Introduction

This thesis aims to give the first extended exposition of (computer-based) *modelling with definitive scripts* (MWDS), a concept first introduced by Beynon in 1983 that has been central to the work of the Empirical Modelling (EM) research group at the University of Warwick for many years. As a *treatise* on modelling with definitive scripts, the thesis will frame the concept of MWDS, discuss its background and history, introduce the tools that have been developed to support it, illustrate and document its use with reference to many diverse examples, compare and contrast MWDS with alternative approaches to modelling and programming, and identify and – to some extent – address problematic issues surrounding its application. Within the EM project, MWDS is seen as playing a central role in a radical shift in perspective on computer use. This thesis is not primarily concerned with justifying this controversial viewpoint, but with assembling and organising the practical evidence that informs it and that supplies the context for its ongoing evaluation. To date, many who practise MWDS become enthusiastic about its potential, whilst many who evaluate the claims of the practitioners without themselves gaining experience of MWDS become frustrated by the difficulty of communication. This treatise should assist those who practise MWDS to interpret and communicate their experience, and to identify issues that require further investigation and clarification. It should also assist those who are sceptical about MWDS to gain understanding or to identify specific points of concern without themselves engaging in practice.

A radical shift in perspective on computer use?

In complementary work associated with the EM project, MWDS has been linked with three related issues that are topical in connection with recent trends in computer use:

- achieving a closer integration between human activity and computer-based technology [Rasme02, Maad02];
- understanding the ways in which the computer and computer-related technologies can serve as physical artefacts [Ness97, Car98, Maad02];
- promoting open-ended and empirically-led development of complex computer-based systems [Ness97, Sun99, Chen02].

The importance of closer integration between human activity and computer-based technology stems from the fact that both expert and novice users are using computers to overcome specific problems and efficiently handle their work in diverse areas. The significance of this integration becomes prominent in areas such as business, where computer technology has a major organisational and social impact, and education, where the role of computer technology in supporting cognitive tasks and learning has yet to be fully understood. The real and potential impact of the computer upon human behaviour, at both individual and social levels, could hardly have been envisaged in the early days of computing.

The need to understand the ways in which the computer and computer-related technologies can serve as physical artefacts stems from the fact that the experiential aspect of computer use is becoming ever more important in interaction with machines. The computer can generate virtual worlds, process music and pictures, and support concurrent interactive activity involving sensory data drawn from the environment and the users themselves. These 'very high-level' activities are quite different in character from the batch processing of textual and numerical input of the early days of computing, where calculations and symbolic manipulations were dominant.

The problems of developing complex large-scale computer-based systems are well-recognised (cf. [Brook95]). Recognition of the role that empirical elements play in complex system development, the need for user involvement in systems requirements capture and design, and the high profile of experiential concerns in modern applications has led to approaches outside the scope of what Terry Winograd identifies in [WF93] as the 'rationalistic tradition':

> [T]he 'rationalistic orientation' … can be depicted in a series of steps:
> 1. Characterise the situation in terms of identifiable objects with well-defined properties.
> 2. Find general rules that apply to the situations in terms of those objects and properties.
> 3. Apply the rules logically to the situation of concern, drawing conclusions about what should be done.

Despite this, it has proved hard to devise approaches to system development that meet Brooks's essential concern for 'conceptual integrity' [Brook95]. For instance, the extent to which object-oriented analysis and object-oriented design can be effectively integrated remains controversial [Kain99].

In the EM framework within which MWDS has been developed, the difficulty of addressing the three issues identified above is (controversially) seen as stemming from the limited support for general computer use that is offered by the classical theory of computation [Turing36, HU79] (cf. Wegner's contention that the technology shift in computing demands a paradigm shift from algorithms to interaction [Weg97]). Our specific reason for questioning the generality of the classical theory of computation is that programs, as classically conceived, necessarily have an intrinsic behavioural specification (whether or not such a specification is made explicit prior to implementation) and are interpreted according to conventions about how variables and states are connected with the external world that are only negotiable prior to execution. These features of the classical conception of a computer program limit the conceptual support that the classical theory of computation can give to integrating human and computing activities and towards treating the computer as a physical artefact. Because they are framed by their preconceived and circumscribed characteristics, classical programs are typically difficult to interpret in relation to their external context at intermediate states of execution. In this respect, they are unlike physical artefacts whose states can be continuously monitored and experienced – a fact that inhibits their adaptation to changes in their environment and use.

MWDS can be regarded as a radical generalisation of the spreadsheet principle. The spreadsheet provides the user with direct access for inspecting and manipulating computer state. It also allows the user to represent his/her observations directly by making use of dependency, and to construct a model with visual and interactive support from the spreadsheet throughout its construction. A spreadsheet can be conceptualised as a classical program by focusing on specifying the behaviours that maintain dependencies between cells, but the distinctive characteristics of a spreadsheet are not adequately expressed in this way. In interpreting spreadsheet use, it is more appropriate to view the spreadsheet as an artefact with which the modeller can interact in an open-ended experimental manner to incrementally embody his/her understanding of an external situation. This puts the emphasis on how dependencies amongst observables external to the spreadsheet are captured by definitions that relate its cells, and on how the evolving relationship between the spreadsheet and the external situation that it represents is experientially mediated state-by-state by the modeller. Viewed from this perspective, the spreadsheet is better suited to integrating cognitive and situational concerns with computer activity than when viewed as a classical program. The user can interact with the spreadsheet with some understanding of how its internal model operates, of how it applies to a particular situation, and of its limitations. The sharp contrast between these two perspectives on a spreadsheet is highlighted by the fact that the dependency maintenance that dominates the classical computational account is 'abstracted away' in modelling the spreadsheet through its dependencies, and that the 'what-if' interaction so essential to the spreadsheet concept is quite unlike classical program 'hacking'.

A full discussion of the potential role that MWDS can play in addressing the three issues identified above is beyond the scope of this thesis. It is clear that the exploitation of MWDS in realistic applications will involve the essential use of other more well-established programming and management techniques. The 'pure' use of MWDS that is illustrated in this thesis is unrepresentative of what would be appropriate in such practical applications, and only hints at the issues that are raised by concurrent and distributed use [ABCY94], and by translation and optimisation [ABCY98]. Though the discussion in the thesis returns to the above three themes at several points, this is by way of exploring – rather than establishing – the credentials of MWDS as an alternative paradigm for computer-based modelling that can do more justice to contemporary computing practice.

Introducing modelling with definitive scripts

Orientation

MWDS is similar in nature to familiar forms of model-building, not necessarily computer-based. These relate to ways in which people understand and investigate phenomena and situations based on what they observe and their implicit knowledge. Typically, they construct a physical model or artefact[1] to represent what they observe, perceive and conceive in the referent[2]. The experience of observing and interacting with the model is designed to invoke the experience of observing and

---

[1] An *artefact* is something viewed as a product of human conception or agency. [DictWeb, source: The American Heritage® Dictionary of the English Language, 4th Copyright © 2000 by Houghton Mifflin Company]. Throughout this thesis, the term 'artefact' is specifically used to designate artefacts that serve a representational purpose and that are physical rather than abstract in nature.

[2] A *referent* is something referred to; the object of a reference [DictWeb, source: WordNet ® 1.6, © 1997 Princeton University].

interacting with its referent, so that the model is interpreted as embodying the modeller's knowledge of the referent. For instance:

- as an experimental scientist studying the phenomenon of electromagnetism, Faraday constructed physical models to represent the way in which the magnetic field was disposed in the neighbourhood of a wire carrying an electric current;
- an engineer, when building a bridge, may make a scale model of the bridge or prototype components needed in its construction, so that he/she can evaluate its design before actually building it;
- a manager, when studying the financial position of his/her company, sets up a spreadsheet in which he/she can simulate the possible outcomes by varying predefined parameters.

In all the above examples of representational artefacts, interacting with the artefact can be viewed as 'imitating interactions with its referent'. A feature of these activities is the monitoring and interpreting of the relationships amongst observables[3] in the artefact and those amongst their counterparts in the referent. In the above examples, Faraday's construction, the model of the bridge and the spreadsheet can be viewed as artefacts that respectively capture the characteristics of electromagnetic phenomenon, the features of the actual bridge and the financial state of the company, as observed by the model builders.

Using interactive artefacts to serve a representational purpose in this way is topical in respect of several contemporary research themes. A full discussion of these themes is beyond the scope of this thesis, but their relevance is noted at several points throughout. They include: the relationship between the experiential/experimental and the logical/theoretical (cf. Gooding [Good90], James [James96] and Sections 2.2, 3.1 and 5.1) and the role of the physical world and embodiment in human interaction and communication (cf. Turner [Turn96], Lakoff [Lakoff88] and Sections 3.1 and 5.1). The specific focus of the thesis is on the practical construction and interpretation of computer-based models that (whilst they relate to these themes) are primarily of interest to us in relation to the semantics of agency and language in computer science.

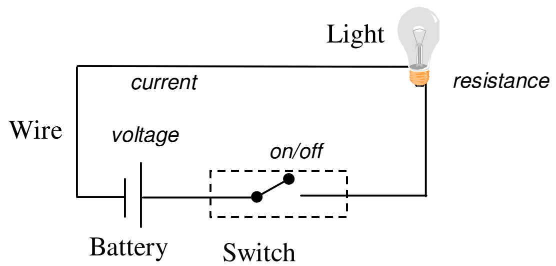
The use of interactive artefacts, as opposed to symbolic mathematical and textual models, for representation has some distinctive characteristics which stem from the fact that the relationship between the artefact and its referent is experientially mediated state-by-state. The

[3] An *observable* is a feature of a referent or an artefact to which an identity and current value can be ascribed. Throughout this thesis, the term 'observable' is used in a broad sense that embraces what can be directly observed by the human senses, scientific observables, and observables of an abstract nature (cf. Section 5.1).

modeller's interpretation can be subjective. It can also depend on context and be subject to change.

The concept of a definitive script

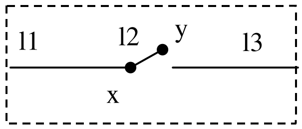
In all three examples of model-building discussed above, a key feature in interacting both with the artefact and its referent is changing the value of one observable, and then observing the consequent changes to other observables (cf. updating a spreadsheet cell, or changing the current in the wire). The relation between the change to one observable and the consequent changes to other observables can be construed as cause-and-effect. The connection between the artefact and the referent is established through the realisation of the similarity between the cause-and-effect relationships in the artefact and the referent. This realisation is only possible if the association between change and consequence is reliable and timely – if not 'immediate'. In order to relate interaction with the artefact and interaction with the referent in this way, it is also important to know which of the observables it is sensible to change directly. For instance, in the manager's spreadsheet, the way in which the profit is computed cannot be changed.




```
light is if battery == OK and switch == ON then ON else OFF
switch is OFF
battery is if voltage >= 1 then OK else NOT OK
current is if switch == ON then voltage/resistance else 0
voltage is 2
resistance is 4
```

Figure 1: A definitive script for a simple electrical circuit

MWDS is based on creating a computer-based artefact that embodies particular cause-and-effect relationships. The principle (to be developed less informally later in the thesis) is illustrated in Figure 1. The set of definitions in Figure 1 (a 'definitive script') records the cause-and-effect relationships (the 'dependencies') between the observables in a simple electrical circuit. The relationships between the values of observables are maintained in a similar way to that used in a spreadsheet. To some extent, the script already serves as an interactive artefact. An intelligent person interacting with the script will be able to recognise at once that as the value of switch is OFF, the value of light is also OFF. He/she will also know that (with the current value of voltage) setting the value of switch to ON also sets the value of light to ON.




```
y is if switch == ON then x+(1,0) else x+(√3/2, 1/2)
l1 is line((0, 0), x)
l2 is line(x, y)
l3 is …
x is (4, 0)
```

Figure 2: Visualisation for a switch

In order to improve the quality of the script as an interactive artefact, the visual representation of the circuit that is depicted in Figure 1 can be attached to the script. The principle behind such visual representation is illustrated in Figure 2, which shows how the disposition of the three lines representing the switch in Figure 1 can be defined in terms of the variable switch. Computer support for MWDS in this way can be given by a system that maintains the dependency between the values of variables as recorded in a script (cf. Figure 1) and as shown in a visual display (cf. Figure 2).

A definitive script focuses on direct representation of what is perceived or observed (based on relationships) on computers. Its emphasis is on capturing state as perceived and experienced, unlike other programming constructs, which tend to describe sophisticated abstract mechanisms to prescribe machine behaviour. In MWDS, we view and interpret the computer-and-definitive-script as a computer-based interactive artefact.

Open development and closed-world paradigms

MWDS offers a rich framework to support and facilitate the modelling activity that is needed for developing a computer artefact to imitate an external referent. A definitive script represents a particular physical state of an artefact in which the potential transitions are performed through redefinitions. The values and the definition of observables can be changed and redefined at any time to reflect the change in the modeller's perspective or the external situation. Subject to maintaining the semantic connection between the artefact and its referent, the interpretation of a definitive script is open-ended. For instance, the visual representation for the switch as depicted in Figure 2 can be readily reinterpreted as a door (cf. Figure 2-6) or a valve (cf. Figure 3-1).

In contrast to traditional 'closed-world' computer representations whose interpretation is unambiguous and objective, MWDS can be seen as supporting the concept of 'open development', in a sense that is explained by Brödner in [Brö95]:

> "…, the 'closed world' paradigm, suggests that all real-world phenomena, the properties and relations of its objects, can ultimately, and at least in principle, be transformed by human cognition into objectified, explicitly stated, propositional knowledge."
>
> "…, the 'open development' paradigm, does not deny the fundamental human ability to form explicit, conceptual, and propositional knowledge, but it contests the completeness of this knowledge. In contrast, it assumes the primary existence of practical experience, a body of tacit knowledge grown with a person's acting in the world." [Brö95]

This distinction between closed-world and open-development paradigms is reflected in two different ways in which dependency has been used in computing applications: closed-world dependency (associated with e.g. declarative constraints) and open-development dependency (associated with definitive scripts).

Context for the thesis

Open-development dependency in applications

A spreadsheet is one of many applications that make use of open-development dependency. The definition of cells by formulae, and the instant automatic recalculation of values based on these formulae, is one form of a definitive script. As in MWDS, a spreadsheet provides an interactive and experiment-based environment in which a modeller can play a role of a designer (to frame a definition) and of a user (to interact with the sheet) at the same time. The interactive and exploratory features (i.e. 'what-if') of a spreadsheet are similar to the concept of 'redefinition' in the definitive approach. Many researchers [BP93, Nar95, Green00] have studied the qualities of the spreadsheet as an artefact, which include powerful features for direct manipulation and their potential use as a basis for an alternative programming culture.

Other computing applications featuring 'definitive principles' (cf. Section 2.1.1) have also been developed. G. Wyvill [Wyvil74] and B. Wyvill [Wyv75] both developed languages for computer-based graphical modelling, which had some characteristics in common with MWDS. Todd's Information System Base Language (ISBL) [Todd76], based on Codd's relational model [Codd70], is another example of an application based on definitive principles. Similar principles are exploited in various software applications such as Agentsheets [AgentWeb], MSWord, Makefile in UNIX and OLE[4].

The Empirical Modelling (EM) project at Warwick

Ongoing research associated with the Empirical Modelling project at Warwick has explored the use of a definitive script as a fundamental means to exploit the computer in modelling and programming. EM is based on three principal concepts – observation, dependency and agency – that are closely related to MWDS. A number of notations and tools have been developed to support MWDS and dependency maintenance in constructing computer-based artefacts. These include the Eden interpreter, first developed by Y. W. Yung in 1987 [YY88, Yung90] and subsequently extended by Y. P. Yung, P-H Sun and A. Ward, and the abstract definitive machine (adm) [Slade90]. The diverse definitive notations that have been developed to serve various representational purposes (cf. Figure 1 and Figure 2) include ARCA [Bey83], Eden [Yung90], DoNaLD [BABH86], Scout [Yung92], Eddi [Tru95] and Sasami [Carter99].

This thesis focuses on the study of a wide range of models and applications that have been developed by several hundred undergraduate and postgraduate students using the above tools and notations. The research draws on about a hundred publications (papers, research reports and theses) by members of the EM research group [EMWeb], and many project reports (cf. References in Appendix B).

Timeliness of the research

The development of MWDS has now reached the stage at which there is enough material for it to be appropriate to gather and review the essential characteristics of the approach. The scope and character of the definitive models that have been developed so far presents many difficulties in classifying, organising and maintaining them. By way of illustration (cf. Figures 3-17 and 5-2), the Digital Watch model {Digital92} – now comprising some 800 definitions distributed over about 6 files in each variant – was originally developed by Beynon in 1992 but only partially completed. The button interface to the digital watch was added, and an analogue clock and a chess clock were integrated into the model by Cartwright in 1995 [BC95]. The additional functionality required to complement the restricted model of state supplied by the statechart was added by Fischer in 2000 [FB01]. A variant of the model, based on his actual digital watch and

[4] Object Linking and Embedding

adapting the statechart to his mental model, was developed by Roe in 2001 as an educational case study [BRWW01]. The development of an alternative model of the digital watch statechart was the focus of a final year project by Carpenter in 1996 in which he implemented an Eden application with a graphical user interface to specify and animate statecharts [Carpen95]. The Digital Watch model was also used as a basic case study relating to work of Horrocks [Horr99] on statecharts in user interface design [BCSW99]. A deep understanding of the principles behind MWDS clarifies the reason why a single model can evolve in such diverse directions in this way. It can also help to promote further development and suggest ways of making the management, organisation and documentation of such model more coherent.

There are other indications that definitive principles are finding practical applications. The work of Nardi [Nar95] on spreadsheets as a framework for end-user computing and that of Repenning on Agentsheets [AgentWeb] point to a potential paradigm shift in software development. The Java maintainer API (JaM2), developed by Richard Cartwright first at Warwick [Car98] and subsequently at the BBC, has been successfully applied to address challenging applications in broadcasting. It is also the basis for Empirical HyperFun geometric modelling tool [HyFunWeb]. Practical applications of definitive models recently developed at Warwick include the Temposcope [BWM+00] (see Chapter 4) and SQL-Eddi, a tool for teaching the principles of relational database query languages [EMWeb].

Content of the thesis

Objective of the thesis

The objective of the thesis is to compile the first treatise on modelling with definitive scripts. As stated in Merriam-Webster's dictionary [DictWeb]:

> "A treatise is a systematic exposition or argument in writing including a methodological discussion of the facts and principles involved and conclusions reached."

The facts about MWDS (cf. Chapters 1 and 2) relate to its history and current status where the development of tools, notations and models is concerned. They also involve situating MWDS in relation to other approaches to modelling with which it shares common features such as an emphasis on dependency or experiential aspects of state.

The principles of MWDS concern the way in which the modelling works, and the characteristics of both the modelling activity (cf. Chapter 4) and the models themselves (cf.

Chapters 2 and 3). The subtle connection and stark contrast between these principles and existing established principles of computer-based modelling that are used to support classical programming is elaborated.

The conclusions reached concern an evaluation of the progress that has been made in clarifying and exploiting the characteristic qualities of MWDS, the identification of its current limitations and possible future directions for development.

Methodology for the discussion in the treatise

The main motivation for the work in this thesis is to collate evidence that relates to a proposed radical shift in perspective on computer use and science. The thesis has been conceived as a systematic staged exposition of the evidence for MWDS as the basis for such a shift. The alternative stance on computer science that we have in mind is in the spirit of Gooding [Good90]:

> "Scientists' descriptions of nature result from two sorts of encounter: they interact with each other and with nature. Philosophy of science has, by and large, failed to give an account of either sort of interaction. Philosophers typically imagine that scientists observe, theorize and experiment in order to produce general knowledge of natural laws, knowledge which can be applied to generate new theories and technologies. This view bifurcates the scientist's world into an empirical world of pre-articulate experience and know-how and another world of talk, thought and argument. Most received philosophies of science focus so exclusively on the literary world of representations that they cannot begin to address the philosophical problems arising from the interaction of these worlds: empirical access as a source of knowledge, meaning and reference, and of course, realism." (p. xi)

Gooding's reappraisal of the relationship between the 'empirical world of pre-articulate experience' and the 'world of talk, thought and argument' is a controversial theme inconsistent with the more mainstream view of philosophers such as Karl Popper [Popp59]. A similar reappraisal – associated with controversy with a similar status – is represented in the work of James [James96] (cf. the view of James's critic F. H. Bradley [Brad30]), Turner [Turn96] (cf. the treatment of metaphor by Gentner and Forbus [MetapWeb]) and Cantwell-Smith [Cant96] (cf. the logicist debate in AI [McDer87]).

MWDS shares with Gooding's work a practical orientation towards relating the empirical to the linguistic. As Gooding writes in his preface [Good90]:

> "My approach to the practice of observational science involved repeating observation and experiments to recover the contingency and uncertainty of real experiment, and to bring out the tacit knowledge and skill concealed behind the laboratory records." (p. xiv)

Our exposition in this treatise on MWDS has been framed around the key issues raised in Gooding's agenda as set out above:

Chapter 1 contrasts two kinds of representation, associated with closed-world and open-development perspectives, that – in our view – are respectively characteristic of Gooding's literary and empirical worlds.

Chapter 2 discusses the fundamental role that MWDS can play in the construction of computer-based artefacts that serve a similar sense-making function to what Gooding [Good90] identifies as *construals* of an observer's personal experience:

> "Construals are a means of interpreting unfamiliar experience and communicating one's trial interpretations. Construals are practical, situational and often concrete. They belong to the pre-verbal context of ostensive practices." (p. 23)

In keeping with this characterisation, the use of MWDS for construal is identified with 'single-agent' modelling: the construction of models whose representational function may be purely subjective by the modeller alone and for the modeller alone. We also consider the use of single-agent modelling in conjunction with 'projection' in developing construals for communication, in developing construals collaboratively and in developing construals that invoke many agents.

Chapter 3 describes an abstract framework for MWDS that reflects the generality and scope of its potential applications. The diverse roles for MWDS in this framework illustrate many interpretations of Gooding's concept of 'empirical access as a source of knowledge, meaning and reference', as is shown with reference to several practical case studies.

The second part of our treatise addresses the relationship between Gooding's empirical and literary worlds. Its central theme is the highly controversial issue of how – if we deny the literary world of representations its traditional privileged status – reasoning is connected with our agency in the world. In the words of Gooding:

> "If we drop the assumption that ratiocination is the only sort of agency worth recognizing then we can look at how reasoning interacts with other activities. We get a very different picture of the relationship between the world and our representation of it. That relationship is based upon our agency in the world." (p. 8)

Chapter 4 focuses on the activity of MWDS from the modeller's perspective, developing two interpretations for our agency in the world: as resembling the use of an instrument, and as the interactive situated construction of artefacts.

Chapter 5 aims to illustrate how MWDS can be viewed as providing 'a very different picture of the relationship between the world and our representation of it' by exploring the relationship between MWDS and classical programming.

Chapter 6 aims to illustrate how MWDS can potentially be viewed as providing empirical roots for logical constructions by interpreting a formal account of a classical algorithm within our definitive modelling framework. In the process, in keeping with the principal goal of the treatise, it hints at a fundamental role for MWDS in an alternative treatment of computer science.

Because of its subject matter, a practical emphasis for our treatise was most appropriate (cf. the complementary abstract treatment of related themes in [Bey99]). The evidence set out in the treatise has been supplied by collating models from previous work to illustrate major themes, by comparing and contrasting models so as to highlight key issues (see especially the pairing of illustrative models in Chapter 3) and by developing new models or extending existing models.

Research contribution

This thesis illustrates the significant characteristics of MWDS through a systematic review and study of existing definitive models developed with the tools and notations introduced by the EM research group at Warwick. The author's principal contribution has been in supplying the essential practical foundation for an illustrated exposition of MWDS much more ambitious in scope than has been attempted before. Supplying this foundation has required the analysis, comprehension and documentation of more than 20 definitive models selected from over 150 models. Several difficulties, both in comprehending definitive scripts and in comprehending the nature of MWDS, emerged during the practical study and investigation of models. Some of these can be attributed to the nature of applications and to individual styles of model-building. The practical research has involved recording, documenting, interpreting and classifying the diverse ways in which MWDS has been exploited in models to date, and supplying several new models and many extensions to existing models that help to expose the key issues. These practical and experimental studies have a crucial role in the thesis since MWDS, being of its essence associated with tacit knowledge and skills in model-building, is intrinsically difficult to describe in the abstract.

The most significant contribution of this thesis is to give the first focussed account of the concept of MWDS and to provide coherent evidence for Beynon's claim that one-agent modelling is primary (cf. Chapters 2 and 3 and [Bey99]). It also establishes the first explicit connections between MWDS (as a non-logicist approach to knowledge representation) and formal approaches to representing state and knowledge. Because of the subtlety of the concepts to be articulated, which are centrally concerned with the limitations of language, several key paragraphs (in Sections 2.1, 3.1 and 5.1) concerning the semantics of MWDS have been composed in close collaboration with my supervisor. The organisation of the thesis has presented me with a major challenge in the systematic exposition of a complex and elusive subject.

The models collated for the thesis are of benefit in themselves. They include several models (viz. the enhanced Heapsort device in Chapter 6, the Monotone Boolean Function model in Chapter 3, the Number model in Chapter 2, the Chocolate Vending Machine model in Chapter 5) especially constructed by the author to elucidate semantic issues in MWDS. Together with the numerous special-purpose extensions of pre-existing models, these extend the repertoire of Interactive Situation Models (ISM) for tutorial use and learning. They also help to highlight problematic issues and hence suggest possible improvements to the current tools and extensions to the principles.

During the preparation of the thesis, I have contributed to refereed publications [BRS00, BCH+01], and research reports [BRSW98, BCRS99]. I also presented the paper [BCSW99] on behalf of my colleagues at Proc. SCI'99 and ISAS'99 in Orlando, USA. These contributions relate to the work described in:

Chapter 3 – the Digital Watch model [BCSW99] and the Lines model [BRSW98];

Chapter 4 – the computer as an instrument [BCH+01] and the applications of ISMs [BCSW99, BCRS99];

Chapter 6 – the ADM Heapsort device [BRS00] and the Heap model [BRSW98];

Chapter 7 – the exploratory extension of the Lines model [BRSW98].