

# **Business and IT Perspectives on AMORE: a Methodology using Object-Orientation in Re-engineering Enterprises**

**Kiran Fernandes, Vinesh Raja, John Keast**  
Warwick Manufacturing Group, University of Warwick, Coventry CV4 7AL

**Meurig Beynon, Pui Shan Chan, Mike Joy**  
Department of Computer Science, University of Warwick, Coventry CV4 7AL

## **Abstract**

This paper is both an outline and a critique of AMORE, A Methodology Based on Object-Orientation for Re-engineering Enterprises that is being developed at the University of Warwick under the auspices of EPSRC grant GR/M02675. The methodology involves two principal phases, which deal with re-engineering from business and software viewpoints respectively. AMORE proposes models and guidelines that are intended to achieve a closer integration between these two viewpoints than is possible with alternative approaches. To this end, it exploits business domain modelling that takes account of the business factors (such as goals, location and organizational structure) that influence the design of processes. The prospects for AMORE are reviewed first with reference to current thinking on BPR and software development for business information systems by IT specialists such as Jacobson and Warboys, and subsequently with reference to business applications of a new computer-based modelling paradigm that are being developed in parallel with AMORE by Beynon and Russ in the Department of Computer Science at the University of Warwick.

**Keywords:** Business Process Re-engineering, Business Modelling, Object-oriented Software Modelling, Organisational Design, Empirical Modelling

## **Introduction**

Business Process Re-engineering (BPR) is an activity that embraces a wide range of disciplines. It has had a central rôle both in motivating interest in BPR, and in suggesting paradigms for its effective application. There remain many controversial issues to be addressed in reconciling the agendas associated with business and IT perspectives on BPR. The difficulties of communication between business and IT specialists are well-recognised. These are highlighted in projects such as AMORE, where there is strong industrial and commercial interest in exploiting methodologies such as object-oriented software engineering (OOSE). Though such methodologies may be perceived by the business analyst as sufficiently mature, there is still considerable academic scepticism about the extent to which the problems of large-scale software development for modern IT systems have been resolved [Bro95, Bro97].

The AMORE project is a useful vehicle for assessing the prospects for successful application of a well-established paradigm for OOSE to structured business modelling from both business and computer science perspectives. It has involved two complementary strands of research. The first and primary goal of the project is to develop a methodology and framework for structured business modelling, and to demonstrate its application to an industrial case-study. The second goal is to understand the relationship between software engineering and BPR in the light of significant intellectual and practical contributions to this field by computer scientists over the last decade (see [Jac92, JEJ94, Bro95, Bro97, WKR99]). The latter goal is here examined with particular emphasis on a new approach to computer-based modelling (Empirical Modelling - EM) that has been developed at Warwick over the last 15 years under the direction of Beynon and Russ (see [Bey99, Bey97, BM99, BM00, BRR00a, BRR00b, RRB00, CRB00a, CRB00b, BWMWRR00, BS98, BRS99, BCSW99, RRR00, RR00] and <http://www.dcs.warwick.ac.uk/modelling> for more details).

This paper is in three principal sections, reflecting the complementary contributions of the two groups of authors, and respectively representing a business and a computer science perspective on the AMORE project. Section 1 describes the current status of our research on developing and applying a methodology for object-oriented re-engineering. Sections 2 and 3 describe progress towards understanding the AMORE agenda from a standpoint that is more closely linked to computer science and to EM.

## **1. AMORE from a Business Perspective**

Today's business world is facing a plethora of managerial and technological changes beyond the capacity of any company to control and absorb them. Customer satisfaction, development of new products, and the introduction of new technologies are some well-known driving forces, but frequent modifications are making them unpredictable. To succeed in this competitive environment, it has been established that companies should employ Business Process Re-engineering (BPR) as a means to change their business processes and radically improve their business. Although there are a number of successful industrial cases to support this proposition, at present it is still used mainly in large companies. Small to medium sized companies may consider it as a risky and expensive activity. Information Technology (IT) is widely regarded as an enabler in most BPR initiatives, and the effectiveness of the IT infrastructure should be closely monitored. In addition, it should be flexible enough to incorporate legacy systems without large cost implications and to allow graceful evolution as part of an ongoing continuous improvement programme.

Within this overall context, this section discusses a methodology for structured business process modelling and analysis using Object Technology. The aim of the methodology is to support rapid business process change and re-engineering incorporating legacy systems. The framework for AMORE has been informed by a comprehensive review of the BPR literature [MKJ98], and is representative of current thinking on BPR from a business-oriented viewpoint. As will be explored in sections 1 and 2, this thinking is in tension with ideas developed about BPR from an IT perspective in some important respects, and is for the most part quite radically different from that suggested by an Empirical Modelling approach.

### **1.1 Motivation and Background for AMORE**

An increasing number of companies are adopting the concepts and principles of BPR. This aims to radically alter many outdated procedures, to reduce costs, and to improve the companies' competitiveness. IT plays an important facilitating role but should not be regarded as the only key to achieving the expected results.

The keen interest in BPR can be attributed to the fact that most leading companies are likely to subscribe to one of the most popular and valuable management dogmas: to stay close to their customers. Recent literature shows that most well-managed and established companies are consistently ahead of their industrial rivals [Cer94]. Although BPR constitutes a well-known managerial approach for increasing business competitiveness, its diffusion is still slow and it mainly concerns large companies. This can be partly explained by poor attempts to gain a complete understanding of the business processes concerned and the subsequent deployment of inappropriate IT support. Researchers studying problems in the implementation of BPR [Jam96a] have pointed out that as many as 50% to 70% of projects have failed to achieve the dramatic results at which they aimed. The key to a successful BPR implementation is to strive to attain a better understanding of the underlying business process through a systematic approach so that a smooth transition from a business view into an Information System (IS) domain can be achieved.

Despite the need for a tight coupling between the BPR practice and the IT perspectives, co-ordinated actions involving BPR and IT practitioners are still rare. This is largely due to a sort of conceptual gap dividing the "world" of IT design and the "world" of BPR techniques. This gap is apparent not only in the frequent poor understanding of the business process view by IT experts, but also in the difficulties that BPR experts face when they try to systematically uncover the business process from the existing software system. In short, one can say that there is an inherent difficulty in transferring knowledge [FRM00] between people designing information systems required for a BPR implementation and their customer (viz., the company undergoing the BPR exercise), due to problems such as the difference in

cultures, languages and problem perception. This distinction accounts for problems in modelling new processes and in controlling their implementation lead-time, costs, quality and completeness.

We argue that any methodology aimed at addressing these problems in order to reduce failure rates of BPR projects must take the organisational, managerial and technological factors into consideration. Our methodology, based on the concepts of object technology, originates from the AMORE project. The AMORE project aims to deliver a methodology which uses distributed computing and component-based development (CBD) to drive and support a flexible IT infrastructure for use in large and medium sized organisations. This will enable software systems and business processes to evolve rapidly, enabling congruence between organisational changes and business focuses. In satisfying this goal, the following objectives are identified:

- the creation of a methodology for structured business process modelling and analysis using object technology to support re-engineering, rapid business process change and the adoption of legacy systems.
- the proposal of a framework, based on the methodology for the integrated use of distributed systems, to support rapid evolution of re-engineered processes.

## 1.2 The Conceptual Approach

A successful implementation of a structured business model requires a comprehensive set of tools for formal representation and thus enabling analysis, a method for embedding organisational and managerial aspects concerning strategic orientation and more importantly, the involvement of customers. Communications between the modellers, designers and the users are essential in reaching a platform of common understanding from which to resolve ambiguous or contentious issues.

Process modelling is the first step in formalising processes [UF96a, UF96b] and in creating a foundation to build on. In this paper, the term 'process' is defined as follows: "A process uses a defined input to produce a defined output and it consists of tasks which are performed by the organisational units. The information that is needed for the execution of a task is stated either explicitly or implicitly." A process is composed of a set of tasks, together with the relationships between these tasks that define their possible orders of execution.

Process modelling by itself is not sufficient to define the whole business domain. Research has concentrated on representing processes by means of a structural architecture. Flow Diagrams, Activity Diagrams, IDEF diagrams are being used to represent processes and the relationships between them. A lot of work has also been done in the area of process translation. Projects such as PIF (Process Interchange Format) [LGJMTY97] and PSL (Process Specification Language) [Psl00] are designed to support the exchange of process descriptions among different process representations. Our belief is that any particular process description format is unlikely to suit the needs of all applications completely.

Software models are primarily used for performing static (Use Case Models) and dynamic process analysis (Process Interaction Models). The objective of software modelling is to represent parts of the real world and its behaviour. Since software models are based on business models, business and software modelling activities are tightly connected. The effective integration of software models with business models is a central objective in BPR; there is always a danger of building software models that do not reflect the real business conditions. This is mainly due to the conceptual differences between business strategies, the real processes, and the software systems that support them. Furthermore, it would be difficult to completely embed all the business knowledge into a software model because there are technological constraints [FRA00a].

Generic models are often used to describe the structure and the behaviour of a process unambiguously [CK092]. They refer to characteristics at a high level of abstraction that are common to companies having approximately the same size and operating in the same industry. Generic models provide building blocks that are efficient and easy to use but are inappropriate for detailed modelling. In order to meet the effectiveness requirements, more detailed models have to be made available and this may subsequently incur higher costs.

One approach to detailed modelling which is directed towards defining basic processes characterising different industries, is given by ARIS (ARchitecture of Integrated Information System) [Sch92]. The focus of ARIS is on the integration of processes by using event-controlled chains. In this architecture, processes are the basic units of analysis and can be tailored to different kinds of companies. Depending on the pursued objectives, ARIS allows the users to define various perspectives without losing the global view of a process. The system focuses on the analysis and modelling of business processes as well as on methods for describing the complexity of a corporate IS environment. In this stream, PROMAX [KR96] offers a more comprehensive set of functional and representational requirements, considering both production and administrative environments.

Another approach [Ner92] takes a different stance and its focus is on objects. The process is defined by the continuous interaction among these objects. This is a bottom-up approach, based on elements constituting the problem space, as opposed to the top-down definition of building blocks. It is flexible in managing highly volatile requirements and continuous paradigm shifts [Ner92], but it may have some limitations when trying to keep systems consistent.

Some researchers have concentrated on defining objects involved in the modelling [GF94]. Their aim is to provide a shared terminology that enterprises can use to define objects, the knowledge they contain and their mutual relationships. The researchers tried to define the meaning of each term in as precise and unambiguous manner as possible. For example, the TOVE (Toronto Virtual Enterprise) [GF94] semantic is implemented in a set of axioms that, by means of knowledge-based simulation, automatically deduces the answer to many common sense questions about the enterprise and defines a set of symbols for depicting terms, or the concepts constructed from terms, in a graphical context. This ensures a consistent representation of relations among entities, and facilitates better communication, coordination, and optimisation of enterprise decisions and processes to achieve higher levels of productivity, flexibility and quality.

An approach that combines both process and object ingredients, is proposed by Jacobson [JEJ94]. It is called Object-oriented Business Engineering and is based on use cases. This approach utilises an external and internal model of business. The former is object-oriented and describes the company and its external environment (e.g. processes that satisfy customer's needs), whereas the latter describes business processes, different tasks (internal processes) they consist of, and the various types of resources that they exploit or produce. The processes are modelled through use cases, whereas the environment is modelled through actors. A use case is a sequence of transactions in a system whose purpose is to yield a result of measurable value. The use case model shows how the external environment interacts with the business, that is, how actors communicate with use cases within the business.

Although each of these approaches attempts to integrate different viewpoints and to support the reuse of components, they do not clearly distinguish between business modelling and software modelling and the constraints they impose. This may increase the risk of low adherence of the final models to the represented processes.

### **1.3 An Introduction to AMORE**

Structured modelling tools such as ARIS and PROMAX which integrate business modelling and system design can provide substantial support for IS development. In general, these tools aim to provide an integrated "abstraction-to-reality" solution for business systems development. Historically, these tools have not lived up to expectations, partly for the following reasons:

1. They do not adequately address business requirements prior to the system design.
2. They do not consider the details of transforming business requirements to system components.
3. They provide no traceability between business requirements and system components.
4. They lack an enterprise-wide understanding of and perspective on what they are attempting to automate.
5. In the case of older tools, they generate code for outdated technology and are clumsy to use.

The "abstraction-to-reality" strategy is not currently implemented in any single toolset. AMORE, the methodology introduced in this paper, is aimed at realising the idea of "abstraction-to-reality". The

outline of AMORE that follows comprises a brief overview of the method, and a brief description of the models and guidelines it involves.

The AMORE methodology is based upon two phases viz. the Business View Phase and the Software View Phase, as represented in Figure 1.

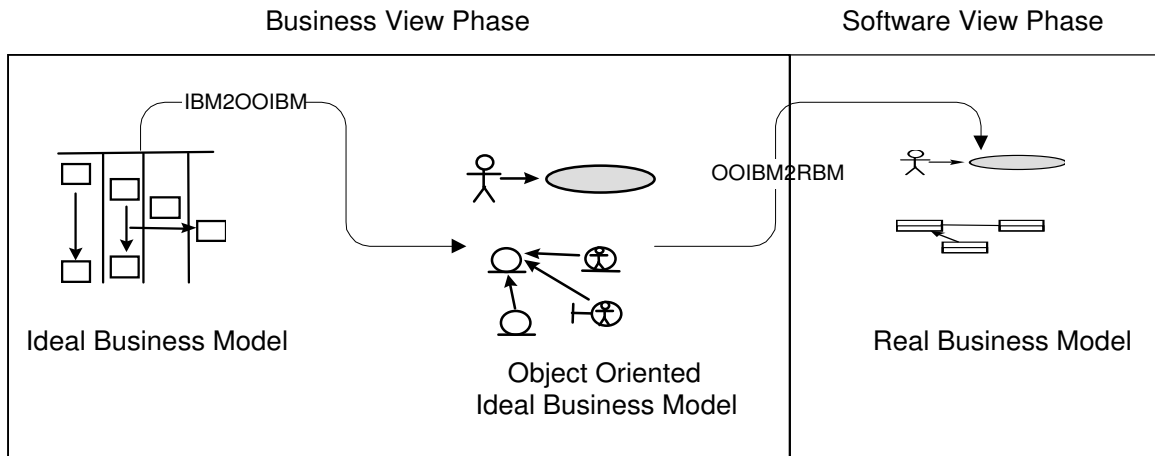


Figure 1: The AMORE overview.

These two phases of AMORE are based on the object-oriented principles and are designed to support re-engineering and rapid business process change (adaptive organisation). AMORE involves the Models and Guidelines displayed in Table 1:

| Models   | Guidelines  |
|--|---|
| <ul style="list-style-type: none"> <li>• Business Domain Model (BDM)</li> <li>• Ideal Business Model (IBM)</li> <li>• Object-oriented Ideal Business Model (OOIBM)</li> <li>• Real Business Model (RBM)</li> </ul> | <ul style="list-style-type: none"> <li>• Guidelines, in a pattern-like format, for mapping the IBM into the OOIBM</li> <li>• Guidelines, in a pattern-like format for mapping the OOIBM into the RBM</li> </ul> |

Table 1. Items comprising AMORE

“Abstraction-to-reality” begins with a clear model of the business concepts and processes. The key management issues considered are the classic business issues of Who? What? When? Where? Why? and How? The answers to these questions will guide process improvements and system development. This is the essential focus of the Business Domain Model. The BDM will then be examined and validated by the customers. The customers will decide whether the organisation should carry out a Business Process Re-engineering exercise. This will lead us into the Business View Phase. The Business View Phase first considers the business concepts and dynamics before deriving and analysing the business domain model. The IBM is the representation of the business domain. This business model is translated into an OOIBM using guidelines. The final step is to translate the OOIBM into a RBM using similar guidelines. The RBM will generate software that aims to support the business domain.

At every modelling stage, there are supporting tools (e.g. to specify goal mapping, business interaction, workflow and organisation mapping) that can translate the models into deliverables such as printed documents, software prototypes and ready-to-run applications. These deliverables can then be discussed with and validated by the customers. As the customers are involved at every stage of the modelling, AMORE can be regarded as a customer-centric structured business model.

This design of AMORE aims to bridge the gap between the business and IT worlds, and to enable the IT and BPR practitioners to move from one view to another in a more effective and efficient way. The next three sections elaborate on the phases of AMORE. They respectively outline the modelling involved in the Business View phase, the modelling involved in the Software View phase, and the Guidelines.

## 1.4 Modelling in the Business View Phase

The Business View phase of AMORE involves three models: the Business Domain Model (BDM), the Ideal Business Model (IBM) and the Object-Oriented Ideal Business Model (OOIBM). Both the IBM and the OOIBM are process models, but the BDM is a richer model that incorporates domain knowledge that informs the business processes. The IBM that is used to initiate the Business View phase is derived from the BDM.

AMORE is based on the premise that the identification of the business processes and their implementation can be treated as separate concerns. This stance is reflected in the concept of Business View and Software View phases.

The Business View phase is initially framed by an exercise in business domain modelling that identifies the IBM. In general, this can be a long and iterative exercise that includes requirements analysis, customer interviews, protocol analysis, a procedural review and a contract review. The scope of the business domain modelling is determined by which process is targeted for re-engineering or re-implementation. Depending on the customer requirements (for instance, whether an increase in productivity of the process is required) it may be necessary to re-engineer one or more processes in order to obtain the IBM. Where necessary, BPR is performed as part of the preliminary business domain modelling activity that precedes the Business View phase.

Where BPR is concerned, AMORE adopts the PAMS (Principles and Methodology Support for Re-engineering the Product Introduction Process) system for re-engineering processes. The PAMS [PAMS] system has been developed as part of the EPSRC funded SPEDE project [MR98][SPEDE]. The PAMS system is:

- an easy-to-follow BPR methodology with the ability to access appropriate software when needed;
- designed to reduce the learning time and increase the working time of newly formed BPR teams;
- designed to support business process improvement teams with easily accessible information about principles, methodology and tools, and techniques for BPR.

Through PAMS, the computer can serve a useful role in the meta-level activities of BPR that are concerned with re-designing work and re-allocating responsibilities.

The focus for the business domain modelling exercise that establishes the IBM is the construction of the Business Domain Model (BDM). The target process that is the subject of the re-engineering activity determines the scale and scope of the BDM. Though the primary goal of the business domain modelling is to arrive at a process model for the IBM, the BDM is not merely a process model.

Traditional business models are typically framed in terms of abstract processes that are divorced from their situated context in the real business world. For example, commercial tools such as ProModel [Pro00], though they aim to represent and simulate a process, do not take the factors that influence the form and character of the process into consideration. Systems that are purely developed using such tools may fail because they take too little account of the real business world.

In AMORE, the BDM developed in the course of re-engineering a process aims to serve two functions: fulfilling the role of conventional state-of-the-art processing modelling tools, and enabling the factors that influence the design of business processes to be recorded. The latter function elevates AMORE from a "process-centric" to a "domain-centric" approach. The potential advantage of a domain-centric viewpoint is that it takes account of the way in which processes depend upon influencing factors, thereby offering a more realistic view of the business domain. For example, when interaction with an external organisation is involved in a business process, a change in that organisation will affect the

underlying business domain. Dependency of this nature is not addressed in traditional process-centric approaches.

The architecture used in developing the BDM requires an extension of conventional representational systems for process modelling. IDEF3 (ICAM DEFinition language) has been the most widely used representational system for process modelling. One of the main drawbacks associated of using IDEF3 is that of translating process information. To create a common representational notation, the PSL (Process Specification Language) [Psl00] (ISO/TC184/SC4 & SC5 V1.0) project has framed the outline for a process specification language. We have developed a novel modified process representational technique termed IDEF\*\* because it is an enhancement of the IDEF3. This novel methodology is CIMOSA-compliant, IDEF3-based and can be integrated into the functional requirements of PSL. It also allows dynamic analysis of aspects of the process such as task and customer interactions. More details of this representation technique can be found in [FR00c].

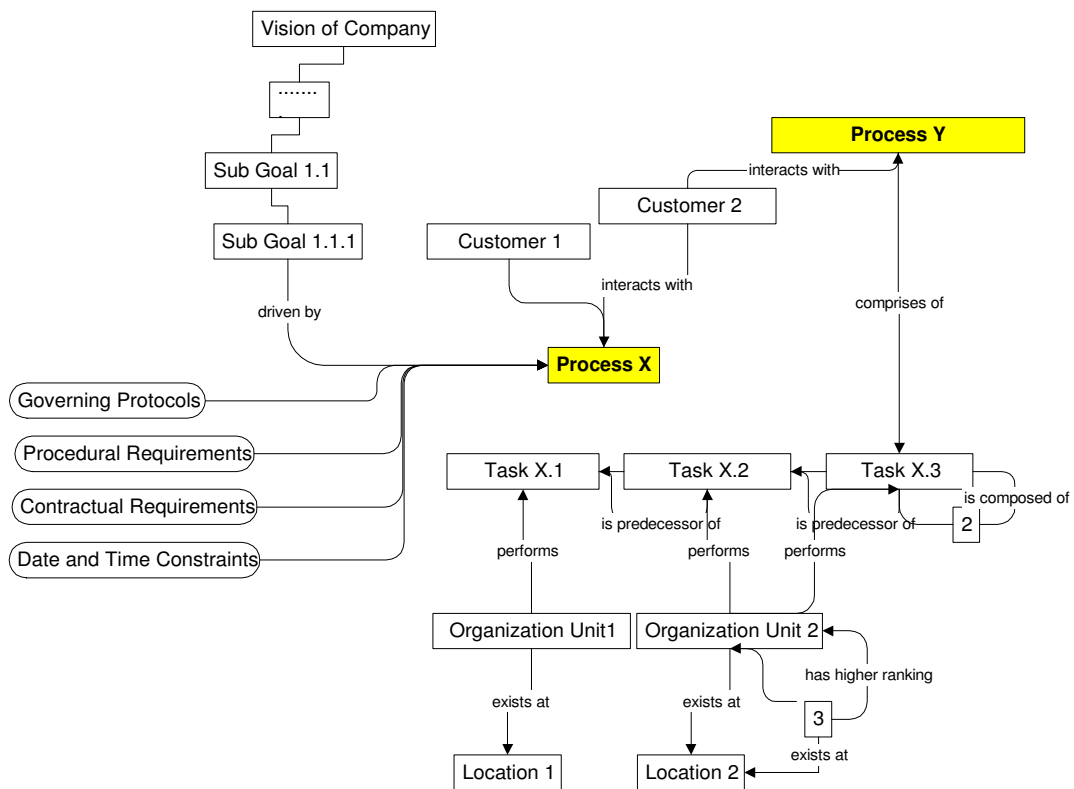


Figure 2: Business Domain Model

Constructing the BDM associated with a targeted process is intended to address a comprehensive range of relevant issues. The modeller must examine the whole process to identify its parts and to understand the interaction among the different subsystems. The BDM should supply an abstract, expressive and rigorous representation of the targeted process that can be used to enhance process comprehension, to validate its behaviour and to evaluate its performance. The business factors that influence the structure of the targeted process must also be recorded (see Figure 2). These include the goals of the processes (and their relationships to the goal of the company), the tasks, location, customers of the process, organisational units, governing protocols, procedural requirements, contract requirements and time constraints. We have developed an ontology for these influencing factors, in which their semantics, synonyms, interrelationship, attributes, and syntactic form are specified.

The rôle of influencing factors in AMORE is illustrated by our treatment of goals. One of the most important and fundamental drivers in a re-engineering enterprise is to understand the goals (“goal mapping”) of the organisation. Typically, the goals of an organisation are derived from the “voices of

the customers". We have developed a procedure by which goal mapping can be considered in the preliminary stages of business domain modelling [FRA00b]. A cost optimisation model for goal mapping, based on work practices of the Space Shuttle Testing Facility at the NASA SSC, has been used to demonstrate the concept of goal mapping and its importance to the BDM.

## 1.5 Modelling in the Business and Software View phases

The business model for a typical large, modern enterprise resides primarily in the minds of its employees and in its software applications. One of the most costly disparities in such an enterprise is that between the mental models of its employees (which reflect the real world they deal with) and the implementation of business processes in its software applications. The Business and Software Views phases in AMORE are aimed at integrating the mental models of employees with the models of the IT applications. The Business View Phase in AMORE involves the elaboration of the Ideal Business Model (IBM) to the Object-Oriented IBM (OOIBM). The Software View Phase then develops the Real Business Model (RBM), which captures the business domain from an IT perspective.

Developing enterprise-wide business and system models prior to software implementation provides a conceptual blueprint that ensures the business users and the software engineers have a common understanding of the new software systems. This is analogous to an aerospace or automotive engineer developing models, blueprints, and precise engineering diagrams before manufacturing complex and costly products.

The IBM, derived from the BDM as described above, describes the business processes as operating in what is seen from a business perspective as an ideal situation. The IBM is specified by a process model derived from the BDM, and is represented using IDEF\*\*. The next step in AMORE is to translate the IBM into an OOIBM. The set of guidelines (IBM2OOIBM) that we are developing to achieve this will be discussed in the next section.

The choice of an Object-Oriented Ideal Business Model as the bridge to software specification is naturally suggested by current software practice as supported by modern techniques and tools. In the course of the last few years, the use of object-orientation has not only gained popularity in the software engineering arena for which it was first conceived, but has also been widely adopted in other areas [FR00b, UF99, UFD97]. AMORE uses the standard UML (Unified Modelling Language) concepts for representing the object-oriented view. The IBM is to be translated to the OOIBM using guidelines defined by AMORE. The OOIBM comprises two related models:

1. The business use case model, which shows the business system from the users' point of view. This is the *white-box* view.
2. The business object model, which is a detailed design of the internal structure of the business system. This is the *black-box* view.

The final step in AMORE is to translate the OOIBM to a Real Business Model (RBM). The set of guidelines (OOIBM2RBM) we are developing to achieve this will be discussed in the next section. The RBM is obtained by translating the business objects derived from the business object model and the use cases into an architecture of interconnected software objects. The end product of this translation is a set of software classes and objects, together with the set of dependencies between these classes (as defined by inheritance and association) and associated user interfaces.

AMORE uses IS use cases and IS logical view to represent the RBM. The RBM reflects the IBM from an IT perspective. The software generated from this model can be used directly to support the business domain. We are currently investigating how to translate this model to produce software that can be incorporated into commercial software such as Windchill [Win00].



## 1.6 Guidelines

Model translation is not trivial, and many approaches have been developed to tackle the problem. These approaches range from simple suggestions and heuristics to pure formal languages such as First-Order Calculus and formal grammar. The former are inadequate in exploiting formalism, whereas the latter are incapable of capturing the fuzziness and the imprecision intrinsic in process mapping. The AMORE approach aims to reconcile the two and to pursue an intermediate level of formalism, through a well-organised list of "how-to" related suggestions structured in a pattern-like form. The structure we use for the mapping guidelines is very similar to the canonical form of patterns due to Alexander [Ale98, Ale64]. It includes the following elements:

- Guideline Name
- Problem
- Context
- Forces
- Solution
- Rationale
- Resulting Context
- Related Guidelines
- Example

The set of guidelines used to map the IBM into OOIBM define the IBM2OOIBM language, whereas the set of guidelines used to map the OOIBM into RBM define the OOIBM2RBM language. The set of models can be treated as a state space. A state transition occurs whenever a guideline is applied to the model. A sequence of states can be regarded as the evolution of a solution. The solutions to a problem are those derived from the final states. The guidelines in AMORE are a means to guide IT practitioners towards a possible solution. It should be noted that it is difficult to formalise whether one state is more important than another as this is heavily dependent on personal judgement and the possession of domain-specific knowledge. For this reason, we propose to structure the guidelines in an ordered sequence so that the users can follow the guidelines in a systematic and logical fashion. To simplify the use of our languages, we will also provide entry-points for users so that the guidelines can be traversed in a particular order.

The transition from the IBM to OOIBM is achieved using the IBM2OOIBM language. This language is partitioned into three classes, associated with the paths depicted in Figure 3:

- IBM to Business Use Case Model Path (IBM2E)
- IBM to Object Model path (IBM2I)
- Business Use case Model to Business Object Model path (I2E)

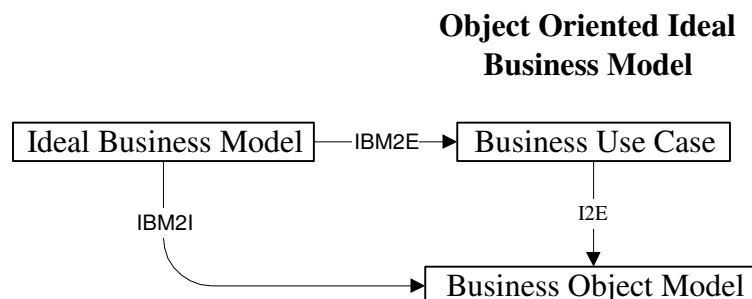


Figure 3: IBM2OOIBM mapping

The mapping guidelines are classified according to the three paths depicted in Figure 3, viz.:

**IBM2E**

The path from the IBM to an external and object-oriented representation of the business (Business Use Case)

**IBM2I**

The path from the IBM to an internal and object-oriented representation of the business (Business Object Model)

**I2E**

The path inside the OOIBM indicates the development of the Business Object Model based on the Business Use Case Model. This is done via an iterative and incremental process and is aimed at refining the object-oriented classes and their relationships. Steps purely representing refinement are outside the scope of this path. Since the guidelines are kept strictly for the purpose of process mapping, there will not be a comprehensive set of guidelines for developing the complete Business Object Model except where the mapping from the Business Use Case Model to the Business Object Model is concerned. A complete set of guidelines for the IBM2OOIBM mapping will be developed. A sample guideline for translating the location to business area is given in Table 2.

|                    |   |
|--------------------|---|
| Problem            | How to define different business areas from the OOIBM with respect to the IBM?  |
| Context            | Path IBM2E<br>You have used “Simplifying system” to reduce the system complexity.   |
| Forces             | Different profit centres have different characteristics and different needs.  |
| Solution           | <ol style="list-style-type: none"> <li>1. Define a business area for each location identified in the IBM which determines a profit centre;</li> <li>2. Give the business area the same name of location, or, alternatively, a representative name connected to location;</li> <li>3. Divide, if necessary, the previously identified use cases into the sub-use cases.</li> </ol> |
| Rationale          | <p>Large business systems are difficult to handle. If a company has several business areas and:</p> <ul style="list-style-type: none"> <li>• each of which being an independent profit centre</li> <li>• and being entirely autonomous</li> </ul> <p>Then there is no real reason to describe them in the same model.</p>   |
| Resulting Context  | <p>In order to avoid information loss due to the application of this guideline, create actors from sub-use cases. See FromSubUseCaseToActor.<br/>You may continue the IBM2OOIBM mapping by applying FromCustomerToActor.</p>  |
| Related Guidelines | <p>FromOrganisationUnitToBusinessArea<br/>FromProcessAndTaskToBusinessArea</p>  |
| Example            | <p>An example may be a multinational company with one product representing its core business; by definition, multinational operates in various countries, and so it is more natural to subdivide the business model into business areas, one for each country.</p>  |

Table 2: Sample Guideline

The OOIBM2RBM language is concerned with the transition from the business perspective towards the underlying IS. As depicted in Figure 4, the guidelines in the language are classified according to four main paths that can be described as follows.

- Business Use Case Model to Information System Use Case Model (BU2IU)
- Business Object Model to Information System Use Case Model (BO2IU)
- Business Object Model to Information System Logical Model (BO2LM)
- Information System Use Case Model to Information System Logical Model (IU2LM)

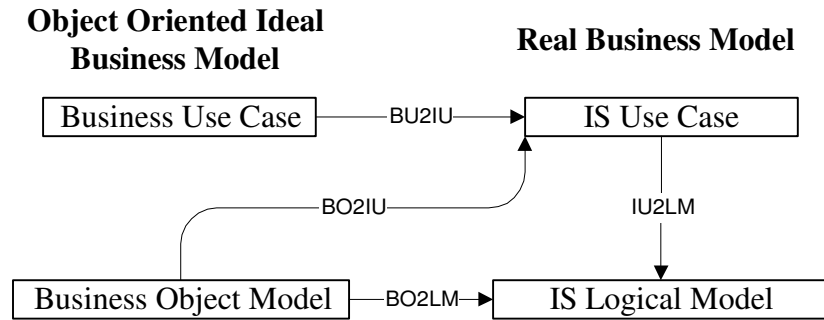


Figure 4: OOIBM2RBM mapping

The guidelines resemble those in the IBM2OOIBM language and are defined by a pattern-like syntax.

## 2. AMORE from an IT Perspective

This section reviews the AMORE proposal from the perspective of current thinking and practice in IT systems development. Useful background and orientation has been drawn from the account of Business Information Systems in Warboys et al [WKR99]. Complex systems made up of people and technologies define the relevant problem domain. AMORE identifies the problems of business modelling within existing approaches, such as that proposed by Jacobson et al [Jac92, JEJ94], as stemming from too much emphasis on software technology, and a failure to align software conception and development to human agency and high-level business concerns. Focussing on re-use of software objects, for instance, does not deal adequately with the human elements in business practice, as represented by goals, skills, instructions, drawings etc. The AMORE project is concerned with applying object-oriented principles in a way that integrates business and software perspectives on BPR. Its emphasis is upon current technologies and mature principles that might be of immediate practical use to industry, and with assessing the prospects for the effective application of best current practice.

Warboys et al [WKR99] take a more sceptical view of current BPR practice. In their view, BPR is typically conducted in such a way that “the nature of the IT systems is not elaborated”, whilst “from a software engineering point of view, it does not contribute to an understanding of how the software system should be constructed and adapted to their environment”. Business analysts can have too naive a perception of the software as a fixed entity which “although central, is not negotiable”. The need to revise this view of software is now widely acknowledged: it is not just that software systems are becoming more significant, but that the form of the software system is changing.

The AMORE project reflects the well-established view that object-orientation can unify the concerns of the business and the software analyst [Jac92, JEJ94]. Where Jacobson et al emphasise the extension of principles for object-oriented software development to business process modelling, AMORE aspires to a more high-level perspective, centred on the business domain. There is no direct counterpart in Jacobson’s approach to the Business View phase in AMORE, but the modelling principles applied in the Software View phase are essentially common to both. In the Software View phase, AMORE aims to promote flexibility through the use of components, design patterns, 3-tier architectures and CORBA interfaces, as complemented by guidelines and standards for good practice. For particular applications (such as manufacturing), these features are already supported by commercial packages (such as Windchill [Win00]).

Architectural frameworks such as Windchill illustrate what Warboys et al characterise as *design for change*. The proponents of design for change accept the inevitability of changes to the business process, and try to establish a framework that can flexibly accommodate software that can be “heterogeneous, of different ages, part bespoke, part general-purpose and dynamic”. Their aim is to support re-engineering by making the mechanisms and architectures that carry the business processes as versatile and adaptable as possible.

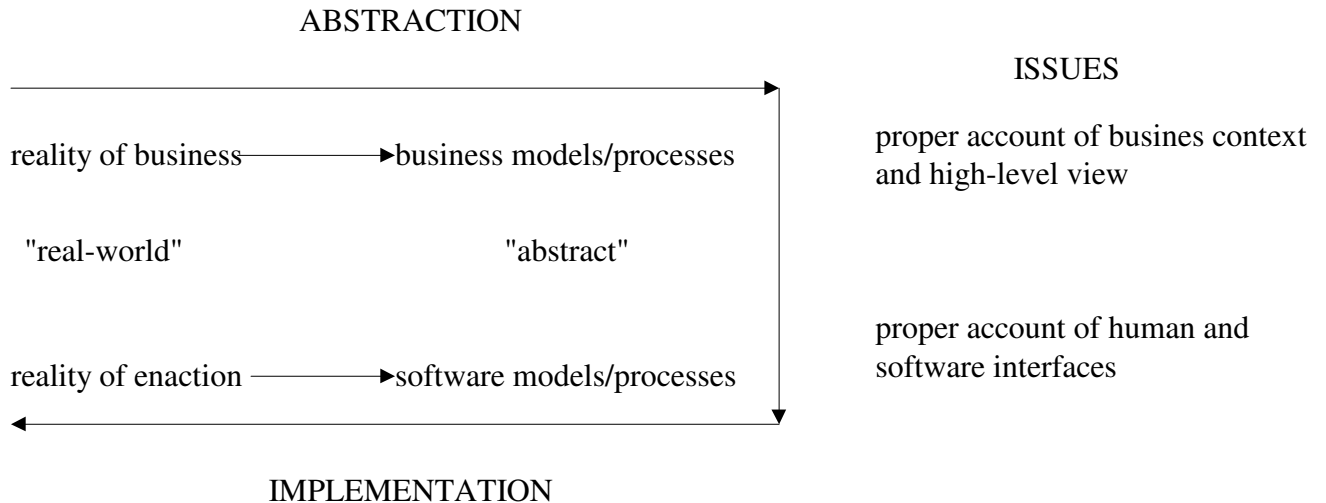


Figure 5: The challenge of realistic system engineering

Figure 5 supplies a useful framework within which to consider different approaches to BPR. A typical methodology involves an abstraction phase in which the business domain is analysed and the new business processes are conceived. These abstractly defined processes are then translated into software specifications and protocols for human users and actors for subsequent implementation. Real-world concerns are represented in two ways in such an exercise: at the initial stage of reappraising the business requirement, and at the final stage of validating the proposed solution. Both types of real-world concern constrain the business processes and software activities, and can provide the motivation for BPR. For instance, BPR can be initiated by a change in business goals, or by the introduction of new technologies.

A design for change paradigm, such as Jacobson's OA [JEJ94], focusses on making the software infrastructure for business processes as readily adaptable to change as possible, no matter what kind of change in business processes may arise. The Business View phase of AMORE aims to address the complementary concern for realism in BPR: relating business processes to their context in the business domain, and documenting the dependency in this relationship. Abstraction and implementation are respectively associated with the Business View and Software View phases in AMORE. Prototyping in the PAMS environment [PAMS] in the Business View phase, and the customer involvement it entails, aims to anchor abstract business processes to the real business context. The adoption of use-case and interaction diagram models for both business processes and software specification aims to integrate the business and software models. Finally, conformance to reality in enaction is to be guaranteed by software prototyping and exposure to users in the Software View phase.

Warboys et al - in common with AMORE - acknowledge that design for change, as in OA, is insufficient to bind abstract business processes to the real business domain. Their Organizational Process Modelling (OPM) approach [WKR99] focusses on 'process' rather than 'object' as its fundamental abstraction. OPM aims to unify business and software views through integrating processes. It is conceived as *design-in change*, in acknowledgement that processes will undergo change, and should be formulated with potential change in mind. This is in contrast to AMORE, where the business processes are established in the Ideal Business Model, and are not subject to revision thereafter. OPM addresses the need for change by developing a meta-process. In this framework, business process modelling designs dynamically changeable and efficient processes, and software process modelling generates the "process knowledgeable software" to support them. A key idea in OPM is that of the *active model*. Such a model is more intimately linked to its subject (its real-world referent) than a traditional passive model - in an active model, "the modelling relationship is maintained even though the elements of the subject may change". The use of active process models in the context of Figure 5 is intended to establish a dynamic connection between the business domain and the implemented business processes. An important feature of OPD is its acknowledgement that "objectives may merge in a bottom-up manner". This possibility is not envisaged in the phase-based and top-down elaboration of business processes in AMORE.

### 3. AMORE from an Empirical Modelling perspective

Empirical Modelling (EM) is a well-established programme of research at Warwick that is led by Beynon and Russ. The principles and tools currently available to support EM are not sufficiently mature to be directly applied to BPR, but a number of pilot studies indicate the potential novelty and interest of such application [BRR00a, BRR00b, RRB00, CRB00a, CRB00b, BWMWRR00, RRR00, RR00]. As an alternative approach to computer-based modelling for complex systems design, EM also provides a useful standpoint from which to review AMORE in its relationship to both OA/OOSE and OPM. The relevance of EM to the AMORE agenda is most evident in relation to the first stage of establishing the initial BDM, but - in the longer term - a well-conceived application of EM to BPR would have a pervasive impact on all the activities that feature in Figure 5.

#### 3.1 Background to Empirical Modelling

Empirical Modelling is a new approach to computer-based modelling. The term *interactive situation model* (ISM) that is used to describe an EM model reflects the fact that its construction proceeds incrementally in association with interaction both with the model and with the external state or situation to which it refers. The epithet ‘empirical’ in EM reflects the emphasis that is placed upon experiment and observation in ISM construction. The fundamental abstractions that inform EM are observables, dependencies and agency. EM is best conceived as a situated activity, in which the construction of the model proceeds in parallel with the development of insight into the relevant domain. The novelty of the approach rests upon using representations that are experiential rather than symbolic in character. That is to say, the ISM is a source of experiences that serve to represent experiences of its referent. Similar principles for knowledge representation are employed by experimental scientists when they create physical artefacts to express their understanding of a phenomenon (an activity that Gooding describes in [Goo90] as making construals), and by designers when they construct an engineering prototype.

The underlying philosophy of EM is akin to that which originally motivated object-oriented programming. In Simula [BDMN79], the aim was to demonstrate that programming could be treated as modelling. Birtwistle [Bir90], one of the original designers of Simula, regards the later association of object-orientation with modular software development [Par72] as a digression from this original agenda. The original objective of Simula is nonetheless still reflected in the aspiration for a seamless connection between object-oriented analysis and object-oriented design. The evidence gained from the EM project suggests that the fundamental abstractions of EM provide a more appropriate foundation for relating modelling to programming than do objects and methods. This is consistent with the ideas of Smith [Smi96], who sees the object as derived from more primitive abstractions. The difficulties in making the transition from OOA to OOD identified in [Kai99] can also be attributed to this source.

The development of EM has been associated with the construction of tools and models for a wide range of applications. These combine the use of principles for dependency maintenance (such as underlie spreadsheets) for computer-aided and concurrent design and decision support with techniques for agent-oriented analysis that address the role of both internal actors and external observers in concurrent system design, modelling, simulation and comprehension. More recent work, stemming from ongoing research by Russ, Beynon and their doctoral students to be reviewed below [BRR00a, BRR00b, RRB00, CRB00a, CRB00b, BWMWRR00, RRR00, RR00], highlights several significant characteristics pointing to applications in business process modelling and software development.

#### 3.2 EM and the comprehension of complex systems

In explaining the relevance of EM to BPR, it is helpful to consider two observations made by Warboys et al in [WKR99]: that “a program can be viewed as a statement of belief”, and that “the ... beliefs upon which behaviour is founded are elusive”. It is a matter of concern that, in approaches such as OA and OPM that promise to deal more effectively with process change, the basis for knowledge about behaviour is not explicitly addressed. In contrast, EM reflects the view that managing change in a

complex system is essentially bound up with beliefs about its behaviour, and can only be effectively treated in a framework in which the representation of this knowledge has the highest priority. Simple commonsense suggests this: we would hardly expect to be able to adapt a system that we had developed without a comprehensive understanding of how actual business process and programs operated in their real-world context, regardless of whether they were modelled using OA or OPM.

In EM, beliefs about behaviour are represented in a characteristic and distinctive manner. In comprehending and developing complex systems, the emphasis is upon postulating agency, dependency and observables and establishing how these are involved in reliable patterns of interaction and system behaviour. The use of the expressions “postulating (rather than *identifying*) agency” and “establishing (rather than *disclosing*) how these are involved in framing behaviour” is significant. It reflects the fact that the way in which a complex system is construed to behave is in the first instance a private and subjective matter. It follows from this that the representation of such beliefs cannot take the form of a propositional specification, but must involve the direct use of representations, such as ISMs, that are experiential in nature. A fuller discussion and justification for this point of view is to be found in the reappraisal of the foundations of computation and intelligence in [Bey99], and is supported by empirical evidence gained from the practical application of EM by several hundred Computer Science students.

In EM, the construction of an ISM is closely linked to comprehension, in that the interactive activity of making sense of the business domain and the development and validation of the ISM are interleaved and interdependent. Once a reliable correspondence between states and transitions of the ISM and those observed in its referent has been established, interaction with the ISM provides the primary representation of the domain understanding - albeit an implicit and non-propositional representation. Prior to establishing reliable state-transition patterns, interaction with the ISM has an experimental character, and is the primary means of enhancing domain understanding. There is a natural correspondence between the particular patterns of agency, dependency and observation that are embodied in an ISM and our expectations of plausible change in the referent. In commonsense terms, which changes in my environment I deem plausible, and which I deem miraculous or absurd is conditioned by my beliefs about how behaviours are framed by agencies, dependencies and observables that are informed by my experience. This explanatory dimension is not represented in current treatments of change in BPR - neither is the familiar informal layering and weighting of beliefs that is associated with experiences of different intensity and length of exposure.

The importance of modelling complex systems is of course widely recognised in all approaches to BPR. As Warboys et al assert [WKR99]: “Modelling (i.e. as in gaining experiential insight) is intrinsic to the comprehension of complexity”. What is lacking in conventional accounts of modelling is a conceptual framework that is as rich and subtle as logic itself has proved to be, and that does justice to the fundamental, rather than incidental role that interaction plays in comprehending real-world complexity.

EM and conventional modelling of complex systems are sharply distinguished by the character of their underlying beliefs about behaviour. In constructing an ISM, the modeller is not making incontrovertible assertions about objective facts, but attempting to imitate their own experience in a computer model. This activity is more tentative and provisional in character than formal specification, and is better suited to the typically imprecise and uncertain character of beliefs, and the awareness of exceptional conditions and singularities with which they are habitually qualified. This is consonant with the observation in Warboys et al that “a process model cannot, except in the most extreme case, be a valid, detailed prescription of the behaviour of people”.

Conventional computer-based modelling trades in abstractions and conceptions of systems that are far too sophisticated in experiential terms to respect the nuances of belief. Adopting the non-logicist perspective of [Bey99], an enormous body of empirical evidence is needed to support beliefs about the objectivity of different agent perceptions, for instance, or to endorse such a sophisticated conception as autopoiesis. Even objects and processes are richly informed by experience: objects through familiarity built up through acquaintance with many occurrences, and processes through repeated sequences of events in which such issues as commonality of perception, extending to an awareness of “location in an abstract process” on the part of some of the participants, play an essential part.

### 3.3 Prospects for EM in BPR

The potential application of EM to BPR is the current focus of research by Russ and Chen [CRB00a]. More details of this research (with particular reference to a warehouse case study introduced by Jacobson in [Jac92, JEJ94]), are given elsewhere [CRB00a, CRB00b]. It is already evident that EM has considerable promise as an approach to BPR, but that developing such an approach would require a radical re-orientation and large-scale investment in fundamental research, training and new tools. The work of Russ and Chen has proceeded in parallel with the AMORE project, and has drawn upon related EM research being carried out by Beynon, Sun, Maad and Rungrattanaubol. This section briefly reviews key aspects of EM of particular relevance to BPR that have emerged from research described in more detail elsewhere. The topics addressed in this work include: the application of EM to distributed, collaborative design support [BS99, BM00]; the use of ISMs in program specification [BRS99] and semi-automatic program construction [ABCY98]; EM as an amethodological approach to requirements understanding [Sun99, BS98] and software development [Sun99, BCSW99]; the potential uses of ISMs for program comprehension [BCSW99] and business systems integration [BM99]; EM as a tool for decision support in business applications [BRR00b, RRB00, RRR00, RR00].

An important feature of this work has been to highlight the crucial role of business process comprehension in BPR. The technical challenge that such comprehension presents is clear from the fact that understanding the role of IT in a business system is just one aspect. Relevant issues are:

- how does the code of a program affect its behaviour in its business context?
- how is the specification of a program shaped by the constraints in the business domain?

Many of the conclusions to which experience of EM points are consistent with those reached in state-of-the-art research on program comprehension [Cha97, CM97]. Comprehension activity emerges as amethodological in character, with an essential need for human involvement, and cognitive input. Interaction is required in order to understand the relationship between business processes and the interfaces to software, and this relationship is subject to evolve in ways that cannot be preconceived [Leh91, BRR00a, BWMWRR00]. The integration of computational paradigms poses particular challenges. Further examination of comprehension in relation to processes, programs and EM is the subject of current work in progress.

The manner in which ISMs are developed is in keeping with the spirit of the ‘active model’: as in the construction of a spreadsheet, respecting the semantic relation between the ISM and its referent is the guiding principle behind the model construction [BRR00a]. At the same time, the semantic relation in EM has a quite different character from that considered in OPM, where the emphasis is on correlating an abstract process model with real-world behaviours as observed or conceived. In the context of Figure 5, the understanding of the business domain and the elaboration of the business solution proceeds in parallel with the development of an ISM. Shaping the semantic relation between the ISM and its subject involves no shift of focus from reality to abstraction, but is established experientially - the computer model is itself interpreted as situated in the world. In a successful BPR exercise, this ISM will encompass both the requisite knowledge of behaviours and supply the prototypes for human and automatic agency.

In the application of EM to BPR, the rôle of an ISM is similar to that of an artefact an engineer might construct to inform the design of a complex system. Such an artefact typically provides specific information, confined to a limited but representative experimental context, that relates to one or more aspects of a complex system design. An ISM resembles such an artefact in that its significance is typically intrinsic in the interactions it admits, and may be only loosely related to the functionality of the whole system. For this reason, the ISMs that are developed in the initial stages of business process comprehension serve only to capture open-ended localised interactions with a real-world situation, rather than circumscribed comprehensive behaviours such as are associated with traditional processes or objects.

EM addresses many of the significant issues raised by Warboys et al in [WKR99]. An EM approach to software development [Sun99] gives great prominence to situated problem-solving activity [Suc87, WKR99]. It promotes active models, as illustrated by the re-engineering of user interfaces through exploiting dynamic statecharts discussed in [BCSW99]. EM lends itself to distributed use that supports “mutual learning” [WKR99], and also helps to address the challenge of integrating the knowledge

and skills of the experienced user and those of the modeller. It unifies the analyses of human and machine agency, allowing BPR to be addressed in a fashion “that contributes to how software systems should be constructed and adapted to their environment”.

As has been discussed, EM also offers an alternative response - one that is based on an experiential rather than a formal account - to Warboys’s call for a theoretical grounding for processes. Such a radical shift in perspective seems to be essential in the face of the difficulty - and arguably the futility - of attempting to express experiential distinctions using formal specifications and definitions. For instance, it is hard to see how a formal ontology can do justice to the different kinds of ‘sameness’ that are represented in discs cut from the same sheet of metal, coins passed in the same transaction, buses on the same route, and rocks deposited in the same meteoric shower.

In due course, it is to be hoped that EM can take account of the two very different perspectives on realism in Figure 5 in ways to which current approaches such as AMORE and OPD aspire. For instance, business goals may be viewed as very sophisticated observables, and autopoietic systems as highly sophisticated ways of construing the interaction of agents. The problems of reconciling object-oriented analysis and object-oriented design [Kai99] are associated with issues of comparable subtlety. Another long-term goal for EM is the automatic or semi-automatic generation of systems from ISMs [ABCY98].

EM encourages a radical reappraisal of the orthodox business perspective. The character and flexibility of ISMs is such that they encourage a different kind of relationship between human and automated business activity. This is illustrated in a recent application of EM to timetabling [BWMWRR00], where the possibility of liberating the user from strict patterns of interaction, and dispensing with the tight constraints of phased administrative processes is examined. The use of ISMs is seen as enabling a mode of operation that is far more loosely tied to process in the routine sense, and encourages a creative, opportunistic, situated problem-solving paradigm.

In conception and application, EM is unlike a methodology, and - like the so-called ‘scientific method’ - is primarily concerned with exploratory activity rather than systematic problem-solving procedures. Its basic presumption is that there is always experiential substance in the subject of which the modeller is unaware, that experience of the subject is a key element in acquiring insight, and that there can be no guarantee that such insight will lead to the attainment of preconceived goals. This agenda calls into question how far re-engineering can be addressed by proposing methodologies, standards and ontologies. It also engages with issues that conventional approaches to modelling typically fail to address. Where it can be successfully applied, EM leads to the construction of models that can still be useful in singular conditions, and can serve as a way to disclose where such singularity may arise. It can also be used to investigate matters of agency and observation, such as what knowledge, perceptions and skills are exercised in a role.

## 4. Concluding remarks

This paper has outlined a methodology for object-oriented re-engineering of enterprises. It is clear that, in the present state of maturity of software engineering, there is no plausible alternative to the object-oriented paradigm for practical industrial application of BPR. AMORE and the OPM school identify a similar weakness in object-oriented methodologies that have been proposed from an IT perspective. Their emphasis on design-for-change focuses too much attention on the flexibility of the mechanisms that support the business processes, and gives too little to the origin and character of the processes themselves. Methodologies such as OPM and [WPS00] try to address this by taking a process-centred approach. AMORE advocates a domain-centred approach, in which the goals and constraints that define the business domain model have a prominent and primary role.

There are certain business domains, such as that of manufacturing systems, where the presence of powerful toolkits (such as Windchill) suggest good prospects for successful application of AMORE. There are also reasons to be sceptical about AMORE as a general and comprehensive framework for BPR. As business information systems evolve from classical enterprise models that were effectively implemented with relational database technologies, they acquire the characteristics of reactive systems that have proved to be a major challenge for software engineering [Bro95, Bro97]. As approaches such as OPM acknowledge, software development for such systems is an activity in which change plays an



essential and integral part. This motivates active models within a design-in change paradigm, where there is scope for the conception of abstract processes to be influenced by their proposed implementation. Whilst AMORE aspires to take greater account of the business domain than the process-centred design-in change paradigm of OPM, it offers no new technical insight beyond orthodox OOSE into how the formidable problems of business process and software development can be attacked.

Empirical Modelling (EM) is a new and radically different approach to complex systems design. Its primary focus is on comprehension, and upon the use of computer-based interactive situation models that represent the way in which aspects of system behaviour are construed in terms of agents, observables and dependencies. Such use of the computer to create experiential representations of knowledge is associated with a non-logicist stance [Bey99] with far-reaching implications for the conceptual framework surrounding complex systems design. In this framework, concepts such as propositions, processes and objects lose their primitive status, and the primary focus of attention shifts from methodologies, procedures and behaviours to situated activity that engages with state-as-experienced through experiment and observation. It is commonly presumed that methodologies for BPR, such as OA, OPM and AMORE itself, can operate within a context in which the subversive influence of what William James terms 'pure experience' [Jam96b] is suppressed. Though EM is as yet too immature for practical application to BPR, it has already provided evidence to challenge this presumption. Research in program comprehension points to similar conclusions [Cha98]. In due course, this may provide a motivation for embracing the radical new agenda that EM entails.

The AMORE methodology is still under development. A case study (rework of parts), taken from British Aerospace System, is currently being implemented [AS00]. Further practical work on this case-study will help to put the thinking in this paper to the test.

## Acknowledgements

We are indebted to British Aerospace Systems, Sun Microsystems, Parametric Technology Corporation and the Engineering and Physical Science Research Council (M02675) for supporting the AMORE project. Our work has also been influenced by valuable input from Steve Russ, Yin-Chang Chen, Soha Maad, Elizabeth Burd and from Phil White and Martin Wright of Trinity Expert Systems. Special thanks are due to Gurprit Bilkhu for her help in preparing this paper.

## References

- [ABCY98] Allderidge, J.A., Beynon, W.M., Cartwright, R.I., Yung, Y-P. *Enabling Technologies for Empirical Modelling in Graphics*. Proceedings of the Eurographics UK, 16<sup>th</sup> Annual Conference, 199-213, 1998.
- [Ale64] Alexander, C. *Notes on The Synthesis of Form*. Harvard University Press, 1964.
- [Ale98] Alexander, C. *The Nature of Order*. New York, Oxford University Press, 1998.
- [AS00] Assman, G. and Stetter, R. *Flexible Processes For Technical Changes – The Integrated Engineering Change Management*. Proceedings of IEEE International Conference on Systems Integration. Incorporated in the proceedings of Integrated Design & Process Technology, 2000, The Society for Design & Process Science, 2000.
- [BCSW99] Beynon, W.M., Cartwright, R., Sun, P-H. and Ward, A. *Interactive Situation Models for Information Systems Development*. Proceedings of 'SCI' 99 and ISAS' 99, Vol. 2, pp. 9-16, Orlando, USA, July 1999.
- [BDMN79] Birtwistle, G., Dahl, O-J., Myrhaug, B., and Nygaard, K. *Simula Begin*. Chartwell-Bratt, 1979.
- [Bey97] Beynon, W.M. *Empirical Modelling for Educational Technology*. Proceedings of the Cognitive Technology' 97, University of Aizu, Japan, IEEE, 54-68, 1997.

- [Bey99] Beynon, W.M. *Empirical Modelling and the Foundations of Artificial Intelligence*. Computation for Metaphors, Analogy and Agents, Lecture Notes in Artificial Intelligence 1562, Springer, 322-364, 1999.
- [Bir90] Birtwistle, G. *Personal Communication*, 1990.
- [BM99] Beynon, W.M. and Maad, S. *Empirical Modelling of Real Life Financial Systems: The need for Integration of Enabling Tools and Technologies*. Proceedings of IEEE the International Conference on Systems Integration. Incorporated in the proceedings of Integrated Design & Process Technology, The Society for Design & Process Science, 1999.
- [BM00] Beynon, W.M. and Maad, S. *Integrated Environments for Virtual Collaboration: an Empirical Modelling Perspective*. Proceedings of the IEEE International Conference on Systems Integration. Incorporated in the proceedings of Integrated Design & Process Technology, The Society for Design & Process Science, Texas, 2000.
- [Bro95] Brooks, F. *The Mythical Man-Month Revisited: Essays on Software Engineering*. Addison-Wesley, 1995.
- [Bro97] Brooks, F. *No Silver Bullet: Essence and Accidents of Software Engineering*, *Computer*, Vol.20, No.4, pp.10-19, 1987.
- [BRR00a] Beynon, W.M., Rasmequan, S. and Russ, S. *The Use of Interactive Situation Models for the Development of Business Solutions*. Proceedings workshop on Perspective in Business Informatics Research (BIR-2000) Rostock, Germany, March 31- April 1, 2000.
- [BRR00b] Beynon, W.M., Rasmequan, S., Russ, S. *A New Paradigm for Computer-Based Decision Support*. To appear in Journal of Decision Support Systems Special Issue, Autumn 2000.
- [BRS99] Beynon, W.M., Rungrattanaubol, J. and Sinclair, J. *Formal Specification from an Observation-Oriented Perspective*. 15th British Colloquium in Theoretical Computer Science, Keele University, April 1999.
- [BS98] Beynon, W.M. and Sun, P-H. *Empirical Modelling: a New Approach to Understanding Requirements*. Proceedings of the 11<sup>th</sup> International Conference on Software Engineering and its Applications, Vol. 3, Paris, December 1998.
- [BS99] Beynon, W.M. and Sun P-H. *Computer-mediated communication: a Distributed Empirical Modelling perspective*. Proceedings of CT' 99, San Francisco, August 1999.
- [BWMWRR00] Beynon, W.M., Ward, A., Maad, S., Wong, A., Rasmequen, S. and Russ, S. *The Temposcope: a Computer Instrument for the Idealist Timetabler*. To appear in the Proceedings of the 3<sup>rd</sup> International Conference on the Practice and Theory of Automated Timetabling, Constance, Germany, August 2000.
- [Cer94] Cerpedes, F.V. *Industrial Marketing: Managing New Requirements*. Sloan Management Review, Vol. 35, pp. 45-60, Spring 1994.
- [Cha98] Chan, P-S. *An Integrated Approach to Program Comprehension: An Application of Software Visualisation*. PhD thesis, Department of Computer Science, University of Durham, 1998.
- [CKO92] Curtis, B., Kelner, M.I. and Over, J. *Process Modeling*. Communications of the ACM., Vol. 35, No. 9, pp. 75-90, 1992.
- [CM97] Chan, P-S. and Munro, M. *PUI: A Tool to Support Program Understanding*. Proceedings of the IEEE 5th International Workshop on Program Comprehension. Dearborn, Michigan. IEEE Computer Society Press. pp. 192-198, May 28-30, 1997.

- [CRB00a] Chen, Y-C., Russ, S. and Beynon, W.M. *Empirical Modelling for Business Process Re-engineering: An Experienced-Based Approach*. Proceedings of the Workshop on Perspectives in Business Informatics Research, Rostock, Germany, 2000.
- [CRB00b] Chen, Y-C., Russ, S., and Beynon, W.M. *Participative Process Modelling*. To appear in the Proceedings of IEEE International Conference on Systems, Man, and Cybernetics, Nashville, Tennessee, USA, October 2000.
- [FR00a] Fernandes, K.J., and Raja, V. *Business and Software Phase Synergy in a Re-engineering Environment*. Proceedings of the 9th Annual Industrial Engineering Research Conference, Cleveland, Ohio, May 21-23, 2000.
- [FR00b] Fernandes, K.J., and Raja, V. *Incorporated Tool Selection System using Object Technology*. International Journal of Machine Tools and Manufacture., Vol. 40, No. 11, pp. 1547-1555, 2000.
- [FR00c] Fernandes, K., and Raja, V. *IDEF\*\*: A Modeling Methodology supporting Business Domains*. Working Paper, 2000.
- [FRA00a] Fernandes, K., Raja, V., and Antony, J. *Knowledge Transfer using Publish Subscribe Technology*. Proceedings of the Knowledge Management Beyond the Hype Conference, Aston University, July 2000.
- [FRA00b] Fernandes, K., Raja, V., and Antony, J. *Optimum Level of Goal Mapping in a Re-engineering Environment*. Submitted for possible publication on 06/07/00.
- [FRM00] Fernandes, K.J., Raja, V.H., and McKinney, L. *Knowledge Transfer in a Cross-Functional Environment*. Proceedings of the Knowledge Management: Concepts and Controversies conference, University of Warwick, pp. 67, February 2000.
- [GF94] Gruninger, M. and Fox, M.S. *The role of competency questions in enterprise engineering*. Proceedings of the IFIP WG5.7 Workshop on Benchmarking – Theory and Practice, pp. 212-221, 1994.
- [Goo90] Gooding, D. *Experiment and the Making of Meaning*. Kluwer, 1990.
- [Jac92] Jacobson, I. *Object-oriented Software Engineering : A Use Case Driven Approach*. Wokingham: ACM Press, Addison-Wesley, 1992.
- [Jam96a] James, G. *Intranets rescue re-engineering*. Datamation, Vol. 42, pp. 37-45, Dec. 1996.
- [Jam96b] James, W. *Essays in Radical Empiricism*, Dover 1996.
- [JEJ94] Jacobson, I., Ericsson, M., and Jacobson, A. *The Object Advantage*. Reading, MA: Addison-Wesley, 1994.
- [Kai99] Kaindl, H. *Difficulties in the Transition from OO Analysis to Design*. In IEEE Software, pp. 94-102, September/October 1999.
- [KR96] Kratz, N., and Rose, T. *Modelling and analysing processes in production and administration*. Computer-Assisted Management and Control in Manufacturing Systems, pp. 122-147, S.G Tzafeta, Ed. Berlin, Germany: Springer-Verlag, 1996.
- [Leh91] Lehman, M.M. *Software Engineering, the software process and their support*. IEEE Software Engineering Journal, Vol. 6, No. 5, September 1991.
- [LGJMTY97] Lee, J., Gruninger, M., Jin, Y., Malone, T., Tate, A., and Yost, G. *PIF: The Process Interchange Format v.1.2*. Technical Report, MIT Centre for Coordination Science, 1997.

- [MKJ98] Motwani, J., Kumar, A., and Jiang, J. *Business Process Re-engineering: A Theoretical Framework and an Integrated Model*. International Journal of Operations & Production Management, Vol. 18, No. 9/10, pp. 964-977, © MCB University Press, 0144-3577, 1998.
- [MR98] McKay, A., and Radnor, Z. *A characterization of a business process*. International Journal on Operations and Production Management., Vol. 18, pp. 924-936, 1998.
- [Ner92] Nerson, J.M. *Applying Object-Oriented Analysis and Design*. Communications of the ACM., Vol. 35, No. 9, pp. 63-74, 1992.
- [PAMS] PAMS, website address [http://www.spede.co.uk/pro\\_over/overview/pams/pams.htm](http://www.spede.co.uk/pro_over/overview/pams/pams.htm).
- [Par72] Parnas, D. *On the criteria to be used in decomposing systems to modules*. Communications of the ACM. Vol. 15, No. 2, pp. 1053-8, 1972.
- [Pro00] ProModel, website address <http://www.promodel.com>.
- [Psl00] The Process Specification Language (PSL): Overview and Version 1.0 Specification" NIST Internal Report (NISTIR) 6459.
- [PW] *Process Weaver (Process management tool)*. Process Weaver website. <http://sdt.cern.ch/ProcessWeaver>.
- [RR00] Rasmequan, S., Russ, S. *Cognitive Artefacts for Decision Support*. To appear in Proceedings of IEEE International Conference on Systems, Man and Cybernetics, Nashville, Tennessee, USA, October 2000.
- [RRB00] Russ, S., Rasmequan, S. and Beynon, W.M. *An Experience-Based Approach to Decision Support Systems*. Position paper for IFIP TC8/ Working Group 8.3, International Conference on Decision Support Through Knowledge Management, Stockholm, Sweden, July 9-11, 2000.
- [RRR00] Rasmequan, S., Roe, C. and Russ, S. *Strategic Decision Support Systems: An Experience-based Approach*. Proceedings of the 18<sup>th</sup> IASTED Conference on Applied Informatics, Innsbruck, Austria, 2000.
- [Sch92] Scheer, A.W. *Architecture of Integrated Information systems: Foundations of Enterprise Modelling*. Heidelberg, Germany: Springer-Verlag, 1992.
- [Smi96] Smith, B.C. *On the Origin of Objects*. MIT Press, 1996.
- [SPEDE] SPEDE, website address <http://www.spede.co.uk>
- [Suc87] Suchman, L.A *Plans and Situated Actions: the problem of human-machine communication*. Cambridge University Press, 1987.
- [Sun99] Sun, P-H. *Distributed Empirical Modelling and its Application to Software System Development*. PhD Thesis, Department of Computer Science, University of Warwick. 1999.
- [UF96a] Usher, J.M., and Fernandes, K.J. *Dynamic Process Planning - The Static Phase*. Journal of Materials Processing Technology, Vol. 61, pp. 53-58, 1996.
- [UF96b] Usher, J.M., and Fernandes, K. *A Two-Phased Approach to Dynamic Process Planning*. Computers and Industrial Engineering, Vol. 31, No. 1&2, pp.173-176, 1996.

- [UF99] Usher, J.M., and Fernandes, J. *An Object-Oriented Application of Tool Selection in Dynamic Process Planning*. International Journal of Production Research, Vol. 37, No. 3, pp. 2879-2894, 1999.
- [UFDS97] Usher, J.M., Fernandes, K.J., Dhage, S. and Sharma, G. *An Object-Oriented Implementation of Tool Selection for Dynamic Process Planning*. Proceedings of the 6th Industrial Engineering Research Conference, 895-900, May 1997.
- [Win00] *WindChill*. Website address, <http://www.ptc.com/products/windchill/index.htm>.
- [WKR99] Warboys, B., Kawalek, P., Robertson, I., and Greenwood, M. *Business Information Systems: a Process Approach*. McGraw-Hill, 1999.
- [WPS00] Weber, H., and Padberg, J. and Sunbul, A. *Evolutionary Development of Business Process Centred Architecture Using Component Technologies*. Proceedings of the Fifth World Conference on Integrated Design & Process Technology, The Society for Design & Process Science, Texas, 2000.