The Computer as Instrument

Meurig Beynon, Yih-Chang Ch'en, Hsing-Wen Hseu, Soha Maad,
Suwanna Rasmequan, Chris Roe, Jaratsri Rungrattanaubol, Steve Russ,
Ashley Ward, and Allan Wong

The Empirical Modelling Research Group, Department of Computer Science,
University of Warwick, Coventry, UK CV4 7AL
http://www.dcs.warwick.ac.uk/modelling/

Abstract. A distinction is drawn and discussed between two modes of computer use: as a tool and as an instrument. The former is typical for the use of a conventional software product, the latter is more appropriate in volatile environments or where close integration of human and computer processes is desirable. An approach to modelling developed at Warwick and based upon the concepts of observable, dependency and agency has led to the construction of open-ended computer-based artefacts called 'interactive situation models' (ISMs).

The experience of constructing these ISMs, and the principles they embody, exemplify very closely the characterisation of instruments as 'maintaining a relationship between aspects of state'. The framework for modelling that we propose and report on here seems well-suited to account for the dual 'tool-instrument' use of computers. It is also sufficiently broad and fundamental to begin the deconstruction of human-computer interaction that is called for in any attempt to understand the implications of computer-based technology for human cognitive processes.

Introduction

Current frameworks for developing technological products reflect a limited conception of their role. In designing such a product, the emphasis is placed on what can be preconceived about its use, as expressed in its functional specification, its optimisation to meet specific functional needs, and the evaluation of its performance by predetermined metrics. This perspective on design is not sufficient to address the agenda of cognitive technology [13]; it takes too little account of the interaction between a technology, its users and environment. For instance, it is well-recognised that developments in technology can be the result of uses of a product outside the scope of those envisaged by its designers.

Such considerations apply in particular to computer-based technologies. Standard software development methodologies begin by identifying the precise roles that the computer has to play (e.g. through the study of use cases [11]), and focus on designing programs to fulfil these roles as efficiently as possible. Because each use of the computer is tightly constrained by specifying such roles, the trend in designing business processes is to prescribe the interaction between human and computer agents exactly, and optimise their operation accordingly. In this respect, traditional software development favours the conception of the computer as a tool, developed specifically to serve a particular purpose.

In practice, business environments and technologies are volatile, and are liable to evolve in ways that subvert the intended preconceived processes. A major concern in modern software engineering is the need to develop software in such a way that it can be readily adapted to changes in its environment, and to the reengineering of business processes. A conception that is better suited to computer use, both in this context and with reference to the agenda of cognitive technology, is that of the computer as instrument.

Our paper will be in three main sections: the first elaborating on the distinction between the tool and instrument perspectives, and the issues concerning human interaction with artefacts it raises; the second outlining principles and tools for computer-based modelling that we have developed in order to address these issues; the third discussing some relevant case studies.

1 Instruments and Tools

The purpose of this section is to highlight key features of tool and instrument use that motivate the principles for computer-based modelling to be introduced and illustrated in Sections 2 and 3.

1.1 What is an instrument?

The term 'instrument' is here being used to refer to a piece of technology that maintains a relationship between two aspects of state. This broad definition is intended to encompass scientific instruments — such as an ammeter, prosthetic devices (such as a pair of spectacles), and musical instruments. An ammeter maintains the position of a needle according to the current flowing in a circuit, a pair of spectacles maintains a relationship between an external scene and the image on the wearer's retina, and a musical instrument maintains a relationship between the emotional state of the performer and an aural effect. The informality of the references made to 'state' and 'maintaining relationships between state' in this characterisation is acknowledged; later sections of the paper will supply more context for their interpretation.

All three examples of instruments mentioned above have a characteristic feature in common: they establish a correspondence between states that is conceptually direct and immediate. A change in current moves the needle. A change in the external scene changes the image on the retina. A change in the performer's emotional state effects a change in the sound emitted by an instrument. A significant distinction between the three examples is the different roles that human agency plays in each case. No human intervention is needed to maintain the position of the needle on the ammeter. A pair of spectacles serves its function through cooperation between human and technology where the human element is typically unconscious. The most effective performance of the musical instrument demands great intensity of awareness and responsiveness in exercising human skills.

Our primary concern is with *interactive* instruments, where the role of the human in maintaining state resembles that of the performer of a musical instrument. Within the exceptionally broad framework of study to be invoked in this paper, other instances of instruments can be interpreted as derived from this most general case, in the sense that — for instance — the ammeter is the product of a sophisticated empirical process arising from human interactions with the world that involved an awareness and responsiveness of comparable subtlety. In what follows, the term 'instrument' will be used to refer to an interactive instrument.

The characterisation of an artefact as a tool or instrument is not to be interpreted as an either-or classification. The surgeon's scalpel can be (at one and the same time) both a tool to perform a function, and the subject of a performance quite as engaging and open to environmental influences as any musician's. The terms 'tool' and 'instrument' are to be regarded as interpretations put upon the use of an artefact. The OED definition of an instrument as 'a tool for delicate work' [12] suggests a similar association between the concept of an instrument and a particular quality of attention required for its use. Potentially the computer can serve as both tool and instrument, and both perspectives may be appropriate at one and the same time. The principal issue to be examined in this paper is: how can we complement our formal view of computation, which favours the computer as tool, to address the potential of the computer as an instrument?

1.2 Characteristics of instruments

The distinction between an instrument and a tool is associated with particular characteristics of use. In practice, the emphasis when using instruments is on exercising personal skills, whilst the use of tools is typically associated with performing a specific function in an organised framework for interaction in which other human agents or observers are involved. Instruments and tools are respectively correlated in this fashion with subjective and objective interactions. For instance, where the pianist is engaged in a highly personal way with their performance, and judges its success in subjective terms, the mechanic wielding a spanner is generally taking a specific action following a well-defined procedure to attain a particular goal that can be objectively validated. The relationship between instruments and tools identified in this paper accounts for this subjective versus objective emphasis in terms of closely related, but more primitive, aspects of interaction with artefacts.

Both tool and instrument use are particular cases of interaction with artefacts. The very concept of identifying an artefact as a tool or as an instrument involves establishing some characteristic mode of interaction with it. The use of a hammer is appropriate to a context where the characteristic action is hitting a target object with the head of the hammer. A piano is normally used by striking the keys with the fingers. In practice, the potential interactions with an instrument are more open-ended in nature, but they are focused around a range of specific skills that can be evaluated by experienced exponents. In the case of the piano, examples of such skills might include the ability to play scales and arpeggios, to harmonise a melody, or to play pieces within a particular genre.

The standard activities associated with tools and instruments in this way — though very diverse in character — have this in common: they are all to some degree examples of ritualisable experience that can be reproduced by a suitably skilled agent. Recognising such ritualisable experience is not necessarily an objective matter — it is enough that the personal experience of the executant acquires a degree of consistency, and reflects authentic knowledge of their own capabilities, the qualities of the artefact and the essential context. It is in this spirit that — whatever the independent judgement of an experienced musician — the amateur pianist speaks of 'playing the Moonlight Sonata' and of 'not being able to play it with the cat on my lap'.

Both tools and instruments are rooted in the use of artefacts associated with activities that are sufficiently familiar, well-rehearsed and practised that they can be repeated and so can reliably carry us to specific goals; moreover, these activities may be sufficiently rich as to be valued in themselves, for the experience they offer in execution, and the promise of unexpected novel interest and delight. The distinction between tool and instrument perspectives is then a matter of emphasis. In tool-like use of an artefact, we are concerned with efficient and reliable progress towards specific goals (possibly sacrificing any concern for satisfying engagement in the activity). In instrument-like use of an artefact, we give greater priority to appreciation of the experience than to achievement of the goal. A balance of both perspectives is often appropriate, as — when playing chess — we want to win, but also want to explore interesting and novel scenarios, or — when playing music — we aim to play accurately, but aspire to emotional intensity.

The most significant characteristics of the use of an instrument rather than a tool can be illustrated with reference to musical performance. The performer experiences interaction with the instrument as a continuous engagement, where feedback from the instrument and the environment is involved. The outcome of the engagement between performer and technology is more than the accomplishment of a preconceived function. The performance will differ according to situation, and be open to influences (such as the acoustics of the hall, the response of the audience, the precise characteristics of the instrument, the mood of the performer) that are shaped through negotiation and evolve dynamically. The unpredictable manner in which these factors are reflected in the physical and mental state of the performer contrasts with the stereotyped and goal-oriented view of state that is expressed in the familiar proverb "for a man who has only a hammer, the whole world looks like a nail". There is also the possibility that a performance ventures beyond preconceived limits — there is scope for spontaneous action, experiment and improvisation.

2 Computer-based modelling for instruments

This section discusses the extent to which Empirical Modelling (EM), an approach to modelling under development at the University of Warwick [20], provides a conceptual framework for studying the use of instruments and practical support for their construction using the computer. The essential concept behind EM is the analysis of experience in terms of agency, dependency and observation and its representation through the construction of computer-based 'interactive situation models' (ISMs) [14]. A number of special-purpose software tools have been developed to support the construction of ISMs, and a large number of such models created through student projects over the last 10–12 years. Experience gained from this modelling activity indicates strong points of connection between interaction with ISMs and interaction with instruments, as characterised above. In particular, the construction of an ISM is a situated activity that can develop in an open-ended fashion in response to the modeller's evolving focus of interest, and involves exploration and experiment.

2.1 Principles of ISM development

The principles of ISM development will be illustrated using a simple exercise in modelling a traditional clock (see Figure 1).

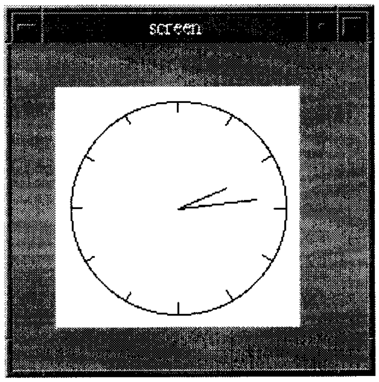

Fig. 1. A simple clock model

This illustration is quite unrepresentative of the scale of ISMs that have been built using EM tools, whose scripts may include several few thousand definitions, but it does indicate the nature of the incremental construction that is involved in creating and using such ISMs. The definitions in the script for this model include the following:

```
openshape clock
within clock {
  real sixthpi
  line eleven, ten, nine, eight, seven, six, five, ..., one
  line noon
  point centre
  real radius
  circle edge
  sixthpi = 0.523599
  radius = 150.0
  eleven = rot(noon, centre, -11 * sixthpi)
  ...
}
```

The variables in this script represent observables in the clock: the rim of the face, represented by the circle `clock/edge`, its centre `clock/centre` and the divisions `eleven`, `ten`, `nine` ... that indicate the hours. A complementary set of definitions represent the dependencies that link the positions of the hour and minute hands to the current time (represented by the variable `clock/t`).

```
within clock {
  line minHand, hourHand
  real minAngle, hourAngle
  real size_minHand, size_hourHand
  int t
  size_minHand, size_hourHand = 0.75, 0.5
  minAngle = (pi div 2.0) - float (t mod 60) * (pi div 30.0)
  hourAngle = (pi div 2.0) - float (t mod 720) * (pi div 360.0)
  minHand = [centre + {size_minHand*radius @ minAngle}, centre]
  hourHand = [centre + {size_hourHand*radius @ hourAngle}, centre]
  centre = {200, 200}
  ...
}
```

Notice how these are specified in such a way that both the position of the minute hand and the hour hand depend on the time via independent definitions. An alternative way to express this dependency that might more aptly describe the physical relationship between the hands of a mechanical clock would express the position of the minute hand as linked to the position of an internal mechanism, and derive the position of the hour hand by a definition representing the chain of cogs that might connect the hour hand to the minute hand.

```
within clock {
  minAngle = (pi div 2.0) - float (t mod 720) * (pi div 30.0)
  hourAngle = (pi div 2.0)  - ((pi div 2.0) - minAngle) div 12.0
  ...
}
```

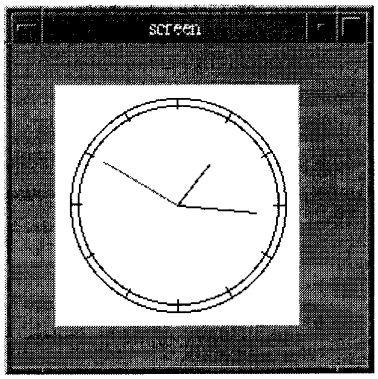

Fig. 2. Clock with details added

Whilst the current time `clock/t` is unspecified, the hands are omitted from the clock face. In specifying this time, the modeller can adopt many different viewpoints. For instance, they may act as if in the role of:

- a user, setting the clock to the current time;
- a designer, seeking to place the hands in a significant configuration;
- the clockmaker who connects the clock mechanism.

When defining the clock mechanism, a simple agent can be introduced to update the clock according to the real time. This is programmed to 'observe' the time on the computer system clock, and to increment the variable `clock/t` every minute.

There are many other instances of potential redefinitions that represent plausible actions on the part of different agents. These effect only very simple changes to the generated display, but nevertheless can correspond to rich thought processes and changes of perspective on the part of the modeller. In the role of a user, the modeller will consider such issues as starting and stopping the clock, or setting the time to reflect a new time zone. In the role of designer, the modeller may consider the appearance of the clock face, the possibility of changing the colour of the hands or adding a second hand (see Figure 2). The modeller can also act in a role that is outside the scope of either the designer or the user, as when reconfiguring the display to a convenient size for demonstration, or adding physically unrealistic features to the clock. Other possibilities include simulating an exceptional event, such as occurs when the minute hand comes loose and hangs vertically. These modifications highlight two fundamental ideas behind EM:

- the construction and structure of scripts mirrors the way in which the modeller construes state-change to occur;

- the modeller's perspective on the script is subject to change from moment to moment, and involves internal human activity (relating to thought processes, situation and agency) that is much richer and more complex than the external computer-based change.

In these respects, constructing an ISM differs from the mathematical approach to creating a model using a computer, where the normal practice is to decide the precise functionality of the model in advance, and to implement from a functional specification. Modelling activity in EM is closer in spirit to creative work in the arts, such as making a sculpture or composing a piece of music. The interaction between the artist's state of mind and the work they are creating is dynamic, and the meaning of the work of art is shaped as it is being developed, as in bricolage [9].

2.2 ISMs as Instruments

There are many ways in which experience of constructing ISMs can illuminate — and has informed — the characterisation of tools and instruments introduced in Section 1. To simplify the discussion, and to avoid technical detail, an ISM will be viewed at a rather high level of abstraction as comprising a definitive script that defines a conceptual state, a display interface made up of one or more screens that embodies some part of this state, together with a collection of agents, each with certain privileges to amend a definition in the script or add a new definition, subject to context and cue. These agents will in general include a variety of human interpreters, who might be in the role of users of the ISM or be one amongst several in a distributed team of modellers. The act of making a redefinition in the script may itself be embodied in an external interaction, such as the movement or an action of the mouse, through a control interface. Where the ISM is not distributed, so that all the state is localised in a single artefact, there is a conceptual role for a locally omnipotent interpreter of the ISM, who is privileged to modify the definitive script directly in whatever fashion they please.

One of the practical aspirations for Empirical Modelling is to develop software tools and/or a more general computer-based technology that can support this 'idealised' vision of an ISM and more. The idealisation reflects the illustrative models that we have constructed in practice, making allowance for the limitations of our current tools. It would clearly be appropriate to extend the concept of embodiment in respect of display and control to take account of more advanced technologies than a typical workstation supplies. For the purposes of this paper, such an extension is not essential, though it is relevant to the issue of using ISMs to construct tools and instruments of the degree of sophistication we are accustomed to see around us.

The characterisation of an instrument as 'maintaining a relationship between aspects of state' is vividly represented in working with ISMs. The concept of shaping the state-as-experienced of an ISM to correspond to that of an external referent is prominent in EM, and in itself characterises an ISM as an interactive instrument. Within an ISM, there are dependencies that maintain the relationship between different subscripts, such as the definitions that link the internal value of the time to the position of the hands, or that determine whether the alarm is ringing with reference to the current time, the alarm time and whether the alarm is set. The agency that is introduced into the clock linking the display to the current time illustrates another mechanism for maintaining relationships between aspects of state.

Analysing what is conceptually involved in the ISM as an instrument reveals the fundamental abstraction to be dependency between states in the physical world. Each such primitive dependency is associated with an experimental observation about how a change to one observable indivisibly effects changes to others. The ISM builds layer upon layer, each based on activities of an instrumental character: the implementation of the dependency maintainer in our interpreter, the compiler for the interpreter, the design of the workstation — at each level, engineered for the maintenance of relationships between state. The significance of such dependency is for the most part hidden from the modeller, but can be exposed — for instance — by substituting a computer too slow to implement an agent that updates in real-time, or to re-evaluate a definition within the lifetime of the modeller.

Viewed in this way, the ISM itself is a complex hierarchical organisation of agency and dependency. Subject to avoiding chains of interdependent definitions of pathological length, there is no practical need to deconstruct the dependencies expressed in definitions by taking the interpreter, the compiler and the hardware into account, but such a deconstruction is essential in order to appreciate the semantics of the ISM as an instrument. In particular, an ISM can refer to relationships between aspects of state embracing observables that are explicit in a definitive script and those in the external environment. It is for this reason that part of the definitive script for the clock can be interpreted as defining "the state of the screen display".

3 Computer-based Instruments and Tools from a Cognitive Technology perspective

The impact of technology upon our cognition is the central theme of Cognitive Technology (CT). Much thinking about computer use and technology necessarily tries to address this issue without taking full account of the complexity of the relationship between the experiences offered by the computer and the experiences of users: how these experiences depend on the physical and social context, on the personal characteristics of the user, and how they are liable to evolve. The concepts of ease-of-use [15] and of invisible computing [16] will no doubt play a significant practical role in exploiting computer-based technology, but — where CT is concerned — they are only one peripheral aspect of a much bigger agenda. The most satisfying activities — such as playing a musical instrument — are not generally easy, and though they eventually involve invisible interaction, they are learned through sometimes painful, sometimes rewarding engagement of mind, body and soul. To understand the use and implications of computer-based technology more fully, it is essential to undertake some deconstruction of human-computer interaction, exposing its empirical roots not only in human experience and technological performance, but also in its physical, social and administrative context. Exploring the potential for marrying human and computer activities through the use of interactive instruments provides an appropriate focus. A key objective is to be able to understand the dual tool-instrument perspectives within a single framework.

3.1 Paradigms for Computer-based Instruments and Tools

The ISM can be seen as an archetype for interactive computer-based instruments. In its essential substance and nature, it is well oriented towards this rôle. A definitive script is an intricate net of observations about relationships between changes to observables — the product of a family of experiments. Within the script, each definition can be viewed as an instrument, maintaining a relationship between one feature of the state and another. Taken as a whole, the definitions in an ISM, each associated with an experimental context, form a tower of dependencies composed hierarchically in a manner resembling the network of experimental observations that validates a well-conceived engineering product. To construct such an ISM, the mind of a human interpreter must visit every composition of such dependencies, construing it with reference to the agency that is to exploit it. This is the justification for making interactive instruments (see Section 1) our primary concern.

Though each ISM has the same characteristic substance, its quality is crucially dependent upon two factors. The first is the way in which the dependencies in the ISM are assembled by the modeller: this relates to the structure of the ISM, empirically established by the modeller according to how they construe its intended behaviour with reference to observables, agency and dependency. The other is the experiential foundation supplied by the constituent experiments. In each case, the reliability with which a relationship between aspects of state can be maintained is an empirical matter. The delicacy of the human control over the instrument is one of these constituents of the experimentally shaped responses of the ISM: it is the basis for the ritualisable experience of the skilled performer.

Numerous ISMs demonstrate these principles practically in relation to modelling real-world phenomena. In that context, the modeller's construal refers most especially to how the phenomenon itself is explained. A simulation of the Clayton Tunnel railway disaster is one case study of this nature [17]. Other research, carried out by Cartwright in collaboration with Adzhiev and Pasko [2], has involved the development of a geometric instrument based on a definitive front-end to the HyperFun geometric modelling environment [3]. In this context, the application builder's construal is concerned with giving the user appropriate control over the geometry described by the geometric modeller.

In contrast, conventional programming paradigms are oriented towards tool-building by computer. The possible contexts of application of the program as a tool are determined by its specification, and the program code is an explicit account of the functions that the tool can perform. Procedural and declarative programming styles approach the characterisation of a tool by specifying its functions explicitly and implicitly respectively, as is indicated by their substance. A procedural program is a complex pattern of sequences of changes to values of observables (an explicit account of a process). A logical or functional specification is a complex aggregate of assertions about relationships between values of observables (the set of predictions of a theory).

3.2 Instruments and Tools in the EM Perspective

Reliability of experience is crucial to the successful development of tools, and to the subagendas of ease-of-use and invisibility in particular. Unlike ISMs, traditional computer programs, being optimised to serve particular functions and operate in specific situations, are constructed in ways that do not necessarily give any insight into the fashion in which the programmer construes the domain (though this is recognised to be highly relevant to the process of identifying a requirement). They are generally designed to exploit the computer's capacity for performing exceedingly complex state-change, and to make the role of the user as clearly defined and simple to enact as possible. These qualities derive from specifying and fashioning the context for the program execution tightly, in somewhat the same manner that a train runs along pre-engineered tracks. In software system development, the analogue of laying track is the identification and contrivance of reliable experience. Providing this essential foundation for software system applications was what first motivated Pi-Hwa Sun to introduce the concept of an ISM [1].

The use of ISMs to trace the activities involved in developing algorithms and processes in environments that initially support only unconstrained and unsystematic interaction is illustrated in two studies. Our study of heapsort [8] shows how an environment in which logical invariants of the algorithm appear as observables can be embedded into an environment similar to that a lecturer might use when introducing the algorithm on a blackboard. A second study illustrates how a manufacturing process and an associated rework process can be fashioned from primitive production and assembly style activities by building an ISM that combines process automation with the possibility of human intervention in managing non-routine rework [18].

The way in which tools are locked into their context of use accounts for their relative inflexibility. A traditional computer program can be versatile, in the sense that it can perform a compendium of diverse functions, like a Swiss Army knife, but it is constrained by the sharply prescribed user-computer boundary, and does not admit open and interactive re-interpretation in use. In contrast, an instrument such as an ISM invites the human interpreter to engage their imagination in whatever ways suit the situation. This potential for an eclectic projection of meanings that can be subjective and provisional onto an ISM is evident even in the simple clock illustration. The result is that re-use in EM is often associated with re-interpretation and a relatively seamless reworking. Indeed, several variations on clocks and digital watches deriving from a single

ISM are featured in previous work: these include ISMs, including distributed ISMs, to represent a combined statechart and digital watch, for a chess clock, and for the explicit state and mental model of an actual digital watch [6], [7]. There is likewise an ISM associated with a family of OXO-like games [5].

The intimacy of instrument and mind is nowhere more apparent than in the ways in which instruments can migrate from the external domain of the technology so as to become invisible to the human interpreter. This is commonplace in everyday technology, as when the use of a lens as a subject in the study of optics leads to the development of spectacles. In our characterisation of the instrument as maintaining a relationship between aspects of state, this can be interpreted as merging one aspect of state with another, enlisting the instrument in the service of the model. In EM terms, this is directly interpretable with reference to partitioning definitive scripts in different ways and so reconfiguring the aspects of state whose relationship is the subject of attention. An instance of this migration from referent to model occurs whenever a fragment of script is first developed in isolation, then embedded into the ISM under construction. It is through such migration that this fragment becomes associated with one of the constituent experiments of the ISM.

EM supplies a useful framework in which to integrate the dual tool-instrument perspectives. Though they have an open, uncircumscribed functionality, our ISMs can be exercised as if they were designed for a specific purpose. In this role, ISMs are not as efficient as conventional programs optimised to this function, and in this sense they can be viewed as instruments for prototyping tools (see [19]). As described in [1], they can also be used to explore the contexts for reliable interaction that precede the specification of tools.

An ISM establishes an intimacy of human-computer association that is quite unlike a conventional program in character. From a CT perspective, the most important implication of this is the way that — like the spreadsheet [10] — it has the power to change the culture of use. In principle, the openness of the ISM allows the human agents to exploit the technology in what is characterised in [4] as an 'idealist' rather than a 'realist' frame of mind. Where the objective of the realist is to use technology to save effort and obtain results automatically, the idealist is primarily motivated by a concern to complete the task in a way that gives satisfaction and achieves results that are highly optimised to the particular situation. The first significant practical application of this concept was the use of the Temposcope [4] to timetable some 120 student project orals in March 2001. It is perhaps encouraging that the administrator who made use of this ISM for the first time this year made no comment on the quality of the software, but declared herself much happier about the resulting timetable than on previous occasions.

Conclusion

It remains to consider more closely the relevance to Cognitive Technology of the computer-based instrument culture associated with EM.

It is surely too much to expect that CT can predict or fully explain the complex interactions between technology, mind and society. It is difficult to imagine how any study could remove all controversy from issues such as the survival of the QWERTY keyboard, how certain musical instruments are forgotten whilst others have become the carriers of an entire musical tradition, or what social conventions are needed to sustain a language. That said, current accounts of technology are not well-suited for the discussion of such concerns, and EM provides an alternative perspective that gives much greater prominence to the empirical roots of knowledge. In particular, as a conceptual framework, EM can help us in studying the emergence of the ritualisable activities that support tools and instruments from our casual and serendipitous interaction with artefacts. As our discussion of the tool and instrument perspectives has demonstrated, the construction of ISMs can also be used to record and explore insights that are difficult to frame in language alone.

It is unclear to what extent CT is concerned with guiding the future development of technology. In so far as CT draws our attention to a complex evolutionary activity, there is a speculative analogy to be made with Darwinian evolution, and the developments — inconceivable to Darwin's contemporaries — that have eventually led to genetic engineering. Studies in CT can certainly guide us, when developing technologies, to anticipate *some* of the unfortunate implications for people and society that are currently unintended and unexpected and to promote technological developments that are more rewarding and potentially less dangerous in human terms. Somewhat paradoxically, the essential rationale for CT is that — no matter how technologies are developed — they will always evolve in ways that take us by surprise. In so far as CT is concerned with helping us to deal with the effects of this evolution, EM is of interest as an approach to developing computer-based technology that acknowledges that requirements change — indeed that there *is* no fixed requirement — and promises to deliver resources that are less prescriptive and integrate more effectively with human activities. Our ongoing research on the Temposcope [4] and Cartwright's research on applying dependency maintenance to interactive TV applications [2] is indicative of the potential here.

In our current state of knowledge, the principal agenda for CT is perhaps to expose and describe the phenomena that we observe in the interaction of technologies with people and societies. It is our belief that the EM approach of construing phenomena in terms of observables, dependency and agency, and embodying these construals in ISMs, is philosophically and practically well-suited for tackling this agenda, and can assist in understanding and developing instruments of mind.

References

1. Sun, P-H., Distributed Empirical Modelling and its Application to Software System Development, PhD thesis, University of Warwick, July 1999.
2. Cartwright, R. I., "Distributed shape modelling with EmpiricalHyperFun", First International Conference on Digital and Academic Liberty of Information, Aizu March 2001, to appear.
3. `http://www.hyperfun.org/`
4. Beynon, W. M., Ward, A., Maad, S., Wong, A., Rasmequan, S., Russ, S., "The Temposcope: a Computer Instrument for the Idealist Timetabler", Proceedings of the Third International Conference on the Practice and Theory of Automated Timetabling, Constance, Germany, August 16–18, 2000.
5. Beynon, W. M., Joy, M. S., "Computer Programming for Noughts-and-Crosses: New Frontiers", Proceedings of PPIG '94, Open University, 27–37, January 1994.
6. Fischer, C. N., Beynon, W. M., "Empirical Modelling of Products", International Conference on Simulation and Multimedia in Engineering Education, Phoenix, Arizona, January 7–11, 2001.
7. Roe, C., Beynon, W. M., Fischer, C. N., "Empirical Modelling for the conceptual design and use of products", International Conference on Simulation and Multimedia in Engineering Education, Phoenix, Arizona, January 7–11, 2001.
8. Beynon, W. M., Rungrattanaubol, J., Sinclair, J., "Formal Specification from an Observation-Oriented Perspective", Proceedings of the Fifteenth British Colloquium in Theoretical Computer Science, Keele University, April 1999.
9. Levi-Strauss, C., "The savage mind", University of Chicago Press, 1966.
10. Nardi, B. A., "A small matter of programming — Perspectives on end user computing", MIT Press, Cambridge, Mass, 1993.
11. Jacobson, I., "Object-oriented software engineering — A use case approach", ACM Press, Addison Wesley, 1992.
12. Concise Oxford Dictionary of Current English, 8th Edition, Clarendon 1990.
13. `http://www.cogtech.org`
14. Beynon, W. M., "Empirical Modelling and the Foundations of Artificial Intelligence", *Proceedings of CMAA'98*, Lecture Notes in AI 1562, Springer, pp322–364, 1999.
15. Roberts, D., Berry, D., Isensee, S., Mullaly, J., "Designing for the User with OVID: Bridging User Interface Design and Software Engineering", Macmillan Technical Publishing, 1998. `http://www.ibm.com/easy/`
16. Norman, D. A., "The Invisible Computer", The MIT Press, October 1999.
17. Beynon, W. M., Sun, P-H., "Computer-mediated communication: a Distributed Empirical Modelling perspective", Proceedings of CT'99, San Francisco, August 1999.
18. Evans, M., Beynon, W. M., Fischer, C., "Empirical Modelling for the logistics of rework in the manufacturing process" COBEM 2001.
19. Allderidge, J., Beynon, M., Cartwright, R., Yung, Y. P., "Enabling Technologies for Empirical Modelling in Graphics", Research Report CS-RR-329, Department of Computer Science, University of Warwick, Coventry, UK, July 1997.
20. `http://www.dcs.warwick.ac.uk/modelling/`