Chapter 3 – An experiential perspective on learning

3.0 Overview of the chapter

In this chapter, we set out the main challenges that are faced in using computers for learning and discuss the broad framework on learning that informs the research in this thesis. In the field of computers for learning, there are two perspectives that need to be considered: that of the educationalist and that of the computer specialist. Each typically has different concerns, and it is the successful marriage of these concerns that will yield positive results. In this chapter, we shall set out a major claim of this thesis: that in respect of learning through building computer models, EM is in general more suitable than other approaches because model construction and development of domain understanding are intimately linked. The latter part of the chapter discusses the theme of EM and learning in detail to conclude that the principles of EM model construction support a wide variety of learning activities. We introduce an Experiential Framework for Learning (EFL) that describes how learning activities are broadly related and are rooted in our personal and private experience. We shall argue that the EFL supports a general view of learning and that the principles of EM are well aligned to supporting the range of activities described in the EFL.

3.1 Challenges for computers for learning

In this thesis, we consider the challenges that are faced in successfully using computers for learning. There are a range of important issues that span computer science and education. The perspectives of the specialists in these two fields differ to such an extent that it is hard to identify a common agenda. Achieving a closer collaboration between all the participants in using computers for learning, namely educationalists, computer specialists and learners, is necessary for successful computer-based learning [diS97b]. In examining the issues from the perspectives of the educationalist and the computer scientist, we consider two rather different ways in which computers are used to support learning through:

- **the use of educational software**. In traditional educational software, a learner interacts with the software but does not modify it. This way of using computers for learning embraces different learning paradigms, including both instructionist and constructionist approaches (these terms are introduced in section 4.1).
- **building of computer-based models**. There are a number of areas in which computers are currently being used for building models to aid learning, including: financial models built in spreadsheets; scale models built in engineering; and prototypes built for software engineering. The primary objective in building these types of models is to acquire domain related knowledge. These models may be constructed using a variety of different programming languages and development environments. Tools for computer-based model building – often developed with children in mind – include programming languages and development environments.

Where the use of educational software is concerned, there are many issues and challenges relating to its development. The perspectives of the educationalist and the computer specialist are traditionally different. When constructing software, the computer scientist is typically concerned with issues such as the usability of the software, requirements specification, and the choice of programming paradigm. The educationalist's focus is on the quality of the learning activities that are supported, and the actual computer implementation is a secondary concern. In the field of educational software – where educationalists and computer specialists have a common interest – the primary concern is typically that the software is 'as easy as possible' to adapt for use in different learning contexts. The demand for adaptable educational software stems from the fact that learners often have different needs, abilities and approaches to learning. Learners' different requirements arise dynamically in the learning context and teachers would ideally like to respond to individual situations as they occur. The particular features of the culture within which learning takes place may also require software to be adaptable [RB02]. The concept of being 'as easy as possible' to adapt is not well defined – we shall take this to mean that a non-computer specialist (such as a teacher) can adapt the software themselves, or that a computer specialist can adapt the software quickly with very little effort. We acknowledge that this requirement is particularly difficult to satisfy given the demands on teachers' time. There may also be a conflict between the expectations of the educationalist and the computer specialist where the qualities of the software are concerned. The focus in software development is on providing polished software products to users [Bey01]. In contrast, a teacher may appreciate the benefits of 'do-it-yourself' software that may lack the sophistication of a commercial product but allows a higher degree of ownership, engagement and adaptability. Such qualities were in evidence in the 'cottage' educational software industry of the 1980's [Ker92]. Given the current focus on object-oriented principles in software development, the favoured way of trying to develop such adaptable software is through object re-use. The merits of EM as an approach to the development of educational software are discussed and illustrated in detail in chapters 5 and 6.

The issues and challenges in using computers for learning through model building are considered in chapters 3 and 4. In such use of computers, we are not concerned with the incidental learning of programming knowledge that is a necessary part of the process, but rather the way in which characteristics of the model construction approach assist or hinder the learning of domain knowledge. In chapter 2 we concluded that the synergy between learning domain knowledge and constructing a computer model depends on the paradigm that is used in the model construction. For instance, in building a spreadsheet – and in EM – the modeller's attention is focused on the semantic relation β that is intimately linked to understanding domain knowledge.

The above discussion motivates the two central claims of this thesis:

1) **EM for computer-based model building**. In respect of learning through the construction of computer-based models, EM is in general more suitable than other approaches because model construction is more intimately linked to the development of domain understanding. This claim is developed in chapters 3 and 4.
2) **EM for the development and use of educational software**. In respect of learning through the use of computer-based models, if EM is used for the development of educational software (following the principles of software development outlined in previous theses [Nes97, Ras01, Won03]) then this software will have qualities that are well-suited to the educational needs of learners and teachers. This claim is discussed in detail in chapters 5 and 6.

In justifying the first claim, we first address the question of how learning and EM are connected (in sections 3.2 – 3.6), and then review learning through model building in relation to established educational theories (in chapter 4).

In section 3.2, we discuss what we mean by 'learning' and delineate the scope of the learning activities that we aspire to support in computer-based exploratory modelling. We informally describe some types of learning that we undertake in the world, including learning skills, learning about situations and learning about artefacts. We use these examples to motivate a broad framework on learning encompassing a variety of learning activities. We have termed this framework an Experiential Framework on Learning (EFL) because it reflects the way in which learning is rooted in our personal and private experience. Within the EFL, we can analyse how computer-based modelling tools can support a wide-ranging view of learning. From section 3.3 onwards, we introduce the principles underlying EM and describe the extent to which it supports the learning activities described in the EFL.

3.2 A perspective on learning

In this section, we discuss learning in detail to explore various types and applications of learning that motivate the learning framework we use in the remainder of the thesis. It is evident that learning on computers cannot replicate the enormous diversity of learning that can take place 'in the world' at the present time – but a major aspiration of using computers for learning is to apply computers in as wide a range as possible of learning situations. In education, critics of children's use of computers for learning stress that the virtual nature of computer reality leads children to less rich and engaging learning experiences (see e.g. [Tal95]). This has been one justification for arguing that young children should not use computers in learning [Hea99, AFC00], and that educational systems based on personal engagement with the world and other people are more beneficial (see e.g. [Opp97], [Opp03], Waldorf education [Aep86], Alliance For Childhood [AFC00]). It cannot be denied, however, that the interactive nature of the computer offers advantages over building real-world artefacts, in that experimentation can often be more easily performed.

It would be impossible to give an authoritative view on exactly what learning is: major debates in psychology centre on how we learn, what knowledge is, and what are the best ways to learn [HO96]. It is the difficulty of giving an objective definition of learning and how it occurs that motivates us to discuss the perspective on learning that informs this thesis. To this end, we shall first describe some examples of learning in abstract terms, then complement this with longer discussions of the learning of skills, learning about artefacts and learning about situations. Our framework for learning, the EFL, is intended to provide a general setting within which diverse learning activities can be discussed.

The definition of the verb to 'learn' from the Oxford English Dictionary is:

> 'gain knowledge of or skill in by study, experience, or being taught' [OED97].

This definition is so broad as to encompass self-directed learning and teacher-directed learning. These two activities are representative of constructionism and instructionism respectively (see section 4.1). In this thesis, the term ‘learning’ is used in a broad sense to embrace any kind of activity that enables us to adapt our future behaviour. One particular difficulty with this broad definition is in identifying separate learning activities that may be taking place concurrently within one and the same situation. For instance, in constructing a computer model, I may be learning new insights about the model’s domain, peculiarities of the modelling tool and better ways of organising my model. A wide variety of different types of learning may also be concurrently represented:

i) **learning as equipping us to respond to questions of fact** – e.g. what is the capital of England?

ii) **learning to understand a concept** – e.g. what is taxation and how is it applied?

iii) **learning about social situations** – e.g. learning the roles and responsibilities of signalmen and drivers in the safe passage of trains through a tunnel.

iv) **learning a physical skill** – e.g. playing the piano, or learning to row.

v) **learning about a real-world artefact** – e.g. learning about the controls of a new watch.

The distinction between these different types of learning is manifest in the different ways in which we would assess whether learning has taken place. From one perspective, the knowledge that two times three is six is a matter of fact. One kind of activity that informs such knowledge is rote learning of multiplication tables. From another perspective, knowing that two times three is six entails knowing the meaning of ‘two’, ‘three’ and ‘times’ and being familiar with many concrete examples of how ‘times’ occurs in practical situations. Such illustrations show the subtlety of the distinctions between different notions of learning and the difficulty of expressing them formally. To address this subtlety, we shall introduce a framework within which to organise the activities associated with different types of learning.

To motivate this framework, we now describe some informal examples of learning. We shall do this with reference to contexts in which the learning process is complex, such as learning a new skill (e.g. a new sport or a musical instrument), learning about an artefact (e.g. a digital watch) or learning about a situation (e.g. being a restaurant manager). In the discussion that follows, it is useful to refer to a learning situation you have been in and reflect on the learning activities that you undertook. I will use two examples of learning skills: firstly that of learning how to row in a boat (an activity that I first attempted to learn two years ago); and secondly that of learning to play the piano (as we discussed in a previous paper [RB02]).

3.2.1 Learning skills, learning about artefacts and learning about situations

Watching an expert performing a skill, honed to near-perfection through innumerable learning experiences belies the difficulty of undertaking it for yourself. Any task can seem easy when performed by somebody who has been through an extensive learning process to reach their advanced standard. The first time you sit in a boat and try and row, or sit at a piano to learn how to play is a daunting experience. You have none of the necessary skills; you have acquired none of the language of the domain. Where conscious learning of a skill is involved, your primary source of knowledge is an initial idea of how you think the skill is executed derived from previous experience of observing others. Howell claims that learning a skill passes through a succession of four stages [How82]:

1) Unconscious incompetence – we don't know that we don't know how to do something.
2) Conscious incompetence – we know we want to do something but we don't know how to do it.

3) Conscious competence – we can do something but only by concentrating fully on it and by focusing on individual parts of the task.
4) Unconscious competence – we know how to do something and can do it automatically whilst concentrating on other things.

A fifth stage has been proposed by Pike [Pik89], namely conscious unconscious competence, which is taken to mean an ability to do a task without thinking about it, yet retain a level of awareness of how it is done that enables you to teach the skill. Performing a skill initially requires a commitment of time and energy to learn the 'basic' skills of the discipline, such as the rudiments of the rowing stroke or performing scales on a piano (cf. moving through conscious incompetence). Through our interactions we move from having to consciously think about each and every element of the stroke to a level where it is a natural, ingrained movement that we can perform without conscious thought, (cf. progressing from stage 1 to stage 4 of Howell's stages). This leaves our minds free to engage in more advanced thoughts, such as 'are we rowing at an appropriate speed and stroke rate to win this race', or 'have we started to play this piece of music at a tempo that is feasible for the most difficult passages'.

In learning, we gradually build up experience of important features of a situation and how they are dependent on each other. For example, the balance of a rowing boat depends on the positions of the oars in such a way that if the heights are the same then the boat is balanced and 'runs' along the water more smoothly. Understanding the 'feel' of the boat running requires experience and experimentation in order for it to become repeatable (cf. Howell's stage 3). Each individual rower (through personal experience) will learn to 'feel' a good stroke and be aware of relationships between the position of the hands on the oar and the feel of the blade in the water. This requires observation of factors such as 'is the blade in the water?', 'is it at the right angle?' and 'am I putting an appropriate amount of force on the oar for its position in the water?'. Over time, experienced oarsmen gain a comprehensive understanding of how their individual movements affect the run of the boat and the performance of the crew as a whole.

During the entire learning process the learner is engaged in non-verbal communication through the use of artefacts and physical hands-on demonstrations. The coach of a rowing crew will demonstrate the particular stroke pattern they are looking for and (on a land-based simulator) will physically control the oarsmen, isolating each part of the stroke to perfect it. The learning of skills often involves a coach who will demonstrate how to perform an aspect of the skill, communicating through physical manipulation, pictorial representations and the use of domain-oriented language.

The novice rower meets language that they have not encountered before that refers to either directly observable features of the skill or skill domain, or more complex culturally situated features of the environment that are meaningful only with reference to that skill. For example, in rowing, there are simple concepts such as ‘bow side’ and ‘stroke side’ that refer to sides of the boat that are accessible to a beginner. These are directly observable and have a definite meaning. There are then terms such as ‘drive’, ‘recovery’, ‘quarter slide’ and ‘backstops’ that are particular positions within the rowing stroke, or particular phases of the stroke. There may be some disagreement over the precise meaning of these terms, but their meaning is unambiguous enough to enable rowers with a modest amount of experience to communicate. However, other terms are imbued with meaning that is more difficult to directly apprehend. For example, the terminology of different oar pressures is a purely individual matter. A coach may ask for a particular training interval to be completed at ‘half pressure’ but it is almost certain that individual rowers (and the coach) will interpret this term in different ways. It is also likely that an individual’s interpretation of such a term will change with their experience. As a rower gains confidence and can apply more pressure to the oar, the concept of half pressure will change.

The specialist language of a domain can also have exceptionally broad and rich cultural connotations. For example, in musical performance – as in rowing – there are different types of language that are appropriate at different competency levels and are directly correlated with the experience of the performer. Over and above this, the intelligent interpretation of music can draw upon diverse kinds of knowledge (e.g. of history and of musical forms) and experience (e.g. of emotions and of symbolic pattern recognition) as explained in [RB02]:

> 'For each level of attainment and genre of piano-playing, there is a pianistic competence and an appropriate level of sophistication in musical language (cf. "Play Middle C", "Play the harmonic scale of C sharp minor", "Play the octave passages in the coda of the Rondo in Beethoven's Waldstein sonata as glissandi"). It is significant that at its most sophisticated the language associated with a culture draws on such extensive experience and so many different sources of knowledge (e.g. in the above instance: music theory – *octave*; classical musical forms – *coda*, *Rondo*; musical history – *Beethoven, Waldstein*; and instrumental techniques – *glissando*) that it is only intelligible to the musical specialist.'

In summary, it is through extensive experience of gaining the necessary skills, identifying patterns of interaction and stimulus-response mechanisms and the acquisition and understanding of the relevant language of the domain that a learner progresses from a complete beginner to an expert in a domain.

Learning a skill is one aspect of learning about situations or artefacts. In learning to face a new situation or to use an unfamiliar artefact, learners are required to correlate the acquisition of new skills with the identification of important features in their context. Learning about artefacts need not commence with a user digesting formal instructions from a manual. The experimental psychologist John Carroll's theory of minimalism suggests that learning is more successful if learners are involved in hands-on tasks and not on reading obstructive instructional materials [Car90].

There is some evidence for Carroll's claim in the empirical observation of typical users of a new product [Nor98]. Exploratory interaction provides an initial understanding of how to use an artefact. We prefer to experiment, noting important observations and building up experience through interaction, without relying on objective prescriptions for how to use the artefact. We create mental models of the artefact under study that inform us throughout our learning [Joh83, Nor83]. This everyday, hands-on, empirical approach to learning contrasts with traditional educational approaches where problems and skills are mediated to the learner through the use of language, and particular emphasis is placed on logical and mathematical thinking as the most important aspect of intelligence [Gar93]. As the concrete examples described in this section illustrate, learning is much more than can be described through formal representable knowledge. Exploratory interaction is a key feature of complex learning situations. We now outline a framework for learning that is informed by the above discussion and that will be used in the remainder of the thesis.

3.2.2 An experiential framework for learning (EFL)

This section introduces an Experiential Framework for Learning (EFL) as a way of classifying learning activities on a spectrum between the private and the public domain. An earlier version of the EFL appeared in [Bey97], and has been adapted from a previous paper on educational technology [RB02] for this thesis. Figure 3.1 shows different categories of learning activity within the EFL. These categories range from concrete to formal learning and are concerned with issues that span empirical and theoretical knowledge. Activities towards the formal end have their foundation in experience-driven activities at the concrete end. This view is consistent with Noss and Hoyles's perspective on learning, as expressed in [NH96]:

> 'Although knowledge is constantly constructed and reconstructed through experience, this same experience also shapes and reforms a global and theoretical perspective'.

The purpose of the EFL is not to portray learning as a simple linear transition from private experience to public knowledge, but rather to express the way in which different learning activities depend upon each other. For instance, the learner can only progress to using symbolic representations meaningfully when they have a degree of experience gained through interaction in the domain. The interdependency between learning activities does not prescribe the learning pattern completely, but it imposes some loose constraints on the order in which they can occur. For instance, the focus of attention typically moves gradually from private experience to public knowledge as we learn about a domain.

private experience / empirical / concrete

interaction with artefacts: identification of persistent features and contexts
practical knowledge: correlations between artefacts, acquisition of skills
identification of dependencies and postulation of independent agency
identification of generic patterns of interaction and stimulus-response mechanisms
non-verbal communication through interaction in a common environment
directly situated uses of language
identification of common experience and objective knowledge
symbolic representations and formal languages: public conventions for interpretation

public knowledge / theoretical / formal

Figure 3.1 – An experiential framework for learning

As our examples of learning about rowing and piano-playing have illustrated, learning begins from private experience. Our preliminary interactions are informed by our previous experience. We begin to understand the persistent and important features of the domain and acquire the practical skills to manipulate them. Our interactions can lead us to understand the dependencies between our actions and events and understand how other agencies can affect the situation. With experience we come to understand that particular patterns of interaction are common and stable and we can communicate within the domain through non-verbal means. We are continually extending and refining our understanding of the situated language of the domain. Learning can eventually lead us to be able to establish the empirical basis for common experience and objective knowledge, which can in turn be representable as formal languages and have public conventions for interpretation.

In learning, there are identifiable ways in which we move from one category of learning activity to another within the EFL. Practising to develop a skill, experimenting to frame a theory or hypothesis and identifying new concepts in deriving new words are characteristic of moving from the empirical to the theoretical within the EFL. Practising to refine and debug skills, experimenting to test theories and hypotheses and devising situations in which to test the integrity of new vocabulary are characteristic of moving from the theoretical to the empirical within the EFL. These characteristic aspects of learning can be regarded as metaphorically ‘moving down and up between levels’ within the EFL in a way that may tend to stability. We may understand a concept and its application so thoroughly that exploratory interaction with it is unnecessary – but it is unnecessary precisely because we possess the experience of interaction with it that informs its use. When we are learning about a new concept, it then becomes important to support the learning activities that enable us to gain the broad base of experience required to interact with it in the fullest possible way.

In understanding the EFL, it is important to consider how it can be applied to support learning in practice.

The EFL is to be viewed as a generic template for learning. The specific character of learning activities in the EFL can be entirely different depending on the context in which learning takes place. Relevant considerations are: the subject domain (e.g. learning to row, to count, to write); the nature of the learning task (e.g. learning the concept of number, learning to use a calculator, learning times tables); the character of the learning environment (e.g. teacher-supported, self study); and the technology available (e.g. physical artefact, computer, virtual reality environment). There are nevertheless general patterns according to which learning activities are organised, as has been explained above.

‘Moving up and down between levels within the EFL in a way that tends to stability’ can be interpreted as negotiation of the semantic relation β (cf. section 2.2.2). Negotiation can be associated with genuine creation and novel discovery (as e.g. in Newton’s discovery of the refraction of light [CW95]). In this context, the learning activities in the EFL are emergent rather than previously understood. Negotiation can also be associated with coming to a common understanding through personal experiment and communication (as e.g. in learning to generate a spectrum using a prism). In this context, the learning activities in the EFL are familiar to the knowledgeable observer.

‘Moving from the empirical towards the theoretical within the EFL’ is a process of *abstraction*. Abstraction is concerned with formalising learning. ‘Moving from the theoretical towards the empirical within the EFL’ is a process of *concretisation*. Concretisation is concerned with gaining familiarity with underpinning activities and experience. For instance, this concretisation may take the form of testing abstract relationships or refining primitive skills.

Concretisation is one aspect of elaboration of the semantic relation β (cf. section 2.2.2). It is associated with enriching the specific experiences that inform a particular learning objective. For instance, in learning to row, diagnosing the difficulties in achieving a smooth stroke may involve working on particular basic elements of the stroke in isolation. A further aspect of elaboration is associated with setting a learning activity in a richer domain context. For instance, a novice may be introduced to rowing on a static machine, and progress via rowing a machine on slides to rowing in a boat on the water. In this example, the learning activity changes from one context to another – the skills become more complex (e.g. balance becomes important) and the terminology is necessarily embellished (e.g. concepts regarding the oar become relevant). In elaboration of this nature, the mapping from the EFL to specific learning activities is hard to formalise as the learning activities in themselves evolve.

As stated earlier, the ‘computers for learning’ agenda must aspire to support the widest possible range of different types of learning. This aspiration cannot be fully realised with existing computer technology: computer-supported interaction and visualisation is limited in comparison with activity in the real world (cf. the accounts of learning to row and play the piano in section 3.2.1). Developments in computing are already introducing richer interaction metaphors that potentially offer support to a wider range of learning activities (see e.g. [RJM+98]). The principles of EM to be introduced and discussed in this thesis are conceived as potentially general enough to embrace computer-related technology as it may develop in the future (cf. the discussion of ubiquitous computing in [Won03]).

In considering EM for learning, we aspire to provide computer support for the whole range of learning activities described in the EFL. In the world, learning often begins from tentative hypotheses, a type of interaction we aspire to support in EM model construction. A computer-based approach to model construction that reflects the EFL must be able to support fluid movement between many different types of learning activities. In the remainder of the chapter, we discuss how the principles of EM model construction (section 3.3, 3.4) match up with the EFL (section 3.6). We shall illustrate EM principles with reference to the construction of a restaurant manager model (section 3.5).

3.3 Learning by experience

Within the EFL, the most primitive learning activities originate from private experience. In this section, we expand on the role of experience in learning by considering Kolb’s theory of experiential learning and relating the EFL to an underlying philosophical attitude of Radical Empiricism first promoted by William James [Jam96].

3.3.1 Experiential learning

The dictionary definition of learning (as cited in section 3.2) is: to 'gain knowledge of or skill in by study, experience, or being taught' [OED97]. Many scholars have emphasised that experience is fundamental to learning. The seminal American educationalist, John Dewey, made the claim that learning has to be grounded in experience [Dew38]. Jean Piaget, in research on children's learning, proposed that children have different stages of learning, from sensori-motor, through concrete learning to abstract learning [Bra78]. Piaget stressed the important experience gained through interaction between the learner and their environment. Kurt Lewin's research in organisational behaviour also emphasised the importance of experience in learning, particularly stressing the active nature of the learner [Lew51]. The ideas of Dewey, Piaget and Lewin underpin David Kolb's well-known experiential learning cycle [Kol84].

Kolb's experiential learning cycle is based on an iterative cycle of four activities, namely concrete experience, reflective observation, abstract conceptualisation and active experimentation. Experience initiates the cycle; as Kolb says [Kol84]:

> 'Immediate personal experience is the focal point for learning, giving life, texture, and subjective personal meaning to abstract concepts and at the same time providing a concrete, publicly shared reference point for testing the implications and validity of ideas created during the learning process'.

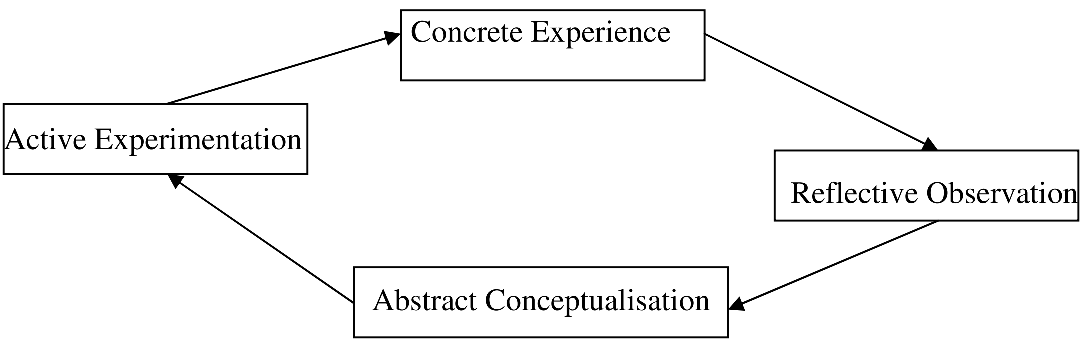

Figure 3.2 – Kolb's experiential learning cycle

In Kolb's cycle (see Figure 3.2), reflection on our personal experience gives rise to new concepts or ideas. These ideas in turn stimulate experiments that typically lead to new experience or new perspectives on our previous experience.

The four activities in Kolb's experiential learning cycle can refer either to private or public activities. Atherton's interpretation of Kolb's cycle makes this private/public distinction [Ath02]. Atherton classes concrete experience and reflective observation as internal activities, and abstract conceptualisation and active experimentation as external activities. This classification is appropriate in certain circumstances. Concrete experience and reflective observation are surely private activities. The nature of abstract conceptualisation and active experimentation depends on the learning context. Abstract conceptualisation may or may not involve concepts that belong to the public domain. Active experimentation may or may not be publicly interpretable or accessible. However, Atherton's interpretation makes it apparent that experiential learning can involve both private and public learning activities.

Kolb's experiential learning cycle is reflected in the learning activities in the EFL. As we pointed out in section 3.2.2, the learning activities associated with the EFL are not necessarily addressed in a rigid sequence; learners will move between activities in a fluid fashion. Concrete experience and reflective observation are closely related to activities at the private end of the EFL, whilst – depending on context – active experimentation and abstract conceptualisation are more closely related to the learning activities at the public end of the EFL.

As Kolb's cycle illustrates, learning can consist of many different types of activities, which draw on our experience, and change our experience. In many ways, learning can be considered to be reclassification of experience. We can understand this with reference to different categories to which experience may belong. Some of our experience is stable and revisitable ("I know how to do this now and I know I can do it again"), whereas parts might be unstable and tentative ("I have done that, but I am not sure how I did it or if I could do it again"). With reference to Howell's learning stages [How82] (cf. section 3.2.1), the process by which our experience migrates from being tentative to reliable is mirrored in the move from stage 2 to stage 4. In terms of Kolb's cycle, the reclassification of experience is mediated by a succession of activities that sees us experiment, reflect and form new ideas. All of our experience is open to reclassification in the light of new insights or of new circumstances to be taken into account. In learning, we always have the possibility of being surprised ("I didn't know that that could happen") and this can lead to new classification for our experience. When we are entirely sure that our experience of some phenomena is reliable, it is in some circumstances appropriate to explore the possibility of 'sharing the experience'. This notion of sharing experience depends on observing the social interaction that underpins inter-subjectivity. Part of the communication difficulty in establishing inter-subjectivity stems from the fact that it is hard for you to relate my account of my experience with your newly forming experience which you do not yet understand ("You are telling me this is true and I am not sure if I believe you until I try it for myself"). Our personal experience of a phenomenon in the social context can be said to be public knowledge when we can share it and others agree that it is true ("I know this and you agree with me").

3.3.2 Radical Empiricism

In this section, we outline William James's 'philosophic attitude' of Radical Empiricism, as first described in his "Essays in Radical Empiricism", first published in 1912 [Jam96]. According to Naur [Nau95], James was the pioneer of the experiential view of knowledge. Radical Empiricism has been considered in connection with EM in previous papers [Bey97, Bey99, Bey03]. The relevance of James's thinking to emphasising the concrete above the abstract in education can be seen in the following quote [Jam96]:

> '… the one thing that is sure is the inadequacy of the extant school solutions. The dissatisfaction with these seems due for the most part to a feeling that they are too abstract and academic. 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'.

Radical Empiricism draws on James's descriptive account of human mental activity in his "Principles of Psychology" [Jam90]. As emphasised by Naur (cf. the entry for **stream of thought** in [Nau01), James's philosophic attitude is distinguished by his readiness to talk about such issues as thought, feeling, association and knowledge by placing them in clear relation to *every person's experience of his or her thoughts and feelings*. Central to James's thinking is the capacity of the mind to make associations between experiences in the stream of thought. James identifies the roots of knowledge in how 'one experience knows another' in the stream of thought [Jam96]. He characterises the relationship between two experiences, one of which knows the other, as being given in experience and not rationally apprehended with reference to some explicit preconceived account.

Beynon [Bey97, Bey03] has made the connection between William James's outlook on experience and the philosophical issues raised by focusing on Cantwell-Smith's semantic relation β as it relates to spreadsheet design and use (cf. section 2.2.3). The key observation is that the evolution of a spreadsheet in design and use is similar in character to that of states of mind in the stream of thought. Changes to the spreadsheet are not to be interpreted as specifying a new spreadsheet, but as reflecting some change in our experience of its referent.

By generalising spreadsheet principles (cf. chapter 2), EM aims to account for the semantic relation β between a computer-based model and its referent in terms of James's notion of one experience knowing another. Experience of the computer model stimulates us to understand it in terms of our experiences of corresponding interaction with its referent. This correspondence leads to a conflation of the external and computer-based experiences, resembling what Turner has characterised as *blending* [Tur96]. The negotiative process of blending gives rise to new insights and directions in which to take the computer-based model, much in the spirit of Levi-

Strauss's bricolage [Lev68], a theme we return to in chapter 4. Negotiational blending of this nature is the vehicle for learning through the reclassification of experience discussed in the previous section. This is consistent with James's view that [Jam96]:

> 'subjectivity and objectivity are affairs not of what an experience is aboriginally made of, but of its classification. Classifications depend on our temporary purposes'.

EM endorses a view of knowledge that is consonant with James's idea that knowledge is rooted in personal experience. The EFL can be viewed as mapping out the activities that account for public knowledge with reference to private experience (cf. [Bey99]). It also represents a particular perspective on learning as 'gaining knowledge'. In his essay 'The experience of activity' [Jam96] James advocates Radical Empiricism as an appropriate philosophical stance from which:

> '… to try and 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'.

EM could be seen as bringing computer support to this agenda. As will be discussed in detail in section 3.4.3, negotiational blending in EM traces the progression of model building through the elements identified in James's quote above: identifying agency, attributing state-changes to those agents, and interpreting agent interaction in global state-based terms (cf. [Bey03]). As will be discussed in the next section, this is reflected in the way that, in EM, the model and the modeller's construal of the referent evolve together.

3.4 Principles of Empirical Modelling

In this section, we describe how the private experience that forms the basis of learning can be utilised in computer-based model construction at the empirical end of the EFL. Our understanding when we begin to construct a model is tentative and requires computer support that does not commit us to build on experience that is at present unstable. As Russ remarks [Rus97]:

> 'where there is no adequate theory we may wish to build models simply in order to aid our understanding; any specific purpose may be unknown, or provisional, and it is then only an impediment to make early commitments to certain properties we wish to preserve in the model'.

EM models that support the agenda above can be regarded as *construals* in the sense of Gooding [Goo90], rather than as conventional programs with preconceived functionality. This shift in perspective stems from focusing on *state-as-experienced* as being prior to *behaviour-as-abstracted* (cf. the distinction between EM and conventional programs described in chapter 4). Furthermore, this shift in perspective requires a different set of key concepts that underlie the modelling process; in EM these are *observables*, *dependency* and *agency*.

3.4.1 Construals

Real-world learning can often involve the making of models to supplement current understanding of a situation and give the opportunity to experimentally comprehend how changes affect it. For example: an engineer creates a prototype to gain fundamental knowledge about an artefact before the construction of the final system; a financial analyst constructs a spreadsheet to understand and explore potential changes to a situation. David Gooding introduces the term *construal* to refer to a concrete artefact that is used to embody evolving understanding of a phenomenon [Goo90]. He developed the idea of a construal from studying the experimental practices that Michael Faraday used in investigating electro-magnetic phenomena. Faraday used physical objects to convey his evolving understanding of electromagnetism. Gooding [Goo90, p22] characterises construals as:

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

> cannot be grasped independently of the exploratory behaviour that produces it or the ostensive practices whereby an observer tries to convey it'.

Gooding emphasises the close connection between the evolving understanding of a referent and the exploratory interactions that are used in developing the construal.

In EM, we observe an external referent, and concurrently build a computer model that metaphorically exhibits similar patterns of observables, dependency and agency [BS98]. 'What-if?' style modelling enables the interrogation of personal construals in testing beliefs about a referent. If experiments return expected results then a modeller's construal is reinforced (cf. stabilising our experience). Unexpected results in experiments serve to change a modeller's construal, because either the construal is mistaken or the referent exhibits some previously unknown characteristic (cf. new insights). A construal in EM is a voyage of discovery, a creative process quite unlike conventional modelling where the emphasis is on the representation of well-understood behaviours. The key features of a construal are that (cf. [Bey99]):

i) it is empirically established. It is informed by past experience and subject to modification in the light of future experience.
ii) it is experimentally mediated. Our experience with it guides its evolution.
iii) the choice of agents is pragmatic (what is deemed to be an agent may be shaped by the context for our investigation of the system); it only accounts for changes of state in the system to a limited degree (the future states of the system are not circumscribed).

A construal must be testable beyond the limits of the expected range of interactions with it [BRS99]. In specifying a conventional program, the modeller has to preconceive its behaviour thereby restricting the exploratory interactions that can be undertaken. In contrast, EM model construction privileges experimental interaction. Interactions can take account of the changing real-world situation; can probe unknown aspects of a referent; and may even be nonsensical in the world. Beynon has described these interactions, which reflect Situation, Ignorance and Nonsense (SIN) respectively, as exhibiting the SIN modelling principle [Bey01]. He claims that this principle is not well supported in classical computer programming, which requires the abstraction of well-understood problems. The SIN principle can also be seen in spreadsheets: the spreadsheet refers to an external situation; there is incomplete understanding; and we can test our understanding by making experimental changes that may be nonsensical.

Building construals using EM is closely associated with learning. The process of model construction is a private learning experience and our construal represents our evolving understanding of a situation [Bey97]. Experiments performed during the early stages of modelling an artefact are tentative and exploratory; they are a reflection of our provisional construal. Modelling dependencies is a prominent aspect of the early stages of EM. Rungrattanaubol [Run02] highlights the significance of modelling of this nature when knowledge is pre-articulate, informal, situational and takes account of personal viewpoints. Such modelling is intimately concerned with state-as-experienced rather than behaviour-as-abstracted, as discussed in the following section.

3.4.2 State-as-experienced and behaviour-as-abstracted

Formal computer science encourages the view that the only significant semantics of a computer program resides in the abstract patterns of behaviour and interaction that it supports. This is consistent with what Brödner has characterised as the ‘closed world’ paradigm:

> ‘ …, the “closed world” paradigm, suggests that all real-world phenomena, the properties and relations of its objects, can ultimately, and at least in principle, be transformed by human cognition into objectified, explicitly stated, propositional knowledge” .’ [Brö95]

To support this ‘closed world’ paradigm, the key requirement is to be able to develop programs which support planned user interactions and preconceived interpretations.

The users of such a program have no choice but to adapt themselves to the features of the program and its interaction style. In contexts where a domain is well understood, this viewpoint is satisfactory – both the designer and the user conceive of the program in a similar way. In situations where knowledge is uncertain, the programmer faces problems because they cannot conceive the abstract behaviour of the referent in its entirety. Beynon [Bey99] suggests that classical computer science has limitations that stem from concentrating on a 'closed world' paradigm.

EM is attempting to supply principles that can support what Brödner identifies as a counterposition – the 'open development' paradigm:

> '…, the "open development" paradigm, does not deny the fundamental human ability to form explicit, conceptual, and propositional knowledge, but it contests the completeness of this knowledge. In contrast, it assumes the primary existence of practical experience, a body of tacit knowledge grown with a person's acting in the world.' [Brö95]

The emphasis on practical experience and on growing knowledge in 'open development' requires an approach to modelling that enables unconstrained interaction with a computer model. This cannot be achieved if explicit behaviours are the primary concern of the modelling process. EM emphasises modelling that is state-based, where the term 'state' is to be understood as referring to 'state-as-experienced' rather than abstract computational state. The term 'state-as-experienced' necessarily refers to the experience of an individual, which may not be objective because our interpretation of the world may well be different from that of another person. State-as-experienced may confound us by changing in unpredictable and uncircumscribed ways, for example through events occurring that are beyond our expectations. State-transitions in EM are constrained only by the modeller's imagination. An open development approach requires a close correlation between the state of the computer model and the state of its external referent that reflects Cantwell-Smith's semantic relation β [Smi97]. Open development in EM has close connections with spreadsheet development that was discussed in section 2.2. For example, a spreadsheet user always interprets the spreadsheet with reference to its current state and in relation to the external situational state to which it refers. Construction of a spreadsheet goes hand in hand with its use; changes can be made on-the-fly as insights occur. Although spreadsheets can be used in a rigid predefined way, circumscribed use occurs only after significant evolutionary development.

The distinction between the concept of state in EM and in traditional computer-based modelling is depicted in Figure 3.3.

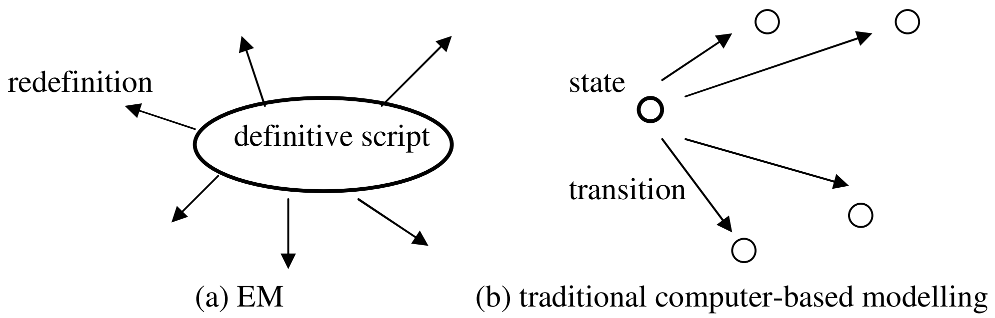

Figure 3.3 – State-based and Behavioural-based views on development processes

The concept of state in Figure 3.3(b) relies upon a circumscription of system behaviour that is characteristic of the closed world paradigm. Each circle depicts an abstract computational state and each edge a valid state-transition consistent with the abstract system behaviour.

In contrast, Figure 3.3(a) depicts the concept of state as it applies to modelling in an open development paradigm. The characteristics of such a state are not defined with reference to preconceived neighbouring states or an abstract behaviour. In keeping with the notion of state-as-experienced, the semantics of the state is implicitly defined by exploring plausible atomic state-transitions in an experimental fashion. It is for this reason that the state is represented by a spreadsheet-like definitive script and possible redefinitions (cf. section 2.4.1), rather than by a configuration of abstract states.

The identification of the plausible atomic state-transitions in Figure 3.3(a) depends on the modeller's viewpoint (who's making the observations?) and notion of indivisibility (when are observations deemed to be simultaneous?) [BC95]. In EM, these issues – which are crucial to changing our perspective on developing models from behaviour-as-abstracted to state-as-experienced – are addressed by introducing special concepts. In the following section we outline the three key concepts that underpin this shift in perspective, namely: *observables*; *dependency*; and *agency*.

3.4.3 Observation, dependency and agency

In this section we describe the EM concepts of observables, dependency and agency.

An **observable** is a feature of the situation or domain that we are modelling to which we can attach an identity (cf. a cell in a spreadsheet). The main requirement of an observable is that it has a current value or status (cf. a value in a spreadsheet). An observable can refer to a physical entity, an abstract entity or a conceptual entity. Examples of observables could be the mass of an object, the status of my bank account, whether I own a car, and the quality of the television reception. Observables can be of different kinds. These include: events (my train has arrived); quantities that are directly or indirectly measurable (the amount of petrol in the tank); booleans (my tank is half-full of petrol). What we deem to be an observable will in general depend on the context and the observer. An observable is understood to be something that an agent can apprehend instantly but such apprehension may be dependent on experience (e.g. this knot is a reef knot).

Observables in the domain are represented in EM by variables in a definitive script. The meaning that the modeller attaches to a variable in the script is negotiated in relation to the referent for the model [Nes97]. The plausible redefinitions for such a variable are those that have counterparts in interaction with the referent (cf. the way in which states acquire their semantics in Figure 3.3(a)). As discussed in [Run02], this gives definitive variables in the script the characteristic qualities of observables rather than abstract programming variables.

A **dependency** is a relationship between observables that expresses how they are indivisibly linked in change. A change to the value of an observable will cause changes to the observables that are dependent on it. For instance: the amount of tax payable is dependent on the current tax rate, personal income and tax-free allowances; the quality of the television reception depends on the weather conditions, distance from transmitter and strength of the signal. Unlike constraints between observables, which express persistent relationships between values in a closed-world, dependencies express the modeller's current expectation about how a change to one variable will affect the value of another. In open development, such expectations are subject to change. In EM, dependencies between observables are represented by definitions in a definitive script.

Observables and dependencies together are used to represent the current state of an EM model. The concept of agency is used to express state-transitions.

An **agent** is an entity in the domain being modelled that is perceived as capable of initiating state-change. The agents identified by the modeller will depend upon their construal and the purpose of the modelling. On this basis, the notion of agency encompasses such diverse possibilities as the manager of a restaurant, the battery of a digital watch or the modeller in the role of experimenter.

In an EM model, an agent is conceived as an entity that is capable of changing the values of observables or dependencies. Such an agent is itself typically associated with a set of observables. In practical EM using TkEden, the actions of agents are represented by redefinitions that may be manual (performed by the modeller via the input window) or automated (through the use of triggered actions as described in section 2.4.3).

In EM, there is a special purpose notation called LSD for describing observables, dependency and agency that can help in classifying agents and their capabilities. The LSD notation was initially motivated by research into the CCITT Specification and Description Language, and offers a way of describing systems at an abstract level [Bey86]. The LSD notation can be used in an Empirical Modelling framework:

i) to guide our evolving understanding of a situation by elaborating the observable elements of a situation and how they are viewed and controlled by agents.

ii) as a documentation tool to be used after model construction to guide others to the important features of the model.

Constructing an LSD account involves the identification of agents and the classification of observables with respect to agents. Each observable belongs to categories that reflect its status with respect to an agent. Observables can be classified into the following categories:

State observables – these are the observables that are associated with the presence of the agent. If the agent ceased to exist, such an observable would disappear. Examples include: the speed of a car (bound to the car) or the number of tables in a restaurant (bound to the restaurant).

Oracle observables – these are the observables to which an agent can respond in the current state of the environment. An agent does not necessarily have privileges to change the values of such observables. Examples include: the colour of a traffic light (an oracle to a car driver) or the time at which a customer phones the restaurant (an oracle to the manager). In certain contexts, a very significant oracle is the 'absolute time', which all agents can be presumed to know but none has the privilege to change [Bey86].

Handle observables – these are the observables that an agent has the privilege to conditionally change. Examples include: the speed of the car (a handle for the driver) or the table allocated to a customer in a restaurant (a handle for the manager). An agent does not necessarily need to have an observable as an oracle in order to change it.

Observables that are related through dependencies in the view of the agent are termed *derivates*. This term reflects the fact that the value of an observable can be derived from the values of other observables. An example of a derivate is: whether there is a table free in a restaurant depends on the occupancy of the tables.

An observable can be classified in different ways with respect to different agents (e.g. I can see the speed you are driving but have no control over it) and may appear in many different categories for a single agent (e.g. to you, the speed of your car is both a handle and an oracle).

The privileges that the agent has to make changes to observables are recorded as a set of guarded sequences of redefinitions. This set of redefinitions is referred to as the agent's *protocol*. Examples of protocols include: if a customer has requested a booking, the manager will allocate an appropriate table; and when a customer leaves the manager will take payment.

The protocols of the agents in an LSD account specify possible state changes that the system of agents can perform. They are not in general construed as circumscribing the behaviour of the system for a variety of reasons that are discussed in detail in [BNO+90]. For instance, an LSD account does not take matters of synchronisation, speed and reliability of response into consideration. A more typical use of an LSD account is in documenting our evolving understanding of a situation so that for instance other privileges could be added to the protocol of an agent.

In developing an EM model, our perspective on agency within the domain evolves with our construal. The way that we perceive agency is related to our experience of the situation and our current purpose for studying it. In [Bey97], Beynon has identified three viewpoints on agency that are representative of how the modeller's perception of agency may develop during model building in EM:

View 1: **The modeller identifies primitive entities as agents**. In this view, every observable or object-like set of observables is potentially an agent, as is the external observer. Any entity that is a cause, cue or trigger for some action on the part of another agent is identified as being an agent.
View 2: **The modeller attributes state-changes to agents**. In this view, the modeller construes specific observables and objects as responsible for particular state changes. This corresponds to understanding how state changes can be correlated with the presence (and action) of a particular agent (e.g. a flag moves only if there is a wind).
View 3: **The modeller identifies a system behaviour**. In this view, the modeller understands the system so comprehensively that it is possible to circumscribe its behaviour; agents act through reliable and objective stimulus-response patterns and have no capacity for surprise. This corresponds to virtual agency in the closed world.

Each of the above viewpoints can be correlated with different learning activities within the EFL. In View 1, we are concerned with identifying entities that can potentially cause state change in a domain, whereas in View 3 we have identified a specific systematic behaviour within the domain and are interested in whether we have identified all the relevant agents together with their actions. Moving from a View 1 to a View 3 perspective is like moving from a broad unfocused view of a domain to a narrow specialised application within the domain. Making this transition requires correlating agents with state changes as described in View 2. EM is concerned with facilitating the transition between a View 1 and View 3 perspective by providing ways of describing agents and actions in an exploratory way so as to embrace the learning that occurs in this process of correlation. In learning about a specific system, the View 1 – View 2 – View 3 transition is enabled in EM by a development environment in which agency in all three viewpoints can co-exist. This means that, even as we aspire to understand a system completely enough to represent it from a View 3 perspective, we can always override automatic operation to take advantage of new insights or experiments. One of the central problems in developing learning environments is that we need to accommodate incomplete and imperfect understanding. This demands support for modelling that is broader than mere specialisation to a View 3 perspective on a system. This is particularly problematic for conventional system development, which focuses on a View 3 perspective.

The model building discussed in this thesis makes use of the TkEden modelling tool (see section 2.4.2). Because this tool does not give full support for modelling agency, we shall not normally make explicit use of object-like abstractions and LSD accounts of situations. As discussed in detail in [Run02, Chapter 3], modelling with TkEden can be construed as taking place within an 'abstract definitive modelling' framework in which there is more comprehensive conceptual support for agency and entities. The way in which agency is implicitly represented in modelling with TkEden will be illustrated in the following section.

3.5 Modelling restaurant management

To illustrate the concepts of model construction introduced in this chapter we describe a case study of a restaurant management model. The restaurant manager model was originally developed in order to investigate decisions that a manager might make regarding the allocation of customers to tables in a restaurant. It is a model that could be used in order to inform decision support within a business context [RRR00]. Decisions on table allocations are an important and imprecise task for a restaurant manager. There is a need to accommodate the particular needs of each client, but both the requirements (numbers of tables, available time) and the resources available (tables, waiting staff, chefs) are changing dynamically. The customers (or potential customers) are agents who can act in unforeseen ways and so make the job of allocating tables more than a merely quantitative exercise in profit maximisation.

Figure 3.4 shows the model in use. The model contains two main display windows, a clock window and three forms which are used to generate customer events. The top window shows a plan view of a fictional restaurant with a total of eight tables of two different sizes. Below this is a window that contains a representation of a booking timetable for the restaurant for an evening. The booking timetable has a record of the current bookings for the evening and can be interpreted to establish when particular tables will be occupied or empty. The vertical red line shows the current time in the evening as displayed on the clock window in the bottom right. The other three windows are used to generate customer events. Potential customers can walk in off the street and request a booking immediately or may telephone the restaurant requiring a booking for a time in the future. It is also possible for customers with bookings to ring up and cancel their booking. Furthermore, when a customer departs the restaurant manager may record customer information for later analysis.

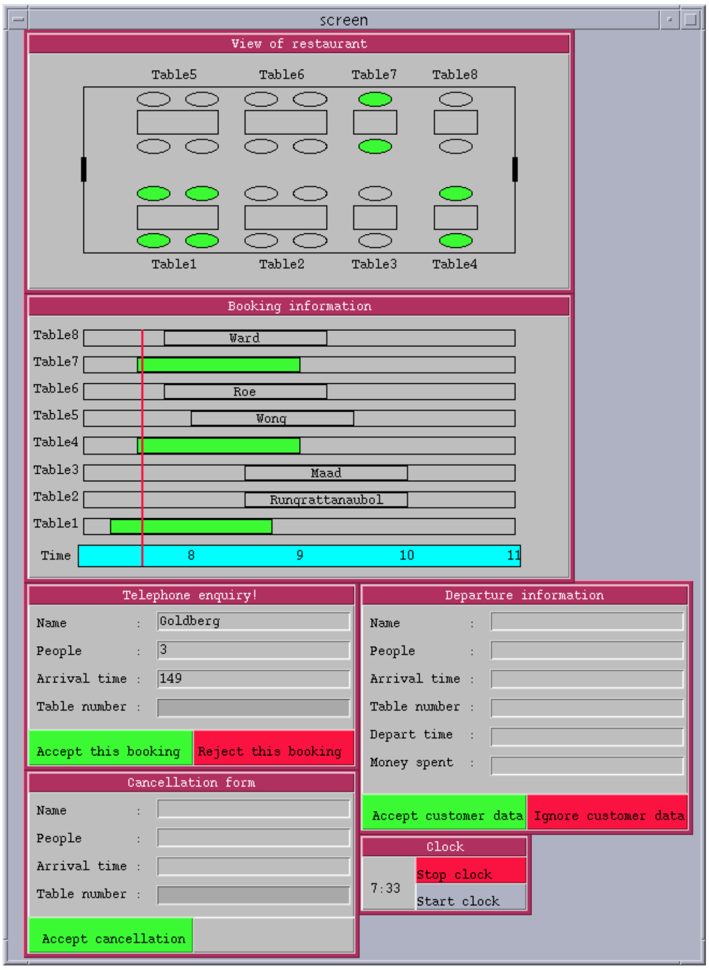

Figure 3.4 – The restaurant manager model

My motivation in constructing the restaurant manager model was to gain an appreciation of the issues involved in restaurant management. For example, during the construction of the model, I – as the modeller – initially allocated tables. Identifying the factors involved in human judgment was important in understanding how to add an automatic table allocation routine (cf. the changing viewpoints of agency discussed in the previous section). This involved understanding complex issues such as the relationship between unused capacity and unused time (cf. section 3.5.2).

The restaurant model had no prior specification, and in the model building I had no specific features in mind – these emerged during the construction of the model as a direct result of interaction with the partial model. This is possible in EM because of the emphasis placed on the representation of state rather than the recognition of abstract patterns of behaviour (see section 3.4.2). Initial model construction focused on identifying the important qualities of a restaurant and building on these basic ideas to shape the more advanced concepts such as automation of table booking (see section 3.5.2). In constructing a conventional program to perform restaurant management we would primarily consider the important actions that the restaurant manager has to perform and set out to automate these (cf. software development based on the analysis of use cases in the spirit of Jacobson [Jac92]). In the construction of the EM model, there is no circumscription of the possible uses to which the model can be put and these remain open throughout the development. For example, we might wish to add waiters to the restaurant model and animate them on the restaurant window, or consider the effect that changing the number of waiters has on customers and fellow staff. In a conventional program, where use is preconceived, it can be much more difficult to alter the interpretation of the partially constructed program flexibly since it is already optimised to serve a particular function.

3.5.1 Experiential learning and the restaurant manager model

The initial construction of the restaurant manager model was guided by my experience of visiting restaurants. Each visit to a restaurant has built up the background knowledge of how restaurants look and function on which I drew in building the model. Some aspects of my experience of restaurants are stable and represent objective knowledge. For instance, customers are allocated to a table, eat, and then depart after a period of time. Other knowledge acquired through experience such as my conception of the thought processes behind how tables are allocated to customers take the form of subjective hypotheses. The initial phase of model construction is concerned with building a computer model that embodies my objective and stable knowledge of restaurants.

Interaction with the model, through experiments and observations, is the source of new insights into restaurant management. New experience is acquired as a direct result of intimate engagement with the restaurant model. As Kolb's cycle indicates, new experience results from reflection, conceptualisation and experimentation. This cycle works repeatedly in our ongoing and increasing experience of restaurant management. Experience matures from tentative hypotheses about aspects of restaurant management to stable experience that we can reliably revisit and reproduce as requested. My stable experience – for instance that there are tables to which customers are allocated – is embodied as part of the restaurant model that I am assured is valid. In summary, stable experience of actual restaurants underpins our initial construction of the model, and our emerging experience gained from the computer-based restaurant model guides its future construction.

3.5.2 Empirical Modelling principles and the restaurant manager model

The EM restaurant model is a construal of restaurant management. Throughout the model building, the computer-based model is always provisional and reflects my current understanding of restaurant management. It is always open to new insight and new exploration based on my emerging understanding of the situation. In developing the restaurant construal, there were three broad phases in the model construction: I initially concentrated on the representation of state; then investigated sensible behaviours through experimental manual redefinitions; and finally automated appropriate behaviours.

There are a number of observables that are important in the restaurant model. The most primitive observables concern the physical characteristics of the restaurant itself. These include: the room layout; the number of tables; and the number of people that can sit at each table. As illustrated in Figure 3.4, the model includes a visualisation of the restaurant with different size tables and a table-booking sheet for an evening. Other relevant observables include: the time of each booking; the number of bookings; the number of people in each booking; and customer preferences (such as smoking or window seats). A possible classification of these observables with respect to the restaurant manager is shown in the LSD account in Listing 3.1.

The modeller initially defines customer events using the forms on the interface (see Figure 3.4). Through creating sample reservations and cancellations, a preliminary appreciation of the booking experience and knowledge that a manager possesses can be gained. Choosing a sensible table for a customer is not a trivial matter. There are many issues to consider, including customer satisfaction, staff morale and past occupancy patterns. Each of these factors will consciously (or subconsciously) influence the decision made by the manager. In justifying his or her decision, a manager may appeal to their tacit knowledge, which may be exceedingly difficult to articulate ("I feel that's the right decision"). Experience gained through the experimental generation of enquiries can lead to insights into the manager's job, even if their task is viewed simply in terms of maximising profit without considering other resources such as numbers of waiters and chefs.

```
agent restaurant_manager {

  oracle
          table1_position, table2_position, ...
          table1_occupancy, ...
          future_bookings
          telephone_ringing
          customer_preferences
          size_of_customer_party
          restaurant_full

  handle
          table1_occupancy, ...
          future_bookings
          table1_position, ...

  derivate
          restaurant_full is (table1_occupancy>0) && ...

  protocol
          table1_occupancy==0
                → table1_occupancy = size_of_customer_party,
                ...
          size_of_customer_party > maxtablesize
                →    table1/2 = table1+table2;
                     table1/2_occupancy  =  size_of_customer_party,
                     ...
 ...
}
```

Listing 3.1 – Part of an LSD specification for the restaurant manager model

The eventual construction of an automated routine to simulate a restaurant manager in allocating tables to potential customers entails deeper insights into the subtle nature of the job. There are obvious considerations, such as ensuring that there is enough time to fit in a particular booking on a table and that there are enough seats for the number of people in a party. The results of applying these simple conditions leave a set of possible tables on which to place a party. As the model is developed, more sophisticated questions emerge. For instance, what is the relationship between unused capacity (such as results from seating a party of 2 at a table for 4) and unused time (as when a table has no people on it for a period of time)? The answer to this question could guide the manager in deciding whether to accept a party of 2 on an empty table for 4. Other factors relevant to the above question may include: the type of restaurant; the type of cuisine; the night of the week; and the geographical area. These factors influence the manager but are very difficult to quantify. The decision to refuse a booking for a couple when a table for four is free might indicate that the restaurant manager believes that a larger party is likely to arrive soon enough to be more profitable. The elements of the model are linked through dependencies so that we can easily perform 'what-if' style queries to see the effects of changing features such as the length of booking slots or the opening hours of the restaurant. Queries of this nature can be made 'on-the-fly', and can support other interactions, such as changing the layout of the restaurant to simulate bookings from large groups.

From a personal viewpoint, experimentation with the model led to personal insights that have enabled me to understand some of the difficulties of a restaurant manager's job, although it would of course be impossible to fully replace a manager's decision-making by an automated routine. The construction of a faithful model of a restaurant, where human agents can play roles as they would in the real world, allows the judgments and insights of the individual to be expressed through interaction. The process of model construction in EM is intimately associated with domain learning.

3.6 Chapter Summary: Empirical Modelling and the EFL

In the final section of this chapter, we discuss connections between EM and the EFL and argue that they are intimately linked. In section 3.2.2, we proposed the EFL as a generic learning framework that can be interpreted with reference to any subject domain. We shall now argue that 'EM supports the EFL', in the sense that it supports learning activities from across the whole of the EFL and enables fluid movement between them.

EM endorses a view of knowledge similar to that proposed in Radical Empiricism (cf. section 3.3). In this view, all knowledge is rooted in the primitive notion of ‘one experience knows another’ in personal experience. In EM, the building of an artefact offers experience that ‘knows’ experience of its referent. In [Bey99], Beynon considers the way in which EM allows objective and theoretical knowledge to be traced to its experiential roots. In [Bey03], Beynon discusses the similarities between the perspectives on knowing represented in EM and in Radical Empiricism. The EFL is also motivated by the idea that learning is rooted in private experience and that abstract activities are grounded in our sound understanding of concrete examples.

EM emphasises interaction and experimentation with artefacts in order to generate and test our construal of a referent (see section 3.4.1). During construction, the modeller is always able to interact with the evolving artefact. As discussed in section 3.4.2, our early interactions are primarily concerned with exploring the current state of the artefact. This is analogous to the experimental changes a spreadsheet modeller might make in order to work out sensible future states. In EM model construction, the emphasis is on interactive exploration of plausible state transitions to increase our understanding: the modeller is exploring the ‘space of sense’ (cf. [Bey01] and section 3.4.2). It is through the occasional verifiably mistaken experiment that appropriate stimulus-response mechanisms and generic patterns of interaction are identified [Bey01]. The account of the construction of a clock model in Appendix D gives a practical illustration of how model construction in EM can be experimental, reflecting our evolving understanding of the referent and the differing purposes to which we may want to put the model. With reference to the EFL, activities of this nature are concerned with the identification of dependencies, generic patterns of interaction and stimulus-response mechanisms at the concrete end.

In section 3.4.3, we discussed how agency in EM can be used to automate reliable behaviours of an artefact and hence move towards the abstract end of the EFL. Because the modeller has the discretion to perform experimental interactions at all times, they can always step back from an abstract behaviour into a concrete situation. The activities described in the previous two sentences are forms of abstraction and concretisation respectively (cf. section 3.2.2).

The above discussion of EM and the EFL leads to two conclusions:

i) EM is well suited to support learning activities at the concrete end of the EFL because of its primary emphasis on experimental interactions with artefacts and the representation of state-as-experienced.

ii) EM can support the fluid movement between learning activities in the EFL due to its ability to integrate the abstract and the concrete within a single modelling environment.

In later sections of the thesis, we will discuss the connections between EM and the EFL from different perspectives. In section 4.6, we consider the links between EM and the EFL with respect to the educational theory of constructionism. In section 6.5, we consider the links between EM and the EFL with reference to sizable EM case studies.