Radical Empiricism, Empirical Modelling and the nature of knowing

Meurig Beynon
Computer Science, University of Warwick, Coventry CV4 7AL

Abstract

This paper explores connections between Radical Empiricism (RE), a philosophic attitude developed by William James at the beginning of the 20th century, and Empirical Modelling (EM), an approach to computer-based modelling that has been developed by the author and his collaborators over a number of years. It focuses in particular on how both RE and EM promote a perspective on the nature of knowing that is radically different from that typically invoked in contemporary approaches to knowledge representation in computing. This is illustrated in detail with reference to the modelling of several scenarios of lift use. Some potential implications for knowledge management are briefly reviewed.

This paper considers the potential significance of William James's philosophic attitude of 'Radical Empiricism (RE)' [13] in relation to contemporary problems of knowledge representation in the information sciences. Current trends in computer technology and use provide a strong motivation for reviewing RE in this light. For instance, as Gooding relates in [11], our understanding of how scientific knowledge relates to interaction with the natural world and with our peers is challenged by the development of virtual reality environments, and the role that virtual experiments have come to play in science. Such considerations have prompted a reappraisal of the fundamental assumptions that inform the logicist approach to knowledge representation in AI, and called into question the extent to which knowledge is mediated by language rather than engagement with the world (cf. Cantwell-Smith [9], Turner [16]). In this connection, the relevance of RE stems from the priority it ascribes to 'pure experience', and its contention that (to paraphrase William James) the whole of the nature of knowing can be put into experiential terms ([13], p.56). The problematic aspect of RE, as identified by Bird in [8], is that it is of its essence inarticulate: " ... James's pure experience has to be such that nothing can be said about it, if it is to fulfil the role for which it is cast". This distances RE both from the mainstream philosophical traditions, and from the received views of computer programming as intrinsically bound up with formal languages and logical specification.

Empirical Modelling (EM) is an approach to computer-based modelling that has been developed by the author and his collaborators at the University of Warwick over several years (for background, see [1,2,3,4,5,6,7,15,17,18]). The product of an EM exercise is first and foremost to be regarded as a source of experience whose interpretation by the modeller is not preconceived, but is to be established in the mind of the modeller through an association between experience of the model and experience external to the model. Knowledge in such a context has the qualities that James attributes to knowledge in [13]: it is a personal awareness on the part of the modeller that one experience stands in a particular relation to another. To borrow James's expression, experience of interaction with the model is 'an experience that knows another' that can act as a substitute for experience external to the model in a definite practical sense ([13], p.61).

This paper explores the extent to which, through model-building using EM, it is possible to track James's exposition of the empirical roots of knowledge, with its emphasis on the fundamental significance in sense-making of our capacity to experience conjunctive relations between things. This exploration touches on many issues topical in modern computing that are addressed in James's account of RE, such as the nature of consciousness ([13], p.132-3), agency ([13], p.178-80, 185-6) and reality ([13], p.159-60). In what sense, and to what extent, it is possible to establish meaningful connections between James's philosophic attitude and EM is itself potentially a controversial issue. The author's justification for proposing such connections stems from his own direct experience – in particular from the way in which James's discussion of 'pure experience' in [13] resonates with the issues involved in a detailed exposition of modelling with definitive scripts. In James's terms, the thrust of the exposition will be to make it plausible that the experience of EM 'knows' pure experience.

The paper is in two principal sections. Section 1 reviews Empirical Modelling (EM) principles and practice. Section 2 discusses William James's philosophic attitude of Radical Empiricism and the parallels that may be drawn between James's account of 'pure experience' and experience of EM. The paper concludes by identifying significant issues in knowledge management for which RE and EM may be seen as particularly relevant.

1. Empirical Modelling

Empirical Modelling (EM) is a term that has been introduced to describe principles and tools that support an unusual kind of computer-based model-building. The development of this approach has been the subject of an extended research programme at the University of Warwick that originated with the design of a notation for interactive graphics some 20 years ago (see [17]). EM is unusual in that it represents a form of modelling oriented towards capturing 'state-as-experienced' and leads to the construction of computer-based artefacts that have no preconceived or formally circumscribed behaviour. It will be helpful to put this claim in context before presenting concrete evidence to illustrate it.

1.1. The nature of EM models

It is a commonplace fundamental notion of computer science that the significant semantics of a computer program is captured in the algorithmic behaviour that it implements. This notion leads on to the idea that all legitimate computer use is necessarily essentially concerned with specifying and implementing abstract behaviours. Whilst computer science acknowledges the need to design interfaces through which the user can direct or monitor computer behaviours, this is typically seen as beyond the remit of the core science of computing. Such a view of core computer science as fundamentally concerned with abstract mathematical concepts of computation and behaviour is curious in view of key trends in the historical development of computing practice.

The use of the computer – or more precisely of computer-related technology – to generate so-called virtual reality (VR) environments epitomises some of the most significant issues. If a VR environment is to have the qualities of a reality such as we experience in our everyday life, the character of the interface to the underlying computer model of state is a crucial rather a peripheral issue. In a real environment, the user can observe the environment in ways that have not been preconceived, and conduct authentic measurements and experiments. It may be possible in principle to conceive how such a real environment can be implemented if we accept a reductionist view of reality and a logicist account of human intelligence. Such a conception is of little interest either to the computer practitioner or to the user, whose view of construction and use is guided by pragmatic concerns. For instance, we should not necessarily expect to have perfect knowledge in order to construct an environment to assist us in the task of 'knowledge management'. In any event, an environment in which only knowledge that has been previously encoded is recoverable is of limited use.

Viewed from the classical computational perspective, the idea of building computer-based models that offer the user more than has been consciously encoded by way of 'use cases' (cf. [12]) seems paradoxical. To see this from another perspective, it is helpful to consider other model-building activities, such as those associated with what Levi-Strauss characterises as bricolage [14]. In bricolage, the modeller's concept of the artefact under construction develops in conjunction with the artefact itself: the modeller gains feedback from experience of the unformed artefact itself, and uses this feedback to guide its further development. To the extent that prototyping plays a part in computer program development, practical computer programming can be viewed in this light. In practice, feedback routinely affects the bricoleur's conception of the developing artefact in a much more radical way than prototyping affects software development. The bricoleur is not constrained by the pre-established conventions for interpretation that typically frame a software product: in contrast, the way in which the artefact is to be interpreted and used is in many aspects ill-defined and open to negotiation throughout its construction. EM may be seen as offering an approach to model-building with the computer that has these key characteristics in common with bricolage.

The classical view of computation is complementary to a classical view of knowledge. The concept of building programs that are optimised to serve a specific narrow function, and of encoding information in formal data structures, promotes a prosaic view of what knowledge and knowledge representation entail. Knowledge is seen as something to be possessed that can be expressed and recorded as a proposition, as in "I know the telephone number of staff member X". This view is appropriate in its proper context, as is implied by the use of the expressions 'perfect knowledge' and 'only knowledge that has been previously encoded' above. It makes sense to speak of recording the telephone numbers of all university staff in a directory, and – in the absence of any further context – it would be absurd to search this directory for those who are blue-eyed and can play the bassoon. On the other hand, it is evident that developing a VR environment involves much more than encoding abstract knowledge about a real-world domain. A key issue is how knowledge about the domain is reflected in the interactive experience that the environment offers to the user.

EM differs from classical computer-based model-building in its relation to both computation and knowledge. The distinction between an EM model and a computer program is analogous to the 'real-world' distinction between an open environment and a circumscribed procedure. To be familiar with an EM model is analogous to being familiar with a city; to be familiar with a computer program is analogous to knowing how to use the underground to get from one station to another. It is helpful to think of the EM model as something organic that grows and changes over time: there is no sense in which being familiar with a city is a limited notion – we explore and observe more of a city over time; the city changes; we change in our response to the city; we change the city. There is no reasonable sense in which familiarity with a city can be comprehensively preconceived; in any moment, we can at most testify to what is familiar in the particular aspect of the city of which we have immediate and direct experience. Though I may tell you with conviction that the cathedral is out of sight but lies just around the corner, I appeal to my memory and to my faith in the permanence of place, and know what only the act of taking us to the cathedral can confirm. An EM model is more general than a computer program in much the same way that 'knowing how to use the underground' sits within the broader framework of 'being familiar with the city'. The analogy also helps to illustrate the ontological distinction between the model and the program. If we take the permanence and reliability of a city environment and the generic and routine nature of our observation of underground transport for granted, then 'knowing how to use the underground' is a skill that can be viewed as independent of any particular city, that can be exercised without engaging with the total experience of 'becoming familiar with the city'. It is possible to imagine how a robot can be programmed with the stimulus-response patterns needed to use the underground, but much harder to conceive what might be meant by programming a robot to get to know a city.

The metaphor of "knowing the city" is helpful when understanding both the way in which an EM model is realised on a computer, and the philosophical stance it reflects on knowledge. As I look out of my study window at this moment, my view of the city of Coventry is limited to the roof of my neighbour's house and the sky above. In my imagination, I can trace the path from the house to the city, though the act of tracing this path is no part of my present direct experience. My knowledge of Coventry invokes the conjunction in my memory of all the direct experiences I have had over many years of different aspects of the city. There is no sense in which all these prior experiences can be taken as one experience, any more than I can be in more than one place at once. From this perspective, the 'perfect knowledge' on which we might aspire to base a classical computer-based model of the city is a mere chimera. What there is to be known of Coventry is more than I can ever experience, and my personal knowledge is established, maintained and revised dynamically through my ongoing interactions with it. Even my current limited view of the city is potentially open to such elaboration of knowledge, now that I notice the pigeon droppings on the crest of the roof, the silver sliver of a distant passing plane, and savour the taste of lukewarm lime juice cordial. As a source of direct experience, an EM model cannot compare with my everyday Coventry environment in its richness, but it offers access to potential experience and knowledge with which I can engage in similar ways. In effect, it supplies an interactive artefact with which to trace a history of experiences that is similar in character to our everyday experience of purposefully and accidentally developing our knowledge of a city. In this activity, the computer serves two significant roles: it serves as a physical artefact, similar to the artefacts of bricolage, that generate direct experiences to which the modeller can make a creative response, and at the same time it allows the modeller to record and elaborate experiences as they are encountered in 'the stream of thought'. In these respects, the computer offers support for managing knowledge as characterised as 'interaction with memory'.

1.2. An illustrative case-study

Empirical Modelling principles and tools were not initially developed with the general characteristics of EM models, as described above, in mind. Our present understanding of the characteristics of EM models was arrived at after first appreciating the difficulties of providing a formal semantics for such models within the classical theoretical framework for computation. Two sources of inspiration have been particularly significant in reaching this understanding: a large body of practical work on model-building and tool development that has been carried out largely by computer science students at Warwick over the last 15 years, and the philosophical writings of William James [13]. This subsection will give a brief sketch of practical EM with reference to a model of a lift initially developed in a summer vacation project in 1994. (The lift model and the **`EDEN`** interpreter which is needed to run the model can be downloaded from the EM website – see [17] and [18]). The next section will then discuss EM in relation to William James's philosophic attitude of Radical Empiricism [13].

Figure 1 depicts our EM model of the lift as it might be displayed in one particular state. The lift car appears on the left, and the five floors that the lift visits on the right. The stick figures are used to indicate the current locations of lift users: a label attached to the head of each stick figure identifies the list of users at each location. The bold vertical lines between the lift shaft and the floors represent the status of the doors between the lift car and each floor: the door on the fourth floor is currently open. The small boxes to the right of the floors and to the left of the lift car are call buttons, and those that are currently selected are indicated in bold.

To interpret Figure 1 as associated with a direct experience of an actual lift, it is helpful to think of looking at a glass lift at the side of a five storey building. Whether this is representative of the direct experience that is of primary interest to the modeller depends upon the role in which the modeller interacts. Examples of possible roles that the modeller might play, possibly concurrently, include: a lift user, designer or analyst, or perhaps even a story teller for whom a lift is a significant location (e.g. as the venue for murder in a detective story). In practice, if we have any familiarity with lifts (even if we are in the position of user X in Figure 1, with limited access to the actual current status of the lift), we bring to our interaction with them a general concept that there are floors and users and a lift car that moves vertically between them, and may be expected to visit the ground floor at some stage in response to pressing the call button.

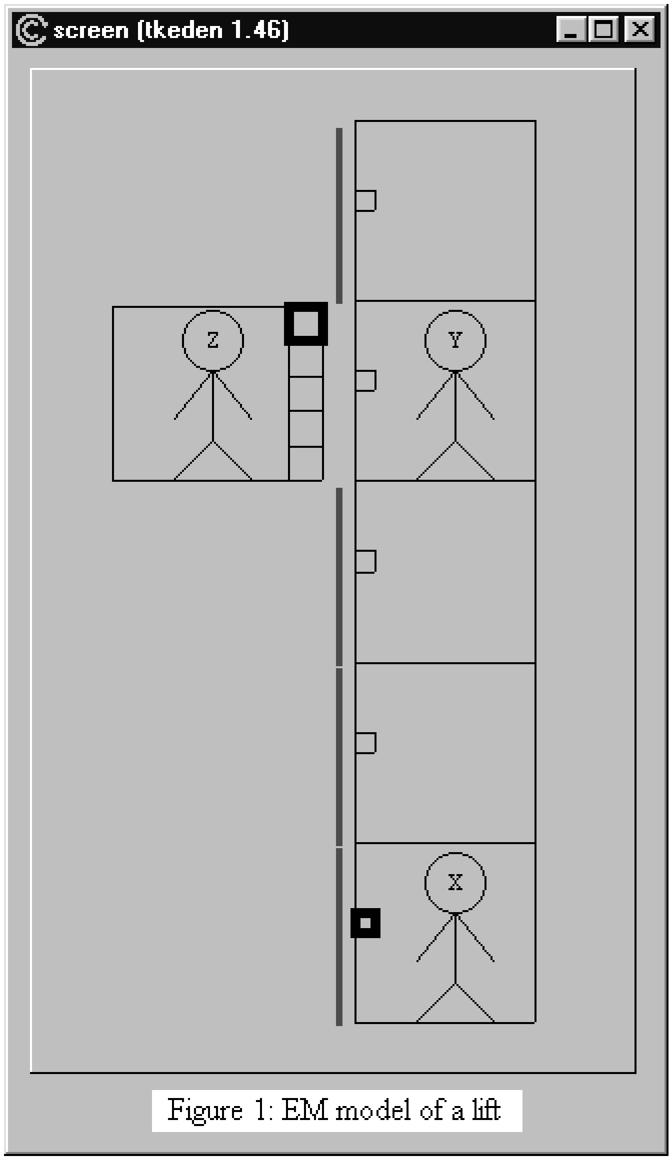

Figure 1: EM model of a lift

In our modelling tool (**EDEN**), the current status of the lift is determined by the current values of its characteristic observables. The names of some of the key observables, as they are recorded in our model-building interpreter, are:

_liftfloor	- where is the lift? [on which floor: 1-5]
_open3	- is the door open at level 3?
_car2	- is button 2 in the car selected?
_button4	- is the call button on floor 4 selected?
locX	- where is user X? [0: in lift, 1-5: on which floor, 7: nowhere]
_inliftX	- is user X in the lift?
_destX	- where is user X intending to go?

The values of most of these observables are directly reflected in features of the display. For instance, **_liftfloor** determines the position of the lift, **_open3** determines how the line that represent the door on floor 3 is depicted and **_button4** determines whether the call button on floor 4 is highlighted. It is characteristic of modelling with **EDEN** that these features of the display can themselves be construed as observables whose values are specified by definitions resembling the definitions that relate the cells of a spreadsheet. Taken together, the definitions of observables, whether explicitly by values, or implicitly via formulae, form a 'definitive script' that specifies the current state of the model. The term 'definitive notation' is used to refer to the underlying notations used to formulate a script. **EDEN** includes a definitive notation for defining the screen layout and for defining planar line drawings. The use of these notations is illustrated in Figure 1.

The definitions in a script express expectations about how the values of observables are linked where 'atomic' changes are concerned. For instance, when the lift moves, anyone in the lift moves with it as part of one and the same state change. The relations linking these changes are called 'dependencies' between observables. For example, using **carpos** and **posX** respectively to represent the positions of the lift car and of user X (as depicted by 2D coordinates in Figure 1), definitions of the following general type express dependencies:

```
carpos is {liftshaftL , (_liftfloor-1) * floorheight}
posX is if _inliftX then carpos else floorX
_inliftX is (locX == 0)
```

The use of a definitive script to represent state-as-experienced is the fundamental technique by which a modeller constructs environments for interaction in EM. Such an environment is typically oriented towards expressing the perspective of a particular agent where both observation and interaction are concerned. In this context, an agent is anything that might be deemed to be responsible for changing the current state of the EM model. The archetypal agent is the modeller who acts to change the state of the model through manually entering redefinitions of observables into the **EDEN** input window. As will be discussed and illustrated later, the lift users and the lift itself can also be viewed and in various different ways animated as agents with some autonomous capacity to change state. In viewing the EM model, it is important to understand its primary role as a representation of state-as-experienced, rather than merely as a vehicle for automating conventional lift behaviour. To return to the analogy introduced above, animated behaviours in EM are in the first instance like bus tours of the city: by default, on such a tour, we observe the city according to a pre-programmed plan, but there is nothing in principle to prevent us from leaving the organised party and continuing our exploration independently.

The idea that the lift model captures state-as-experienced can only be appreciated by considering the possible modes of interaction with it that are open to the modeller. To understand this, it is helpful to reflect on the distinction between the abstract interaction that is involved in using the underground and that involved in making the same journey above ground. In the former context, the traveller can get by by pattern matching, relying on the way in which the underground environment itself is built to encode knowledge about locations and directions in place names, keywords and icons. In the latter context, the personal knowledge of the traveller – their ability to recognise landmarks, to maintain a sense of direction, to devise strategies for crossing roads, to be able to consult other people – is paramount: though the environment itself has embedded knowledge in the form of signposts and street-names, the logistics of their use are by no means fully systematised. In practice, both opportunistic and exploratory human interpretation and pre-coded knowledge play their part in navigation in a city environment. In a similar way, interaction with the EM model of the lift can either give prominence to how human intelligence informs state transition or support routine and automatic interaction. What is more, commitment to changing the model is not necessarily required in playing these different interactive roles, and either can be highlighted according to purpose.

The openness of the modeller's agency in EM is the platform for intelligent interaction. The modeller is free to redefine any of the observables in the script subject only to being able to interpret the consequences of such redefinition. As a simple example, a redefinition such as

```
carpos is {liftshaftL , 0}
```

relocates the lift car at the base of the lift shaft as if it were out of service. Note that this redefinition in some sense 'deals intelligently with user Z': in the context of Figure 1, it would move user Z within the lift no matter where the lift was placed. Actions of this open-ended nature are analogous to the unconstrained actions that the traveller can make in the city environment, that may result in reaching the destination inadvertently, might lead to getting lost or to being run over.

To illustrate how knowledge about meaningful interaction can be exposed without being encoded in an EM model, the lift model has been adapted and used to animate the unusual instance of real-life lift use described in Box 1. For this purpose, the modeller has only to compile a sequence of elementary redefinitions of observables to reflect the actions of the users and the lift as they occur. By playing through this sequence of redefinitions, the modeller can visit each situation as it arises, and reflect upon the perceptions and motivations of the various agents involved. The modeller can then construct a narrative to document their informal understanding of the scenario. The **`EDEN`** interpreter provides a procedural construct for playing out this sequence of redefinitions as if they were being entered one-by-one by the modeller. Such a sequence can be accompanied by a commentary that gives more insight into the complex combination of observation and logic that guides each lift user's actions.

By way of context, the events described in this scenario occurred some fifteen years ago, and have been a source of puzzlement to the author ever since. Constructing the model has finally enabled me to reconcile my assured personal knowledge of the circumstances of the interaction with a plausible objective account of the behaviour of three sane lift users. To dramatise the distinction between these two perspectives on the events, I shall complement the personal account in Box 1 with a further analysis in which my role is described in the third person, as that of 'user I'.

I am at a conference in the Netherlands.
I arrive late at night and hardly notice where my room is.
Next morning, I notice that my room is on the top floor.
I walk down to breakfast thinking about my talk later on.

After breakfast I meet two other delegates X and Y.
We get in the lift to return to our rooms.
X presses the button for floor 4.
Y says he is on the floor above X, and selects floor 5.
Since the top button is selected, I don't press a button.

We talk as we ascend. The lift stops. The door opens.
The floor numbers aren't clearly marked.
I say to X 'this must be floor 4' – he gets out.
Y and I carry on talking.

When the lift next stops, the floor is still unclear.
I say to Y 'X is on the floor below you; this is your floor'.
Y gets out. I think something is not quite right.
I think 'is this the top floor?' and 'should I get out?'.
I'm unsure, but notice that the button for floor 5 is still lit.
I proceed to the top floor which is the next floor, floor 5.
When I get out of the lift, I can't find my room.

There's no room where my room is on floor 5.
I walk down to floor 4, and pass Y on his way to floor 5.
When I reach floor 4, I meet X coming up from floor 3

How did I manage to get all three of us to the wrong floor?

Box 1 : Travails in a lift

Figure 2 depicts a critical moment in the scenario, at which it becomes clear to user I that there is some conflict between what I observes and the way in which I had conceived X and Y's interaction with the lift up to this point. I supposes that his room is on the top floor, and realises that the lift has not yet reached the top floor. On the other hand, on entering the lift, I observed the selection of just two buttons by co-users X and Y, and noted that one of these corresponded to the top floor. It would seem from this that I should have got out at the same time as Y. It was just after Y got out of the lift that I realised that perhaps he had misled X, and that the lift had in fact stopped at floor 3 when called by an impatient user Z. At that point, it was apparent to I that both X and Y had got out at the wrong floors, but not that he himself was also about to travel upwards to the wrong floor. The extra detail of the profile of the building that has been introduced in Figure 2 represents the other element in the situation by which I himself was misled into supposing that his room was on floor 5 rather than floor 4: in the vicinity of I's room, floor 4 was the top floor. The pragmatic nature of the decision that I makes at this point, so different in character from the pure reasoning that is sometimes imputed to human agents (cf. the Mensa problem discussed in [1]), is emphasised by the real time constraints on decision imposed by the lift itself.

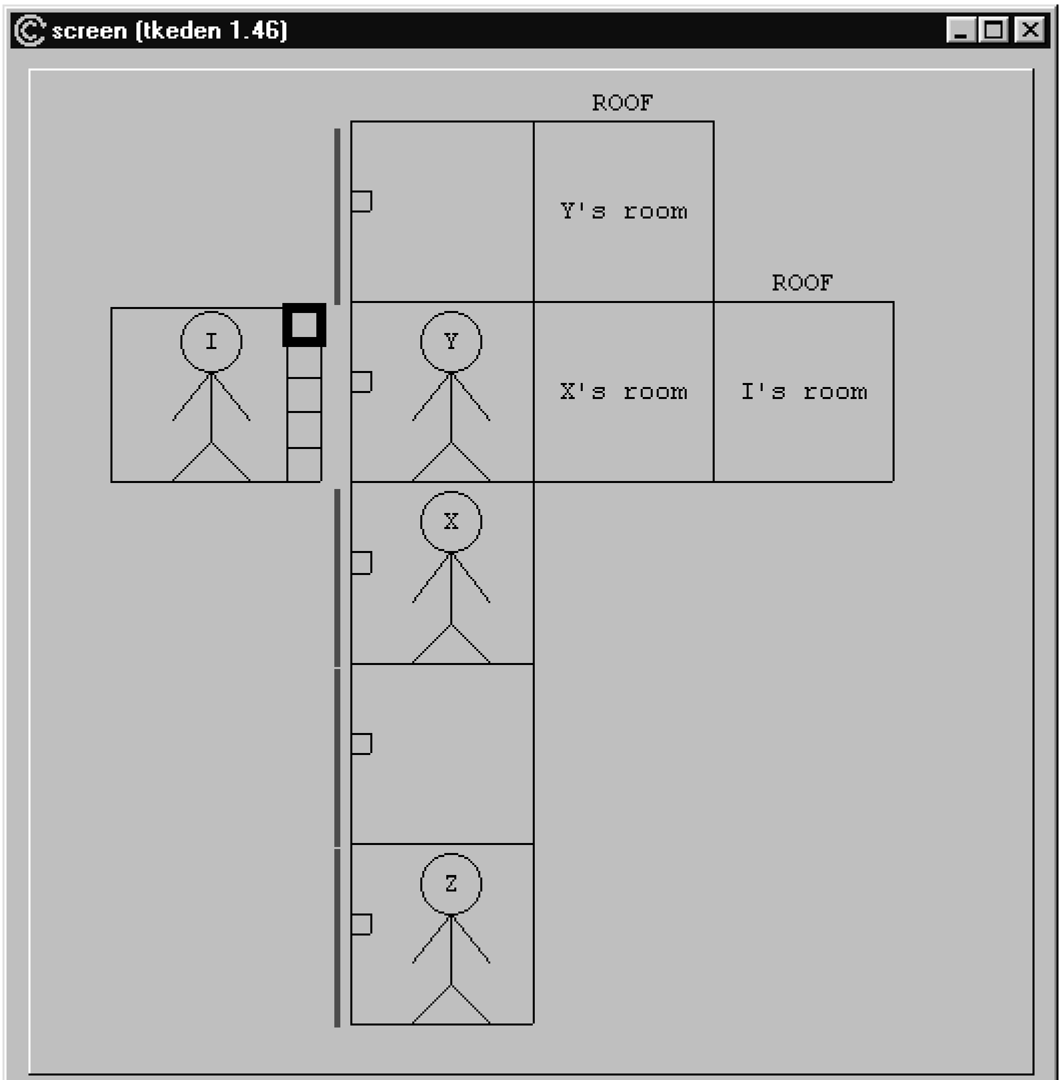

Figure 2: Embellished EM model of a lift

Figure 2 can be interpreted as the particular specialisation of the vanilla model of the lift depicted in Figure 1 that I ideally should have had in mind in using the lift. In practice, I was not able to construct this model in real-time, and acted with a different conception of the situation. A crucial aspect of the EM model is that it relies essentially upon physical artefacts to account faithfully for I's misunderstanding. The geometric content of the model, limited as it is as far as realistic details of the physical environment are concerned, has rich experiential significance. It is to remembered observation of physical models such as are depicted in Figures 1 and 2 that we appeal in using a lift, and the experience of interacting with these models that exposes their role in representing our knowledge of lifts. Needless to say, many attempts have been made to abstract the essence from such experiential representations in logicist approaches to knowledge representation (cf. [10]). In practice, there are many other pertinent issues connected with the experience of lift use that influence the behaviour of users, many of which are only tacitly acknowledged even in our account of the scenario described in Box 1. These issues include: the nature of the experience – the time it takes a lift to move between one floor and the next, the relative time it takes to ascend 3 rather than 4 floors, the acceleration of the lift car; the nature of the auxiliary observation that might have affected user responses, as determined by the status of the buttons of the lift, the visual cues to distinguish different floors, whether and how the users' room keys and the lift car buttons were numbered; the implicit conventions framing the lift design and use, such as whether the ground floor was deemed to be floor 0 or floor 1, the fact that the buttons in the lift were ordered top to bottom in a 1-dimensional array in correspondence with the five floors, or that a call button in the lift was visibly de-selected when the door opened at the corresponding floor.

As the above discussion illustrates, the EM model of the lift offers insight into real world state-transitions about which the automation of the perfect lift user can give no information. Interaction with Figure 2 offers a plausible explanation for corporate behaviour that is absurdly different from the ideal outcome to be expected, but it also shows that – taking appropriate observational and experiential factors into account – the individual actions of the users are not so very far from sensible as they might appear. Logic alone cannot account for the scenarios that lie near to normality 'in the neighbourhood of sense'. In exploring such scenarios fully, there is no alternative but to allow the modeller to engage as freely as possible in the state-changing activity. In the spirit of Gooding's definition of the term, an EM model can be interpreted as a *construal* [11]. In constructing such a model using EM principles, our aim is to embody the patterns of observables, dependency and agency that we deem to be characteristic of a situation. The product of this activity typically has a personal, provisional and particular character that is intimately connected with the ways in which we choose to interact with it. In all these respects, it is unlike a formal specification.

In developing a construal, the modeller is not only concerned with modelling state change as it is observed from an external perspective. A further analysis of the sequence of redefinitions that are made in realising the scenario in Box 1 shows how certain groups of redefinitions can be attributed to different agents. The movement of the lift and the manipulation of the lift doors are part of the automatic behaviour of the lift system itself. Entering and leaving the lift and the selection of call buttons inside and outside the lift are actions for which the lift users are responsible. To refine the model so that it better reflects our understanding of normal lift operation, it is appropriate to distinguish between realistic transitions and transitions that are pure fantasy, such as might involve simulating a user entering the lift before the door is open, or the lift jumping between floors. EM includes techniques for enriching the lift model so as to give special prominence to construals suited to different perspectives and purposes, some of which may include automated behaviour, or reflect several viewpoints concurrently. It is important to realise that the specialisation of an EM model does not involve excluding singular actions on the part of the modeller. In some circumstances, such as when the lift cable breaks, it may be appropriate for the lift to exhibit abnormal behaviours that can only be simulated by the intervention of the modeller in the role of a super-agent. The **EDEN** interpreter is designed to support the opportunistic interaction by the modeller that is required for this purpose, but whether such privileges for interaction are exercised is entirely at the modeller's discretion. A brief review of some relevant EM techniques is given here – for more details, consult the references at [17].

In elaborating construals, a significant role is played by an informal special-purpose notation, called **LSD**, that can be used to classify the observables in a situation with respect to an agent. The aim of such a classification is to document the roles that observables play in determining the interaction of an agent. Listing 1 is an **LSD** account of a person X in the role of a prototypical user of the lift depicted in Figure 2:

```
agent person (X) {
   state  loc[X]
   role liftuser {
      state   dest[X]
      oracle  open[*], dest[X], loc[X], pos[X]
      handle  loc[X], dest[X]
      derivate
         pos[X] is if loc[X]==0 then liftfloor else loc[X],
         LIVEliftuser[X] is 0<=loc[X]<=5
      protocol
         loc[X]==0 and open[dest[X]] and pos[X] == dest[X]
               -> loc[X] = pos[X],
         true  -> dest[X] = i ,                     (1<=i<=5)
         ...
   }
   ...
}
```

Listing 1: An **LSD** account of a prototypical lift user

In this account, the observables **loc[X]** and **dest[X]** refer to the location and destination floor of user X respectively. The values of these observables are defined according to the same conventions used to define **locX** and **destX** in the scripts discussed above. The location of X is classified as a *state* for X as a person, since observation of this location is meaningless in the absence of X, and the destination floor of X is classified as a state for X in the role of lift user, since the concept of X's destination floor is meaningful only when X acts in this role. Whether or not X is currently playing the **liftuser** role is reflected in the boolean value of the special observable $\mathbf{LIVE}_{\mathbf{liftuser[X]}}$, which is true or false according to the current value of

loc[X]. As a lift user, X has access to certain observables as *oracles*, either in the sense that she can observe them directly, or that she can observe them at certain times, and may retain some notion of their current values. She typically knows her destination floor. She also knows whether she is in the lift, or at a floor, as determined by the oracle **loc[X]**. If X is in the lift, she has some notion of what floor the lift is currently at, as represented by the observable **pos[X]**. The observable **pos[X]** is classified as a *derivate* because of the dependency that defines it in terms of the current lift position and the value of **loc[X]**. The observables over which X can conditionally exercise control are classified as *handles*: they include X's location and destination. The conditions under which X can redefine these observables are set out in her *protocol*. The two example privileges specified for X indicate that if X is in the lift, the lift is at her destination floor and the door is open, then X can step out of the lift, and that moreover X can change her mind about her destination floor at any time.

An **LSD** account is not in general to be interpreted as a specification, but rather as documenting the characteristics of observables as they are experienced by the modeller in the real situation, and (if the EM model is sufficiently convincing) as they are experienced in the associated EM model. In effect, an **LSD** account is intended to complement an artefact with which the modeller can interact, and is not to be viewed as an alternative form of representation. An **LSD** account can play a significant role in guiding the development of several different kinds of EM model for a multi-agent system such as a lift. These include:

Concurrent modelling of user perspectives on lift use. An **LSD** account can be seen as describing the system as viewed from the perspectives of its various users. These views can then supply the basis for a distributed EM model in which the personal construal of each agent is modelled as a client in a client-server configuration, and the corporate behaviour of the agents is developed by managing their interaction via the server. For instance, using a distributed variant of the **EDEN** interpreter, it would be possible to set up an EM model similar to that depicted in Figure 2 on a server, and to create EM models on four clients to recreate the interaction of the lift users X, Y, Z and I in the scenario as it appears from their four different perspectives. In such a model, there will be nothing that corresponds to the lift users X, Y and I from the perspective of user Z, since Z merely presses the call button on floor 3 then descends to floor 2. The models for X, Y and I will be based on different variants of the prototypical lift user account given above. For instance, the oracles **dest[X]**, **dest[Y]** and **dest[I]** have to be construed quite differently: user X has a correct perception that **dest[X]** is 4, but (when encouraged by I), supposes that **pos[X]** is 4 when it is in fact 3; user Y has an oracle to **dest[X]** which takes its value from the floor on which X gets out, and interprets **dest[Y]** as a derivate defined by **dest[X]+1**; user I has the number of the top floor of the building as an additional observable topfloor and interprets the oracle **dest[I]** as a derivate defined by **topfloor**. A distributed model of this nature can provide a more vivid reconstruction of a scenario than the basic EM model depicted in Figure 2, as has been illustrated elsewhere in our previous research in the reconstruction of historic railway accident developed by Sun [4].

Modelling the lift as a reactive system. An **LSD** account can be used in analysing the stimulus-response behaviour of automatic agents as they are construed to respond and act upon observables in their environment. Activity of this nature has a fundamental role in experimental science. It is also an important aspect of reactive systems development, where it is associated with the exploration and specification of context that precedes the design of control software. In applying EM principles to such activities, the most significant feature is making the connection between model-building from a personal subjective perspective and what is interpreted as objective observation of a system by an external observer. A full discussion of the technical issues involved is beyond the scope of this paper – it has been one of the primary concerns of the EM project as a whole (cf. [1,4,6,15]). The basic concept is that of treating the activity of automatic agents as it were being carried out by a human agent with the appropriate perceptions and state-changing capabilities. By way of illustration, the lift system can be viewed as automatically carrying out the sequences of redefinitions required to reset the call buttons and open and close the doors on visiting a floor to which it is called. The stimulus for this operation is the presence of the lift car at a floor that is associated with a selected call button. An **LSD** account of this role of the lift system would (for instance) identify oracles – such as the status of call buttons, handles – such as the status of the doors, and include an action to manipulate the doors and call buttons appropriately in its protocol. A similar analysis can be used to develop a protocol to prescribe the motion of the lift. The **EDEN** interpreter includes features, such as procedures that are triggered by changes to specified observables, that can be introduced into the EM model of the lift to automate the lift system protocols. Such features make it possible to implement a lift simulation through an incremental process of extending the EM model in which the modeller's role involves shaping the behaviour to accord with realistic observation and interaction. The delay after which the doors close can be adjusted, for instance, and the selection of buttons by lift users simulated by direct mouse actions. The significant point here is that the **LSD** account has no formal operational semantics, but documents actual interaction with a computer-based artefact for which behaviours can be developed in much the same incremental and empirical fashion that an engineer might construct a prototype. In this process of empirical refinement of the EM model, the scope for extension is open: simple extensions that feature more realistic lift motion, implement a lift scheduling algorithm, and introduce prototypical users based on the **LSD** account in Listing 1 can be found in the **liftBeynon2003** directory of the EM archive [18]. Further extensions for this model might involve introducing greater physical realism by way of modelling lift car velocity and acceleration, adding 3-dimensional visualisation, or linking the model to special-purpose hardware that could simulate the impact on the user of forces generated by the lift motion.

The above description and illustration of EM sets the scene for the discussion of Radical Empiricism that follows. To fully appreciate this discussion, it is most helpful to have some experience of the nature of EM as a practical activity. Without such experience, it is difficult to appreciate the conceptual distinction between traditional computer programs and EM models that is described in section 1.1. It is in particular important to realise that all the lift models discussed in this section are to be regarded as part of a single open-ended conceptual process of exploration that can be seen as resembling the exploration of a city. Each model is informally associated with a particular way of viewing a lift situation and of organising the transitions between one situation and the next, but each is apprehended – even when, left to itself, it is executing a particular pattern of behaviour – state-by-state, in such a way that the modeller can choose to intervene to redirect the experience perhaps with a view to its reinterpretation. In this respect, the individuality of the different lift models is not defined objectively, but only with reference to what experiences are coherent for the individual modeller.

2. Radical Empiricism

William James's Essays in Radical Empiricism [13] was first published in 1912. The potential relevance of James's 'philosophic attitude' to a discussion of alternatives to a logicist framework for knowledge representation is apparent throughout these essays. In 'The World of Pure Experience', for instance, when discussing the philosophic atmosphere of his time, James refers to 'a feeling that [the extant school-solutions] are too abstract and academic', and goes on to write: 'Life is confused and superabundant, and what the younger generation appears to crave is more of the temperament of life in its philosophy, even though it were at some cost of logical rigor and of formal purity'. Empirical Modelling is motivated by a perceived need to develop methods of computer-based modelling that can do more justice to life in its confusion and superabundance than can the rational formal accounts of agent interaction that underlie typical computer programs. The distinction between these different views of human agency has been illustrated above when contrasting the farcical scenario of lift use associated with Box 1 with the prosaic and predictable behaviour that is attributed to prototypical lift users in Listing 1. This section explores other respects in which the principles and techniques of EM can be related to thinking developed by James in [13]. Our overall aim is to make it plausible that James's philosophical stance supplies a foundational framework in which to examine issues that seem paradoxical from more conventional philosophical perspectives.

2.1. Philosophical foundations for EM

Some background motivation for our discussion can be found by thinking in general terms about what kind of philosophical foundations are appropriate for EM. One of the most characteristic activities in EM is the construction of a computer-based artefact (for instance, the lift model in Figure 2) that embodies patterns of observables, dependencies and agency that can be identified in an external situation to which the artefact refers (for instance, the specific use of the actual lift described in the scenario in Box 1). From a traditional computer science viewpoint, where there is an underlying presumption that all computer-based modelling can be accounted for by using the universal abstractions that rest ultimately upon the classical theory of computation, it is usual to propose that EM reduces to classical programming through a correspondence of the following general kind: an observable is a procedural variable, a dependency is a constraint relation, an agent is a sophisticated abstraction such as an active object. The renunciation of each of these proposed reductions has been the focus of special attention in our previous work (see for instance: the discussion of variables in [5], of constraints in [15] and of agency in [1]). Broadly, our counter to this suggestion is similar in all three cases: that the notion of 'observable' refers to a feature of a situation that is experienced as having an identity and current status or value; that a 'dependency' is more than a perceived abstract relationship between values of variables and expresses the modeller's expectations about the immediate consequences of changing the values of observables in a situation; that 'agency' entails a potential for action that is of a truly experimental character, in that the possibility of taking the action has not been preconceived, and that no prior commitment to the possible interpretation of its consequences has been made. What links each of these counterproposals is the context in which our interaction with the computer model is conceived to be occurring: a context resembling that through which I as-of-now experience the city of Coventry through the sight of my neighbour's house and the sound of his lawnmower.

Our rejection of a conventional interpretation for EM has another significant element that is closely linked to Brian Cantwell Smith's critique of classical computational semantics in [9]. The proposed reduction of an EM model to a classical program purports to attribute an abstract behaviour to the EM model, in accordance with the ways in which variables, constraints and active objects might be used to specify behaviours in a conventional approach to model-building. The development of an EM model does not rely upon the identification of an abstract behaviour that can be embedded into the environment through formal symbolic associations. This is to revisit our previous observation, that an EM model does not in the first instance specify an activity, like travelling about a city using the underground, for which – thanks to the traveller's training in symbol recognition, and the careful engineering and signposting of the underground environment – limited experience of the city is required. On the contrary, the EM model offers itself to the modeller as a state to be experienced, where the correspondence between the features of the model and those of its referent are to be directly established, explored and enhanced through interaction. This perspective has to be understood with reference to a philosophical position that assumes no given absolute knowledge, in the same spirit that (in my current context) I cannot be absolutely sure that all I remember of the city of Coventry will be there to experience when I set off to visit the railway station.

As Cantwell Smith observes in [9], the inadequacy of the classical view of computation is conspicuously exposed in emerging computing practice. The wide range of applications for EM principles that we have identified to date highlights both the potential of EM and the challenge of understanding its semantics. The pragmatic view of the semantics of modelling 'real-world objects' that serves well enough in traditional engineering design can to some extent be adapted to an exercise such as using EM to model an actual lift. The concept of a direct correspondence between experience of an EM model and the experience of the real-world situation it represents requires more justification when the EM model is a spreadsheet, and the situation a financial scenario. It is also more difficult to argue for a direct correspondence between observables for an EM model to represent a lift that is under design and has yet to be built. Further complications arise if we consider the status of the EM model of a lift that we might create in a virtual reality, where the relationships between observables are no longer subject to familiar physical laws and constraints. EM has been used to build models that have the experiential characteristics of the data structures (such as the heap) that lie behind standard algorithms (such as heapsort), and thereby to generate an environment in which to explore the design of algorithms [6]. In other contexts, it is apparent that the use of EM principles is not directly concerned with issues of external representation. For instance, our tools now enable us to design new definitive notations – in particular, for graphics – within the same paradigm that we use to construct models, and – for such a notation – successful design is concerned with how the syntax of the new notation affects the correspondence between the structure of a script and the graphical image that it produces on the computer screen.

A feature common to all these applications of EM is the construction of a computer-based artefact that in some respects has the qualities of an instrument (cf. [3]). The term 'instrument' is used here to express the idea that there is a reliable correlation between the interactions of the modeller with artefact and its associated changes of state. This correlation may be explicitly engineered by the modeller, or learned through skill acquired in mastery of the instrument. Explicit engineering is prominent in the case of a financial instrument such as spreadsheet, where the definitions of cells are contrived to express known relationships between observables. Skilful interaction is more prominent in relation to a musical instrument, such as the violin, where the tiniest nuances in the movement of the bow can be used to control the sound generated. In broad terms, the account of EM that we have sought to develop in our project represents EM models as instruments simultaneously under development and in use by the modeller, in which both explicit engineering and skilful interaction have a role to play. Through explicit formulation of dependencies and through experimental redefinition, the modeller develops an understanding of how interactions with an EM model reliably effect changes to its state. In this account of EM, the interaction with the instrument does not acquire meaning as a result of a complex abstract process of off-line decoding; it directly evokes a parallel experience because of the perceived similarity between the effects of interaction with the instrument and the familiar effects of interaction in another context. It is in just this manner that the movement of geometric elements of the simple drawing in Figure 2 evokes familiar interactions with an actual lift.

The correspondence between one experience and another that underpins EM is different in character from the realistic modelling of behaviour and appearance that is commonplace in routine prototyping of software and engineering systems. For instance, our primary concern is not with simulating the lift dynamics as accurately as possible by analysing the forces acting on the lift in detail and applying Newton's laws, nor with developing a virtual reality model that imitates the user's visual and sensory experience as faithfully as possible. As the lift model illustrates, EM can serve to establish such similarities, but the connection between an EM model and the situation to which it refers is more direct and primitive in nature. The correspondence between interactions with the EM model and interactions with its referent is itself a matter of immediate experience, subject to confirmation, exploration or possibly even refutation by the modeller 'as of now'. A correspondence of this nature is rooted in the notions of 'observable', 'dependency' and 'agency' as they have been discussed above: it has a concrete and dynamic rather than abstract and static quality. In immediate experience, the agency of the modeller has the capacity to confirm or confound expectations, to create or destroy observables, to make or break dependencies. In these respects, it resembles the agency of the experimental scientist, whose actions can be used to test her construal.

The potential role that language and symbolic conventions can play in this context is a delicate issue. It is self-evident that our apprehension of correspondences in immediate experience can be mediated by language, as when we respond to the value in the column of the spreadsheet that is headed 'balance' or 'profit'. It is quite another matter to argue – as some philosophical traditions appear to do – that language plays an essential role in all correspondences between experiences. A core idea in EM is that not all correspondences between one experience and another can be established by symbolic conventions – at some level, correspondences must be made through direct experience without reference to language. The expectations of the experimental scientist may well entail much sophisticated theory, but the primitive correspondences that provide the grounding for such theory are arguably beyond words and equations. Experience in applying EM principles and tools suggest the further hypothesis that agency, dependency and observation have a fundamental role in such primitive correspondences.

2.2. Radical Empiricism from an EM perspective

Experience of EM, and reflection about its semantics, endorses a philosophical position that has strong points of connection with William James's Radical Empiricism. The primitive correspondences between experiences at the core of EM are necessarily personal experiences, and in the first instance are associated with subjective and provisional knowledge. This establishes priorities that are in line with those of RE: to account for logic and theory in terms of observation and experiment, rather than to account for observation and experiment in terms of logic and theory. Practical experience of EM reinforces this perspective, demonstrating how EM can be accompanied by transitions from personal to public, subjective to objective, and provisional to assured (cf. [1] for more discussion of these issues). This section discusses how some key ideas of RE are helpful in elaborating on what is involved in EM.

In [13], James remarks that, in RE, "the relations that connect experiences must themselves be experienced relations, and any kind of relation experienced must be accounted as 'real' as anything else in the system". One of his primary concerns is that traditional empiricists emphasise the disjunction in experience "leaving things permanently disjoined", whilst the rationalists remedy this disjunction "by their Absolutes ..., or whatever other fictitious agencies of union they may have employed". For James, both conjunctive and disjunctive relations should be deemed to be given in experience. Amongst these, he includes "the most intimate of conjunctive relations, the passing of one experience into another when they belong to the same self".

James's outlook helps to explain the difficulty we have encountered in understanding and communicating the nature of EM models. The typical computer scientist has a natural desire to attribute a characteristic set of discrete states and behaviours to the EM model of the lift in Figure 1, and to view it as a structure or system. In the mind of the modeller, the character of the EM model is more elusive. We can experience the state of the lift model as it is depicted in Figure 1, but interpret this in relation to all the other experiences of the states that have brought the model to this point. By the same token, what we experience in the current state of the model stands in an open, yet to be explored, relation to other experiences we can get by interacting with the model. What we understand by the model is an unfolding conjunction of experiences that cannot be circumscribed but is apprehended as a single relation.

An important aspect of James's analysis is the implicit emphasis it places on the authenticity of the personal experience of the observer. This distances RE from the perspective with which empiricism is ordinarily associated, where the 'reality of the given world' is seen as the primary source of knowledge. As our practical experience of EM has shown, it is implausible that we can give a good account of EM without regarding reality as constructed by experience. Though it is convenient in describing EM to adopt everyday terminology, and speak of 'the modeller's state of mind', and 'the real-world situation', this should not be understood as subscribing to a Descartian dualism. This accords with James's observation that "subjectivity and objectivity are affairs not of what an experience is aboriginally made of, but of its classification. Classifications depend on our temporary purposes ... ". In observing the development of the EM model of the lift, we are led to think quite as much about the evolving state of mind of the modeller as about realistic changes to the state of the actual lift, whether these take the form of a movement from one floor to the next, or adding labels to the call buttons in the lift car. This is keeping with our perception in EM that, contrary to the received view that a computer model of lift should only – perhaps even can only – be constructed with a specific goal and behaviour in mind, our EM model of the lift is somewhat neutral to purpose. Certainly, the experiences of the lift model that would be generated through the interaction of modellers acting in the roles of lift users, lift designers, lift analysts, or story tellers would be quite different, and reflect many different kinds of conjunctive relation in their minds.

Of particular relevance to EM is James's contention that "the first great pitfall from which [RE] will save us is an artificial conception of the relations between knower and known" [13]. This issue relates directly to the discussion above concerning how correspondences between experiences are established. The artificiality to which James alludes here stems from what is perceived by other philosophers as the 'indefeasibly dualistic' structure of experience [13]. For James, there is no such duality: "All the while, in the very bosom of the finite experience, every conjunction required to make the relation intelligible is given in full" [13].

James's philosophical position is helpful in understanding the nature of EM. As has been argued above, the fact that the EM model of a lift stands in a special relation to an actual lift (viz. in what James characterises as 'the relation between knower and known' [13]) is not illuminated by classifying the one as a 'mental model' and the other as a 'real-world object'. What matters is that they are two portions of experience that are related in a way that is itself experienced by the modeller (cf. [13]). This leads us to a characterisation of the modeller that matches our experience of EM activity well: that of an agent who generates an experience that knows another. In the same spirit, EM principles and tools can be seen as assisting the generation of this experience. In this context, our choice of the word 'model' belies our faithfulness to James's account of knowing, as it is commonly associated with what James rejects by way of "representative theories".

The simplicity of the relation of knowing, as James describes it, is consistent with our experience of how readily EM models can be combined and reinterpreted. James rejects the duplicity of experience that distinguishes 'consciousness' from 'content' : "Experience ... has no such inner duplicity; and the separation of it into consciousness and content comes, not by way of subtraction, but by way of addition – the addition, to a given concrete piece of it, of other sets of experiences, in connection with which severally its use or function may be of different kinds". This explanation accounts for the way in which the object of study in one exercise in EM seamlessly becomes a part of the EM model in another; for the fact that an EM artefact, like an artefact in bricolage, can be developed without a referent in mind; for the ambiguity about what features of an EM model serve a significant representational role (cf. the bold lines indicating the lift doors, the size of the call button panel in the lift, the presence of the roofs in Figure 2). In each of these contexts, the precise character of the EM model is only shaped by the nature of the interaction of the modeller as it unfolds at her discretion – in particular, by how the values of observables are interpreted and changed, and how these changes to observables are interpreted. By way of illustration, a student who used EM to simulate bread baking in an oven chose to represent the oven by borrowing a simple line drawing to represent the floor plan of a room (see **`roomviewerYung1991`** at [18]), and labelled his simulation by substituting 'Bread baking simulation' for a warning message that notified the user when a table obstructed the door. This meant that the identifying label could be changed by relocating an invisible table – a functionality that the student did not document, and was unacknowledged in his personal interpretation of the model.

Though both RE and EM share a central concern for rooting knowledge in personal experience, their agendas are quite different, and explicit connections are hard to make. Despite this, RE is helpful in developing a deeper understanding of the primitive concepts of EM, for which formal logical foundations cannot be supplied.

James's account of pure experience provides a most appropriate setting in which to explain the notion of an observable in EM. The appropriate sense of being an observable in EM is 'having an identity' and 'having a value that can be directly experienced', where the term 'is directly experienced' refers to the capability of the human interpreter in the given context to apprehend immediately. There is a most significant distinction between this notion of an observable and what traditional empiricism might deem to be an observable. To the infant, the symbol '2' on the lift button is a mere pattern of sensation that signifies nothing; to a young child, it will be associated with the idea of 'some pair of objects'; to the competent lift user, it is known to refer to a specific floor. James's account of knowing dispenses with the idea that "seeing the symbol '2'", "thinking of two floors" and "imagining going to the second floor" are categorically different kinds of experience whose association in the mind of the lift user must be explained by "fictitious agencies of union". When situated in the lift, the lift user experiences the conjunctive relation that connects all three elements of his experience of the button labelled '2'. The uniform way in which observables are treated in an EM model, regardless of the level of sophistication of the observation involved so long only as it is immediate (cf. labelling the call button by 'II', or "The smallest prime number"), is consistent with James's outlook. The fact that in EM, as in life, the scope and significance of such observables is dynamically established as experience is acquired – perhaps from moment to moment, as a child might be taught *in situ* to connect the symbol '2' with a pair of floors – is also in keeping with James's conception of knowledge: "Why insist that knowing is a static relation out of time when it practically seems so much a function of our active life?" ([13], p.75).

In EM, there is an intimate relationship between the notions of dependency and agency. Dependency is in effect a way of binding together all the consequences that are deemed to be an integral part of a single action. In EM models, this concept is realised practically through modelling with definitive scripts (cf. [15]), where a typical atomic action involves redefining an observable or introducing / removing a cluster of observables. An analysis of the notion of dependency reveals a number of respects in which it is related to agency. Whether a dependency is identified in a particular context in general depends on the perspective and agency that the modeller has in mind. The position of the lift car might be seen as dependent on the position of the lift cable, but this view is entirely appropriate only if we assume an idealised mechanical model (e.g. making no allowance for the initial extension of the cable under load), discount the discrepancies between the exact positions of the lift car within the tolerances allowed in the design of the lift shaft, and decide not to model the behaviour of the lift cable at the atomic level. A naive mechanical model of the dependency may need to be modified if the possibility of thermal effects is admitted, and is no longer appropriate if the lift cable is presumed to become slack or to break. Nor is dependency necessarily associated with synchronisation of change; it has more to do with what we informally understand by causation, as when a doctor declares that a living person has been fatally wounded. This leads us to view dependency as framing what consequences of an action are an inevitable effect of the action, and cannot be allayed by the intervention of any other agent. There is also an important distinction between the notion of dependency that is often invoked in analysing concurrency in traditional procedural programs, which broadly relates to how the current value of a variable depends on previous assignments to other variables, and the indivisible characteristic of the dependency that in general binds many changes to one atomic change in EM.

There is no explicit reference to dependency in James's account of pure experience. However, amongst the conjunctive relations he lists "relations of activity, tying terms into series involving change, tendency, resistance and the causal order generally". One of the most significant features of EM is that it provides an alternative to the classical procedural model of state, where the values of variables are treated as discrete and independent: the dependencies in a script are not assertions about values alone, but embody expectations about responses to interaction that are themselves a part of state. To the extent that these expectations are relations between observables that are directly experienced, it seems appropriate to classify dependencies as conjunctive relations of activity. A plausible analogy likens the disjoint terms of traditional empiricism to the discrete variables of procedural programming, and the conjunctive relations of Radical

Empiricism to the dependencies in modelling with definitive scripts. The elaborate programming mechanisms that are required to maintain dependencies between variables in a procedural representation of state echo James's comment about the complex ways in which rationalists remedy disjunction.

James's perspective on the relationships between transitions in experience is helpful in clarifying the interpretations of interactions and dependencies in EM models. In the EM model depicted in Figure 1, for instance, the lift is conceived as 'moving from one floor to the next' though there is no explicit attempt to model continuous lift motion. Within this naive lift model, the location of the lift user is thought of as defined by 'in the lift', 'on a particular floor' or 'not in the vicinity of the lift'. This 'user location' cannot be identified with an actual physical location, but is nonetheless deemed to be a meaningful observable. In this context, just exactly where the lift user is standing in – or in the vicinity of – the lift, or what posture they adopt, is not significant. From this, it may appear that the dependency by which "the position of the user is determined by that of the lift" is to be construed in a different way from the dependency that links the position of the lift car to that of the end of the supporting cable. The correlation between the movement of the end of the cable and that of the lift car is far more exact, and is easily interpreted with reference to atomic 'infinitesimal' change. To conceive the movement of the lift between one floor and the next (as modelled by `_liftfloor++;`) as an atomic change is to presume that no matter what the lift user does within the lift, they will be moved with the lift from one floor to the next. In the context, the relationship between the continuous motion of the lift and the discrete movement from floor to floor can be interpreted as illustrating what James identifies as substitution, whereby "an experience that knows another can figure as its representative, not in any quasi-miraculous 'epistemological' sense, but in the definite practical sense of being its substitute in various operations, sometimes physical and sometimes mental, which lead us to its associates and results". The natural way in which these two experiences of a lift can co-exist subject to appropriate assumptions about the modeller's motivation is illustrated in the extension of the EM model mentioned above (cf. **liftBeynon2003**). For this purpose, we need only introduce a new observable **liftfloorheight** that (say) can assume integer values from **1** to **5*N** (where **N=10** for instance) corresponding to lift positions at or between floors, and define **_liftfloor** to be the integer part of **liftfloorheight/N**. Such a model is appropriate for many practical purposes, but would not serve for a murder mystery in which we might well be asked to imagine that a person ascending in the lift from floor 3 fails to arrive at floor 4, or meet the need the engineer may have to consider non-integral positions for **_liftfloor** for a lift that is malfunctioning.

As the above discussions indicate, the character of an EM model is moulded by the way that the modeller chooses to interact with it, and how this interaction reflects the modeller's evolving presumptions about the agency to be taken into consideration. In much the same way that each visit to Coventry serves both to fulfil familiar expectations and functions and to introduce what is changed or was previously unknown, interaction with an EM model involves both creation and use. In contrast, the formal specification of computer models requires a commitment to modes of agency and interpretation (i.e. in 'creating' the model) that cannot be reappraised in the subsequent course of interaction with the model (i.e. in 'using' the model). This accounts for a fundamental difference in orientation between EM and traditional computer-based modelling. In the former context, our agenda is 'finding a good construal', for which the key question is 'given that this is what we're interested in achieving, what assumptions about agency in the world is it necessary and appropriate to make?'. In the latter context, our agenda is 'optimising our use of known resources for familiar purposes', for which the key question is 'given that this is the agency in the world, how do we best exploit it?'. The premise for the former agenda is ignorance of the world, and for the latter, knowledge of the world.

From a philosophical perspective, RE can be viewed as endorsing this shift in engineering priorities that EM promotes. In "The Experience of Activity", James [13] remarks: "... the healthy thing for philosophy is to leave off grubbing underground for what effects effectuation, or what makes actions act, and to try to solve the concrete questions of where effectuation in this world is located, of which things are the true causal agents there, and of what the more remote effects consist". The thrust of agent-oriented analysis in EM is arguably well-aligned to James's recommended agenda in its concern for identifying agency, attributing state-change to agents and interpreting agent interaction in global state-based terms. In contrast to the mainstream traditions of research on agent-oriented modelling and programming, EM favours a concept of 'agency' that is more than any circumscribed preconceived rationalised interaction, and is oriented towards a pragmatic dynamic shaping of construals. This emphasis is consistent with James's recommendation that, in examining 'the real facts of activity', and arbitrating between whether our actions are programmed by a higher authority, are an expression of free will, or emerge from the corporate behaviour of more primitive agents, we should evaluate our responses to the question "Whose is the real activity?" by asking "What will be the actual results?" [13]. In principle, EM provides a practical framework within which to tackle this agenda, supplying environments in which to explore 'possible construals' and to situate the negotiation of meaning.

As the discussion of the possible extensions of the EM lift model illustrates, the virtue of such an environment is that it is a source of experience that is rich to the point of incoherence. As James observes: "Experiences come on an enormous scale, and if we take them all together, they come in a chaos of incommensurable relations that we can not straighten out. We have to abstract different groups of them, and handle these separately if we are to talk of them at all." [13]. In the EM model of the lift, we can accommodate the possibility that the lift scenario described in Box 1 occurred not because – as I hypothesised – there was a user Z, but simply because the lift control was faulty. We can dramatise I's predicament in deciding whether to get out of the lift on floor 4 without needing to resolve the logical inconsistencies in his perception and pursuing these to their contradictory conclusions. In contrast, conventional programming, like traditional empiricism, has no satisfactory way to handle the incompleteness of knowledge. Without such means, it cannot do justice to James's conception of ourselves as 'virtual knowers', or his observation that "To continue thinking unchallenged is, ninety nine times out of a hundred, our practical substitute for knowing in the completed sense."

Conclusion

The perspective on the nature of knowing that RE and EM endorse has direct practical relevance for current trends in knowledge management (cf. [7]). To date, the successful application of computers in management has relied to a large extent on exploiting what can be objectified and expressed in formal notations (as in a relational database, or an expert system). As we seek to make yet more sophisticated use of computer technology, and as this technology itself potentially embraces broader aspects of the total business experience, so the limitations of widely accepted philosophies of information science are being exposed. The problems we face are epitomised by the difficulties of negotiating ontologies and standardising formal representations and procedures for communication and re-use. This paper attributes these problems to the barriers that traditional philosophical frameworks for cognition and computation place between words and concepts and the experience that informs them.

The fundamental role that experience plays in informing concepts, as emphasised by both RE and EM, is patent in the everyday situations within which knowledge management has to function. Consider the experience that leads us as we grow up to identify one and the same entity as "a person", "a man", "a doctor", and "a paediatrician", or to appreciate the distinction between 'learning to speak French' and 'being French'. In attempting to capture such concepts in formal ontologies without reference to experience of artefacts, it is arguably impossible to do justice to the role of tacit knowledge, and to reflect the subtle nuances concerning the perceptibility, reproducibility and stability of the underpinning experience. As we aspire to provide computer support for 'experience management', and accommodate such broad perspectives on knowledge such as 'mimetics' affords, there is ever more need to take explicit account of personal experience and to understand this in relation to our interaction with the natural world and with other people. RE in conjunction with EM potentially offers a philosophical and practical framework within which to give proper prominence both to direct experience and the personal stream of thought, and to the distinctions between knowledge as 'socially accepted' and knowledge as 'experientially validated'.

For the sceptical reader, a major intellectual objection to engaging with the thesis of this paper is that it represents RE and EM as essential alternative fundamental philosophical and computational perspectives. To acknowledge that concepts such as 'theory', 'reasoning' and 'reality' have their significant place within this perspective is not enough to deflect such scepticism. In this context, a key issue is that we have become inured to interpreting our interactions with computing technology in a narrow sense that we would (arguably) not entertain as appropriate in relation to other experiences, such as playing or listening to a musical instrument. Indeed, the influence of computational theory on cognitive science has been such that there is a tendency towards construing all interaction as a form of computation. James's characterisation of the nature of knowing is of crucial significance in this connection: it identifies the relationship between one experience and another not as rationally apprehended and explicable with reference to preconceived criteria for similarity, but as itself given in experience. For the sceptic, a useful first step towards appreciating this view of what it is to know is to distinguish the EM model of a lift from an orthodox simplified formal model of a lift with a prescribed and preconceived interpretation and functionality. In the longer term, in the author's opinion, the possibility of wider acceptance of RE and EM as a new framework for knowledge management does not rely upon such intellectual assent: it potentially offers such benefits in terms of the quality of the results and experience it can offer that its practical application will be justification in itself.

Acknowledgements: I am indebted to all the participants in the EM project for their various contributions and in particular to Cheryl Sidebotham for the basic EM lift model. I am especially indebted to Steve Russ for his role in promoting my interest in the philosophical aspects of EM and its application to issues of knowledge management, and for presenting an early version of this paper at the Workshop on Philosophy and Knowledge Management at Lucerne in April 2003. I also wish to thank Bertin Klein and Hans-Joachim Petsche for their help and encouragement.

References

1. W. M. Beynon, Empirical Modelling and the Foundations of AI, *Computation for Metaphors, Analogy and Agents*, LNAI 1562, Springer, 322-364, 1999
2. W. M. Beynon, EM for Educational Technology, Proc. Cognitive Technology '97, University of Aizu, Japan, IEEE, 54-68, 1997
3. W.M.Beynon et al, The Computer as Instrument, Proc. CT 2001, LNAI 2117, Springer 2001, 476-489
4. W.M.Beynon, P-H Sun, Computer-mediated communication: a distributed Empirical Modelling perspective, Proc. CT 1999, San Francisco, August 1999, 115-132
5. W.M.Beynon, S.B.Russ, The development and use of variables in mathematics and computer science, *The Mathematical Revolution Inspired by Computing*, IMA Conf. Series 30, 25-95, 1991
6. W.M.Beynon, J.Rungrattanaubol, J.Sinclair, Formal Specification from an Observation-Oriented Perspective, JUCS, Vol. 6 (4), 2000, 407-421
7. W.M.Beynon, S. Rasmequan, S.B.Russ, A new paradigm for computer-based decision support, Decision Support Systems, vol 33, no2, 2002, 127-142
8. G. Bird, William James, Routledge and Kegan Paul, 1986
9. B. Cantwell-Smith, The Foundations of Computing, http://www.ageofsig.org/people/bcsmith/print/smith-foundtns.pdf (accessed 7/8/03)
10. M.L.Ginsberg, D.E.Smith, Reasoning about Action I: A Possible Worlds Approach, Artificial Intelligence 35 (1988), 165-195
11. D. Gooding, Experiment and the making of meaning, Kluwer 1990
12. I. Jacobson, M. Christerson, P. Jonsson, G. Overgaard. Object-oriented Software Engineering: A Use-Case Driven Approach, Addison-Wesley 1992
13. William James, Essays in Radical Empiricism, Bison Books, 1996
14. C. Levi-Strauss, The Savage Mind, University of Chicago Press, 1966
15. J. Rungrattanaubol, A Treatise on Modelling with Definitive Scripts, PhD Thesis, Computer Science Department, University of Warwick , April 2002
16. M. Turner, The Literary Mind, Oxford University Press, 1996
17. The EM website at URL http://www.dcs.warwick.ac.uk/modelling/
18. The EM archive at URL http://empublic.dcs.warwick.ac.uk/projects/