Towards a New MODDE of Improving Legal Decision Support Design

TUNDE MEIKLE
University of Ballarat/Australian Catholic University, Ballarat, Victoria, Australia. E-mail: t.meikle@aquinas.acu.edu.au

ABSTRACT This paper contributes to the ebb and flow of innovations in decision support system design by presenting the MODDE methodology (Model Of Decision support System Design and Evaluation). It utilizes a simple conceptualisation of the software engineering process, and aims to provide computer consultants with guidance to the way they may include consideration of three qualitative dimensions of high level autonomous decision making. These three dimensions—discretion, consistency and resolution—have arisen directly from work in an applied legal domain. MODDE facilitates systematic attention to these dimensions in a way not previously done and thus complements existing software and requirements engineering practices.

Introduction

It appears that the tide to sweep legal decision-makers into the electronic age moves inexorably onwards (Susskind, 2000). A sign of this is the swelling attention to the use of computers and, by aggregation, the Internet, in the service of justice. The ebb and flow of progress in decision support systems also continues. At times it is buoyed by a new technical or philosophical idea, while at other times progress is becalmed for the want of an elusive solution to a critical element.

This paper contributes to that ebb and flow. It outlines the development of a methodology for advancing the design of decision support systems developed to assist legal decision-makers. It is important to emphasise that it is assumed here that not all legal decision-makers are lawyers. In fact people may make substantively legal decisions with little or no professional legal training. This is the case, for example, in the applied domain of this project.

The specific innovation presented here is MODDE (Model Of Decision support System Design and Evaluation). It utilizes a simple conceptualisation of the software engineering process, and aims to provide engineers with guidance to the way they may include consideration of three qualitative dimensions of the decision-making environment. These three dimensions have arisen directly from work in an applied legal domain. MODDE facilitates systematic attention to these dimensions in a way not done previously and thus complements existing software and requirements engineering practices.

Being informed by various disciplines such as, but not limited to cognitive and social psychology, software engineering, jurisprudence, semiotics and evalu-
ation, the strategies employed by MODDE reflect different perspectives. It is anticipated that this rich view will result in an increase in the fit of software to the user and their environment. Evaluation of the MODDE framework is a continuing process with feedback being gained from legal domain experts about the relevance of factors and the indicators used to characterize decision making in the domain.

This paper begins by placing the applied domain of this project into its legal context. A description of the inspiration for this project and an explanation of the systems approach and the qualitative perspective taken in this project follow. After that is a summary of current software and requirements engineering concepts and a description of the complementary place of MODDE. An outline of the development and structure of MODDE follows this. In the final section are suggestions for the future of MODDE.

**Legal decision-making contexts**

Legal decision making proceeds in many environments, each with their own characteristics. These may be broadly simplified here to be judicial, adversarial, inquisitorial or administrative environments. They vary in terms of the parties involved and the processes used in the making of legal decisions, but in each case legal decision making results in a chain of legal argument in support of a particular outcome. Not every setting requires the same transparency of this chain of argument (also referred to as the reasons for the decision); often they are largely left implicit.

In judicial environments the decision-maker (or there may be more than one) is tasked with formulating a summative decision upon the hearing of a range of information. Depending on the court and the audience, that decision will have a greater or lesser impact beyond the parties immediately involved. At the highest level, these decisions may have the power to influence all others thereafter.

An adversarial legal environment may be usefully contrasted with the inquisitorial one in ways apart from the cultural conventions of the continents in which one method is preferred to the other. Most importantly, for the purpose of the nature of the legal decision-making task, these environments differ such that in the adversarial situation each party must prepare their legal argument with a view to the actual challenge they may face from their adversary. They may be very good at this anticipation or they may be very poor. In the inquisitorial setting, however, although the legal decision-maker also seeks the truth from its inquiry, it does not do so by the process of legal thrust and parry in the adversarial way. It could be said however that in an inquisitorial setting there might exist an imagined adversary or challenger that has the potential to take a contrary view. This challenger role could be even taken by a potential appellate court, and is an important construct that accounts for the mindfulness of legal decision-makers to create sound arguments supporting their outcomes even in an inquisitorial setting.

In the administrative environment, legal decision making may appear much less involved with the creation of an original chain of legal argument and more with the application of and navigation through existing legal provisions for a presenting case. The decisions in this environment have been among the first to be automated as they tend to be dominated by well-articulated rules.
Inspiration for MODDE

The idea to develop the methodology described in MODDE arose from an ongoing project with the Refugee Review Tribunal (Victoria) (RRT). That project aimed to initially explore the feasibility of using an argumentation structure based on that devised by Stephen Toulmin (1958) to model legal argument in what is usually described as a highly discretionary domain. It has resulted in the development of a version of a software shell called EMBRACE (Yearwood & Stranieri, 1999).

The cooperation was gained of individuals that possessed a high degree of expertise to act in the role of domain expert. From the domain experts has come considerable information about the declarative, procedural and tacit knowledge with which they make decisions. Mostly declarative aspects are used in the knowledge base of EMBRACE and have formed the basis for the knowledge-in-the-machine. The implementation has been dependent on appropriate technical methodologies.

Of parallel interest to the work on EMBRACE as a test of the applicability of a theoretical tool, is the establishment of fit between the expected users, their environment and the software. Santos & Caravalho (1998), for example, encourage a view that computer-based systems ought to support many aspects of work including the structural, social, political and symbolic. Therefore our domain experts have provided further information on procedural and tacit aspects of their decision making. These aspects form the areas of interest dealt with by MODDE, the implementation of which is dependent on software engineering methodology related to fit.

It is from this close collaboration that three significant dimensions have become clear requirements of the software. They are the preservation of discretion, the maintenance of consistency and the provision of information in an appropriate way that is called resolution in MODDE (also described in Meikle & Yearwood, 2001). It is not suggested here that these three dimensions are the only important ones that may exist. Though they are the ones raised by experts in our specific application domain and therefore form the major focus of this work.

Fit is both a matter of intention and evaluation. Evaluation of fit can be achieved through examining user satisfaction in its many permutations (Hall & Zelaznikow, 2001). Intentional fit is achieved through an appropriate software engineering process involving thorough investigation and understanding of the domain, generation of effective software requirements, and creation of a suitable software solution—all within a professional approach that permits responsiveness to changing circumstances and is primarily user-focused (for some interesting viewpoints, see Damodaran, 1996; Kaulio & Karlsson, 1998; Tata, 1998).

Given that software engineering is reasonably sophisticated in its ability to borrow investigatory and confirmatory techniques from other disciplines it is not entirely a new endeavour. It could thus be expected to have taken some account of the more fuzzy aspects of human decision making as well as the more concrete ones that relate to declarative knowledge. However, it appears that articulating tacit aspects of decision making has remained a challenge to software engineers. MODDE aims to go some way towards assisting with this.
Methods of investigation

In a complex domain with both human and organisational elements, a technique that permits attention to many facets is needed. Accordingly an ecological systems approach (Bronfenbrenner, 1979) has been adapted to focus attention on the multiple layers and players in the environment. It forms part of the underlying conceptual framework of MODDE. The levels of the ecological approach cover the aspects to which software system designs should be responsive. In the context of computer-based decision support, the ecological levels reflect the individual, the integrated-functional, the organisational and the societal layers of the software user’s environment. This approach invites systematic attention to the multiple interrelationships that exist in the application domain. Thus it is a convenient and thorough way to begin understanding the decision-making environment for which one is creating software.

Only one level will be described here, that which focuses upon the individual, their processes and experiences of decision making. A treatment of the other levels is beyond the scope of this paper. The individual level is however sufficient to highlight how MODDE aims to complement existing practices, and permits a special focus on the phenomenology of decision making integral to the design of computational support systems.

Various inquiry techniques have been used with members in our domain to gain and validate information and impressions. These include collaborative reflection where professional expertise experience and insight is shared, compared and critiqued; observation and discussion where work, documents, environment and decision-making steps are demonstrated and made available for review; and collegiate critical development of concepts where propositions are presented, applied to observations, critically reviewed and modified.

Such depth research can be a process of discovery (Kleining & Witt, 2001) for both the researcher and the researched, and as a result insights and changes in beliefs can develop in each. Therefore some variation in the sophistication of self-understanding by domain experts can be expected over the course of a project that spans years. The next section deals with the central focus of MODDE: the individual decision-maker.

At the individual level

The smallest organisational unit of decision making at the RRT is the individual—the RRT member. At the RRT decisions are made individually with each member receiving an applicant’s case papers that they then work through in their individual systematic way to reach a conclusion about the applicant’s refugee status. As the cases are merit review cases the member has the decision of the primary decision-maker before them in the papers as well as original material. They also carry out whatever research they feel is demanded by the case in order to be fully informed and to enable them to effectively justify their decision.

The document that they produce is referred to as a ‘determination’. It varies considerably in length, though has a standard structure. Work on them has recently been greatly assisted by the use of a complex template, although individuals have had simpler ones or their own for some time.

The member is obliged to put in the determination the chain of argument that supports their decision but need not include arguments and evidence that
support the contrary view. Recent legislation has strengthened the power of a
decision made by an RRT member and this has made it less vulnerable to appeal
on spurious grounds that act to interfere with, rather than facilitate natural
justice.

Appeal is, however, possible on points of law and it is with this in mind that
members take pains to include in their determination evidence that they have
considered all current legal rulings in addition to the direct facts of the case. This
mindfulness of potential challenge is what I refer to here as the ‘imagined
challenger’ as opposed to the actual real challenge of a courtroom adversary.

At the individual level, the member acts relatively autonomously, and in fact
the experience of autonomy appears to be a quality that is dearly held. From it
spring a number of other features that are intrinsic to individual decision
making and as long as the organisation maintains its procedure of individual
members making decisions, then those features need also to be maintained. They
give rise to the particular way the three dimensions of discretion, consistency and
resolution are operationalised.

The individual does not act without context however. The legal context may
be one that is inquisitorial but the organisational one is also important to
consider. A detailed treatment of the other ecological levels is beyond the scope
of this article, although mention of the effect and interrelatedness of elements at
different levels does merit mentioning as they remind the reader of the wider
influences upon the individual that can often be missed in conventional ap-
proaches that take a reduced view.

For example, the individual RRT member, while making decisions au-
tonomously, is exposed to the ideas of other decision-makers, the expectations of
applicants, and current foreign and economic policy. They all have their
influence and may impact upon the RRT member’s experience of or actual
discretion, their expectation of consistency and the informational need, or resol-
ution.

In other legal contexts the way legal decision-makers carry out their work will
also be the result of multiple influences at various ecological levels. Each context
is assumed to be unique, and different requirements of their decision support
software are likely, in relation to these three dimensions. But, in order for the
resultant software to be fully and effectively utilised by users it must conform
to their needs often in ways that are hard to articulate. These three dimensions
are just such an example and they pose a non-trivial challenge in the computing
context.

The software engineering challenge

Leach (2000) described software engineering as ‘a systematic procedure that is
used in the context of a generally accepted set of goals for the analysis, design,
implementation, testing and maintenance of software’ and ‘the software pro-
duced should be efficient, reliable, usable, modifiable, portable, testable,
reusable, maintainable, interoperable, and correct’ (p. 9).

The various tasks involved in a typical project according to Leach include:
analysis of the problem; determination of requirements; design of the software;
coding of the software solution; testing and integration of the code; installation
and delivery of the software; documentation; maintenance; quality assurance;
training; resource estimation; and project management (p. 11).
One feature that this list has in common with most guidebooks to software and requirements engineering is the absence of a direct reference to evaluation (Budgen, 1994; Davis, 1993; Ince, 1989; Jackson, 1995; Leach, 2000; Maciaszek, 2001; Robertson & Robertson, 1999). There is usually an awareness of the need for a means to assess whether the project is progressing as expected, often expressed as ‘testing’ (not necessarily the same thing as evaluation), and there is often considerable space devoted to establishing metrics for measuring this progress. Metrics are not, however, generally useful for qualitative characteristics.

The list offered by Leach is reasonably comprehensive and its elements are usually found in what is called the ‘software lifecycle’. Good examples of lifecycles can be found in Leach (2000). There is still a linear and step-wise nature to many lifecycles proposed where the elements follow one another with testing often being last. Maciaszek (2001, p. 16) presents what is a progressive view in saying, ‘in our opinion, [planning and testing] are not separate lifecycle phases because they span the whole lifecycle’. It is this latter view that is adopted in MODDE as it facilitates a means of integrated evaluation of the achievement of the qualitative aspects it elucidates by the use of a spatial representation. Also in common with Maciaszek, MODDE assumes an iterative approach to software development.

Design is, of itself, a complex process with requirements often included within it rather than as a separate task. That is the perspective taken in MODDE. Design is a wide-ranging concept that is described well by Budgen (1994, p. 5) who tells us that ‘the approach used in design ... begins by assuming the end result and then seeks ways of bringing this about’.

The major aspect of software design that is of interest here is that of requirements analysis and specification that ‘aims to determine, analyse and negotiate requirements with the customers’ (Maciaszek, 2001, p. 17). The requirements phase distinguishes between domain functions and constraints, produces documentation and underpins later phases of the development lifecycle. The first of these is dissected further.

Domain functions tend to be categorised into ‘functional’, ‘nonfunctional’ and ‘constraint’ categories. According to Robertson & Robertson (1999) functional requirements are those things that the product must do, nonfunctional requirements are the qualities the product must have, while constraints are the global issues that shape the requirements.

When considering the software being developed for the RRT the functional requirements may include a need to:

- represent refugee law accurately,
- represent reasoning and legal argument,
- assist decision making and decision formulation, and
- assist document drafting and information retrieval.

Domain knowledge is used in meeting the functional requirements of EMBRACE and has been represented using a modified Toulmin argumentation structure (TAS) and is implemented as the Generic Argument Structure (GAS) in EMBRACE. The member uses the GAS to create a specific Actual Argument Structure (AAS) for an applicant. The AAS links to a document drafter facility in EMBRACE that assists in the construction of a determination. Stranieri et al. (2000) describe the reasons for the choice of this method of knowledge
representation. Domain experts have confirmed the value of the GAS as a sound representation of the domain knowledge they use, and the GAS is continually open to review through ongoing consultation. Their feedback has also prompted modifications to the TAS.

Nonfunctional requirements would include features such as look and feel, usability, performance, operation, maintainability, security, cultural and political, and legal. On the whole they affect the appearance, ease of use and sensitivity of the software to its context. Few of these have been considered in EMBRACE at this time. Constraints such as the product purpose, being a statement about the intention of the software to solve a specific problem, or those related to a solution, such as those related to compatibility with other operating systems are also not considered in detail by EMBRACE at this time.

However, upon examining these three categories in more detail it is apparent that the dimensions revealed by domain experts at the RRT as requirements do not find an easy home. As these dimensions act at multiple levels of the domain environment, most of all at an overarching contextual level, it is proposed that an additional category—that of metarequirements—be considered (Meikle, 2001). This would place qualitative and organisational considerations related to the nature of the decision-making process into a special category when developing decision support software.

A timely warning is given by Robertson & Robertson (1999, p. 170) for the conceptualisation of metarequirements when they write, ‘if you can’t measure a requirement then it is not really a requirement’. But qualitative aspects are notoriously difficult to articulate let alone measure. MODDE does, however, provide a means of ordinarily measuring concepts in order to facilitate describing the present, the future aim and the state at any particular point, aided by the spatial representation. The following section describes other aspects of the development and structure of MODDE not already covered.

**Development and structure of MODDE**

The lifecycle of legal decision support software (DSS) is assumed to follow a simple three phase iterative process (see Figure 1). Phase A involves coming to understand the domain and its decision-makers, it is during this time that the specification of metarequirements is developed. Phase B matches the specifications to computational techniques that are appropriate to achieve those aims, it will therefore necessitate trialing, prototyping and review, and possibly a revisiting of Phase A techniques. Phase C is the longer term implementation of the DSS and includes final evaluation of how well it has delivered with respect to the three qualitative dimensions.

It is assumed that the understandings reached in Phase A may be revisited at any time for confirmation and also if a deviation is found or is suspected. Work in the field by a software engineer may have misunderstood the requirements as meant by the users, and this may need clarifying. This is part of the iterative nature of MODDE, as well as the way evaluation is integrated and ongoing throughout the software lifecycle. It is not just a summative process but instead has more in common with action research as described by Owen (1993).

Continuous monitoring ought to optimise the likelihood that the resultant system is useful for the users and that it reflects the expectations of them as decision-makers. An optimal system should assist the user to make their
decision while avoiding unintended effects upon what is meant to be normal decision-making processes. In other words, it should help not hinder. Establishing what this means in practice is not an easy task, but one that is assisted by using the protocol provided in MODDE.

The MODDE protocol takes the form of documentation that guides the software engineer to incorporate the three qualitative constructs of discretion, consistency and resolution into the DSS. Central to this is the simple conceptualisation of the three constructs.

Each is assumed to have theoretical extremes (making them dimensional). There is a neutral centre point from which the two extremes extend. Also given this means of representation, the dimensions may be depicted as axes of a three-dimensional space (see Figure 2).

This also permits the idea of plotting the position of the decision-making domain on the axes in terms of its level of attributes for the three dimensions. Comparison of that plot over time and between phases can then be made (see Figure 3).

In the case of discretion, it is possible to say theoretically that a legal decision-making context may be characterised as being at the extreme ‘strong discretion’ end, or even at the extreme ‘weak discretion’ end. The terms ‘strong’

**Figure 1.** The phases of system development assumed in MODDE.

**Figure 2.** The three MODDE dimensions permit the location of a decision-making domain in an octant.
Figure 3. The plot of a decision support system may be monitored through development phases. It should always remain similar to the position of the decision-making domain. Variations may emerge as at Phase C.

and ‘weak’ have been borrowed from Dworkin’s (1977) discussion of discretion. In a similar way it is possible that a legal decision-making context may vary from the theoretical extreme of ‘high consistency’ to the other extreme of ‘low consistency’. Resolution may also range out to the theoretical extremes of ‘high resolution’ and ‘low resolution’. To take each one briefly in turn will illustrate their operational meanings.

The strength of discretion implies something about the choices available to the decision-maker. This is determined by a number of contributing characteristics of the decision-making environment that are explored by the computer consultant when they investigate the application domain. The place of the decision-maker for which the software is intended in the organisational hierarchy and the open-textured (Hart, 1961) nature of the legislation that guides practice are examples of factors that will affect the extent of discretion that may be exercised. At the ‘strong discretion’ extreme there would be no constraining rules and maximum liberty to decide on a case in whatever way the decision-maker saw fit, there being no need to adhere to precedents. At the other extreme, there would be maximum constraints upon the choices available to the decision-maker such as might exist in a highly proscribed area of law where rules dominate and are unambiguous. Domain experts at the RRT consider their decision making as being towards the ‘stronger discretion’ of neutral, though not at the extreme.

‘Consistency’ is a more difficult concept to define in some ways. It might be assumed to always vary inversely with discretion. A moment’s reflection upon geometry will show that this is not a necessary theoretical assumption even though it may be a practical likelihood. In practice, if this is so, it may simply mean that decision-making domains will tend to cluster in the two corresponding octants. At the core of consistency in legal decision making lies the idea of ‘similar processes’. This means applying the law the same way to each case. This is where the relation to discretion is strongest. The greater the liberty (as may occur in strong discretion) the less it may be that the same process of applying the law occurs each time. This may or may not be true and has not yet been tested empirically. The RRT domain experts regard themselves as having fairly strong discretion but aim to apply the law the same way each time. This achieves a particular form of consistency of process. They regard their decision making as being characterised by fairly ‘high consistency’. The outcome of a case is
dependent on both the legal process and the facts. It appears to be unhelpful to dwell on consistency as the demonstration of same outcomes from same cases for there are no cases that are ultimately the same (Lovegrove, 1999). What can be controlled better is the process applied. Also if consistency is an aim in order to facilitate prediction of their personal outcome by a new applicant, then the more those processes are standardised and made transparent, for example, the more consistent they will be.

The third dimension, ‘resolution’, is a particularly complex concept. It revolves around the informational need of the decision-maker. It is here that the concept of ‘usability’ as explained in Barron et al. (1999) becomes relevant. Drawing on an understanding of semiotics and the assumption that language, and its aggregation as information, is not presented value-free, the meaning of information and the need of decision-makers that it meets is acknowledged as being context-specific. It is necessary to determine the illocutionary effects of machine outputs to the user as part of the consideration of fit.

One characteristic of the RRT members that springs from their legitimate autonomy is that they do not want a mechanical judge in the machine before them (see also Leith, 1998). The idea that their decision making may be directed towards one or another outcome by computational means is unwelcome, however intelligent it appears. This sentiment is echoed in research on lawyer’s opinions on artificial intelligence. In the context of the RRT then, what is required is that the informational needs are met in a timely fashion, presenting an appropriate text fragment depending on context, with an enhanced search facility and a suitable alert when information changes. The means by which this is achieved ought not to be delimited by any assumption within the machine that a particular solution is likely. The user must retain the entire locus of control over direction, even including changing the information parameters. At the ‘low resolution’ extreme, information would be presented with a very large ‘halo’ around the potential solution to the problem (or outcome to a case). At the ‘high resolution’ end, by contrast, the information presented would be tightly solution-focused, and would present only information that pertained to the optimal (probably machine-generated) solution. The RRT members consider themselves as being in a decision-making environment that is at the relatively ‘low resolution’ end.

As a result of consultation with members using a more detailed protocol than is shown here, the decision-makers in the RRT characterised their domain as having ‘strong discretion’, ‘high consistency’ and ‘low resolution’. These concepts and how these qualitative aspects are supported in the design of EMBRACE are discussed in other places (see, for example, Meikle & Yearwood, 2001).

The final construct used in MODDE that promotes greater understanding between software developers and their application domain collaborators is that of domain ideal and domain real. These have been invoked to help explain the difference between what the computer consultant may discover is the present perception of the domain characteristics that may be determined and plotted (the ‘domain real’), and the anticipated state that may also be discovered and plotted (the ‘domain ideal’). It must be something made clear contractually between the parties when developing software, just what characteristics are to be preserved as they are, and what are planned to change. This is not the same as the inevitable unintended changes that will occur purely as a result of working
on a project to develop software that necessitates an examination of what one does and what one wants. Attending to this clarifies the type and amount of change that is to be supported by the software, and may also point to non-computational interventions needed in the organisational setting to facilitate the adoption of the new software.

Final comments

MODDE is a project under development and appears to have potential beyond the legal decision-making context. Early inquiries suggest that at the most generic level, the concepts of ‘discretion’, ‘consistency’ and ‘resolution’ have relevance to high-level autonomous decision-makers in business management and medicine as well. Both these contexts are rich areas for decision support software research and development and they may benefit from the concepts in MODDE to assist with the requirements specification process, ultimately resulting in a better fit between the software, the user and their environment.

References


