogeneous and multiple heterogeneous datasets will need to consider data fusion at a number of levels. As this is predictably the case for any future wide-ranging agricultural DSS, new research will be needed into how best to structure these various forms of data fusion so that fusion at one level does not clash with fusion at another level, if both sets of data fusion use the same source data.

These papers, as well as others on heterogeneous datasets, such as (62) are quick to point out the complexities involved with meaningful heterogeneous data fusion. With this in mind, and the fact that not many previous agricultural DSS have used fused heterogeneous data, the author anticipates that there is much room for research here. In addition to this, the author notes that when giving examples of what the authors of such papers consider to be heterogeneous data, their examples are often limited to digital data in different spatial formats such as 2D and point source or temporal formats such as 1 second and 1 day timestep data. Little mention is made of extremely heterogeneous data such as soil moisture point data and annual calendar holiday data. The author believes that there are few systems that attempt to fuse such different data.

Despite heterogeneous data fusion appearing to be most useful (there is obvious utility in comparing wide ranging datasets like calendar data and soil moisture probe data to help one in making a decision), the author was not able to find detailed examples of heterogeneous data fusion during the course of writing this literature review. The statement “feature vectors extracted from original datasets” given in (26) was most as detailed as the descriptions got. This is in contrast to homogeneous data fusion literature which readily identifies neural networks, genetic algorithms etc as technical methods commonly used. The author believes that this is symptomatic of the fact that heterogeneous data fusion is very difficult and little can be said generally of the process at a technical level without actually delving into the specifics of particular projects. This once again emphasises the complexity of heterogeneous data fusion.
Information fusion is sometimes thought of as a form of data fusion. Information fusion relates to data fusion at the highest level; that at which data is finally viewed by humans. Papers such as (45), although military focussed, are useful to anyone that plans on using data from multiple sources as it states that good standardization of input data can lead to far less expensive fusion techniques needing to be undertaken. The paper suggests that input data standardization is in fact one of the largest challenges faced by US military systems that rely on more than one of the vast range of data sets that the US military has generated over the past few decades. A specific example from (45) that may be illuminating is that a single body in the US military, often known as the Global Command & Control System - Maritime can also be found on various databases listed as the AN/USQ-119(V)3 and the JCIS or C2PC which would require any system that needed to use more than one of these databases to run unnecessary processes to operate correctly.

The author’s impression is that while information fusion may not present technical problems equalling the magnitude of heterogeneous data fusion, it is a significant problem to overcome where diverse data sources, created at different times by different people are used, as is the case in the agricultural sector. Papers such as (45) and thoughts on information standardisation generally point out that problems relating to information fusion may always be with us as long as separate entities provide data that is not commonly arranged for potential future for combined use.

Separate to the technical problems involved with data fusion and one that will definitely be important for irrigation DSS is that of at what level to fuse data for optimal user interpretation? Different tasks (such as asking for timing advice for a single irrigation or for a advice on laying out a whole season’s scheduling plan) may require different levels of data fusion for even a single user. Potentially a user would want to choose a fusion level to suit their particular task’s requirements.

2.4.2.2 Data Analysis

Rule-based and case-based reasoning are two techniques, originally from the world of artificial intelligence (AI) that have found widespread use in the world of
decision theory in the last two decades. These forms of reasoning and hybrids of the two are currently used as the 'engines' that drive the decision support predictions for some of the largest DSS in operation, such as CORMS AI.

Rule-based reasoning (RBR) is a familiar concept for software developers and has been used since the very early days of computers. It is concerned with a set of rules that dictate a system’s output. Those rules typically follow a logical IF-THEN-ELSE statement structure. A simple example is when a user, purchasing online tickets to an 18+ event, is asked to enter in their age, the website would respond to the input by reading it and performing an IF-THEN-ELSE function on the value as in the pseudocode given below.

IF age_entered < 18

THEN print ‘‘Sorry, you are too young for this event’’

ELSE print ‘‘You are able to attend, please enter further details’’

RBR, as seen in a DSS environment typically consists of a set of defined rules that current facts are tested against to lead to an output. Rules can be cascaded so that the operation of some rules is conditional upon the operation of other rules for example if the event used in the example above was also limited to people under 30, an age input of equal to or greater than 18 would then need to be checked to see if it was smaller than 30. An age of smaller than 18 would not need to be secondarily checked in this way. In this way RBR can be used to recognise patterns and classify events with the great challenge lying in determining rules for a particular environment.

CORMS AI uses 16 distinct rules that assess incoming environmental data and present different information to system users, based on the conditional outputs of those rules. In this case, as in most RBR implementations, it is the staff with knowledge of the decision area, not of decision support software, who determine the rules. Expert knowledge of the types of data collected by the system, how to interpret that data and the outputs required, was needed to generate the 16 rules and this knowledge was supplied by the NOAA (the CORMS AI host
organisation) staff, not the RBR system design staff.

An example of a possible agricultural DSS using a rule-based system is one where a red light flashed when a truck being filled with earth was about to become overburdened by weight: in this instance, visualising the earth level in the truck may not be sufficient to predict the truck’s mass due to different earth densities so that a pressure sensor could be used to check the weight of the truck every second to see if it had reached a certain value. When it had done so, the light flashes to tell the operator to stop filling, regardless of how full the truck appears.¹

RBR will see much use in any DSS at many levels due to its fundamental use in computer systems. With regard to a reasoning engine that might use a rule-based approach for decision support in future irrigation DSS, defining what rules to use and what condition levels to use within those rules becomes the main task of the DSS designer. It is at this point, the definition of a rules-based reasoning engine’s rules, that the link between decision-making criteria from the unsupported decision making environment and the DSS environment is explicitly made and technical design meets non-technical requirements in terms of support utility.

Many RBR engines for irrigation DSS have been used and most of them focus almost exclusively on scientific criteria for irrigation management, such as evaporation rates and soil moisture. For a recent example of a category 5 DSS that uses RBR in this way but ignores non-scientific information, see WaterSense[34]. Much work has also been done on holistic approaches to irrigation decision making with recent reports such as [46] that attempt to quantify as many rules, scientific and otherwise, stated and hereto not stated that are used by irrigators in decision making but none of these studies have yielded a holistic irrigation DSS that supports non-scientific rules.

¹This is a simplistic example used to illustrate RBR systems however the author anticipates that, due to the complexity of their nature[46], decisions in agricultural environments will be considerably harder to generate rules for and that those rules may not be as ‘clear cut’ as the one in the example above. Potentially logic other than IF-THEN-ELSE logic (Boolean logic) may have to be applied to situations to generate rules that cater for less clearly defined requirements.
A very interesting research question that could be asked is: given the recent advances in data format standardisation and networking standardisation as well as computing power and computer literacy, how exhaustive a list of rules can be made for an irrigation decision making reasoning engine, in the first instance of purely scientific criteria and in the second for all criteria? Would the bottleneck in the DSS’s abilities be found in the ability of DSS designers to define a comprehensive set of rules to be used, the accuracy of the rule thresholds or the heterogeneous data fusion required to allow a single collection of rules to operate in a single coding environment?

To rephrase the above more generally; multi criteria decision making with respect to irrigation is seeing much research recently and many factors, from the scientific to possibly the spiritual are known to affect irrigation decision making. How realistic is it, in terms of the decision theory and the actual execution of that theory’s results in computer code to assemble all of those known factors into a single rule base?

There is little work in existence that allows irrigators to define their own rules for decision support. The APSIM\(^1\) modelling framework allows users to customise their decision support to some extent by entering rules into the system\(^{21}\) however APSIM is not a decision support tool for irrigators and the rules able to be entered are scientifically based only. Potentially a DSS that allowed for a vast range of user-defined rules to be entered could help address the question of bottleneck location stated above.

The author considers Bayesian Networks, a probabilistic dependency network used to gauge the effects of changing multiple inputs on an output, often graphically, as a form of RBR. Initially he had planned to consider Bayesian Networks in full in this literature review but with the thought that they can be classified as one type of RBR, no more needs to be said about them here as they are therefore not a topic on their own but a RBR sub-topic. Their inclusion in or exclusion from future DSS will be considered elsewhere.

The use of case-based reasoning is much more recent in computing than the use of RBR and may help address some of the problems encountered by that

\(^{1}\text{http://www.apsim.info/}\)
older approach. It is adapted from work in human decision making theory, such as Wittgenstein’s *Philosophical Investigations*\(^\text{(71)}\), already referred to in this literature review in Section 2.2.2.1\(^1\). Case-based reasoning (CBR) is a computer’s equivalent of everyday human decision-making processes in which humans rely on their memory of previous experiences to help them make choices when faced with new problems. It is defined by (1) to be taking place when a system is:

> “able to utilize the specific knowledge of previously experienced, concrete problem situations (cases). A new problem is solved by finding a similar past case, and reusing it in the new problem situation. A second important difference [to other reasoning techniques such as RBR] is that CBR is also an approach to incremental, sustained learning, since a new experience is retained each time a problem has been solved, making it immediately available for future problems.”

(1) describes the process of case-based reasoning as taking place in the following 4 steps:

1. **Retrieve**: Given a target problem, retrieve data from and about previous cases that are relevant to solving it. The full range of data for a case is a description of a problem as well as it’s solutions as well as, importantly, annotations about how the solution was derived.

2. **Reuse**: use the data from the previous case to solve the current problem. This may involve modifying the solution to fit the new situation.

3. **Revise**: After adapting the previous case to the current situation, test the new solution and, if necessary, revise it.

\(^1\)The link here is that Wittgenstein did not think it was useful to attempt to create a formal definition of any concept in the natural world, as there are always myriad ways to classify something, and therefore natural world concepts are best understood in context such as in instances of their use, in other words 'cases'
4. **Retain:** After the case has been solved, store the problem description, solution description and data about how the solution was reached as a new case for future use.

CBR is held to be more complex than RBR to implement. Where in RBR the challenge for the implementation designer is to decide what characteristics of decision scenarios must be quantified in rules and what the choice thresholds in those rules are, in CBR the designer must decide what characteristics of decision scenarios need to be used as case characteristics. In addition, the CBR designer must also deal with how to build some sort of case base memory structure that has the ability to be dynamically altered in order to 'learn' from new cases stored in it. (1) details a standard method that had been employed for dealing with the additional challenge which is as follows: Cases are parameterized and stored in a hierarchical structure that abstracts common features from similar cases into 'generalised episodes' (GE). As new cases come to hand they are compared initially with the most general GE and then progressively more specific GEs until finally they are compared to cases. Once certain parameters are matched, other parameters are compared to GEs for further matching. Newly solved cases are slotted into the case base as far down the hierarchical GE tree as possible with new GEs created if the new case matches cases that, until this new addition, did not have a match. For this method’s efficacy, much work must be put into how generalised a GE can be and how many levels of GEs the case base can be allowed to have.

The author notes that there are now numerous papers published on CBR that list numerous, more recent, techniques for facing this challenge but time prevents this author from examining the pros and cons of the various other systems described. Already though, the author notes that questions exist here as to how many CBR case bases have been developed and tested in the irrigation sector.
with irrigators’ individual irrigation events considered cases and how many systems have been developed that compare irrigators’ individual irrigation events to those of other individuals in a case base that therefore acts to extract benchmarked ‘best practice’ trends. The author was not able to find much literature that detailed studies of this find at all.

Many examples of decision support engines that use CBR that are quite different from the CORMS AI hybrid implementation are recorded in (1). Two of them are the HYPO system that was designed to use CBR to interpret courtroom situations in the USA, based on legal case cases and the CASEY system that attempted to predict heart failure in patients that used medical test results as parameters for individual patient ‘cases’. This range of CBR application is so wide and the few examples noted above are so different that the author suggests CBR could be adapted to the irrigation sector with no more difficulty than to any other sector and that perhaps poor uptake of the technology is due to facts other than the nature of irrigation itself.

*Prima facie* the author believes it is not unreasonable to think that CBR may assist greatly in irrigation decision support especially when one considers the yearly repetitive but variable nature of irrigation. From this speculation comes the question: how comprehensive a CBR engine can be built for irrigation DSS and would that engine be subject to the same bottlenecks as those faced by a rule-based engine?

A corollary to this question is: what advantages do hybrids of case- and rule-based reasoning engines provide? (65) investigates this very question in relation to the CORMS AI DSS for the US Ports Authority where the considerations given to using rule-based, case-based and hybrid systems were given before a final selection was made; a hybrid system was finally chosen in an attempt to harness the power of both RBR and CBR. It seems, from the process given in (65), that answering questions such as whether to use RBR, CBR or both is a very large and extremely expensive task. Great expenses incurred in the instance of selection for the CORMS AI system were largely due to the magnitude of the physical network of sensors and therefore their produced data, the real-time nature of the

44
DSS and the requirement that it be used by staff members on a large scale.

The author contacted the 3 companies short-listed for the implementation of the CORMS AI DSS mentioned above in order to gain some insight into the sorts of CBR/hybrid systems that commercial software vendors provide. The author was interested particularly in the range of possible applications for the companies’ products and general costs of the CBR systems. The three companies were MindBox, whose CBR platform ‘ARTEnterprise’ was eventually chosen for CORMS AI, Kaidara Software who make the ‘Kaidara Advisor’ program and Gensym who make the ‘G2’ CBR platform. Gensym was contacted by way of their Australian partner firm Daesim.

A sales representative from MindBox informed the author that

“there is really no generic ARTEnterprise implementation – the efforts, rates and costs all depend upon the specifics. Generally we implement mission-critical applications and the shortest timeframe is rarely less than 3 months with a team of 3 people. This time would include requirements analysis as well as testing. Our longest implementations are around 8 months – with 25% of the time being in testing. ... our rates are typically mid-range software consulting rates” (54)

The author believes that this sets a minimum figure for the cost of MindBox ARTEnterprise implementations at around $AU75k for the cheapest implementation of the an ARTEnterprise system. This was the cheapest of the 3 companies’ products. In addition to the information above, the MindBox sales representative noted that they have had little contact with the agricultural sector and certainly none with irrigation.

Kaidara Software responded to the author’s request for information by saying that, like MindBox, they do not have a standard implementation for their CBR systems (59). In terms of pricing, Kaidara stated that “Regardless of the application the pricing for our type of system ranges from about US$250 to US$400k. With the majority of the cost related to software licenses (50%)”. Relating to
the nature of the product, Kaidara said: “the majority of our use cases [system implementations] relate to solving problems for cars, planes, medical equipment and high tech gear”. Kaidara could not think of any implementations of their software in the agricultural sector.

The final company’s product was rated as such by (65):

“Gensyms G2 product was by far the most impressive of the tools we evaluated. However, we estimate that the impact to the organization would be significant if this product was selected. Investment in this tool could be justified if it was decided that the overarching methodology for managing the PORTS and NWLON data reporting and quality control systems were to undergo a dramatic redesign. The power of this tool for modelling and graphically displaying the attributes (in real time) of any system (potentially from the Space Shuttle to PORTS), and warning of data quality or system failures is impressive.”

The author engaged in telephone conversation with a Daesim representative who was very familiar with Gensym’s G2(69). He was told that the G2 license fee was US$55k and a basic installation total cost would be somewhere around $US200k. The author was informed that, within the commercial CBR vendor community, the general attitude was that knowledge mapping, the process of defining the attributes of cases in a DSS environment, was the hardest part of a CBR expert system (enterprise DSS) implementation. In addition, in response to questions about G2 and Daesim experience in the agricultural sector, the Daesim representative stated that they had done little work in that region but that the assumption by the author that irrigation was a very different field for decision making to other fields which have seen enterprise DSS usage, was most likely unfounded. The Daesim representative stated that multiple use cases, some with little training, in terms of DSS end users are catered for by expert systems in medical, insurance and other sectors.
2.4 DSS Back-end systems

From the interaction with DSS/enterprise system software vendors, the author believes that the irrigation sector has not seen commercial CBR DSS usage and if it were to, it would cost hundreds of thousands of dollars to use a system, such as those mentioned above, that are currently used in other business sectors. In addition to this, purchasing could only take place once the considerable task of knowledge mapping was completed. The author believes there would be much value in looking into the processes that the medical and others sectors undertake for knowledge mapping. Investigations of this sort are outside the scope of this literature review due to the sheer volume and complexity of work in the field but may be inside the scope of this PhD if reasoning engines are pursued.

It is not possible for the author, at this stage in his PhD to enter into more detail about CBR as it is a very complex field with much research and many implementations to consider.

2.4.3 Section Summary

- For physical connectivity, TPC/IP networks are becoming all pervasive. This suggests future DSS back-ends should use TCP/IP.
- Data storage is not thought to present many challenges or opportunities for research for irrigation DSS.
- DSS system, user and 3rd party calendars are not currently considered datasets and perhaps could be.
- Research into the extent of data sources available is needed by the author.
- Research into likely future datasets is needed before a DSS is constructed if it is to have forward compatibility.
- XML is the data format standard of choice where heterogeneous systems are used.
- Data fusion timing will need to be carefully considered if a DSS is to be accessed via multiple devices that have different calculation abilities, such as SmartPhones and powerful desktop computers.
• Research is needed into using data fusion at multiple levels concurrently if a future DSS is to use both homogeneous and heterogeneous datasets.

• Few, if any, DSS exist that attempt to fuse radically different data sets such as calendars, soil moisture levels and yield monitor results.

• Information fusion may prove to be a very large task for a DSS designer who plans on using many heterogeneous data sets and this is usually the case in the agricultural sector.

• Rule bases containing rules derived from non-scientific as well as scientific decision criteria have not been used for irrigation scheduling.

• No DSS exist that allow irrigators to enter in their own rules into a rule base.

• Few off-the-shelf reasoning software packages, if any, have been used in the irrigation sector.

• Little work, if any, has been done to use irrigation events as cases in a case-based reasoning engine.

• Has CBR ever been used with irrigation benchmarking? The author believes not.

• Commercial CBR software and it’s installation is very expensive.

• Knowledge mapping for the irrigation sector could be undertaken and may learn from the medical and other sectors.

• Diverse use cases have been catered for by commercial CBR systems in the medical and other sectors and this may prove of use in irrigation DSS use cases.
2.5 DSS Front-end systems

User interface (UI) or front-end subsystems are an essential part of DSS design as all DSS must eventually present their calculations or predictions to a human user to assist in decision making.

This section addresses front-end subsystems of DSS, or those systems that interface the DSS with end users. This section will consider the existing and possible future ranges of physical interface mediums, problems and techniques for presenting information from a DSS to users, ways in which users may feed information back to a DSS and also interfaces for different user groups. The ability of a future DSS to be used by multiple user groups (use cases) and to use different delivery mechanisms was mentioned in Section 2.3 but little detail was given. The interface components of that this be explored here.

2.5.1 Interface presentation types

Interface theory has necessarily been a part of computing theory since computing theory’s inception in the 1940s due to the fact that computers must eventually interface with humans if they are to be used. Early computing systems used punch cards, paper printouts, lights, segmented displays, command line consoles (such as Microsoft DOS) and text-based graphics as some of the more common interface types. Since the advent of the VGA monitor in the 1980s, graphical user interface (GUI) systems have been used extensively to the point where most computer users only use GUIs like Microsoft Windows or Apple’s OSX.

Recently, with the advent of mobile phones, there has been a re-visitation of text-based, console, simple GUI (such as menu-driven interfaces) and other interface types due to the limited computing capacity of the phones and their capacity to physically display many pixels and colours. A recent search of the cheapest mobile phone found it to contain a basic, menu-driven, GUI. This shows that basic GUI-enabled mobile devices are now universally available. The current top end SmartPhone use Microsoft’s Windows Mobile 5. This interface is used in a similar manner as the familiar Windows for desktop computers (Windows 2000, Windows XP or Windows Vista). This indicates that mobile devices can now display content in a way similar to desktop computers.
Vast resources are placed by companies into the development of mobile interfaces with recent versions of Windows Mobile and Symbian by Microsoft and Nokia respectively being both easy to use without training and feature packed in comparison to earlier versions.

2.5.2 Information Presentation Theory

This subsection considers the theory of data and information presentation, rather than the physical media used for presentation. The author considers this section to be of fundamental importance to DSS, for the way information is presented to a user is critical to the usefulness of the system.

2.5.2.1 Examples of DSS Graphing

Many desktop-based DSS use graphical means to convey information. Custom software in the DSS sector can be generated via software packages, used for calculation that includes graphical packages. Such an interface for fisheries management, described in (63). It uses the program MATLAB’s graphical modules to produce spatial maps and graphs. In this particular example, very little detail of the GUI mechanism or it’s construction process were given which suggests that the authors felt it was not a significant part of the overall DSS. The images supplied in the paper show an interface that only someone familiar with the system could use as it is quite complex and this is also the case in (52) which supplies the DSS user with numerous graphs tracking wetland salt loads that firstly only have meaning for skilled staff (environmental scientists) and secondly require some specific training to interpret effectively.

In a paper entitled “Interface for displaying probabilistic river forecasts to decision makers” (25) the authors present a custom-built interface to display river height data which highlights, with alarm boxes, rivers that have reached certain levels chosen by the users. The various graphs and maps for this interface were presented to the user via a web browser but were generated by specialised software, the details of which were not given. As the ‘interface’ is the main work of the paper, more detail is given of it than in the previous paper mentioned but still little analysis of the design of the individual graphs is given. The author believes
that this is a result of the designers of the interface believing that the order of the
graphs and which graphs to show when, which is covered in the paper, is the only
tasks of interest to the interface designer as the individual graphs are of a type
that is well understood by the target user groups of the system. This final point
in the thinking, that the graphs will be understood by the target audience must
be a result of the DSS being either designed for an already trained audience, or
one that will be trained, or that the concepts to display are simple enough to be
understood by anyone.

Reports and papers, such as (58), (40) and (42) are examples of DSS that
use very simple tables and graphs, like one might find in Microsoft Excel, to help
irrigators decide when to irrigate. Little effort, in these papers, was placed in
making the graphs and tables that they present particularly easy to use: the
authors seem to have assumed that basic tables and graphs were inherently easy
to use.

The author asks whether there may be value in research looking at users’
understanding of individual graphs? This would be separate to looking at which
factors in a multi factor decision making environment would be most useful to dis-
play graphically which the author would consider part of the knowledge mapping
task mentioned in Section 2.4.2.2. DSS outputs that, in the past were displayed
as numbers, basic graphs and simple tables may be better displayed in a way
that is more intuitive for untrained individuals to use via a number of methods.
This could even avoid encountering some non-rational decisions that paradoxes
in decision theory bring out due to decision makers misinterpreting statistics and
probabilities or by placing undue emphasis on certain parts of basic front-ends.

In contrast to the simple font-end systems cited above, the irriGATE proposed
project, (60), answers the question ‘when to irrigate, based on evapotranspira-
tion?’ with the graph shown in Figure 2.4.

The author believes this control (by control the author means a single device
for communication between an internet server and a user viewing a webpage
through an internet browser such as text boxes, graphs, sounds and buttons)
2.5 DSS Front-end systems

Figure 2.4: The irriGATE project’s evapotranspiration graph does not lend itself to ease of use due to it’s complexity. The graph is easily understood by agricultural scientists but perhaps not understood without training by irrigators. If the target users of a DSS that employed such graphs as these were able to be trained then they may be able to gain considerable benefit from such graphs but if not then perhaps the graph would be bewildering for them.

Another example of irrigation DSS graphing is given in Figure 2.5. This image, from the APSIM-based sugar cane front end WaterSense (34) shows soil water deficit at different depths on the same timeline as irrigation events, ETo, rain and tolerance boundaries set by the irrigator for his or her reference. This graph has a zero level at it’s centre line with positive water events (rain and irrigations) above the line and negative events (crop water use) below.

The author believes that the interface in Figure 2.5 is more intuitive (easier to use by untrained people), despite having more display options, than that given in Figure 2.4 due to it’s positive/negative treatment of water balance, rather than a cumulative graph just referenced to ETo. However, the author feels that Figure 2.5 gives a scientific-, rather than irrigator-focussed perspective on crop water balance and therefore may also not be as easily used as it could be. The authors of WaterSence state that it is a research tool, rather than a marketable product.
but this author believes that for a DSS to be truly effective, it must be as easy to use as possible, regardless of whether it is to be marketed or used for research.

The author believes that the example of a user's interface page given in Figure 2.5 and the website (17), which contains many graphs similar to Figure 2.5, represent the current state-of-the-art in irrigation information and data visualisation. The author could not find papers that specifically researched the usability of DSS front-ends and information presentation or the cognitive aspects of information interpretation in the agricultural or irrigation fields. The author believes that there has been little work done in utilising the large amount of research in the cognitive informatics area. Cognitive informatics is a part of the informatics research in a number of organisations including the University of Edinburgh’s School of Informatics, the Institute for Computing, Information, and Cognitive Systems at the University of British Columbia and many others.

The author poses the following questions: What improvements to irrigation DSS graphs and front-ends generally can be made that will yield increased ease of use by untrained persons? What fields of study need to be considered to help with this task and will increased ease of use and understanding make a significant difference to DSS uptake and DSS results? How are these concepts tested?
In partial answer to those questions above, the author states that there is significant work in the field of technical information presentation for both desktop and internet front-ends by companies such as Microsoft which has so far not been used in irrigation. One example of this is the currently unreleased Windows Presentation Foundation(10), a graphical package in-built into machines running Microsoft Vista but also available to machines running .NET version 3. This package allows for 3D graphing and will potentially enable data to be displayed in novel ways. Further examples of new technology that may allow for better graphing are given in Subsection 2.5.2.2. The author also believes that there may be work in the fields such as cybernetics, cognition and psychology that may have use in this area but these are currently both outside his past experience and the scope of this literature review. Certainly no reference to these areas of study have been given in irrigation DSS literature that include graphical front-ends of which the author is aware. These areas may be investigated, at least for potential, at a later stage in the PhD. The author believes that these concepts may be testable but that it does not appear they have been in the literature so far. Answers as to how the author may intend to test the utility of these concepts will appear in this PhD’s methodology.

2.5.2.2 Web 2.0 graphical technologies

There has been a trend towards using web browsers as interfaces for DSS, even for those that do not use the internet, and some examples of DSS that do this were given in the section above. The author supposes that this is due to the ease with which HTML can be created and also the fact that all operating systems now support a web browser of some sort that displays HTML in a standard way (usually compliant with the World Wide Web Consortium’s\(^1\) standards for HTML, XML etc).

In the last 2 to 3 years there has been a drastic paradigm shift in internet technologies generally known as Web 2.0 which was originally mentioned in Section 2.3.2. One major part of this is the use of new programming procedures,
such as AJAX, to facilitate interactive display over the internet that makes web pages more like desktop applications.

Garrett writes:

“Desktop applications have a richness and responsiveness that has seemed out of reach on the Web. The same simplicity that enabled the Web’s rapid proliferation also creates a gap between the experiences [designers] can provide and the experiences users can get from a desktop application.”(20)

AJAX, which stands for Asynchronous JavaScript And XML, is not a new technology but a new procedure for using existing technologies, hence the JavaScript and XML references in the acronym. Figure 2.6 shows where the AJAX ‘engine’ sits in relation to normal internet communication between a browser and a server.

The AJAX-enabled web page loads more data into a browser than a user will need the first time they look at a web page and call extra information from the server asynchronously (i.e. not when the user clicks a button but at a time optimised for efficient data loading). The result is that an internet browser can cache information that a user may be likely to look at and, so when a user chooses to look at that information, it is displayed instantaneously, rather than a whole new web page being loaded. Google Maps uses AJAX to achieve smooth map scrolling in its map area of the screen by pre loading and streaming maps to the browser asynchronously. These technologies can be applied to any device that displays HTML and is JavaScript-compatible which includes most modern mobile phones and SmartPhones.

This could mean that an irrigator views a graph that displays only some of the information available, such as time axes with just rain events, and then click check boxes to layer on new data series, such as ETo, irrigations etc. Another possibility is that an irrigator may ‘drag and drop’ items onto an interactive graph in event-time (the timing of user-driven events that interact with the webpage, such as mouse clicks). AJAX can also be used to automatically update data on
Figure 2.6: The traditional model for web applications (left) compared to the Ajax model (right). *(Image from (20))"
a web page, rather than waiting for a whole page to refresh. In this way graphs and counters may change and increase, in real-time, while a user looks at a page.

These properties of Web 2.0 interfaces, although they have been known and used in the desktop interface environment previously are revolutionary in terms of the possibilities that they offer category 5 and above DSS. What value can be given to DSS front-ends as a result?

2.5.3 Past and Current interface media

The front end subsystems of DSS have traditionally been restricted to a small number of media. These have primarily been desktop computers and local network devices, such as printers. Past and current DSS, category 1 - 5, were all presented with the decision maker interfacing with a 'personal computer' in Section 2.3. This 'personal computer' interface could be seen by the viewer to be a:

• standard desktop program - e.g. data viewed and graphed in Microsoft’s Excel
• custom desktop program - those provided by specialist software vendors like Kaidara or MindBox mentioned in Subsection 2.4.2.2
• custom sensor interface programs - those provided by equipment manufacturers, such as Sentek or EnviroScan
• web browser-based interface - website and website applications, such as Wateright(17)

The author believes this covers almost all past and current types of DSS interfaces with a few notable exceptions, one of these is presented in (5) where a cockpit visualisation system is used to guide pilots in an aircraft carrier landing. This is a good example of a highly customised DSS front end that was built for a single application.
Another exception within the irrigation sector that has been built using standard programming development platforms is that given by (31). This irrigation DSS is a:

kinematic wave simulation model capable of simulating the surface irrigation event in real time, in the field, while the irrigation is occurring. This portable system consists of two key components; a Pocket PC or SmartPhone device capable of running Windows Mobile and a GPS which will interface to these devices.

In a separate vein to the device mentioned above and one that uses GPS connectivity, are those quasi-DSS systems now being use for navigation in motorcars. While these devices may not be considered by some people as an interactive DSS in that they just present information to a user, they can make some predictions about where/how to travel and the author believes that their uptake may foster other, vehicular-based, DSS in the near future. These devices have become accessible to a great range of people in the last year or two (2005, 2006) due to drastic drops in price. An example of such a system is TomTom GO 910 for just over $AU1000.

Email has often been used with desktop and internet-based programs as a complimentary form of UI. Due to the integration of emails into the internet, the ease with which emails can be sent from both desktop programs and internet websites and the fact that they are currently free means their use is very common. Internet server administration DSS, used to monitor server, ‘health’ can send emails to server administration personnel if events, such as ‘disk full’ happen or are likely to happen. The author has not found DSS that rely solely on emails.

In addition to email, other messaging services, such as the widely used SMS are gaining in popularity with new, wide-ranging, SMS services appearing to spring up overnight at the time of writing. The author has found businesses using SMS to dispatch field work jobs or collect information from field workers and import it directly into business databases, (36), SMS used to deliver predicted tram times at various stops on Melbourne’s tram networks etc but the author has not seen any complex DSS use SMS or related services yet.
The author believes that there is the potential for a DSS to use just email or just SMS for some form of decision support, such as an drip irrigation SMS service that uses SMS only to communicate with an irrigator. This service could be designed for stand-alone operation, it could be integrated into a multiple interface DSS as one of its use cases (see next section).

Still regarding SMS, (57) describe huge water efficiency gains that can be made by farmers adapting scientific irrigation scheduling methods based on evapotranspiration (ET), the directives for irrigation from which the receive via the mobile phone-based short messaging service (SMS). This author feels that the tools detailed in these papers, as excellent as they may be, are unlikely to see large-scale adoption by irrigators due to the reasons given above. Despite this paper describing interfaces for complex models, the workings of which are hidden from the end users, there are a series of inputs that are required from the irrigators as well as the need for participating irrigators to attend several workshops to enable them to understand the tool.

There appears to be few instances of DSS that have multiple front-ends in the literature. The author could not find papers that reported DSS with multiple front-ends of different types, other than desktop and internet browser-based front ends or a single mobile computing front end for a single type (make) of device.

2.5.4 Mobile device front-ends

(3) states that, as of November 2005, there were at least a billion mobile phones in the world, therefore they are “a near constant companion of a significant percentage of the world’s population”. The number of phones would have grown considerably, by anybody’s estimate, by the time of this writing. In addition to phone numbers, mobile computing capacity, mobile data storage and mobile network bandwidths have increased dramatically to the point that mobile devices can now connect to the internet and other networks with reasonable speed to allow data transfer in useful time frames and utilise their processing power to run DSS back-end systems of great complexity. For these reasons as well as the fact that
irrigators are often 'out of the office' and 'in the field', the author believes that DSS and quasi-DSS made available on mobile devices have great potential.

As mentioned above, past DSS have been restricted to using only one front-end delivery platform due to the difficulty of designing multiple effective front-ends for a single DSS system. There are many organisations that have DSS or DSS-like systems, such as postage tracking (Australia Post gives some of it’s delivery staff personal digital assistants (PDAs)), order processing, fault reporting etc, that operate on perhaps desktop computers (in any or all of the ways detailed above) and a single mobile platform. In this case, an organisation might purchase a single brand of PDA or SmartPhone for some of their employees and then have a DSS-like program front-end or a stand-alone program (meaning the entire DSS, including front-end) written specifically for that device.

The term 'SmartPhone' is gaining frequent use. (72) defines a SmartPhone as a device that:

“combines the functions of a cellular phone and a handheld computer in a single device. It differs from a normal phone in that it has an operating system and local storage, so users can add and store information, send and receive email, and install programs to the phone as they could with a PDA. A smartphone gives users the best of both worlds—it has much the same capabilities as a hand-held computer and the communications ability of a mobile phone.”

Questions for DSS front-ends that may appear on mobile devices could be:

1. how easily can integrated development environments (IDE) create front-ends for mobile devices with different screen characteristics?

2. how should DSS architecture utilise the computational capacity of mobile devices?

3. how effective is network coverage in the Australian irrigation districts?

4. given the hardware and software tools that SmartPhones possess, what possibility is there for the realisation of user-defined data sets (UDDS) via SmartPhones?
5. where are the bottlenecks in regard to mobile device capacity generally and also specifically in relation to interfaces?

6. will irrigators be interested in using DSS on mobile devices?

Regarding the first question; recently Integrated Development Environments (IDEs) have been created that have simplified the task of software deployment in general, including software for front-end subsystems. There are a range of IDEs with some being open source and free, such as Eclipse\(^1\), and others being proprietary and quite expensive, such as Microsoft’s Visual Studio. These IDEs present a developer with a vast range of tools, sample code, pre-packaged code bundles and testing environments that vastly increase the pace and simplicity with which software, including interfaces for different, devices may be written.

The author feels it is useful to mention three aspects of software that are integral to IDE usage and DSS front-end diversification. These are 1. the standardisation of device operating systems, 2. the advent of ‘Virtual Machine’-based software languages, such as used in the Eclipse IDE and 3. multiple language support used in both Eclipse and Visual Studio. With regards to the first point, the standardisation of device operating systems, many small computing devices are now sufficiently advanced to warrant operating systems. Operating systems (OS) are software programs that control the interactions of programs with other programs and hardware. The average computer user is familiar with the Windows range of OS from Microsoft. Standardisation of OS eases the deployment of software across multiple platforms. An example: Microsoft have released Windows Mobile, a cut-down version of the familiar Windows OS for mobile devices. Programs can be written for standard Windows machines and then ‘ported’ (translated and transported) to Windows Mobile devices easily. The second point, that concerning ‘Virtual Machines’, is related to OS standardisation in that ‘Virtual Machines’ (VM) are software programs that can connect to a range of OS and act as another level of standard OS enabling similar functionality to that which can be achieved with OS standardisation. Multiple language support is a feature of most modern IDEs (those created since about 2001) that allows them to either create programs in multiple languages or translate code from one language.

\(^1\)http://www.eclipse.org/
to another. Eclipse allows the use of C/C++, Python, Perl, Ruby, Fortran and others and Visual Studio allows the use of many imitation languages that are very similar to commonly used languages. This allows partial programs or program bundles to be reused on different devices with only the device-specific parts needing to be rewritten.

Regarding the second question “how should DSS architecture utilise the computational capacity of mobile devices?”; DSS can utilise mobile devices in a number of ways, for example the mobile device can be thought of as a ‘thin client’ where all computation is done by DSS server and other network devices and minimal information is then sent from and to the mobile device. This method would be most appropriate for situations where a large amount of data processing would need to be carried out (such as CBR prediction calculations and data fusion). A ‘thick client’, where much computation is done by the mobile device, would be another operational mode. A stand-alone mobile DSS would be the logical extent of ‘thick client’ modes. The question of which mode to choose for operation can probably only be answered for specific applications requiring research to uncover them.

Regarding the 3rd question “how effective is network coverage in the Australian irrigation districts?”: the simple answer is that in Australia, for areas with recoverable operations, normally associated with high value horticulture such as citrus and grapes, coverage is almost universal.

Regarding the 4th question “given the hardware and software tools that SmartPhones possess, what possibility is there for the realisation of user-defined data sets (UDDS) via SmartPhones?”: firstly “By the last quarter of 2004, about 75 percent of mobile phones in Japan were camera phones”(3). Could cameras be used to take in-field images that are calibrated in some way and then fed back into a centralised, or at least non-mobile device-based DSS? The author believes that the discussion about UDDS and mobile devices will be a large and ever expanding one. Significant work is needed here as the literature is scarce for although specialised people, such as scientists and consultants, are known to use
mobile computers in a number of ways, the author cannot find any papers that explore the use of mobile devices, by non-specialised staff in any of the following areas:

- how mobile devices can be used by irrigators to collect in-field data from field sensors and other equipment
- to what extent location information, collected by GPS or other locating systems attached to mobile devices can be used as a real-time UDDS
- how current or near future hardware add-ons to standard devices, such as cameras and probes, can be used to collect in-field data which could be termed a UDDS

Regarding the 5th question “where are the bottlenecks in regard to mobile device capacity generally and also specifically in relation to interfaces?”: the following quote gives a commonly-held view of the situation:

“The major limitations of m-commerce [mobile commerce], as viewed today, are small screens on wireless devices, limited processing power, modest memory, restricted power consumption, poor voice quality, low-speed data transmission, non-ubiquitous coverage, unproven security, scarce bandwidth, and possible health hazards. In view of the fact that mobile computing is accelerating at a rate faster than Moores law, and according to Edholms law of bandwidth, wireless transmission rates also follow Moores law, many of these limitations are expected to diminish, if not being eliminated, over time.”(23)

In relation to mobile device computational power: restrictions placed on potential DSS operations, such as performance modelling, calculating, data manipulations and visualisation output, due to the mobile devices’ processing power and memory are unlikely to be the major factor affecting overall performance and uptake. The author feels the chip data processing power of devices such as the i-mate JasJam, which is one of the standard SmartPhones for use with Telstra’s
NextG network mentioned above, is enough to ensure that it can perform beyond the capabilities of other parts of the total system.

In relation to modest memory, the author believes that this will also not constitute the bottleneck in total system performance. The reasoning is that much of the data used in DSS is textual or binary and of relatively small size compared to the data currently in use in the entertainment industry such as sound and video files. For example the i-mate JasJam can be fitted with a 2 GB memory chip for very low cost (AU$50) to indicate the cost and size of storage memory currently available.

In relation to low-speed data transmission, non-ubiquitous coverage, the author believes that 3G networks in the Murrumbidgee irrigation area are under utilised in terms of scientific data transfer due to the same reason as given in the last point: the entertainment industry, the driver of 3G technology, requires far more bandwidth for video and audio than is required for much scientific data.

The author’s overall opinion on mobile device bottlenecks is that they are to be found in the size of the display (screens) and their ease of use. Non-interface hardware are not the limiting factor. The author believes that, based on the industry’s rapid development, it is reasonable to assume that the physical requirements of mobile devices, that they fit the pocket, will ensure a maximum screen size of something in the order of 1200 x 600 mm in perpetuity, or at least for long enough to be of importance to this research.

Regarding the 6th question “will irrigators be interested in using DSS on mobile devices?”: with such strong support for mobile devices in metropolitan areas and amongst younger generations as well as the sheer scale of infrastructure and device investment and development, the author believes that widespread mobile device usage in Australia will ubiquitous in the coming half decade with certainty. There may be initial resistance to the technologies amongst older growers but this can surely do nothing other than lessen as the CDMA network yields to a 3G network (NextG). For Australia, the existing rural coverage CDMA mobile

---

1the author is interested in 3D holographic displays that may come to be found on mobile devices
network will be replaced with a 3G network (called ‘NextG’ by the provider, Telstra Corporation). The CDMA network is already widely used by irrigators in Australia for voice calls and basic services such as SMS sending. The speed of this network is considerably faster than that of the previous CDMA network and all devices that are sold to work on this network are internet enabled. The base mobile phones that can be purchased for use with the new network are verging on, or are, SmartPhones. Anecdotally the author believes that practically all the irrigators in the Griffith, NSW, irrigation area formerly used CDMA phones and that they will or have already changed to NextG. This suggests that Smartphone capabilities will soon be universal in this area and others and that the services will be up taken by irrigators, it is only a matter of time. As a final note in this area, new mobile phone/internet interfacing has allowed for the linking of SMS text to web addresses that allow users to simply click on an SMS’s text and have their phone’s inbuilt web browser automatically navigate to a web page(7). This may allow for rich content to be sent to a mobile user in a way that is as easy to access as the already well used SMS.

Regardless of the answers to the above question, the author believes that SmartPhones need to be considered in future DSS front ends or even as platforms for complete DSS.

2.5.5 Front-ends for different use cases

Some DSS have more than one type of standard user (the Wateright website(17) allows users to choose many different functions that are quite separate. These different functions could be seen as different use cases) but most DSS consider only a single user group and most DSS provide for only one ‘mode’ of decision maker usage. For example, the recent weather forecast-based DSS described in (42) has only irrigation advisors as use cases and the irrigation advisors only use the DSS via a web browser on computers connected to the internet.

The author believes that, due to the fact that the set of potential irrigation DSS users is not primarily comprised of employees that can be instructed directly
in how to use a DSS, as would be the case of employees in a company or government department, and that as the computing skills level of irrigators can be assumed to vary enormously, one or a few use cases may not be sufficient to see wide DSS usage in situations other than those in which it is mandated\(^1\) This point was initially addressed in Subsection 2.3.2 and was accompanied with Figure ?? showing multiple user interface use cases.

Due to rich web content, sites such as Wateright, mentioned above, provide both generic information for irrigators as well as providing specific different service options. The author believes that it useful to consider all the modes of information presentation as ‘use cases’. This is a standard technique in software engineering and allows the DSS designer to model and enhance the particular experiences of the system experienced by different types of system user. The provision of generic information by a DSS can be considered to be a ‘broadcast interface’ use case. This most basic of DSS functionality sees the DSS simply display information, potentially using different UI platforms, such as web browser, SMS etc and does not allow for user interaction. The author feels it appropriate to classify users who input information into the system and eliciting specific responses from the system on an ad-hoc basis and for who are not uniquely identified by the system in that the system retains no memory of their previous interactions with it or other data of that user such as location etc, as ‘anonymous users’. There may be multiple distinct anonymous user use case based on different on-off services that are offered by the DSS. Further use cases could be considered ‘account users’. These would be users who log in to the DSS in some way. This may be done seamlessly without requiring a user’s direct action by recognising a user’s mobile device number, IP address or other device ID as opposed to the standard log in screen. These ‘account users’ could also potentially access a single DSS, or aspects of a single DSS, with different interface devices and mediums so that a single irrigator could both log in to a DSS’s web page front end and interact with the DSS via SMS.

The author has not found papers that explicitly detail distinct DSS use cases to even the minimal detail given above but recent conversations with commercial

\(^1\) water ordering over the internet is becoming more commonplace\(^{58}\) in Australia and many soon be mandatory in some areas. This may force irrigators to increase their computing skill.
CBR software vendors have highlighted the fact that there may be similarities between CBR software, including front-end subsystems, in the medical and irrigation sectors.

2.5.6 Front-end decision making process guidance

DSS front-ends in the literature that facilitate users making a single decision have mostly assumed a stepwise, serial decision making process (termed a 'linear' process in decision theory(70)) in that they have desktop screens or web pages that require a user to progress linearly through in order to gain information that leads to the utilisation of decision support. As noted in Subsection 2.2, humans do not naturally make decisions in a linear manner(70). The author enquires believes that little work has explicitly outlines DSS front-ends that facilitate non-linear decision support. A DSS that had a front-end that allowed information to be entered, provisionally results viewed, then updated information revised and re-entered would offer some such support. (17) does offer some support for non purely stepwise decision processes with predicted results generated after initial values are entered that can be used to revise a user’s inputs. Graphical Bayesian Networks (BN) can offer a similar form of aid for individual decision support in that inputs to it can be initially entered, potential results viewed and then revised inputs changed with the addition of system learning through Bayesian statistics(47). Results for systems using BNs, if they are computer based, such as Microsoft’s graphical BN(11) can be viewed in event-time. This could allow an irrigator to enter in values, such as the time and amount of irrigation events, and have the DSS display resulting soil moisture from which the irrigator could revise their inputs and run the DSS model again to view updated outputs. (53) describes a BN for decision analysis in the irrigation industry but the system is not really designed to be used directly by decision makers - it’s results need interpreting and it is not a real-time or event-time system. The model is graphically quite complex in that is has many parameters and events modelled and would not be easily understood by untrained users.

(17) does not extend support for non purely stepwise decisions beyond individual decisions and therefore do not allow an irrigator to establish a season-long
irrigation schedule that could then be updated and revised with time.

Subsection 2.5.2.2 mentioned the use of Web 2.0 technologies that would allow the use of web pages in a manner similar to that of desktop applications by allowing for instant updating, 'drag-and-drop' functionality and smooth user interactions. Potentially work that is done in terms of non-linear decision support for desktop applications could also be used on both web browser and mobile device front ends.

### 2.5.7 Section Summary

- There may be value in research looking at optimising users’ cognitive understanding of graphs.

- The author believes that improvements to irrigation DSS graphs and front-ends generally can be made that will yield increased ease of use by untrained persons.

- Multiple fields of study need to be considered when researching how to best allow untrained persons to use irrigation DSS.

- Increased ease of use and understanding will potentially make a significant difference to DSS uptake and DSS usage results.

- Web 2.0 technologies for front-ends are revolutionary in terms of the possibilities that they offer category 5 and above DSS for real-time and event-time interaction.

- Traditional DSS front-ends delivery media have been limited due to the cost and difficulty of creating cross-platform and cross-device interfaces but modern programming frameworks and environments, IDEs, are allowing for greater ease in multiple, different front-end creation.

- There may be the possibility for DSS to use just SMS, just email or combinations of SMS, and email with other, more commonly seen, interfaces.

- SmartPhones need to be considered in future DSS front ends or even as platforms for complete DSS.
2.5 DSS Front-end systems

- Few DSS are described that cater for more than one use case and design principles based on for catering for a range of potential DSS use cases are not apparent in the literature.

- Non-stepwise (non-linear) decision making processes have not been well catered for in past DSS. Graphical Bayesian Networks may be used to help in this regard along with Web 2.0 technologies but these have not been used for direct use by decision makers.
The adoption of scientific irrigation scheduling tools in Australia is relatively low (46) despite the relatively advanced scientific state of Australian farming. At least some of the barriers to the increased use of scientific irrigation scheduling tools are related to perceived notions of the use of such tools being overly complex or requiring too much of a learning curve to begin with, for example (34) is a sophisticated web interface for an APSIM-based sugar cane model to assist with irrigation scheduling which allows irrigators to use the APSIM-based model by simply entering in certain variables into web pages and receiving graphical and textual outputs. The utility of such a system, in terms of water use efficiency (WUE) is proved in a number of papers(35; 66).

A review of DSS and other software built for irrigation in Australia was undertaken to examine aspects of the tools and their projects such as uptake. The majority of the tools listed here were taken directly from a technical report entitled “Inventory of Australian Software Tools for On Farm water Management”(33). Tools more recent than those listed in the report but considered here were:

- APSIM\(^1\)
- FIDO v.2\(^2\)
- Pocket SIRMOD(30)
- WRON\(^3\)

Aspects of the tools considered were derived from findings in the previous four sections and were:

- whether or not the software caters specifically for irrigators, as opposed to consultants or researchers
- whether or not the software caters for different use cases (or more generally, different levels of use)

\(^1\)http://www.apsim.info/
\(^2\)http://www.ncea.org.au/Irrigation/FIDO/Project_FIDO.htm
\(^3\)http://wron.net.au/
• whether or not it had a basic, cut down version
• whether or not the software caters for benchmarking
• whether or not it is currently still in use
• whether or not it used Case-Based Reasoning (no tools were found to use CBR therefore this point is not tabulated)

Table 2.3 and 2.4 list the DSS investigated and the four assessment points. Results from the review are as follows:

• there are many software packages that cater specifically for irrigators and irrigation scheduling
• no software caters for multiple use cases
• only the WRON ‘introductory’ software caters for very basic use that does not require some training or some sort of understanding process
• only one package, the ‘Probe for Windows’, tabled as ‘Probe’, reports to have benchmarking applications currently under way
• many software packages ran for only a few years or seasons. Few had long term use as a goal and even these numbered their users in the hundreds at most

The impression one receives from the results reported above is that there are a plethora of irrigation DSS and modelling software available in Australia but few

---

1 the WRON’s ‘widgets’ showing dam height have not been updated for 6 months and are really only demonstrations of applications, rather than working DSS
2 The benchmark results from consortia of growers in New Zealand using the Probe for Windows are apparently posted onto the internet at http://www.irrigationscheduling.co.nz/ although the author could not gain access to the site to see whether it was currently running or not. This was also the only example of a tool, currently in use, that reports water usage from the farm level to the district level. The WRON claims to support multiple level (district and total river system) water use in the future but does not claim to support farm level to district level reporting.
## 2.6 Australian Irrigation DSS

Table 2.3: Australian Irrigation DSS

<table>
<thead>
<tr>
<th>Name</th>
<th>for irrig.</th>
<th>use cases</th>
<th>basic</th>
<th>bench</th>
<th>in use</th>
</tr>
</thead>
<tbody>
<tr>
<td>APSIM</td>
<td>y</td>
<td>n</td>
<td>n</td>
<td>n</td>
<td>y</td>
</tr>
<tr>
<td>DamEas$y</td>
<td>y</td>
<td>n</td>
<td>n</td>
<td>n</td>
<td>n</td>
</tr>
<tr>
<td>FIDO v.2</td>
<td>y</td>
<td>n</td>
<td>n</td>
<td>n</td>
<td>n</td>
</tr>
<tr>
<td>FlowCast</td>
<td>n</td>
<td>n</td>
<td>n</td>
<td>n</td>
<td>n</td>
</tr>
<tr>
<td>HydroLOGIC</td>
<td>y</td>
<td>n</td>
<td>n</td>
<td>n</td>
<td>&lt;240</td>
</tr>
<tr>
<td>IRES</td>
<td>y</td>
<td>n</td>
<td>n</td>
<td>n</td>
<td>trial</td>
</tr>
<tr>
<td>IrriMAX</td>
<td>n</td>
<td>n</td>
<td>n/a</td>
<td>n</td>
<td>y</td>
</tr>
<tr>
<td>Magpie</td>
<td>n</td>
<td>n</td>
<td>n/a</td>
<td>n</td>
<td>&lt;350</td>
</tr>
<tr>
<td>MaizeMan</td>
<td>y</td>
<td>n</td>
<td>n</td>
<td>n</td>
<td>v.small</td>
</tr>
<tr>
<td>Micromet</td>
<td>n</td>
<td>n</td>
<td>n</td>
<td>n</td>
<td>some</td>
</tr>
<tr>
<td>Multi-Log</td>
<td>y</td>
<td>n</td>
<td>n</td>
<td>n</td>
<td>y</td>
</tr>
<tr>
<td>Probe</td>
<td>y</td>
<td>n</td>
<td>n</td>
<td>y</td>
<td>y</td>
</tr>
</tbody>
</table>
### Table 2.4: Australian Irrigation DSS cont.

<table>
<thead>
<tr>
<th>Name</th>
<th>for irrig.</th>
<th>use cases</th>
<th>basic</th>
<th>bench</th>
<th>in use</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pocket SIRMOD</td>
<td>y</td>
<td>n</td>
<td>n</td>
<td>n</td>
<td>n</td>
</tr>
<tr>
<td>SWAGMAN</td>
<td>n</td>
<td>n</td>
<td>n</td>
<td>n</td>
<td>small</td>
</tr>
<tr>
<td>VineLOGIC</td>
<td>y</td>
<td>n</td>
<td>n</td>
<td>n</td>
<td>100</td>
</tr>
<tr>
<td>WaterBalance</td>
<td>y</td>
<td>n</td>
<td>n</td>
<td>n</td>
<td>n</td>
</tr>
<tr>
<td>WaterSense</td>
<td>y</td>
<td>n</td>
<td>n</td>
<td>n</td>
<td>y</td>
</tr>
<tr>
<td>WaterSupply</td>
<td>n</td>
<td>n</td>
<td>n</td>
<td>n</td>
<td>small</td>
</tr>
<tr>
<td>WaterTrack</td>
<td>y</td>
<td>n</td>
<td>n</td>
<td>n</td>
<td>y</td>
</tr>
<tr>
<td>WeatherFare</td>
<td>y</td>
<td>n</td>
<td>n</td>
<td>n</td>
<td>n</td>
</tr>
<tr>
<td>WRON</td>
<td>n</td>
<td>n</td>
<td>y</td>
<td>n</td>
<td>n</td>
</tr>
</tbody>
</table>
are for irrigators to use for scheduling in an other than research fashion. None of the packages really allow for a spectrum of usage levels, apart from those that just present information. In addition to this, none report multiple front end delivery formats; all are either desktop or web browser based with just one example of a PocketPC backed application.

2.6.1 Section Summary

- There are many irrigation DSS and modelling software packages in Australia.

- None of the irrigation DSS and modelling software packages in Australia cater for multiple use cases, basic access and benchmarking.

- Many of the packages have short lives and were not expected and did not achieve use in numbers greater than a few hundred.

2.7 Conclusions

This section draws on the ‘Section Summary’ in each of the four previous sections and determines which of the points raised in those sections are both worthy of research in that they will lead to new scientific knowledge and are practical for the author to research within the confines of a PhD. Also presented here are further investigations into some of these points.

2.7.1 Decision Theory conclusions

With regard to Section 2.2: Single-themed aspects of decision outcomes, such as water use efficiency in the irrigation context should not be solely used to assess the utility of decisions made, the best outcome for a decision maker may not be the best scientific outcome and DSS designers must make all efforts to avoid DSS users encountering decision theory paradoxes by presenting information to them as if they were ‘on the side’ of the decision makers.
The final point is that there may be great value in presenting users with 'facilitative' DSS, rather than 'directive' DSS.

From reviewing these points, the author believes they may be bundled into principles for DSS design and that a DSS that allows users to designate their desired outcomes could be the mechanism to make effective use of those principles. To achieve a DSS that allows for this level of user interaction and definition a new paradigm in DSS design must be achieved that is akin to an artificial intelligence system that 'learns' from its user. Research in this area may be undertaken by the development of a 6th generation of DSS that allows for much user interaction and also includes a Rule-Based Reasoning (RBR) engine that allows a user to define their own rules or a Case-Based Reasoning (CBR) engine that can either learn from past cases, allow users to enter dummy, ideal, cases or both. This is entirely possible within the scope of this research and the author considers that research into this area could yield both new science in terms of how reasoning engines can learn or interact with un-trained personnel and in terms of how machine reasoning in the agricultural sector could capture so-far unquantified or parameterised rules that irrigators currently use for irrigation decisions.

The final point from the decision theory section, that there may be value in providing 'facilitative' as well as 'directive' decision support can be partially addressed by a combination of current broadcast agricultural information that contains no direct advice, such as weather bulletins that irrigators receive (these can be seen as 'facilitative' to some extent) and 'directive' DSS such as WaterSense\(^\text{(34)}\).

It has not however been addressed in direct studies where some irrigators are given facilitative information and others are given directives, based on the same information. The author believes that there would be great worth in undertaking such a trial. A potential trial, entirely possible within the scope of a PhD, could send local ETo information to one set of irrigators with drip irrigated vines and calculated dripper times, based on the same local ETo information to another set of irrigators with similar crop set ups. This would lead to new knowledge about the differences in perceived maximally efficient irrigation decisions where 'maximally efficient' is measured only scientifically and actual irrigation decisions where other factors influence the decisions being made but where the scientific knowledge that could be used to calculate the 'maximally efficient' outcome is
known to the irrigator. It could also lead to new knowledge about how well received different forms of decision support is by irrigators and even new knowledge about irrigators understanding of the concept of ET.

2.7.2 Decision Support Systems conclusions

With regard to Section 2.3: five major points were noted for further examination. The first was that there exists few DSS with multiple interface media and that recent technologies could simplify the process of generating multiple interface media for a DSS with a single back-end subsystem. Research into the development and deployment of multiple interface media would certainly be novel for DSS in the irrigation sector and probably novel for DSS work generally if it encompassed a substantial range of media. Research into effectively using subsets of information from a single DSS back-end subsystem on computationally-limited or display-limited media (such as textual information sent via SMS when spatial values are calculated by the DSS) would present a developer with design challenges but would also require new science in relation to accuracy and utility of the information subsets. A comprehensive range of interface media could be used in conjunction with 'facilitative' versus 'directive' DSS testing.

The second point from Section 2.3 is that there are no examples of DSS for irrigation that expressly cater for a range of different user experiences or interaction levels, known as use cases. Research into a DSS that does this the author believes to be fundamental if irrigators with their broad range of experiences and skills are to effectively use DSS. New science can be created in classifying potential use cases for irrigation DSS and that the deployment of multiple interface media may be able to be used in generating a DSS that effectively caters for different use cases. The author believes that research into different use cases should be done while developing new DSS and that regarding his research environment amongst both agricultural researchers and many irrigators that it is entirely possible.

The next two summary points from Section 2.3, that regarding the ability of a DSS user to choose datasources and that regarding the ability of a DSS user to create a user-defined dataset to feed back into a DSS are both worthy of investigation with the first of these two points being much simpler than the second
to investigate (more discussion about the feasibility of potential work relation to these points is given below).

The final point, that lessons from decision theory must be learned was discussed with regard to Section 2.2 above.

2.7.3 DSS Back-end systems conclusions

With regard to Section 2.4: it was noted that TCP/IP connections are becoming all pervasive. There is, therefore, little value in research into other forms of networking for DSS back end systems within the context of irrigation decision support and informatics research. There may be value in considering data sources, such as older sensors that do not currently communicate over TCP/IP but could do so and future datasets that could possibly communicate via TCP/IP but that work would not constitute new science within the context of irrigation decision making or informatics for a PhD but rather product engineering. Similarly there is little value in research into data storage.

Research into the extent of available and potential future data sources for irrigation decision support could be undertaken informally along with the development of future DSS but would not constitute a separate study.

An instance of a potential future datasource, that of 3rd party calendars, was noted. Research into multiple future data source particulars is outside the scope of research into DSS although investigations into a limited number of potential future datasources, such as 3rd party calendars, would be very useful (this may be an option for research once other more immediate challenges are addressed) and could lead to generic design principles that allow for the inclusion of other future data sources.

Investigations into the use of XML will not yield new science but will have to be undertaken at least to some extent by any future DSS designer due to the ubiquitous use of XML as a data transfer format on the internet. XML will need to be understood quite well to effectively be used as a research tool.

Research into data fusion is likely to be technically difficult and unlikely to yield new science in relation to irrigation decision support. It is likely that the
principles of data fusion and the theory relating to data fusion levels may be of
type in systems with multiple user interface media or systems with multiple
data source usage such as future DSS, therefore investigations may be undertaken
here during the process of future DSS design but it should not be undertaken for
its own sake.

Research into RBR that encapsulates rules beyond those of simple water balance
as preferred by technicians and researchers could lead to the generation of
a system that emulates the entirety of irrigator decision making. This would be
a quantification of work such as (46) that was undertaken to understand the to-
tal arena of irrigation decision making, not just the biophysical plant-water-soil
system. If achieved, this could lead to greatly enhanced decision support. It may
also lead to the generation of knowledge as to how irrigators’ irrigation practices
differ from the water balance ‘optimal’ practice through the analysis of other
irrigator-defined rules. Fundamental to this would be the ability for irrigators to
create their own rules for a RBR rule base.

Research into CBR for irrigation decision support has the potential to achieve
new goals compared to research into RBR. These new goals are the generation of
a DSS that uses irrigation events, their intended and actual outcomes to provide
references for decision support and the ability for it to ‘learn’ better methods of
decision support as more cases are added to the case base. This in itself would
be new science but it would also lead to a plethora of further novel investigations
based on the case base, some of which are surely not yet perceivable. The task of
establishing parameters for cases in a case base system from individual irrigation
events is probably not an easy task but the author feels that it may be of such
merit that much effort would not be wasted were it directed into this area.

Research into how CBR is deployed in sectors such as the medical and legal
sectors may help with this task and has not been undertaken so far as evidenced
by the lack of published work on any aspect of CBR for irrigation. This research
would not lead directly to new science but it may allow those who undertake
such investigations to then generate novel CBR systems for irrigation and for
this reason it should be pursued.
2.7.4 DSS Front-end systems conclusions

With regard to Section 2.5: there appears to be value in research looking at optimising users’ cognitive understanding of graphs and other information delivery tools. Cognitive research of this kind has not permeated the world of irrigation decision support so that currently the benefits that it may offer are not known. Existing irrigation DSS display very complex, multi variable, graphs to users and work in relation to streamlining the delivery of that information has not been undertaken despite the widely held belief among researchers that one of the reasons why irrigation DSS are not seeing much usage is that irrigators simply do not understand the information they present. There would be great scientific value in researching how large a contribution to irrigator understanding cognitive optimisation of graphs could make and this may also improve uptake.

It was noted that in relation to the above point and that of generally designing easier to use DSS front-ends was a multi disciplinary field involving studies from agricultural science, IT, engineering, ergonomics and psychology. It may not be possible to undertake either the diverse training required or collaboration needed to utilise all of those fields within the context of the author’s PhD. However, some institutions specialise in such multidisciplinary work for DSS, such as the School of Informatics at the University of Edinburgh\(^1\). Potentially visiting such an organisation should be considered in the course of the PhD.

Web 2.0 technologies were noted to offer possibilities for DSS front-ends. Research into the technical aspects of Web 2.0 technologies is outside the scope of irrigation decision support however applications derived from such technologies could present opportunities, such as the ability of a system to allow a user to define DSS rules, data source selection and even data sets, that are themselves novel and new to science both in terms of irrigation decision support and of DSS generally.

Three related points were that DSS have been limited in their range of front-end delivery media but that recent technologies may remove some of this limitation, there may be the possibility for DSS to use just SMS or email for interfacing

\(^1\)[http://www.inf.ed.ac.uk/](http://www.inf.ed.ac.uk/)
and that SmartPhones need to be considered for DSS front ends. The SMS delivery of information is quite easy to implement using application development frameworks and the implementation of graphical decision support could be undertaken via SmartPhones, also using similar frameworks. The author believes that the ability to cross test DSS delivery media and the resulting information that may be gained from examining utilisation figures on different platforms is one that should be investigated as it will lead to fundamental shifts in the way information is presented for DSS. Studies that compare different DSS delivery media are entirely possible within the context of a PhD.

The author has already undertaken DSS development that utilises just SMS as a DSS delivery media and ongoing research into this work is feasible for the author to undertake in collaboration with other researchers.

The two final points noted in the front-end section summary were: few DSS cater for more than one use case despite different use cases existing among the potential users of an irrigation DSS and that non-stepwise (non-linear) decision making has not been catered for by DSS so far but that Web 2.0 technologies may enable such catering. Regarding the first point, the need to investigate DSS that cater for different use cases has been addressed above in relation to DSS and regarding the second point the author believes that functionality that allows non-stepwise decision making should be investigated as part of an overall investigation into Web 2.0 technologies for DSS. The utility of allowing non-stepwise decision making with DSS is currently unknown due to the lack of study in the area although decision theorists have indicated that non-stepwise decision making is the norm and should be facilitated as much as possible. Research into allowing this may enable a hereto underutilised aspect of decision theory to be utilised, for the first time, in DSS.

### 2.7.5 Australian Irrigation DSS conclusions

With regard to Section 2.6: three points were noted and these were that there were many irrigation DSS packages in Australia, none catered for multiple use cases, basic access and benchmarking and most had short, non commercial lives
and those that did only numbered a few hundred users. This suggests that at least some of the problems associated with poor uptake in irrigation DSS in Australia are those associated with usability and implementation programs, rather than agricultural science knowledge.\footnote{the author cannot investigate the accuracy of the various DSS here but assumes, due to the plethora of packages that have been made, that there are measurably positive outcomes from the packages}. The corollary to this is that if increased DSS uptake in Australian irrigation is to be achieved, research is needed into the DSS presentation and implementation, rather than the workings of irrigation DSS. Further to this, as there are no examples of DSS that cater for multiple use cases, basic access and benchmarking, research into such DSS would be new to science.
2.7 Conclusions

2.7.6 Final conclusions

The key hypothesis upon which this literature review was based is that:

“Informatics science can be applied to irrigation to deliver better water use efficiency and productivity”

In addition to this, some sentences from a summary of this project are:

“software models and most importantly integrated software engines tying together other technologies ... [could] ... provide a powerful irrigation water management tool for informing and predicting irrigation behaviour and radically increasing water use efficiency from paddock to irrigation region scales”

This review of literature shows that there are large gaps in knowledge within current irrigation DSS research in Australia relating to:

1. Use of decision theory DSS
2. DSS that allow users to customise the support they receive
3. DSS back-ends that are highly flexible and adaptable to multiple data source usage
4. The use of case-based reasoning in irrigation
5. How data, other than classical biophysical soil-plant-water data, may be captured and used to assist in decision support
6. Different physical and conceptual front-ends data delivery platforms
7. DSS that cater for a range of use cases
8. How DSS may achieve wider irrigator usage
Conclusions 1,2,3,6 & 7 can be grouped into a broader point which is that current irrigation DSS do not allow users to customise the support they receive. Currently they cannot have input into DSS, either about how they receive support or about what data is fed into the system to generate that support. Improving the design of current DSS by allowing users to be able to have such input would require not only the development of tools that would allow such customisation to take place but also, if it is to be broad reaching, the development of a new DSS conceptual framework that allows for system back and front-end flexibility and much user interaction. Such a framework may also be used to address conclusion 5.

Regarding conclusion 5: to allow individuals to include different non-biophysical data in decision support would be the ultimate goal of a flexible and interactive DSS, thus addressing conclusion 5 would also address the new DSS conceptual framework requirement of conclusions 1,2,3,6 & 7. Rendering decision support using data other than classical biophysical soil-plant-water data, could partly be achieved in a piecemeal fashion with the addition of individual rules concerning specific non-biophysical data (such as factoring in water price versus the cost of not watering in terms of reduced yield) but this approach would not allow for individual customisation unless a comprehensive set of specific non-biophysical data rules were created that quantified every non-biophysical data set that one could suppose an irrigator might wish to employ in decision making. Such a set would not only be vast but may never be comprehensive as surely the non-biophysical data which irrigators include in their decision making varies with time. The author therefore believes such a task to be unachievable. What is needed is a new model of DSS support, which is fundamentally different from previous DSS models in irrigation, that it is open ended as to the data sources used and one that allows users to define those data sources.

The author believes that a combination of the conceptual framework of CBR and new IT tools, such as Web 2.0 technologies, that can be used to enhance both the front-ends and back-ends of DSS as reported in this literature review, may be used to address Conclusion 5 and in so doing also address conclusions 1,2,3,6 & 7. The fact presented in conclusion 4 and the absence of Web 2.0 technologies
in existing irrigation DSS tells us that no previous irrigation DSS designers have attempted such.

Conclusion 8 is the overall question that this author believes will be better answered by addressing conclusions 1 - 7.
2.8 Proposed Research Question

The conclusions from the previous section are to be addressed by testing the following hypothesis:

“A decision support system with new approaches to irrigator interaction and external systems connectivity, in combination with an empirical reasoning framework, can be created to enhance objective irrigation decision making.”

The author proposes objectives to be used as a basis for experiments to test the hypothesis. To test the new approaches to irrigator interaction we could:

- Determine the utility minimally interactive 'facilitative' versus 'directive' DSS
- Determine the extent to which a DSS can accept user-defined input
- Determine the extent to which a DSS can incorporate user-defined rules

To test the new approaches to external systems connectivity we could:

- Determine the extent to which XML specifications may be adopted by irrigation data source for their interoperability and 'discoverability' over the internet

To test an empirical reasoning framework we could:

- Determine whether CBR can provide a rigorous model for irrigation scheduling and provide decision support
Tools for Improving Water Use Efficiency: Irrigation Informatics implemented via SMS

Car, N.J. 1, 2, 3, Christen, E.W. 2, 3, Hornbuckle, J.W. 2, 3, Bartlett, C. 4 and Moore, G. 1

1 Department of Civil and Environmental Engineering, The University of Melbourne, Victoria
2 Griffith Research Laboratories, CSIRO Land & Water, Griffith, NSW
3 CRC for Irrigation Futures
4 Esendex Pty Ltd, Melbourne, Victoria

Email: nicholas.car@csiro.au

Keywords: Irrigation, informatics, Decision support systems, Short Messaging Service, SMS, mobile

EXTENDED ABSTRACT

Irrigation accounts for between 60 and 70% of all consumptive water use in Australia. Recent emphasis has been placed on improving the efficiency and performance of irrigation systems for improving water use productivity, therefore ensuring the best use of this limited resource. One way of achieving this is for irrigators to use objective scientific data to schedule their irrigations. In this regard Decision Support Systems (DSS) that model soil, plant and weather conditions and provide both timing and volume advice can be used. A focus of the Cooperative Research Centre for Irrigation Futures’ Irrigation Informatics project is achieving DSS use.

Many DSS have been developed to help with irrigation scheduling in Australia such as CSIRO’s WaterSense, Destiny and MaizeMan. The biophysical modelling that they use is advanced and can lead to great water use efficiency (WUE) gains, for example due to the use of WaterSense, “cane farmers in the Ord reduced their annual applications of irrigation water to sugarcane from 35 to 40 mega litres per hectare to an average of 21 mega litres per hectare without loss of sugar production” (Sugar Research and Development Corp, 2007)

Despite this, they have seen very poor uptake (Hayman 2004 and Inman-Bamber 2005). Two reasons for this are thought to be that irrigators perceive DSS as difficult to use and that computer-based DSS information is not readily available to an irrigator when it is most needed.

This paper describes the use of the Short Messaging Service (SMS), familiar to most cellular phone users, to deliver biophysical data to irrigators in a format with high end-user utility. The system addresses the two reasons thought to contribute to poor DSS uptake mentioned above by presenting it on a mobile delivery platform thereby ensuring it can be accessed where and when needed.

The system uses reference crop evapotranspiration (ET0) measurements, along with empirically determined crop coefficients, to model actual crop water use which is delivered to irrigators via SMS. All model calculations are undertaken on a remote server with inputs taken from local weather stations or satellite services thereby minimising the information required from the irrigator.

This paper presents the SMS makeup and presentation, followed by a description of the system architecture to be used for experiments in the 2007/2008 irrigation season. This is followed by the design of experiments for the 2007/2008 irrigation season to test the end-user utility of SMS given in thee parts: 1) An experiment to test DSS communication via SMS against other forms of communication, namely the internet, fax and email, 2) An experiment using a series of SMS formats to test ‘facilitative’ versus ‘directive’ modes of decision support, 3) An experiment to gauge the extent to which SMS can be used interactively between the irrigator and a DSS.

Preliminary feedback on many of the ideas presented here was collected from several irrigators and information about how the systems have been modified as a result is given.

Finally this paper suggests future SMS and related mobile computing functions, how SMS communication may be added to existing DSS to enhance their functionality, as well as how SMS fits into a new generation of informatics tools for agricultural DSS.
1. INTRODUCTION

Recent emphasis has been placed on improving the efficiency and performance of irrigation systems for improving water use productivity, therefore ensuring the best use of Australia’s limited resource that is water. Decision support systems (DSS) that model soil, plant and weather conditions can be used to calculate when an irrigator should next irrigate, based on objective assessments of crop water requirements, and it is thought that if DSS-derived irrigation schedules were followed, water savings could be achieved through efficiency gains.

Australian Bureau of Statistics data indicates that, as of 2003, only one in five irrigators undertook any form of objective decision making (Montagu et al. 2006). Additionally, although there are no direct statistics available on the usage of irrigation DSS, numbers are known to be very low with the most popular DSS, such as the APSIM-derived YieldProphet, only seeing usage in the order of a few hundreds (Inman-Bamber 2005). Knowing these figures, the authors’ assessment of the problems facing agricultural decision support in Australia is that the greatest are the poor rates of objective decision making and poor DSS uptake. Due to these problems, minimal water savings are actually realised despite evidence that the usage of objective decision making and DSS may lead to such without a loss in productivity (Sugar Research and Development Corp 2007).

Some of the reasons for the poor uptake of agricultural DSS including irrigation DSS, are related to the perceived difficulty of use and the inability of DSS to present themselves as relevant to a particular user through the deployment of personalised information (Hayman 2004).

The Short Messaging Service (SMS) is a communications service that is currently seeing rapid deployment in many sectors from field worker job scheduling to people using matchmaking ‘flirt’ services. The medium is appealing to many people due to the facts that it is both close to real-time, widely available and cheap. This availability is underpinned by large area cellular phone coverage and the SMS being supported on virtually all cellular phones. Australian cellular phone ownership is about one phone per adult person (IDC Australia 2005) and a future of increased cellular phone usage in Australia seems certain, based on the number of mobile telecommunications service providers and third party service providers using the platform.

With the roll out of rural area coverage of cellular phone services such as Telstra’s NextG and Optus’ 3G, we are able to assume that most agriculturalists, such as irrigators, have, or will soon have the ability to use the simplest cellular service, SMS.

SMS has been used to assist in irrigation decision support in South Africa (Singles 2005) but the extent to which SMS may be used interactively for agricultural/irrigation decision support has not been tested and the issue of poor uptake has not been previously addressed through the use of a mobile electronic communication medium.

2. THE SHORT MESSAGING SERVICE

The Short Messaging Service (SMS) available on 2nd and 3rd generation cellular phones allows small packets of textual information to be sent between cellular phones and base stations. Most SMS messages (SMSes) are sent from one phone to another, via a base station, however, SMSes may also be generated singly or in batches by machines other than phones and fed to cellular network base stations for distribution to phones via an SMS ‘gateway’. The SMS services described in this paper are generated in this way.

2.1. SMS presentation

An SMS consists of a sender and receiver address (the phone numbers), a sent timestamp, a delivered timestamp and a body of up to 160 ASCII text characters. This limits standard SMS functionality to the sending of a few lines of text. Other forms of SMS are mentioned in section 8, ‘Future Work’.

Figure 1 depicts an SMS, sent from the DSS described here to a standard mobile phone. This SMS is delivering suggested dripper run times, in minutes.

![Figure 1](image)
2.2. Back-end architecture

The SMS scheduling messages described in this paper travel from the authors’ server, known as the ‘irriGATEWAY’ server, to a commercial SMS gateway server, then to a mobile provider’s base station and from there on to a user’s phone.

Figure 2 shows a basic schema of the authors’ DSS architecture.

The coding for this DSS system was done entirely in Microsoft’s C#.NET language (http://msdn2.microsoft.com/en-gb/vcsharp/default.aspx) through the Microsoft Visual Studio.NET 2005 (http://msdn2.microsoft.com/en-gb/vstudio/default.aspx) desktop environment. A Web Services component which facilitates communication between the irriGATEWAY internet server and the commercial SMS gateway, provided by Esendex Pty Ltd, connects internet applications to cellular networks and uses an external interface written by Esendex. This is necessary due to the propriety nature of cellular networks that deliver SMS. A similar action is required to send facsimiles. This is not required for communication via internet methods, such as web page or email, as those media are inherently handled by internet servers.

The database for the experiments related here is a MySQL (http://www.mysql.com/) 5.0 database. Database management was carried out using the SQLYog utility (http://www.webyog.com/en/).

3. THE DECISION SUPPORT SYSTEM

Evapotranspiration values for a reference crop (ETo) can be used by crop models, in conjunction with empirically determined crop factors (Kc) to generate a particular crop’s ET value (ETc) which is then used to determine crop water requirements. Provision of daily ETo or ETc values to irrigators helps them to determine how much irrigation water to apply to their crop and is a well known method for providing decision support (Montagu et al. 2006).

If ET values specific to an irrigator’s location can be used in providing decision support to that irrigator, a measure of personalisation can be achieved. Currently there are two ways of sourcing ET values for specific locations in Australia: 1. the SILO meteorological service (http://www.bom.gov.au/silo/), which interpolates ET values from the Australian Bureau of Meteorology across all of Australia, and 2. automatic weather stations (AWS) in various locations which measure variables needed to calculate ET locally.

In its basic mode, decision support to irrigators via SMS described here consists of collecting ETo values from either SILO or AWS networks and sending them out to irrigators on a daily basis, with daily and cumulative values up to a week. In the next, more advanced mode of decision support, daily ETc values for specific crops are supplied to irrigators. The most advanced mode caters for farms using drip irrigation in which dripper run times, based on ETc values and the water delivery specifications of the drip irrigation system, are presented to an irrigator.

3.1. Models

The model used by the DSS to generate decision support values, other than the simple ETo values, is an ET/rainfall/irrigation water balance. In its non-interactive mode, the model simply sums ETc values and rainfall events (expressed negatively) to determine daily crop water requirements. In this mode, no reference is made to irrigation events as the system has no way of learning of them. The daily crop water requirement is presented to
irrigators as a cumulative millimetre value for the previous 1 to 7 days. A zero value is presented if negative to relate ‘no need for irrigation’. The system’s cumulative water balance is presented only for the last 7 days as it is the authors’ understanding that irrigators in the initial experiment locality will irrigate at daily to weekly intervals and thus reset their water balance to zero in that time.

In interactive mode, the model receives irrigation events from irrigators via SMS and calculates a continuous cumulative ET/rainfall/irrigation water balance for the whole season.

The drip irrigation system run times for the most advanced form of decision support are generated as detailed in calculations below.

3.2. Data, calculations and values

The ETo data used for the current trial (described below) are taken from SILO and are generated using the Penman-Montieth ET algorithm, as are the ETo data proposed to be used in South Australia from AWS.

The ETo data used for the trials in Griffith region, taken from AWS, are generated using the Penman-Meyer equation.

ETc is determined using $ET_c = K_e ETo$.

Dripper run times are calculated by knowing the amount of water that is required from the water balance and knowledge of the irrigation system output rate which needs to be provided by the irrigator.

The purpose of calculating dripper run times is that instead of presenting crop water requirements to an irrigator, expressed in millimetres, the DSS is able to present irrigation system run times which is how irrigation systems are operated in practice.

4. SMS EXPERIMENTS

Three experiments and one pilot trial are detailed here. The current trial (4.1) is the first implementation of ET-via-SMS and was started in March, 2007, in conjunction with University of Southern Queensland and Qld Department of Natural Resources and Water. The three experiments (4.2, 4.3 and 4.4) are scheduled for the South East Australian 2007/2008 irrigation season starting in approximately September 2007 and running until March 2008. They are to be conducted in parallel in the Murrumbidgee Irrigation Area. Initially 10 to 15 irrigators growing vines with surface drip irrigation only will be used. Irrigators will be engaged through an initial meeting arranged with the Murrumbidgee Horticulture Council during which instructions for use will be given as well as an information brochure detailing how to measure their drip system’s application capacity. Rain gauges & measuring cylinders to be used to measure local rainfall and the growers’ system capacity will also be given. Informal meeting will then be conducted with individual growers by the authors to assess their initial understanding and initial use of the system.

During the season, the authors will test the accuracy of the simple water balance model by comparing it to scheduling information generated by running the CSIRO’s comprehensive soil-water balance model WaterSense (Inman-Bamber et al. 2004). Some results from WaterSense will be presented to growers for comment.

All SMS interactions will be recorded throughout the season followed up with post season debriefings to determine whether or not growers used the system to guide irrigations. Holding post season debriefings with each irrigator, with all the irrigations, rainfall, ETc and ETo data available, will determine whether the DSS provided benefit in improving irrigation scheduling. Overall, this experimentation will reveal if irrigators find value in putting some (minimal) time into communicating with a simple water balance model.

The aim of the pilot trial is to iron out system problems before the 2007/2008 season and also to gauge the perceived utility of ETo values by a group of irrigators.

The overall aim of the three proposed experiments is to test whether a simple water balance DSS may deliver information with high utility to irrigators, utility here being defined as ‘the quality of being of practical use’. The objectives of the three experiments are:

- To determine the relative utility of delivery platforms (traditional fax, emerging email, mobile SMS and internet website).
- To determine the utility of ‘facilitative’ v. ‘directive’ decision support
- To determine the utility of a minimalist interactive, mobile, DSS

By ‘facilitative’ decision support the authors mean support that presents data to a decision maker that can be used in making a decision but does not
prescribe an action. ‘Directive’ support does not prescribe an action such as “irrigate for 3 hours on Monday”.

4.1. Current trial

The current trial operates in the basic mode, that of sending daily ETo values. It delivers them to 12 irrigators in South East Queensland with the ETo values taken from the SILO service. The irrigators can then use the values as a reference to determine relative values for irrigation events.

The system has had significant technical problems relating to ETo data access and this has retarded the assessment of the utility of the service. Qualitative feedback from the trial will be forthcoming in early November 2007 therefore not available at the time of writing but system adjustments, made on the basis of this feedback and used to continue the trial into the 2007/2008 irrigation season, will be presented at the MODSIM07 conference.

4.2. SMS v. other communication

The method used to determine the relative utility of delivery platforms will be:
1. Collect ETo values for an area
2. Send the ETo values irrigators in several formats:
   a. Facsimile
   b. Email
   c. Mobile Phone Short Messaging Service (SMS)
   d. Internet web page
3. Evaluate the irrigator’s relative receptiveness to the information formats.

This trial should reveal which of the data presentation formats the irrigators find most useful. Irrigators with a range of technical competency and communications devices will be selected but all the irrigators selected will need to be able to access at least two of the communications methods for comparison. This part of the experiment will only be run for the first month of the irrigation season (September, 2007) after which time a survey of the participants will be conducted. After that point, the participants will be moved into the other experiments.

4.3. Facilitative v. directive

The method used to determine the utility of ‘facilitative’ v. ‘directive’ decision will be:
1. Collect ETo and rainfall values for an area. Either grower or AWS supplies rain.
2. Send three different pieces of decision support information, in each of the formats used in establishing the first objective, to comparable irrigators in three different groups. The different pieces of information will be:
   a. The previous day’s ETo
   b. A calculated water balance value
   c. Dripper run times calculated from the water balance value, and the specifications of the drip system
3. Evaluate the different irrigator’s perceived utility of the three different information types

This methodology should reveal which form of decision support - from ‘facilitative’ to ‘directive’ - is preferred by irrigators. It will run for the entire irrigation season with irrigator surveys after 2 weeks, 1.5 months and the full season.

4.4. Extent of interactivity

The method used to determine the utility of a minimalist interactive, mobile, DSS will be:
1. Collect ETo and rainfall values for an area
2. Calculate a water balance for each irrigation unit using ETo, either local rainfall or rainfall notifications provided by the irrigator and notifications of irrigation events sent in by the irrigator
3. Send dripper run times, based on the water balance to irrigators
4. Evaluate the different irrigator’s perceived utility of the decision support

This methodology requires irrigators to initiate their SMS service with the irriGATEWAY server. This is done by a ‘start service’ SMS that contains the drip system’s emitter spacing, row spacing, and application rate. This is then followed by an acknowledgement SMS from the irriGATEWAY server after which confirmation SMS from the irrigator is required.

To facilitate this process, the authors have developed a brochure detailing the steps that are required to be undertaken by an irrigator to measure their drip system delivery rate. The brochure is delivered to the irrigator in a measuring vial, shown in Figure 3, which can be used for the application rate calculation.
5. FURTHER EXPERIMENTATION

It is intended that the qualitative feedback results from the irrigators will be used to improve the system for a renewed round of experiments in the 2008/2009 irrigation season. By locating the entire DSS on a server and not requiring irrigators to store and use any local software, the water balance model may be tuned to present better decision support without the irrigator’s involvement.

The same DSS base will be used for the proposed further experiments in ‘Future Work’.

6. PRELIMINARY FEEDBACK

Some feedback on the viability of the systems to be tested over the 07/08 irrigation season has already been garnered from several irrigators in the NSW Riverina who will not take part in the experiment. This feedback led to:

1. The possibility for irrigators to use either their own rainfall measurements in the interactivity experiment or those of a local weather station.

2. The ability for irrigators to run the systems here for multiple irrigation management units. This was achieved by allowing irrigators to prefix the values of their irrigation and rainfall event notifications sent to the DSS via SMS with a letter, or number, to signify the appropriate IMU. An example would be that of an irrigator irrigating their IMU labelled ‘d’ for 230 minutes sends an SMS reading: ‘i d 230’ to the DSS. A 13 mm rainfall event on IMU ‘6’ would be expressed as ‘r 6 13’. The decision support for irrigators with multiple IMUs would be presented in a similar fashion so that an irrigator with 3 IMUs, A, B and C, would receive the text in Figure 4.

3. The choice of daily SMS delivery time being by each irrigator, rather than one fixed time.

NOTE: the SMS in Figure 4 uses 154 ASCII characters, as both whitespace and linebreak characters are counted, as opposed to the visible 100 ASCII characters and is close to the maximum length for SMS.

7. CONCLUSIONS

SMS offers the DSS designer the possibility of providing simple, real-time and mobile decision support to agriculturalists. The abilities of SMS to be used as both a non-interactive and an interactive DSS platform have not before been tested and this is the focus of the systems and experiments outlined here. An initial system design and experimental regime to test some aspects of SMS use have been presented.

8. FUTURE WORK

The authors believe that there is scope for further irrigation decision support via SMS. Currently only a water balance has been used to provide decision support and no reference has been made
to advanced crop modelling. Once the utility of the basic model’s decision support is known, more comprehensive crop models may be tested. Currently it is not known whether a more comprehensive model will add value to the decision support as the utility bottleneck may exist elsewhere such as with user uptake of any form of SMS DSS.

There may be the possibility of adding SMS functionality to existing or new non-SMS-based DSS. An example that can be imagined would be the use of SMS by a platform such as the CSIRO’s WaterSense. Figure 5 shows a water balance graph mock-up, similar to those seen in programs like WaterSense, with SMS sending times indicated. In this way a DSS with a graphical interface may use SMS to alert DSS users to particular events, such as when modelled soil water levels reach a certain level. This could compliment the current DSS’s functions.

Figure 5. A Mock-up of a water balance DSS graph with SMS sending times when water levels in a soil profile reach a user defined critical level.

Future DSS delivery mechanisms based on the increased interoperability between mobile and internet applications may be envisaged.

The authors also believe there is enormous scope for the use of the SMS-related Multimedia Messaging Services (MMS). An MMS is actually an SMS with additional metadata which allows a web-enabled phone to use the MMS as a hyperlink to navigate to a web page. Such pages, if specifically designed for mobile devices, present the DSS designer with a range of possibilities, such as presenting graphs like that in Figure 5, to an irrigator in the field.

A DSS could use an MMS service to alert a user to modelled events in close to real-time, in the same way as it is suggested WaterSense might use SMS, but with the possibility of the MMS connecting the user to an internet-style DSS interface or additionally spatial information. An example of a web page formatted for mobile devices that presents the last 7 day’s ETo values for Griffith, NSW, is given at http://irrigateway.net/dev/mobile/MobPlot.aspx. This page can be viewed on both mobile and non-mobile devices and may be connected to via MMS.

9. REFERENCES

Hayman, P. (2004), Decision support systems in Australian dryland farming: A promising past, a disappointing present and uncertain future. 4th International Crop Science Congress, Brisbane, Australia, Crop Science Society of America.


Inman-Bamber, G. and S. Attard (2005), Inventory of Australian Software Tools for On Farm Water Management, CRC-IF.


Sugar Research and Development Corp. (2007), How WaterSense is winning in the west. SRDC Update, April 2007 from http://www.srdc.gov.au
Towards a new generation of Irrigation Decision Support Systems–Irrigation Informatics?

Car, N.J. 1, 2, 3, Christen, E.W. 2, 3, 4, Hornbuckle, J.W. 2, 3 and Moore, G. 1

1 Department of Civil and Environmental Engineering, The University of Melbourne, Victoria, Australia
2 Griffith Research Laboratories, CSIRO Land & Water, Griffith, NSW, Australia
3 CRC for Irrigation Futures
4 Adjunct Assoc. Prof. University of New England, Armidale, NSW, Australia
Email: nicholas.car@csiro.au

Keywords: Decision support systems, Web 2.0, irrigation, informatics, irrigation scheduling, Web services, XML, AJAX, Sensor Web Enablement

EXTENDED ABSTRACT

This paper proposes a new generation of Decision Support Systems (DSS) that leverages Web Services and Web 2.0 technologies to allow for new possibilities in the areas of irrigation decision support.

A new classification system for existing DSS, based on their ‘network paradigm’ is presented with current systems being placed in categories 1 to 5. Category 1 DSS are those with no networking abilities and are typically represented by desktop applications. Category 2 (C2) are those with direct network links to local equipment, such as sensors in a paddock. Category 3 (C3) DSS use local area networks to access data from such sources as databases and networked sensors. C4 DSS use large, proprietary and purpose-built, networks, such as SCADA networks, to collect data as well as using resources available to C3 DSS. C5, use the internet to access multiple instances of the resources available to C4 DSS. We present some examples of DSS in each of these categories for illustration.

A further category, 6, is proposed here that uses new internet software technology to extend DSS functionality into uncharted waters. Technologies such as extensible Mark-up Language (XML) and Web Services are proposed to allow DSS to provide different types of support to users at many different levels, to allow for the addition of User Defined Data Sets (UDDS) and to utilise the power of machine-to-machine communications over the internet.

We suggest how potential DSS, using some of the technologies mentioned here, may help counter the poor uptake of DSS in Australian agriculture by addressing one of its supposed root causes: that of the lack of user customisation. We propose that in addition to this, a category 6 DSS may be used in a way that no DSS has currently been used and that is in irrigation benchmarking. Further to this: we suggest how a C6 DSS used for irrigation support may present usage metrics for use by 3rd parties, such as water supply companies.

We then propose back-end architecture for a C6 DSS that utilises technologies such as XML-based Web Services, live, online databases and data fusion to bring together and interpret data from distributed providers. We relate how flexible back-end architecture may allow DSS to provide very customizable decision support and how sophisticated networking may be used to generate benchmarking data.

Next we look at how new approaches to interface design using recent ‘Web 2.0’ technologies, such as AJAX, provide the tools needed by developers to create DSS front ends that can effectively use the DSS back-ends discussed above.
INTRODUCTION

The concept of computerised DSS has been around since the late 1970’s and early attempts to classify them focussed on their architecture (Sprague 1980). Five architectural components have typically been used for classification and they are 1. the database management system, 2. the model base management system, 3. the knowledge engine, 4. the user interface and 5. the user (Gachet and Haettenschwiler 2003). Classification in this way gives no clue as to what a particular DSS does. Recently classification has been undertaken using the DSS user’s ‘assistance method’ (Power 2007) and categories in this classification are

1. Model driven, eg. use mathematical, economic or other models
2. Communication driven, eg. Microsoft’s Net Meeting designed for group decision support
3. Data-driven, eg. singular or multiple sources to a user to assist them in decision making
4. Document driven i.e. storage and processing technologies for document retrieval and analysis
5. Knowledge driven DSS. – uses expert-derived rule base or expert system reasoning, and presents ‘knowledge’ to users, rather than just data

DSS in agriculture typically present as model, data and knowledge-driven DSS.

A new classification method is presented here: classification by networking paradigm. It is more related to both classifications by ‘assistance method’ and by architecture but groups DSS by the types of information that they are able to provide to the DSS user.

DSS related to natural resource management in agriculture have been used to choose crop types, set planting dates and determine fertiliser application rates. For irrigation DSS, some provide ‘tactical’ support that typically helps the user decide when and with how much water to irrigate – an example for sugarcane is WaterSense (Inman-Bamber, Webb et al. 2006) – while others provide ‘strategic’ decision support that helps users to plan irrigation systems and make long-term irrigation decisions. An example such, used for assisting in on-farm water storage design is DAMEASY (Lisson et al., 2003). For ‘tactical’ irrigation DSS, support is often derived from meteorological data, meteorological predictions and crop modelling. The software systems and concepts proposed here relate only to ‘tactical’ irrigation DSS.

Despite many DSS having been created for use specifically in Australian agriculture, there has been very poor uptake (Hayman 2004). Users for any type of agricultural DSS rarely number more than a few hundred (Inman-Bamber and Attard 2005) and our research indicates that many do not run for more than a few ‘test’ seasons. Reasons for this are thought to relate to the idea that although DSS provide good scientific advice, the are not able, or perceived to be able, to provide ‘real world’ support (Hayman 2004). Such ‘real world’ support would be support that is based on all of the factors that decision makers use, not just scientific biophysical factors. These factors may be economic or social, or of some other nature but are certainly not viewed as trivial by the agriculturalist. Our own research in the irrigation scheduling DSS area suggests that factors such as the price of electricity to run irrigation pumps outweighs at least some of the benefits of ‘scientifically optimum’ irrigation scheduling which would see pumps run at peak power price times.

In recent years, since approximately 2001, a new paradigm in internet technologies, termed ‘Web 2.0’ (Forrest 2006) and exemplified by websites and other internet applications that allow users to perform tasks that involve extensive interaction with remote computing resources but with limited technical knowledge required, has emerged. It is providing unparalleled access for non-technical people to server resources, databases and other networked information providers.

Such access could allow irrigators to choose their own data sources for decision support, rather than relying on a predetermined set and, if the total set of data sources available for them to choose from included both biophysical and non-biophysical data sets, then they may achieve better ‘real world’ support.

1. DSS CLASSIFICATION

1.1. The Network Paradigm

Classification by networking paradigm focuses on the abilities of a DSS to present data and information from different resources to a user and also how that data is presented. The word ‘networking’ is used in both technical and conceptual senses. Classification in this way places existing DSS in the following categories:

1. None – standalone desktop application.
2. Single Link – desktop applications that collect data from a single machine, logger, or sensor.
3. LAN - desktop/intranet-based DSS with information only from local area network (LAN) resources such as other computers,
local network databases and local network sensors.

4. **Enterprise Network** – DSS using large, proprietary and purpose-built, networks, such as (Supervisory Control and Data Acquisition) SCADA networks, to collect data as well as the resources available to C3 DSS.

5. **Internet** – use the internet to access multiple, possibly remote, instances of the resources available to C4 DSS and purpose-built data sets presenting information over the internet.

Figures 1 and 2 show schemas of DSS categories 1 to 5.

**Figure 1. Schema of Category 1-4 DSS**

An example of C1 DSS is the comprehensive crop management tool APSIM 5.3 by APSRU ([http://www.apsim.info/apsim/](http://www.apsim.info/apsim/)) which models many different factors affecting crop performance using data entered into and stored on the DSS user’s personal computer. irriMAX™ ([http://www.sentek.com.au/products/irrimax.asp?language=en](http://www.sentek.com.au/products/irrimax.asp?language=en)) by Sentek is an example of a C2 DSS that presents soil moisture information from field sensors to a PC user. An example of a C3 DSS is ‘Probe for Windows’ by Research Services New England, which presents soil moisture data from multiple, different probes connected to via a network on a desktop PC. C4 DSS would likely be seen on large corporate farms and urban irrigation systems. An example is ‘ET Drive’ by the South Australian-based company Micromet which uses local evapotranspiration (ET) values, connected via a radio link, in conjunction with database information to control urban irrigation systems. An example of a C5 DSS is WaterSense, a sugarcane irrigation scheduling tool developed by the CSIRO. (Inman-Bamber, Webb et al. 2006). This DSS uses the internet to present its user interface to irrigators as well as using the internet to download remotely calculated ET data.

**Figure 2. Schema of Category 5 DSS**

The authors propose a further category of DSS – category 6 *Interoperably connected* – to be used to describe future DSS that use Web 2.0 and Semantic Web technologies. Figure 3 shows the schema of a category 6 DSS.

**Figure 3. Schema of a Category 6 DSS**

The three features that distinguish a C6 DSS from a C5 DSS are new DSS features, **multiple interfaces**, **user-defined data sources** (UDDS) and **Web Services resources**. These three features are made possible by new technologies in internet and software engineering and result in new forms...
of networking, thus deserving a new category in the ‘network paradigm’ classification.

1.2. Category 6 Features

Multiple interfaces

Traditionally DSS front-ends have been restricted to desktop programs and internet browsers viewed on PCs. There is at least one example of a mobile computer-based, model-driven, irrigation DSS (Hornbuckle, Christen et al. 2005) that is a C1 DSS on a mobile computer but there are no examples of C2 – 5 DSS that use interfaces other than PCs and none whatsoever that use multiple physical interfaces.

Having a DSS with a primary PC based interface and secondary mobile interfaces will allow for greater flexibility in the DSS’s use. In this way any irrigator may view comprehensive advice on a PC in the morning, before going into the paddocks, and then view a simpler version of the same or updated advice later in the field on a mobile phone or SmartPhone at the time when they are most likely to act upon it by doing something like turning on irrigation pumps.

In addition to past DSS being restricted to using a single physical interface, most have also been restricted to delivering support in one mode to one type of user. For example WaterSense only allows users to receive scheduling advice based on a predefined set of input parameters. It does not allow users with varying degrees of scientific and IT understanding to receive decision support in other ways. It also requires that training of potential users be put in place before use. Potentially a DSS could present very simple decision support information to an untrained user and then allow them to ‘opt in’ for further modes of more complex decision support.

User-defined Data Sets

A UDDS is envisaged to be a data set, such as a local or remote database, or data source, such as a local soil moisture probe or local automatic weather station, that is added to the pool of data sources used by the DSS by the unassisted DSS user. The key is that the UDDS would be added after design time, so the user would be able to add datasets and data sources that the user values that were not specifically catered for by the DSS designers. This is contrasted to the current situation where, even if it were possible for a user to add a new data set to their instance of a DSS after design time, and mostly this is not possible, significant 3rd party involvement (perhaps expensive software consulting) would be needed. Allowing UDDS could have the effect of allowing users to customise the support they receive and thus make it more applicable to their particular situation

Web Services Resources

Capitalised ‘Web Services’ refer to a software system designed to facilitate machine-to-machine interactions over the internet. One may see Web Services in operation where one has a desktop ‘widget’ program that communicates with Amazon.com and each day updates a display of the last 10 books published on a particular topic. Web Services are realised as programmatic functions, also known as methods, hosted on a remote server machine that can be accessed by a client application. This is similar to a web server’s web pages being accessed by a client using an internet browser but with far more possibilities for the types of data delivered and presentation. The aspects of Web Services technology that set them apart from other internet technologies are those relating to service description and discoverability which allow Web Services to be used by a client machine without the need for human involvement during setup or operation.

The concept of Web Services resources is similar to that of UDDS and would only differ in location of the data provider (remote-via-internet for Web Services resources and local for a UDDS) and that Web Service-enabled data sets could be relevant and available to many DSS users.

2. NEW DSS BACK-END SYSTEMS

2.1. Architecture for C6 DSS

For the proposed C6 DSS, back-end architecture needs to be able to provide:

a. The ability to create different subsets of the total decision support information set. A particular subset would be chosen based on the application used to view decision support.

b. The ability to cope with new, unforeseen, data sets and sources relevant to decisions by processes also unforeseen by DSS designers.

c. The ability to ‘mix and match’, at least to some extent, the data sources selected by the users to be used in generating decision support.

With the advent of widespread Internet Protocol (IP) interoperable communications, such as 3rd
generation (3G) cellular networks, it is now easier to present DSS information on mobile devices than before, albeit with additional carrier’s fees. Additionally other internet-to-wireless services, such as the cellular phone Short Messaging Service (SMS) gateways and the Multimedia Messaging Service (MMS) that use SMS with internet web pages, offer the possibility for DSS to use mobile non web-based data delivery channels. 3G networks and Web Service protocols also allow ‘widget’ programs on many platforms to present data gathered via the internet in many different ways including cellular phone Java applications.

Great advances in software programming frameworks, such as Microsoft’s .NET and the open source Java platform Eclipse, allow back-end code used for PC-based programs to easily be ported to mobile devices such as SmartPhones. The authors are currently testing .NET-based mobile font-ends to DSS. Some of their work can be viewed at http://irrigateway.net/dev/mobile.

There are significant technical obstacles that need to be overcome in order to create a DSS that exhibits characteristics b. and c. The main obstacle is that a DSS needs to understand the format of the data presented to it which, in the case of a range of potential data sources or sets is a very large task. A second obstacle is how a DSS may then use such data, once it has understood it. Two methods to address the first obstacle are now proposed.

The first involves writing software ‘drivers’ for each of the possible resources that a user may potentially add to their DSS. There is precedence for this, for example Research Services New England’s ‘Probe for Windows’ implements drivers for many (they claim practically all) of the soil moisture probes available in Australia. Since there is a limited range of potential biophysical data sources available to DSS users, this may be possible for them. It may also be possible to write drivers for most of the remote biophysical data sets available for DSS use in Australia, for example ET readings from Queensland’s Dept. of Natural Resources and Water’s remote interpolated ET service known as SILO project. This method is not comprehensive, will always lag behind new data set establishment and requires much effort on behalf of the DSS designers. The authors believe it to be of limited use.

A second method that could be used to overcome the technical obstacle of DSSes understanding data formats would be the widespread standardisation of data source service description and data interchange formats. If data sources and their data sets are available and presented in a standardised way, their addition to a DSS’s reasoning engine could be much simplified.

The Open Geospatial Consortium (OGC) is working on project called Sensor Web Enablement (SWE) which is a group of standards and protocols “specifying interoperability interfaces and metadata encodings that enable real time integration of heterogeneous sensor webs into the information infrastructure.”(Open Geospatial Consortium 2007). An implementation of a C6 DSS could allow a user to connect any standards-compliant data source so that broad use of standards as SWE would offer much data source and data set choice available to the DSS user.

One of the potential aspects of SWE is that it allows data sources to be ‘discoverable’ thus allowing machines, such as DSS, to find and use them without human involvement. An example situation using a version of SWE could be seen if a DSS user wished to run a decision support advice generation scenario using up-to-date evapotranspiration (ET) information (used for irrigation scheduling from SILO and compare the results to the same scenario using up-to-date ET data from a local, web-enabled, Automatic Weather Station (AWS). They would be able to do so by simply selecting a different ET source on their DSS front-end without any knowledge of how that source were connected to their DSS.

SWE deals only with data from sensors, however there are many non sensor-derived data sets that may be able to be used by a DSS. To connect to such datasets and to make sense of the data they present would require a data interchange format similar to those provided by SWE. A specification that was a superset of SWE’s sensor specifications that included other non-sensor-based data sources or a specification that was an addition to SWE’s specifications is needed. This specification, whatever form it took, would describe non-biophysical data sources (such as economic data sources), calculated data sets and also historical data sets. This would then provide DSS designers with a comprehensive specification of existing and potential data sources that are or could be used by DSS. If new data sources were designed to present data in a standards compliant way, a DSS design may be able to pre-emptively allow for their use as they would be reasonably similar to existing data sources to connect to.

SWE uses Web Services protocols for connecting to data sources and sets over the internet. If both sensor-based and non sensor-based data were presented in a standards compliant manner, Web Services resources may all be accessed in a one
way as if through a portal. Such direct connectivity of sensors to the internet and therefore to internet-enabled tools, such as DSS, allow for information sharing which may be used for benchmarking, ground truthing remote equipment and the interpolation of results to areas were there are no local sensors.

The authors believe methods for addressing the second obstacle, mentioned above, require a new form of DSS reasoning engine and will not be considered here.

2.2. Architecture to enable Benchmarking

Something that has not been done by current agricultural DSS of any category in Australia is to provide instances of decision support advice or the recorded outcomes of that advice’s use to multiple users for group learning. If a DSS were inherently connected to a network or the internet it could provide this functionality. By the phrase ‘inherently connected’ the author means ‘designed with connectivity in mind for all aspects of the DSS operation’. This is in contrast to the design of all extant agricultural DSS in Australia which mostly act as standalone software systems that, if connected to remote resources at all, use connectivity to achieve small, singular, tasks such as the retrieval or delivery of data only. An inherently connected DSS could, for example, collect data from multiple sources (perhaps UDDS and Web Services resource via a portal) and store it remotely to the DSS user, perhaps on a server. It could then run models and compute derived personalised and non-personalised data, also on a server, and then store that data, along with the user’s usage metrics to be used in decision support for other users.

Such a DSS would, in many senses consider system users as data sources rather than users and in doing so could be enabled to learn from such users by pattern recognition techniques as is currently done with standard data sources. In addition to benchmarking, such architecture would allow for:

1. New science, in both the agricultural sciences and information engineering fields, to be added to the DSS without a new product or version release (this would be achieved in the same way that Microsoft upgraded Windows XP with the addition of ‘service packs’ downloadable over the web, rather than requiring customers to purchase new full versions).
2. The possibility for accumulated usage data to be analysed by researchers that may then be fed back into the DSS to further enhance decision support or used elsewhere.
3. The possibility for irrigation decisions on different scales to benefit from a single information repository, for example irrigation companies producing better estimates of their growers’ needs thought the enhanced ability to monitor irrigation demands.

3. NEW FRONT-END SYSTEMS

3.1. Modular Design

A concept of modular presentation, whereby base data, from sensors and non sensors, data from models and datasets derived from the base data was available for independent presentation, would allow a user to customise the decision support they received. Such modularisation could possibly be achieved by designing a display ‘shell’ that could then be altered for many sorts of data. Users could perhaps even create their own module, from such a shell which may provide the front-end to UDDS.

A website that has made some progress towards displaying any datasets that any user may upload is Swivel.com. This website, while not a scientific research tool, nonetheless leads the way in terms of heterogeneous data fusion. An agricultural DSS could emulate some aspects of Swivel.com in attempts to provide for the display of different types of data.

3.2. AJAX and cached presentation

AJAX, an acronym for Asynchronous JavaScript And XML it is an approach to client-server programming for internet resources that uses the power of modern internet browsers to access server data ‘asynchronously’, that is at times other than client request times (Garrett 2005). This is in contrast to the approach taken by most web page designers including DSS designers that use web-based presentation which typically sees a new page loaded when requested by a user through the internet browser on the client computer. This approach allows applications viewed thought internet browsers to act similarly to desktop applications by updating parts of the screen without reloading the entire screen. This approach to web design is used extensively by companies such as Microsoft Inc. and Google Inc.
The authors postulate that a DSS running calculations on a server remote to the DSS user could use AJAX methodology to present numerous data sets and modelled scenarios to a user, seemingly instantaneously. For example, a user might wish to firstly view modelled irrigation event timings that maximise water use efficiency and then secondarily view modelled irrigation event timings that favour a reduction in electrical power costs at the expense of water use efficiency. To seamlessly generate both sets of outcomes, at run-time, may not be possible for a desktop computer due to performance limitations but may be possible on a very fast server or server cluster. Displaying both those outcome sets seemingly instantaneously would not be possible using standard internet design techniques but would be so using the AJAX approach.

4. CONCLUSION

A new generation of decision support systems can be conceptualised for use in irrigation in Australia. Such a generation of DSS would hopefully be better placed to address some of the large problems facing DSS in agriculture today, particularly that of poor uptake by allowing for greater flexibility in the way data is presented and information that is displayed. By allowing for greater user control of the DSS interface and data sources used, the DSS should be able to address more of a particular users’ wants and this may encourage uptake. Potentially such a customisable DSS could also be run by a by a consultant on behalf of a n irrigator and tailored to that irrigator’s needs, rather than the consultant relying on interpreting results for that irrigator from a ‘one size fits all’ DSS.

Some properties of new generation DSS are based on technologies, such as Web Services, that have not yet seen use in the agricultural DSS arena and are thought to be readily achievable, while other potential properties need further informatics investigation and research to achieve realisation.

5. REFERENCES


Appendix E

SEQ ET₀ SMS Trial 1st Year Report

E.1 Scope

This report is an informal report that has been generated from interactions with, and surveys of, the irrigators, extension personnel and scientists involved with the South-East Queensland ET₀ delivered by SMS trial. It has not undergone scientific peer review and is produced only to informally deliver preliminary results to interested parties. Formal publications on the results from this trial will be combined with results from the irrigation scheduling SMS trials conducted in Griffith, NSW, by the author over the 2007/2008 irrigation season.

E.2 Aim of the trial

The aim of this trial was to test the utility to irrigators in delivering them evapotranspiration (ET) information via the mobile phone Short Messaging service (SMS). Reference evapotranspiration, ET₀, was delivered.

As the author had no knowledge of the irrigator’s scheduling practices before commencing this trial, no assumptions were made as to how or if they may use the information.

E.3 Methods and Materials

E.3.1 Data

The data delivered to the irrigators was a single ET₀ value for the previous day. Each ET₀ value was calculated by the SILO project¹ using the FAO56² method using numerical weather variables from the Bureau of Meteorology’s weather stations. The calculated values were then interpolated between the weather stations to give approximate ET₀ values, on a 5km grid, which allowed values to be generated for each irrigator’s location.

The data was then sent daily to irrigators’ mobile phones via a cellular networks/Internet gateway interface.

¹ http://www.bom.gov.au/silo/
² http://www.fao.org/docrep/X0490E/x0490e00.htm
E.3.2 Participants

The original participants in the trial were as follows:

- 3 Mixed Horticulturists
- 5 Nursery Growers
- 4 Turf Growers
- 2 additional recipients (the author and a research scientist)

The participants, except for the author, were spread over a large area of South Eastern Queensland and were not all in similar climatic zones. Irrigator participants were selected by Steve Raine, through grower groups, on the likelihood of them being interested in the trial.

E.3.3 Feedback methods

The participants completed an initial survey in October/November 2007 in which they were asked about their use of ET data and other information in their irrigation scheduling decisions. They were also asked about their understanding of crop coefficients and their use with $\text{ET}_0$. Finally, they were asked questions about the timing of the messages sent, whether or not they were receiving them and whether or not they were experiencing any technical difficulties with the service.

The irrigators were then all contacted by the author in mid-January 2008 and some individuals again in February and March 2008 for informal feedback and finally all were again contacted in April 2008 for a final survey.

Additional feedback was given to the author by extension officers and industry development officers who regularly communicate with the irrigators involved.

E.4 Results

E.4.1 Usage Statistics

The trial started sending data to irrigators in May 2007. The following table details when irrigators received the data and for those that discontinued with the trial, their given reasons.

<table>
<thead>
<tr>
<th>Participant</th>
<th>Irrigator Type</th>
<th>Service Status</th>
<th>Reason for discontinuation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Horticulture</td>
<td>Continuing</td>
<td>n/a</td>
</tr>
<tr>
<td>2</td>
<td>Horticulture</td>
<td>Discontinued, early 2008</td>
<td>The irrigator did not feel that the ET values were accurate enough, when compared to his observations of local weather, to be used for</td>
</tr>
</tbody>
</table>
irrigation scheduling and has resorted to scheduling by weather data.

<table>
<thead>
<tr>
<th>No.</th>
<th>Horticultural Unit</th>
<th>Status</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>Horticulture</td>
<td>Continuing</td>
<td>n/a</td>
</tr>
<tr>
<td>4</td>
<td>Nursery</td>
<td>Paused over winter 2008</td>
<td>Not irrigating over winter.</td>
</tr>
<tr>
<td>5</td>
<td>Nursery</td>
<td>Continuing</td>
<td>n/a</td>
</tr>
<tr>
<td>6</td>
<td>Nursery</td>
<td>Continuing</td>
<td>n/a</td>
</tr>
<tr>
<td>7</td>
<td>Nursery</td>
<td>Continuing</td>
<td>n/a</td>
</tr>
<tr>
<td>8</td>
<td>Nursery</td>
<td>Discontinued, early 2008</td>
<td>Same as irrigator No. 2.</td>
</tr>
<tr>
<td>9</td>
<td>Turf</td>
<td>Paused over winter 2008</td>
<td>Not irrigating sufficiently over winter to require service.</td>
</tr>
<tr>
<td>10</td>
<td>Turf</td>
<td>Never started</td>
<td>Unknown technical problems have prevented this irrigator from ever receiving the data via SMS.</td>
</tr>
<tr>
<td>11</td>
<td>Turf</td>
<td>Continuing</td>
<td>n/a</td>
</tr>
<tr>
<td>12</td>
<td>Turf</td>
<td>Continuing</td>
<td>n/a</td>
</tr>
</tbody>
</table>

Table 1 shows that most of the irrigators involved with the trial are happy to receive the daily ET$_0$ values via SMS.

### E.4.2 ETo Data usage

Before the trial, only one irrigator claimed to regularly use ET data in his day-to-day irrigation scheduling decision making. That grower was involved with a DPI Queensland weather station trial that has now ended. Another irrigator used long term ET averages to roughly estimate relative irrigation volumes per month.

All of the irrigators in the trial understood the differences between ET$_0$ and crop ET and some had previous experience with calculating crop water requirements from ET$_0$.

Five of the 9 irrigators who are continuing with the trial claimed to factor in the ET$_0$ values they received via SMS directly into their irrigation decision making. A sixth irrigator claimed that the data 'helped' with decision making and a seventh said that he would use the data if it was more accurate (compared with the irrigator's observations of local weather). Of the final 2 growers, one didn't use the values and paused the service over the 2008 winter months and would restart the service on the condition that there was a data accuracy improvement and the other thought the data was mostly useless (too inaccurate for use) from the start of the trial. However, this final irrigator was still interested in receiving the SMSes over the irrigation season, even when asked directly whether to discontinue, which says that he perceived at least some worth in the data.

### E.4.3 How the advice was used

Five of the 7 irrigators who indicated that the ET$_0$ via SMS was useful to them gave specific answers to the question of how they used the data. One of the irrigators used the data to place an upper bound on irrigation volumes by assuming that the ET$_0$ value was higher than
the crop ET for any of his crops. Another kept note of the cumulative ET\(_0\) value over a week and then adjusted his standard irrigation volumes either up or down according to the relative variation in estimated crop water use. The remaining three irrigators calculated daily evaporation mean values for each week and adjusted irrigation values accordingly. One of these three irrigators calculated irrigation pump run times from the daily mean values using crop coefficients that he has developed for his crops and application system values.

The irrigator who found the values useful but did not directly use them in his irrigation scheduling decision process (he said they 'helped') varied his irrigation volumes according to measured temperature variations. He said that the idea of using ET values appealed to him but that he doubted the accuracy of the values and therefore was hesitant to use them.

Four of the seven irrigators who indicated that the ET\(_0\) via SMS was useful to them compared the ET\(_0\) values to soil moisture probe readings and other information sources and all of the 7-value experience, weather observations and plant observations in their decision making. All of the four who compared ET\(_0\) via SMS to their soil moisture probe readings claimed that the values where somewhat correlated or 'followed broadly'.

Two of the five irrigators that used the data in their decision process state that they are now applying less water than they did before starting the trial. One of the irrigators used the data to roughly calculate dam evaporation losses over a year. This allowed him to successfully convince his manager as to the cost effectiveness of purchasing dam covers.

### E.4.4 Irrigator Learnings

Several of the irrigators in the trial claimed to have learned that it was possible to use ET values for scheduling from their experiences. One claimed that he was surprised how ET varied compared to his own weather observations.

It seems that while most of the irrigators knew the theoretical applications of ET, most also either didn't have access to ET data or hadn't ever attempted to use it directly for scheduling before.

One irrigator also stated that he has learned to structured his whole scheduling decision making process more and that receiving the daily ET values has been the foci for that.

### E.4.5 Irrigator requests & suggestions

From the irrigators who already used the service in some way for scheduling, five would like to receive weekly totals and/or averages, several would rather receive the ET data, with average and totals via email and two expressed an interest in receiving long-term averages via email too. When it was suggested that predicted ET may be a future possibility, all irrigators currently using the system expressed interest in receiving it.

Changes to the timing of the SMSes was made so that the irrigators received the messages at 6:30 am on weekdays and 9:30 am on weekends. All irrigators indicated that early morning was the best time to receive data.
E.5 Discussion

It is clear from the survey and phone interview results that the accuracy, or the perceived accuracy, of the ET₀ data was a major problem for the system. All the irrigators who either left the experiment or didn’t really use the data cited their perception of its accuracy as their biggest concern. Comments like “it (the ET) doesn’t change as much as I would have thought for our variable coastal weather” or “it (the ET₀) was still sitting at 3.5mm when it was pouring with rain” were relayed to the author on many occasions.

The utility of the data delivery via SMS is great. All users of the data liked this format and were happy to receive ET₀ in this way either all year round or over the main irrigation season. Some irrigators would have been just as happy with weekly emails but at least one of the irrigators who wanted emails would still like to have the daily SMS as well.

The impression that the author gets is that the daily receipt of data in this direct-to-person way had a way of bringing the scheduling decision making process into focus for the irrigators involved, which they seemed to appreciate. Feedback comments indicated that the irrigators were prompted to contemplate the possible impact of weather and ET on their scheduling decisions more as a result of using the service, even when they did not readily accept the accuracy of the ET data delivered.

Much discussion was entertained by the author at the time of the trial’s inception as to the merits of delivering crop-specific ET data however the irrigators in this trial were all growing a multitude of crops and therefore were all happy with simply receiving a reference ET value (ET₀). It appears that currently irrigators either prefer their own fudge factors for particular crops, as opposed to those developed by other bodies, or that their irrigation techniques are not precise enough to be able to effectively use crop-specific ET values. The one or two irrigators who were interested in crop-specific irrigation volumes calculated from ET₀ and crop coefficients suggested that they would not necessarily be able to use the figures any more effectively than ET₀ presently.

E.6 Conclusions

ET data, when delivered directly to some irrigators, is of at least some use in their scheduling decision making process. The utility of using ET data is much enhanced by effective delivery tools with SMS being reasonably well valued by all and with email probably a valuable medium also. Irrigators who had heard of ET theory but did not use ET in their scheduling decisions were likely to do so, at least to some extent, with easily accessible data.

In its current mode of usage, SILO data is not perceived to be accurate enough for many irrigators to use happily.

Despite many technical problems causing days of no service, irrigators still saw benefit in receiving the data when possible which indicates that some data is better than no data, even when received intermittently.

Further assessments of how the irrigators could or would use data that they perceived to be accurate needs to be undertaken before trialling their use of other ET applications, such as
crop-specific ET. If this is not done, there may be confusion as to where the utility limits of the system lie.

E.7 Future work

New work to be undertaken will initially aim at comparing SILO data use to that of local weather station data use. A trial will be setup where irrigators in close proximity (i.e. similar climate conditions) will receive either SILO or weather station data and their perceived accuracy assessed.

Data delivery format changes will be made that give irrigators the following:

- weekly mean/total ET (email or SMS)
- long-term average ET (email or web)

This has been done in the Griffith ET via SMS trials by the author and looks to be of great use to the irrigators.

Pending the results of comparing SILO and weather station data, work into the delivery of crop-specific evaporation values, rather than reference ET to irrigators may be undertaken. This has been trialled successfully in Griffith for vines and can be extended to many other crops. Delivering such information appears to add much value to the decision support tool but requires either participants, or parties, local to the participants to engage in on-ground work to estimate crop coefficients. Additionally, irrigators who already have crop coefficients that they feel comfortable using may wish to supply these values to a future version of the SMS service which would then crop water requirements from them and ET₀.
Appendix F

Crop Coefficient estimation from mobile phone images

This appendix describes an experiment carried out in 2008/2009 to determine the feasibility of using mobile phone imagery for crop coefficient estimation. This documentation of this experiment was completed in 2017 using original results and references from 2008/2009 with no additions. This work is contextualised in Chapter 4.

F.1. Aim

To generate crop coefficient values for field crops from images of them taken by a farmer with a mobile phone in order for them to act as drop-in replacements for satellite-derived values as needed.

F.2. Hypothesis

Image manipulation techniques can allow us to generate more accurate crop coefficient values than a farmer can estimate using imagery acquired by farmers using their mobile phone cameras.

F.3. Introduction

Remote control aircraft-mounted digital cameras have been used to acquire simple images of crop fields when satellite-based imagery is not available for many years (Stombaugh et al. 2003) with effective measurements of crop type, based on canopy shape, able to be processed from such imagery using sophisticated image processing techniques calibrated with ground-truthing (Oberthür et al. 2007).

Given that crop canopy size can be used to estimate crop coefficient values ($K_c$) for vegetable crops (Trout and Gartung 2006), it is thought that it would be possible to derive $K_c$ values from photographs of crops from above using methods other than remote controlled aircraft which usually prove too difficult or costly for non-specialist use.¹

As of about 2008/2009, Australian irrigators were commonly in possession of mobile phone-based digital cameras. Mobile phones had cameras before this date and some irrigators owned

¹ This experiment was conducted well before the profusion in ‘drones’ or cheap, easy to use, hover-capable remote-controlled aircraft.
them but camera phone ownership grew dramatically over this period. It was thought that such cameras with the internet connectivity that 3rd-generation mobile phone network technology (so-called "3G") could be used to acquire images of crops that might be able to be processed remotely to the acquirer (on an Internet-attached server) and from which $K_c$ values could be generated.

### F.4. Methods

#### F.4.1. Waterbalance calculations

Due to the focus of this experiment on accurately obtaining of crop canopy cover from mobile phone images, the relationship used to generate $K_c$ values from canopy cover was taken to be as per (Trout and Gartung 2006). $K_c$ values were then to be used in to calculate waterbalance calculations as per previous IrrI SatSMS methods (Car et al. 2012) following the standard FAO56-based reference evaporation calculations (Allen et al. 1998). In this way, $K_c$ values from mobile phone imagery would be a drop-in replacement for LANDSAT imagery-derived $K_c$ values.

#### F.4.2. General process flow

The general process flow of information within a system to acquire images of crops, process them to produce $K_c$ values and then to both return the calculated result to the image acquirer for checking and to the crop waterbalance database was specified as per Figure 1. The accept/reject looping is described in Section 1.4.8.

![Figure 1: General process flow for this experiment's data](image)

#### F.4.3. Imaging device

The mobile phone camera initially used in these experiments was intended to represent the average mobile phone camera available to irrigators in the trial area at the time. The phone was a Samsung SCH-U350 flip phone, then available as a standard phone on the Telstra Australian

---

2 See Chapter 2 in this thesis for the phone ownership growth figures.
mobile phone network provider’s NextG network which supplied the Griffith, NSW, area with 3rd-generation mobile phone coverage. Its camera was a 1.3-megapixel device.

F.4.4. Image acquisition

The simplest imaginable case of acquiring a mobile phone photo of a crop from which $k_c$ values might be able to be generated was that of an irrigator able to be positioned significantly higher than a low crop’s canopy. This would approximate images taken by aerial camera which were known to be able to be used and would allow photos taken to both cover significant ground area for good representative samples of the crop of interest and to avoid obfuscation of the image by some on-ground objects.

In previous work in this PhD, turf (grass) farmers in South East Queensland (SEQ) were included in a trial of IrriSatSMS (see Appendix C). Such irrigators both worked with the lowest of crops (turf is usually only a few centimetres tall) and often used lateral move or pivot irrigators to irrigate their turf which could be used as a ladder in the field to obtain extra height for additional ground coverage by photo, see Figure 2. This method was termed the Turf Irrigation System method.

![Figure 2: Sketch of a generic centre pivot irrigation machine with three elevated locations (A), (B) & (C) on the support leg indicated from which an irrigator might take photos of a low crop (turf)](image)

Acquiring images of winegrape, orange or prune crops was considered to be too difficult due to the height of the crop compared with any feasible imaging positions: irrigators of these crops involved in earlier PhD work all used drip systems and thus their fields had no elevated locations from which photos above the crops could be taken.

Around the Griffith NSW areas, chickpea and tomato crops were grown by irrigators known to this PhD’s author’s supervisors. These crops were furrow irrigated and no elevated field equipment was permanently present in the crop fields. A decision was made to trial standing photo acquisition of these relatively low crops to test the simplest of image acquisition methods. This method was known as simple the Standing method.

A third method was tested too: for chickpea and tomato crops, the image acquirer could drive their vehicle to various points around the crop field and mount the vehicle, taking images from approximately 4m high (1.75m vehicle elevation + 2.25m irrigator with up-stretched arm). This method was known as the Vehicle method.

F.4.5. Image transport

The previous IrriSatSMS experiments used an SMS/Internet gateway service from 2007 (see Chapter 3) and the provider of that service implemented MMS – Multimedia Messaging Service – capability in 2008. This capability then allowed a mobile phone user to take a photo, create an
MMS message and attach the photo and then send it to a virtual phone number, the same number used by the SMS experiments, and then to have that MMS made available to an Internet-facing web server via Web Services from the service company for processing. Photos taken with the SCH-U350 flip phone and a variety of other contemporary phones seemed to be delivered fairly instantaneously and without issue to the target web server via the gateway service wherever there was Telstra NextG reception. No special arrangements were made for extended reception (such as attachable aerials that the SCH-U350 flip phone was capable of connecting to) since trial sites with NextG reception were able to be found.

F.4.6. Image colour processing

MMS messaged bearing images were received by the image processing server as a simple JPEG image of approximately 200 – 500KB in size when the usual SCH-U350 flip phone was used. The image, once received, would then be processed in order to determine the percentage of it that was “green” (qualifying descriptions below) which would then act as a surrogate for canopy cover\(^3\), as per the method in (Purcell 2000). Figure 3 is one of the images in (Purcell 2000) showing canopy cover estimates from photos of crops using the “green” classification method.

Before an integrated image processing system was implemented that could receive the image, process it and return results to the image acquirer in a reasonable amount of time (less than a minute), isolated tests of image processing were undertaken in order for this author to understand the task and to obtain results similar to those shown in Figure 3.

\[^3\] Distinguishing between crop and non-crop green – perhaps inter-crop weeds – as is done in other crop image processing systems, such as that shown in (Oberthür et al. 2007), was not considered important in this experiment as non-crop greenery still uses water and thus contributes to the crop coefficient that must be catered for, at least partially.
Figure 3: after (Purcell 2000) Digital images of soy bean canopies representing different plant populations in the field (right) and after (left) scanning. The area highlighted in green in the left column of the figure was divided by the total area of the field of view, and this value was used to calculate the fractional canopy coverage values that are shown.

F.4.6.1. Photoshop method

First, green selection from field crop images using Adobe Photoshop’s colour range selection tool⁴ was tried as the author was familiar with Photoshop before this trial began and wanted to have a manually-specified reference for comparisons with automated “green” extraction.

F.4.6.2. MATLAB RGB method

Next, the MATLAB programming language was used to process test images due to this author’s laboratory’s familiarity with MATLAB and its relatively pre-packaged image processing toolkit⁵. The toolkit contained functions for various image element extraction, including colour.

Several methods of determining “green” were tested in MATLAB. The first being simply a green/not-green using a weighted sum of the Red v. Green values in the Red Green Blue values for each JPEG image pixel.

⁴ See the help page for the tool within the current version of Adobe’s Photoshop: https://helpx.adobe.com/photoshop/using/selecting-color-range-image.html. The tool, in 2008, was essentially the same.

⁵ A 2007 version of the toolkit described at https://au.mathworks.com/products/image.html
F.4.6.3. MATLAB HSL method

The next method tried was processing using different colour representation systems, mainly the Hue, Saturation, Lightness (HSL)\(^6\) system. The first reason for this was to ascertain whether using systems other than RGB made it easier to distinguish plant/not plant than simple green/not green in RGB. The second reason was to try to cater for over-saturated images; an issue easily understood when one examines digital photographs taken on phones without sophisticated lenses and filters and by non-expert photographers which can often be over exposed.

The impetus to use HSL was taken from (Purcell 2000) who used it for crop/soil determination and reported some small issues with high light levels. We expected greater issues as we did not plan to constrain image taking to midday as (Purcell 2000) did and angled sun rays (early morning or late afternoon) seemed to increase the chances of over exposure.

F.4.6.4. MATLAB k-means clustering method

The final method tried was that of k-means clustering\(^7\) with pixel values recorded using L*a*b colour representation system\(^8\) initially implemented in MATLAB. This is a well-known colour selection method and MATLAB implements a toolkit for it\(^9\). The purpose of conversion to L*a*b is to then be able to implement Euclidian distance measurement of pixels on the a & b axes which indicate proportions of red/green, blue/yellow. This is thought to give better results for colour separation than RGB-based distance metrics, such as that used in Section D.5.3.2.

This method was trialled because it is a known method for separating images into a fixed number of colour categories, k, which is particularly useful when k > 2, which is required for accurate green/not green results for images containing substantial portions of a non-ground, non-crop area, such as those containing sky. Since the intention of this image processing was to cater for essentially unconstrained photos obtained by image acquirers, non-ground, non-crop areas would have to be automatically handled, rather than excluded by image positioning methods.

F.4.6.5. C# implementation

Finally, image processing methods trialled in Photoshop and MATLAB were re-implemented using Microsoft’s C#.NET programming framework so that they could be implemented within the image processing server that already used C# code to receive SMS and MMS messages and to deliver website content. MATLAB is not able to perform all these tasks and to just implement an image processing routine in MATLAB that could then be accessed by the server was both costly and more difficult than re-coding the methods.

Additionally, work was done to ensure that C# processing was fast, which involved re-coding the basic C# image processing using unsafe (unmanaged) code\(^10\). The C#.NET framework, circa 2008, did not natively allow for direct image bit manipulation using pointers (unmanaged code) but this was required for useful processing times.

---

\(^6\)A general description of the colour model: [https://en.wikipedia.org/wiki/HSL_and_HSV](https://en.wikipedia.org/wiki/HSL_and_HSV)

\(^7\)K-means clustering is “a least-squares partitioning method that divide a collection of objects into K groups” ([https://www.cs.utah.edu/~arul/report/node18.html](https://www.cs.utah.edu/~arul/report/node18.html)) that can be used for colour selection in images by treating image pixels as vector objects, usually in 3 dimensions, RGB, LAB, HSL etc.

\(^8\)See [https://en.wikipedia.org/wiki/Lab_color_space](https://en.wikipedia.org/wiki/Lab_color_space) for details.


\(^10\)See [https://www.codeproject.com/articles/1099/writing-unsafe-code-using-c](https://www.codeproject.com/articles/1099/writing-unsafe-code-using-c) for an explanation
F.4.7.  Image rectification

For images that were not taken perpendicular to the earth, which in this experiment were all other than those using the Turf Irrigation System method, image rectification (warping) was necessary in order to remove oblique bias.

![Figure 4: A sketch showing a photo taken from the standing position (A) and the theoretical position which would give the best results for canopy cover estimation (B).](image)

Initial rectification trials were done "by hand", that is oblique laboratory and field images were opened in Photoshop and rectified using the Warp tool\(^\text{11}\) with the rectification amounts manually selected.

Automated rectification trials first used red markers placed on the ground in a rectangle, first in the laboratory and then later in the field, amongst crops, to be used to establish parallel lines from which rectification angles could be determined and bias corrected for by image stretching. Figure 5 shows some laboratory and field marker setups. The initial workflow to rectify an image using markers is shown in Figure 5.

![Figure 5: Initial workflow to process an image to determine canopy cover, including rectification.](image)

Following Figure 5: initially colour segmentation was used to establish the location of the red markers in the laboratory or field (red was used as bright red is not a commonly occurring colour in crop fields) then, knowing that four markers were sought, if four were identified using centre-of-mass clustering (k-means clustering, k = 4) using the X & Y coordinates of the red pixels only. This method frequently produced useless results, see Figure 11 B, so a modification to the workflow was introduced to start each centre of mass point in a different quadrant of the image, given that the four markers were required to each be positioned within a quadrant by the image acquirer.

\(^{11}\) See the relevant Photoshop documentation [https://helpx.adobe.com/photoshop/using/warp-images-shapes-paths.html](https://helpx.adobe.com/photoshop/using/warp-images-shapes-paths.html)
In order to acquire an image of a representative part of a field, red markers were placed in a portion of the field that would tessellate if repeated (usually centre of crop inter row to centre of crop inter row, as per Figure 6 B & C).

Due to problems with the use of red field markers discovered in trials (obscuration by crop, inaccurate placement within a photo, annoyance of their use to irrigators), a marker-independent image rectification technique was sought and that of vanishing point correction was settled upon for use.

Vanishing point correction (Gallagher 2005) allows for the automated warping of an image to remove “tilt” by using features within the image to trace vertical lines in the scene which, while in reality are parallel (such as crop rows) converge towards a “vanishing point” when viewed obliquely, as per the crop lines in Figure 6, B & C. An investigation was also made into using two and three images of a crop area for uncalibrated image rectification, as per (Sun 2003).

F.4.8. System timing

MMS and SMS, as per their protocol specifications, have no guaranteed delivery time so networks such as Telstra's NextG discourage reliance on specific send/receipt times. Despite this, it was hoped that a full cycle of image acquisition, sending, processing and response receipt should take an amount of time that an image acquirer could sensibly wait for in the field in order to check the processed value against their assumption while still at the image site.

An effort was made to carry out image processing in under a time that allowed the full basic process flow of Figure 1 to take place in less than a minute. As related in Section 1.4.6, this resulted in firstly re coding image processing in the server available toolkit C# and then further re coding using faster unsafe code in C#.

F.4.9. Feedback

The feedback information flow, as shown in Figure 1, was implemented to both allow for value checking during system testing and also in anticipation of it being part of a final production
system which would continuously check calculations against assumptions of canopy cover. An automated validation feedback mechanism was implemented that allowed an image acquirer to respond to an SMS containing processing results with a simple indication of accept or reject. If reject was selected, the image acquirer could either take a follow-up image which would then start a fresh accept/reject process or they could send their own estimate of canopy cover as a percentage which could then be compared with the rejected processed result and perhaps directly used within the waterbalance system.

F.5. Results

Results for each of the methods in the previous section are given and overall results are given at the end of this section.

F.5.1. Image acquisition

Images were able to be acquired via all three methods: Turf Irrigation System, Standing & Vehicle. Very large area images of turf that could have been obtained via high placement on irrigation systems were not obtained due to safety and insurance reasons (this author was not insured on irrigator’s farms). Both this author and irrigators were able to obtain low irrigation system images (human height + 0.5m). Work with SEQ turf farmers was discontinued with no testing beyond the collection of a few test images in one season was made. This was due to SEQ irrigator’s perceptions of SILO inaccuracies (see Appendix C) and thus their disinterest in, as they saw it, related experiments and also the requirement for the author to focus his time on crops local to the Griffith, NSW area due to the lower cost in working with them (no great travel required).

Standing images could only be sensibly obtained at eye-level, not with arms upstretched. Stretching resulted in erratic directions of acquisition.

Vehicle image acquisition was tested and found possible but not pursued due to rectification issues, see Section D.5.4

F.5.2. Image transport

Image transport and response SMS transport proceeded easily with almost all of the hundreds of test MMS and SMS messages taking 10 seconds or less from sending to receiving. Only one or two messages were delayed and, when they were, they were delayed for more than 12 hours – until after midnight of the day on which they were taken.

Image presentation to the processing server by the SMS/Internet gateway service worked without fault for the lifetime of the experiment and also much longer afterwards as the IrriSatSMS system continued.

F.5.3. Image colour processing

The results shown in this section are of an image taken in November, 2011 in a tomato field. The original image is shown in Figure 6 A. Similar results were obtained for turf and for soybeans from September 2007 until December, 2008. This timeframe allowed for testing with tomato and soybean crops at all stages of growth.
F.5.3.1. Photoshop method

Using Adobe Photoshop’s colour selection tool, images like those in Figure 6A, one is able to select a location on the image of a particular colour and then auto-select all areas of the image of the same colour. A range of colours can be selected. With a range of areas within the crop foliage selected, perhaps 10 different points, auto-selection, followed by green/not green determination leads to images like that in Figure 6 B. Note that in the example in Figure 6 B, the metal awl is determined to be green. This is due to it being, as determined by Photoshop, more green than not green.

In general, it was easy to select what looked like green to the human eye doing this manual work and overall green selections were easily able to be varied by simply selecting more or fewer colours from which Photoshop then undertook auto-selection with. This work provided a good benchmark for automated methods.

F.5.3.2. MATLAB RGB method

Initial results from an automated green/not green determination are shown in Figure 7 C. Here MATLAB was used to read each pixel in the photo and to determine whether they were green or not green by using a simple linear equation with the 0 – 255 Red, Green & Blue pixel integer values of the form:

\[ \text{Thresh} = G - \frac{1}{2} (R + B) \]

The pixel was determined to be green if Thresh (threshold) was greater than some value which was initially zero but then varied in experiments. This equation was chosen due to the 3-axis system of colour representation that, by design, represented “green” values, as seen by the human eye, with a higher Green count than a combination of Red & Blue counts.

Figure 7 C uses a Thresh value of 10 and produces results similar to the manual results in Figure 7 B. Actually, better results are shown in this example: more of the metal awl is treated as not green and so is more shadow than in Figure 7 C than Figure 7 B.

Green/not green determination was straightforward when photos included only crop and earth or crop, earth and other non-green features such as sky or concrete. No specific handling of sun exposed versus shaded green areas was undertaken but, as per images like Figure 7 A, inclusion or exclusion of areas of shade only varied the total canopy cover area by a few percentage points.

Overall, automated canopy cover percentage determined using this method for simple, relatively non-oblique photos was easily as accurate as any manual measures of canopy cover.
The determination of green/not green using colour representation schemes such as HSL and HSV (Hue, Saturation & Value – similar to HSL) seemed to be no more accurate than the simple use of RGB, as reported in the section above. Figure 8 shows the photo shown in Figure 7 A with different transformations applied. The first, altered hue (A) highlights colour differences and considering images with hue values altered could perhaps lead to more optimal green/not green determination but this was not found to be useful as green/not green determination using RGB values seemed sufficiently accurate. The second, (B) shows the image as it appears in an over-saturated photo (high saturation value) which emulated a mobile phone digital camera in bright sunlight. Automatically determining that a photo contained a high saturation value and then transforming to one with a lower value to then apply HSL or RDB green/not green assessments seemed not to be useful: the few canopy cover percentage point differences this transformation changed were not able to be determined to be important over about 10 trials of photos of the same crop area in different light conditions (with and without over saturation).
Figure 8: HSL representations of the image in Figure 7 A with A showing altered hue and B showing altered saturation; high saturation emulating over exposure.

F.5.3.4. MATLAB k-means clustering method

Results from the use of k-means colour clustering with $k = 3$ are shown for the photo in Figure 7 A in Figure 9 with the intermediate step of conversion to $L^*a^*b$ representation given in A and the trinary result image (black, white & grey) shown in B. Here the three clusters, automatically determined, approximate to earth (grey), crop (white) and shadow (black). $k = 3$ clustering for essentially crop/not crop images offered no accuracy advantage over RGB clustering due to the limited area covered by shadow for low crops.

$k = 3$ clustering seemed to be promising for taller crops that cast substantial shadow, such as late season tomatoes, but further investigations into the accuracies of $k = 3$ clustering versus RGB determination were not carried out. See the discussion in D.6.

Figure 9: A. the image from Figure 7 A represented in a false colour representation (extended ranges) of the $L^*a^*b$ colour system to highlight the red/green and blue/yellow categorisation of pixels and B. the same image with k-means colour clustering with $k = 3$.

$k = 3$ clustering was proved to be accurate at sky removal and this was pursued. Figure 10 shows a photo of a low soybean crop (A) with green/not green results processed using $k = 3$ which has automatically, and accurately, treated both earth and sky as not green. Background trees in the photo are partially treated as green.
An attempt was made to cater for the edge but real case of processing images with no green in them at all. The absolute values of the final centres of mass (the three clusters) were examined and if none of them was vaguely green, the image would be assumed to be of bare ground. This would then prompt the system to send the image acquirer a "please resent" SMS message with a note to say that the system found no green. In practice this was entirely avoided by simply requiring the image acquirer to perceive some green before sending.

F.5.3.5. C# implementation

The use of C# code to replicate the results of MATLAB RGB green/not green processing was successful with results being indistinguishable. This was possible due to the ability of the author to inspect the MATLAB and C# code and compare calculations on a pixel-by-pixel basis.

Results for k-means clustering implementations in MATLAB & C# were similar but not identical. This was due to differences in the k-means algorithms used by the clustering toolkits in MATLAB & C# used by the author. While different, results were typically within 10% of one another which meant that it was not possible to determine whether the MATLAB or C# results were more accurate due to the field measurements of canopy cover used for accuracy evaluation themselves not being accurate to more than 10%.

As noted in Methods section D.4.6.5: the initial C# implementation, while able to produce accurate results and able to be implemented on the image processing server, required processing times of 30 – 60 seconds which was deemed too slow to be of use to image acquirers wishing for timely feedback after sending an MMS from the field. Reimplementation of C# image processing (both RGB and k-means clustering) using unmanaged code sped image processing up by a factor of 5 – 10 meaning image processing could take place in 3 – 6 seconds.

F.5.4. Image rectification

Initial image rectification by hand in Photoshop proved straightforward: images were able to be warped for parallel lines to appear parallel.

F.5.4.1. Marker trials

For rectification using markers: while RGB-based, red/not red selection for determining marker pixels was straightforward centre-of-pixel-mass clustering frequently yielded problematic results. Even when red/not red selection yielded results that a human could easily identify as
being 4 markers, centre of mass calculations often yielded useless results, such as those in Figure 11 B, which was due to the mathematics of average centre-of-mass calculations when some markers appeared larger and thus yielded more red pixels than others. This was practically always the case, meaning that centres-of-mass could be determined outside marker locations.

The solution to this was not to change red/not red tolerances, as initially planned for as per the workflow in Figure 5 but to implement independent, quadrat-based, marker centre-of-mass searches whereby the photo was split into 4 parts and a red marker sought in each quadrat independently. This did then constrain image acquisition by requiring the image acquirer to place a marker in each quadrat.

Figure 11: Three images showing A. a bare field (no crops) with four red markers placed in it, B. marker centroid estimation error where, while colour segmentation has worked and the markers are easily distinguished from the background, centroid calculations (white crosses) have conflated separate markers. C. correct centroid estimates using forced image area segmentation with quadrants indicated with green overlay lines.
Results from the marker-based rectification of small mobile phone images, as opposed to larger non-mobile phone digital camera images used in testing, looked like that show in Figure 13.

F.5.4.2. **Vanishing point correction**

Despite successful image rectification using markers by the author, problems were encountered when others were asked to replicate these results. Effective marker placement and especially placement of markers within image quadrats proved difficult for others to accomplish. In addition, irrigators reported expected annoyance at the prospect of having to carry four red markers if the markers were as large as the red balls shown in Figure 13. These were the reasons *vanishing point corrections* were attempted.

A workflow presented as a sequence of images in Figure 14 originally presented at a conference in 2010 by this author (Car et al. 2010) was thought to be sensible for vanishing point corrections. Sky removal – sky could be detected using $k = 3$ k-means clustering – was seemingly necessary in order to ensure that accurate edges (see the white lines superimposed on Figure 14 C) could be found for image cropping and rectification. It was thought that using half the image (the half closest to the image acquirer) would reduce taller crop distortions.
Figure 14: a sequence of images (a – d) showing an intended workflow for vanishing point correction canopy cover estimates.

No sensible results were recorded using vanishing point correction methods. This author was not able to either replicate the edge detection and multiple matrix transformations necessary for this family of rectification techniques to work or to find code toolkits that implemented them. Other than the preparatory step of sky removal, as per Figure 14 B, no other steps in the Figure 14 workflow were able to be implemented in an automated fashion with MATLAB or C#.

F.5.5. **System timing**

Once an image acquirer had sent an MMS message, a feedback SMS was usually received by them in less than a minute, which was thought to be a reasonable time for them to have to wait for results while in the field. MMS message sending usually took less than 10 seconds, as did each of the following stages: image receipt by server, server image processing (when the faster unsafe C# code was used), server SMS response and image acquirer SMS receipt.

Only very occasionally were SMS or MMS messages not sent or delivered within these times: this was only encountered once in many months of field MMS + response tests. Failures to send/receive were seen on a few other occasions (3 in total) in the longer running IrriSatSMS experiments (see Chapter 3) and were all attributed to Telstra network issues which were only very occasional but completely unavoidable.

F.5.6. **Feedback**

Feedback from this author on automated processing canopy cover figures was somewhat inconclusive: the author was inconsistent with canopy cover estimates often overestimating small percentages of cover – in the case of the image in Figure 14 with a calculated canopy cover of 4.9%, estimated 10%, double the calculated amount – and with larger crops proved to be only
with 10% of the measured estimate. These inconsistencies invalidated any comparisons of the authors estimates to calculated results.

Figure 15: C#-based, automated green/not green pixel classification of the rectified photo in Figure 13 B with the canopy cover (white) calculated at 4.9%.

Feedback from others on processing results was not easily obtained. Estimates from the tomato and soybean field owners were within 15% of manually measured figures (measured approximately with a tape measure) and were within about the same range of processed results.

F.5.7. Overall results

Canopy cover results from marker-rectified, phone images where within 15% of manually measured values for the tomato crop from 5 – 70% canopy cover. 95 crops looked like those in Figure 13 and 70% tomato crops looked like those in Figure 16 A. Results for the soybean were within 15% from 5 – 30% manually measured crop cover, after which point the crop was abandoned by the farmer for water supply reasons. The abandoned crop soon failed and presented as per Figure 16 B.

Figure 16: A. the tomato crop late in the 2008/2009 season and B. the soybean crop abandoned in last season.

F.6. Discussion

A large amount of IT/software work was undertaken for this experiment and a lot of field measurements and photographs were taken but they did not allow for accurate testing of the hypothesis.
The author did not have a clear plan to test different crops at different growth stages and under different field conditions that would be required to obtain meaningful results of processed versus measured canopy cover values. Furthermore, systematic canopy cover estimate gathering at the time of photo acquisition was not undertaken for the testing required to address the hypothesis. This means that even if reliable results were obtained from processing, another season's worth of data would have needed to have been collected to undertake this comparison.

Of the IT work undertaken, much of it was essentially peripheral to the testing of the difficult part of the experiment, that of obtaining accurate results from image processing. System integration of processing, processing timing improvements and automated workflows to handle estimates and system/image acquirer communication are far more relevant to the task of turning an accurate but unusable (slow or clumsy to use) system into a usable one and should not have been undertaken until the difficult part of the process was proved.

Investigations into turf farmers was abandoned too soon and without results. Rectification would have likely not been necessary to get useful results from turf farms thus work with those irrigators could have far more easily have focussed on processed versus estimated canopy cover.

Investigations into better k-means shadow removal wasn't investigated due to the lack of significantly shading crop after the abandonment of the soybean crop which has started to develop significant shade. The tomato crop even at 70% canopy cover seemed not to cast much shadow outside its leaf cover area.

Vanishing point correction techniques, as per the method shown in (Sun 2003), even if able to be implemented by the author, would have required 3 images to be taken which would have imposed effort on the image acquirer and, more importantly, would have introduced a possible source of error with unusable images possibly being taken (images that could not be cross correlated for vanishing point calculations).

Although strictly outside the terms of reference of this experiment's hypothesis it is relevant to the experiment's aim to know that author did not have clear thoughts on the accuracy of crop coefficients necessary to create useful waterbalance calculations. Generalised crop-specific growth curves can be implemented for season-long waterbalance calculations and even regionalised for crop types per area, such as tomatoes in the Murrumbidgee Irrigation area. If better-than-estimated canopy cover percentages, and thus crop coefficients, for crops were able to be produced from image processing, the author could not say whether these would have lent utility to waterbalance calculations over and above just date-shift calibration of these general and regionalised growth curves.

F.7. Conclusion

The hypothesis was not proved or disproved. An inconclusive result was obtained due to uncertainty in the comparison of the processed results of crop canopy cover versus irrigator estimates. Estimates of crop canopy cover from 5 – 70% with a 15% margin of error This uncertainty was a result of poor datasets: two crops, neither measured for their full lifecycle – one for less than half – and only for one season.

F.8. References


Decision-making need and a domain

Decision Support Systems (DSS) encode expert knowledge and perhaps data for decisions to help users attain best practice. Few DSS cater for different decision scenarios or even variations within a scenario.

Standardised decision modelling would allow us to articulate many decision types within a domain and variations within a type consistently perhaps allowing DSS designers to better cater for decision ranges.

For irrigation decisions by smallholding farmers, we would like to characterise decisions they make in a standardised way, knowing that many factors affect their overall practice (Whittenbury & Davidson, 2009).

Case-Based Reasoning with decisions

A way to provide support for a expert system is to compare them to previous ones using Case-Based Reasoning (Aamodt & Plaza, 1994). Current cases are matched for similarity to previous one whose results must be known, then best practice advice is offered with the current case then stored for future use. Typically CBR systems use a cycle, see Fig. 1, and require a similarity metric to compare cases.

Using DecPROV and Semantic Web modelling generally, schemaless RDF triplestores can be used to store decisions and the standardised SPARQL query language used to compare them. For example, a query could find decision outputs (an Answer) sharing datasets of Type X as an input (see Listing 1).

Current Work

• Categorising known online irrigation-relevant data sources
  • So decisions using similar input data can be selected for
  • Testing the modelling power of DecPROV
  • Does it cover all/many irrigation decisions?
  • Does using DecPROV improve data provenance generally to assist with other questions such as those about data quality?
  • Establishing a range of similarity measures as SPARQL queries
  • Storing anonymised decision instances and listing them publicly (persistent URIs)
  • So they can be found, referred to and reused in CBR and generally

Future Work

• Once a full CBR cycle is implemented, providing CBR decision support
• Discovering hitherto unknown decision making patterns in irrigation to inform future non-CBR decision support systems
• Expanding the use of DecPROV to other domains

References


Appendix H

Code for OWL diagrams in Chapter 1, Introduction


This code is available both here and also in a code source control repository online at http://github.com/nicholascar/phd/.

Code for Figure 1

> The decision to Decide about PhD topic modelled as an OWL diagram containing DecPROV ontology classes (N J Car 2017) developed within this PhD and described in Chapter 7.

@prefix : <http://github.com/nicholascar/phd/ch01/choosing-the-phd-topic#> .
@prefix dc: <http://purl.org/dc/elements/1.1/> .
@prefix dct: <http://purl.org/dc/terms/> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix xml: <http://www.w3.org/XML/1998/namespace> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix prov: <http://www.w3.org/ns/prov#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix decprov: <https://promsns.org/def/decprov#> .
@base <http://github.com/nicholascar/phd/ch01/choosing-the-phd-topic> .
This OWL document contains modelling of the decision, by Nicholas Car's supervisors, to choose a PhD topic for him.

This ontology is presented in Nicholas Car's PhD thesis Chapter 1: Introduction.
prov:qualifiedStart rdf:type owl:AnnotationProperty .

###  http://www.w3.org/ns/prov#used
prov:used rdf:type owl:AnnotationProperty .

###  http://www.w3.org/ns/prov#wasAttributedTo
prov:wasAttributedTo rdf:type owl:AnnotationProperty .

###  http://www.w3.org/ns/prov#wasDerivedFrom
prov:wasDerivedFrom rdf:type owl:AnnotationProperty .

###  http://www.w3.org/ns/prov#wasGeneratedBy
prov:wasGeneratedBy rdf:type owl:AnnotationProperty .

###  http://www.w3.org/ns/prov#wasStartedBy
prov:wasStartedBy rdf:type owl:AnnotationProperty .

###  http://xmlns.com/foaf/0.1/member
foaf:member rdf:type owl:AnnotationProperty .

#  Classes

###  http://xmlns.com/foaf/0.1/Group
foaf:Group rdf:type owl:Class .
In this OptionSelection, the PhD Supervisors decided to select studying about DSS due to both the Requirements for it - it being both Supervisable and Implementable - being able to be met and no other OptionSelections being able to satisfy their Requirements.

Choose studying about DSS

This OptionSelection did not produce the final decision - to study sensor networks - as the PhD supervisors thought that while a topic was supervisable (as per one Requirement) it was unlikely that it would be Implementable within the lifetime of the PhD and the CRC for Irrigation Futures, given CSIRO & University staff capacity.

Implementable for DSS
###  http://github.com/nicholascar/phd/ch01/choosing-the-phd-topic#DSSSupervisable
:DSSSupervisable rdf:type owl:NamedIndividual ,
decprov:Requirement ;
rdfs:label "Supervisable for DSS"@en .

###  http://github.com/nicholascar/phd/ch01/choosing-the-phd-topic#Decide
:Decide rdf:type owl:NamedIndividual ,
decprov:DecisionMaking ;
rdfs:comment "This was the main decision making activity in which the PhD Supervisors decided the PhD topic"@en ;
rdfs:label "Decide about PhD topic"@en ;
prov:wasStartedBy :What .

###  http://github.com/nicholascar/phd/ch01/choosing-the-phd-topic#DssIsImplementable
:DssIsImplementable rdf:type owl:NamedIndividual ,
prov:Entity ;
prov:wasDerivedFrom :DSSImplementable ;
prov:wasGeneratedBy :ChooseDSS .

###  http://github.com/nicholascar/phd/ch01/choosing-the-phd-topic#DssIsSupervisable
:DssIsSupervisable rdf:type owl:NamedIndividual ,
prov:Entity ;
prov:wasDerivedFrom :DSSSupervisable ;
prov:wasGeneratedBy :ChooseDSS .

###  http://github.com/nicholascar/phd/ch01/choosing-the-phd-topic#EvanChristen
:EvanChristen rdf:type owl:NamedIndividual ,
foaf:Person ;
rdfs:label "Evan Christen"@en ;
foaf:member :PhdSupervisors .

### http://github.com/nicholascar/phd/ch01/choosing-the-phd-topic#GrahamMoore
:GrahamMoore rdf:type owl:NamedIndividual,
    foaf:Person ;
    rdfs:label "Graham Moore"@en ;
    foaf:member :PhdSupervisors .

### http://github.com/nicholascar/phd/ch01/choosing-the-phd-topic#IrrigInfoForDSS
:IrrigInfoForDSS rdf:type owl:NamedIndividual ,
    decprov:Answer ;
    rdfs:label "Informatics for Irrigation DSS"@en ;
    prov:wasAttributedTo :PhdSupervisors ;
    prov:wasDerivedFrom :What ;
    prov:wasGeneratedBy :ChooseDSS .

### http://github.com/nicholascar/phd/ch01/choosing-the-phd-topic#IvanMareels
:IvanMareels rdf:type owl:NamedIndividual ,
    foaf:Person ;
    rdfs:label "Ivan Mareels"@en ;
    foaf:member :PhdSupervisors .

### http://github.com/nicholascar/phd/ch01/choosing-the-phd-topic#PhdSupervisors
:PhdSupervisors rdf:type owl:NamedIndividual ,
    foaf:Group .
### http://github.com/nicholascar/phd/ch01/choosing-the-phd-topic#SNSupervisable
:SNSupervisable rdf:type owl:NamedIndividual,
    decprov:Requirement;

### http://github.com/nicholascar/phd/ch01/choosing-the-phd-topic#SnIsSupervisable
:SnIsSupervisable rdf:type owl:NamedIndividual,
    prov:Entity;
    prov:wasDerivedFrom :SNSupervisable;
    prov:wasGeneratedBy :ChooseSensor .

### http://github.com/nicholascar/phd/ch01/choosing-the-phd-topic#What
:What rdf:type owl:NamedIndividual,
    decprov:Question;
    rdfs:label "What should this PhD be about?"@en .

### Generated by the OWL API (version 4.2.8.20170104-2310) https://github.com/owlcs/owlapi

Code for Figure 2
-----------------
> General outline of the progress of this PhD’s experiment topics from initial ideas driven by the selected PhD topic (see Figure 1) to this document modelled as an OWL diagram containing PROV ontology (Lebo, Sahoo, and McGuinness 2013) & DecPROV classes. All relationships are defined in PROV-O.
This RDF document contains modelling of the changing experiment topics selected for this PhD, by Nicholas Car and his supervisors.

This ontology is presented in Nicholas Car's PhD thesis Chapter 1: Introduction.

```r
rdfs:comment "This RDF document contains modelling of the changing experiment topics selected for this PhD, by Nicholas Car and his supervisors."

This ontology is presented in Nicholas Car's PhD thesis Chapter 1: Introduction."
```

### Individuals

```r
# Individuals

### http://github.com/nicholascar/phd/ch01/choosing-the-phd-topic#IrrigInfoForDSS
<http://github.com/nicholascar/phd/ch01/choosing-the-phd-topic#IrrigInfoForDSS> rdf:type owl:NamedIndividual ,
decprov:Answer ;

rdfs:label "Informatics for Irrigation DSS"@en .
```

```r
# Individuals

### http://github.com/nicholascar/phd/ch01/experiment-topic-progression#ExperimentTopicsAfterLiteratureReview
:ExperimentTopicsAfterLiteratureReview rdf:type owl:NamedIndividual ,
prov:Collection ;
prov:used :LiteratureReviewSomeExperimentation ;

rdfs:label "Experiment Topics After Literature Review"@en .
```

```r
# Individuals

### http://github.com/nicholascar/phd/ch01/experiment-topic-progression#FinalExperimentTopics
:FinalExperimentTopics rdf:type owl:NamedIndividual ,
prov:Collection ;
prov:wasGeneratedBy :SubstantivePhdWork ;

rdfs:label "Final Experiment Topics"@en .
```

269
The progression of this PhD's Experiment topics from the initial topics to this document's Chapters modelled as an OWL diagram containing PROV-O & DecPROV classes. All relationships are defined in PROV-O, unidentified relationships are wasDerivedFrom. This diagram is an extension to Figure 2.

Code for Figure 3
--------------------
> The progression of this PhD’s Experiment topics from the initial topics to this document’s Chapters modelled as an OWL diagram containing PROV-O & DecPROV classes. All relationships are defined in PROV-O, unidentified relationships are wasDerivedFrom. This diagram is an extension to Figure 2.
RDF document contains detailed modelling of the changing experiment topics selected for this PhD, by Nicholas Car and his supervisors. It is a more detailed version of the RDF document <http://github.com/nicholascar/phd/ch01/experiment-topic-progression>.

This ontology is presented in Nicholas Car's PhD thesis Chapter 1: Introduction.
###  http://www.w3.org/ns/prov#hadMember
prov:hadMember rdf:type owl:AnnotationProperty .

###  http://www.w3.org/ns/prov#wasDerivedFrom
prov:wasDerivedFrom rdf:type owl:AnnotationProperty .

###  http://www.w3.org/ns/prov#wasGeneratedBy
prov:wasGeneratedBy rdf:type owl:AnnotationProperty .

###  http://www.w3.org/ns/prov#wasInfluencedBy
prov:wasInfluencedBy rdf:type owl:AnnotationProperty .

###  http://www.w3.org/ns/prov#wasInvalidatedBy
prov:wasInvalidatedBy rdf:type owl:AnnotationProperty .

# Classes

###  https://promsns.org/def/decprov#DecisionMaking
decprov:DecisionMaking rdf:type owl:Class .

# Individuals
### http://github.com/nicholascar/phd/ch01/experiment-topic-progression#ExperimentTopicsAfterLiteratureReview

<http://github.com/nicholascar/phd/ch01/experiment-topic-progression#ExperimentTopicsAfterLiteratureReview> rdf:type owl:NamedIndividual ,
prov:Collection;

rdfs:label "Experiment Topics After Literature Review"@en ;
prov:hadMember :BetterDecisionTheoryUse ,
:BetterIrrigationScienceUse ,
:Customisation ,
:NewDssTools ,
:UserDefinedData .

### http://github.com/nicholascar/phd/ch01/experiment-topic-progression#FinalExperimentTopics

<http://github.com/nicholascar/phd/ch01/experiment-topic-progression#FinalExperimentTopics> rdf:type owl:NamedIndividual ,
prov:Collection ;

"Final Experiment Topics"@en ;
prov:hadMember :BetterDecisionTheoryUse ,
:Customisation2 ,
:DssAdoptionAnalysis ,
:EmpiricalDss,
:NewDssTools.

### http://github.com/nicholascar/phd/ch01/experiment-topic-progression#InitialExperimentTopics
<http://github.com/nicholascar/phd/ch01/experiment-topic-progression#InitialExperimentTopics>
rdf:type owl:NamedIndividual,
prov:Collection;

rdfs:label "Initial Experiment Topics"@en;
prov:hadMember :EmpiricalDss,
:FacilitativeVDirective,
:InteroperabilityAndDiscovery,
:UserDefinedInput,

### http://github.com/nicholascar/phd/ch01/experiment-topic-progression#PhdDocument
<http://github.com/nicholascar/phd/ch01/experiment-topic-progression#PhdDocument> rdf:type owl:NamedIndividual,
prov:Collection;

rdfs:label "PhD Document"@en;
prov:hadMember :Ch3,
rdfs:label "Chapter 3"@en ;
prov:wasInfluencedBy :NewDssTools .

###  http://github.com/nicholascar/phd/ch01/experiment-topic-progression-detailed#Ch4
:Ch4 rdf:type owl:NamedIndividual ,
   prov:Entity ;
   rdfs:label "Chapter 4"@en ;
   prov:wasInfluencedBy :NewDssTools .

###  http://github.com/nicholascar/phd/ch01/experiment-topic-progression-detailed#Ch5
:Ch5 rdf:type owl:NamedIndividual ,
   prov:Entity ;
   rdfs:label "Chapter 5"@en ;
   prov:wasInfluencedBy :DssAdoptionAnalysis .

###  http://github.com/nicholascar/phd/ch01/experiment-topic-progression-detailed#Ch6
:Ch6 rdf:type owl:NamedIndividual ,
   prov:Entity ;
   rdfs:label "Chapter 6"@en ;
   prov:wasInfluencedBy :Customisation2 , :NewDssTools , :UserDefinedData .

###  http://github.com/nicholascar/phd/ch01/experiment-topic-progression-detailed#Ch7
:Ch7 rdf:type owl:NamedIndividual ,
   prov:Entity ;
   rdfs:label "Chapter 7"@en ;
   prov:wasInfluencedBy :EmpiricalDss ;
   :BetterDecisionTheoryUse .
### http://github.com/nicholascar/phd/ch01/experiment-topic-progression-detailed#Customisation
:Customisation rdf:type owl:NamedIndividual ,
        prov:Entity ;
    rdfs:label "Customisation"@en ;
    prov:wasDerivedFrom :UserDefinedInput ,

### http://github.com/nicholascar/phd/ch01/experiment-topic-progression-detailed#Customisation2
:Customisation2 rdf:type owl:NamedIndividual ,
        prov:Entity ;
    rdfs:label "Customisation 2"@en ;
    prov:wasDerivedFrom :UserDefinedData ;
    prov:wasGeneratedBy :DecidedCandidateNotSkilledInTopic ;
    prov:wasRevisionOf :Customisation .

### http://github.com/nicholascar/phd/ch01/experiment-topic-progression-detailed#DecidedCandidateNotSkilledInTopic
:DecidedCandidateNotSkilledInTopic rdf:type owl:NamedIndividual ,
        decprov:DecisionMaking ;
    rdfs:label "Decided Candidate Not Skilled In Topic"@en .

### http://github.com/nicholascar/phd/ch01/experiment-topic-progression-detailed#DecidedGatheringResponsesOnerous
:DecidedGatheringResponsesOnerous rdf:type owl:NamedIndividual ,
        decprov:DecisionMaking ;
    rdfs:label "Decided Gathering Responses Onerous"@en .
### http://github.com/nicholascar/phd/ch01/experiment-topic-progression-detailed#DecidedToDirectlyTestNewTech
:DecidedToDirectlyTestNewTech rdf:type owl:NamedIndividual ,
decprov:DecisionMaking ;
rdfs:label "Decided To Directly Test New Tech"@en .

### http://github.com/nicholascar/phd/ch01/experiment-topic-progression-detailed#DecidedToIncreaseNaturalSciencesWork
:DecidedToIncreaseNaturalSciencesWork rdf:type owl:NamedIndividual ,
decprov:DecisionMaking ;
rdfs:label "Decided To Increase Natural Sciences Work"@en .

### http://github.com/nicholascar/phd/ch01/experiment-topic-progression-detailed#DecidedToInvestigateDssOrigins
:DecidedToInvestigateDssOrigins rdf:type owl:NamedIndividual ,
decprov:DecisionMaking ;
rdfs:label "Decided To Investigate DSS Origins"@en .

### http://github.com/nicholascar/phd/ch01/experiment-topic-progression-detailed#DecidedToRelateLessonsFromIrriSatsms
:DecidedToRelateLessonsFromIrriSatsms rdf:type owl:NamedIndividual ,
decprov:DecisionMaking ;
prov:wasInformedBy :DecidedCandidateNotSkilledInTopic ;
rdfs:label "Decided To Relate Lessons From IrriSatSMS"@en .

### http://github.com/nicholascar/phd/ch01/experiment-topic-progression-detailed#DecisionTheoryUse
:DecisionTheoryUse rdf:type owl:NamedIndividual ,
   prov:Entity ;
 rdfs:label "Decision Theory Use"@en .

###  http://github.com/nicholascar/phd/ch01/experiment-topic-progression-detailed#DssAdoptionAnalysis
:DssAdoptionAnalysis rdf:type owl:NamedIndividual ,
   prov:Entity ;
 rdfs:label "DSS Adoption Analysis"@en ;
 prov:wasGeneratedBy :DecidedToRelateLessonsFromIrriSatsms .

###  http://github.com/nicholascar/phd/ch01/experiment-topic-progression-detailed#EmpiricalDss
:EmpiricalDss rdf:type owl:NamedIndividual ,
   prov:Entity ;
 rdfs:label "Empirical DSS"@en .

###  http://github.com/nicholascar/phd/ch01/experiment-topic-progression-detailed#FacilitativeVDirective
:FacilitativeVDirective rdf:type owl:NamedIndividual ,
   prov:Entity ;
 rdfs:label "Facilitative v. Directive"@en ;
 prov:wasInvalidatedBy :DecidedGatheringResponsesOnerous .

###  http://github.com/nicholascar/phd/ch01/experiment-topic-progression-detailed#InteroperabilityAndDiscovery
:InteroperabilityAndDiscovery rdf:type owl:NamedIndividual ,
   prov:Entity ;
 rdfs:label "Interoperability & Discovery"@en .
### http://github.com/nicholascar/phd/ch01/experiment-topic-progression-detailed#NewDssTools
:NewDssTools rdf:type owl:NamedIndividual ,
    prov:Entity ;
    rdfs:label "New DSS Tools"@en ;
    prov:wasGeneratedBy :DecidedToDirectlyTestNewTech .

### http://github.com/nicholascar/phd/ch01/experiment-topic-progression-detailed#UserDefinedData
:UserDefinedData rdf:type owl:NamedIndividual ,
    prov:Entity ;
    rdfs:label "User-Defined Data"@en ;
    prov:wasDerivedFrom :InteroperabilityAndDiscovery ,
    :UserDefinedInput .

### http://github.com/nicholascar/phd/ch01/experiment-topic-progression-detailed#UserDefinedInput
:UserDefinedInput rdf:type owl:NamedIndividual ,
    prov:Entity ;
    rdfs:label "User-Defined Input"@en .

### http://github.com/nicholascar/phd/ch01/experiment-topic-progression-detailed#UserDefinedRules
:UserDefinedRules rdf:type owl:NamedIndividual ,
    prov:Entity ;

### Generated by the OWL API (version 4.2.8.20170104-2310) https://github.com/owlcs/owlapi
Minerva Access is the Institutional Repository of The University of Melbourne

Author/s:
Car, Nicholas John

Title:
Irrigation informatics: addressing poor irrigation decision support system uptake

Date:
2018

Persistent Link:
http://hdl.handle.net/11343/228849

File Description:
Final thesis file

Terms and Conditions:
Terms and Conditions: Copyright in works deposited in Minerva Access is retained by the copyright owner. The work may not be altered without permission from the copyright owner. Readers may only download, print and save electronic copies of whole works for their own personal non-commercial use. Any use that exceeds these limits requires permission from the copyright owner. Attribution is essential when quoting or paraphrasing from these works.