Systems, Signs & Actions
An International Journal on
Communication, Information Technology and Work
Vol. 1 (2005), No. 1, pp. 38–61
http://www.sysiac.org/

Requirements Reengineering from Legacy Information Systems Using Semiotic Techniques

Kecheng Liu
Informatics Research Centre, The University of Reading, UK

Abstract

Recovering users and systems requirements from existing information systems can be valuable in legacy systems reengineering. AMBOLS, a semiotic approach to requirements reengineering, considers recovering the requirements of an existing (legacy) system under the restriction that only the operational system and its current stakeholders are accessible, while the original system documentation may be inaccurate or unavailable. The AMBOLS approach is adopted which consists of investigative activities at three major stages with a set of techniques for analysis and representation. The techniques consider a legacy system from the perspectives of different stakeholders, from its interaction with the users in the business context, and from information content and processes of system operations. The results of analysis from these perspectives are synthesised for derivation of the requirements. A case study of a university library system is used to provide examples. Conclusions are drawn on the need and effectiveness of such an approach to reverse requirements engineering, and further work is suggested.

Keywords: Requirements engineering, Legacy systems reengineering, Information systems, Organisational semiotics, Semantics

Received: 24 May 2004; Revised: 11 January 2005; Accepted: 25 February 2005
Accepting Editor-in-Chief: Peter Bøgh Andersen

1 Introduction

As the business scope expands and business processes change, many IT systems become inadequate in terms of capacity and functionality. However, these "legacy systems" may provide critical information and essential support for business operations as many of them have long been integrated into the core business processes, and therefore cannot be simply scrapped.

Different from routine software maintenance, typical legacy systems modernisation involves coarser-grained, higher level and structural changes which can be realised by a combination of renovation activities from both bottom-up and top-down (Weiderman *et al.* 1997a; 1997b). While the former approach reconstructs the system design by analysing the source code, the top-down methods treat the system as a black box and "examining the inputs and outputs of a legacy system within an operating context", which is preferable for achieving high-level systems understandings (Tilley 1998; Weiderman *et al.* 1997b).

The semiotic approach to requirement engineering, Analysing and Modelling the Behaviour of Legacy Systems (AMBOLS) (Liu *et al.* 1999a; Liu *et al.* 1999b), aims to derive user requirements for a legacy system through observation of the system and its users during operation, and by interviewing its stakeholders about what capabilities it provides to them. The derivation is to be made under the assumption that nothing but the operational system and its users are capable of providing relevant information. We assume that any documentation, including source code, is out-of-date or inaccurate. We also assume that there is no one available who has insight technical knowledge of the system. These conditions are common in commerce and government. Our aim in the project is to construct a top-down black-box reverse requirements engineering approach.

The AMBOLS approach is also useful for systems replacement. The recovered legacy system requirements can serve as a reference point for the new systems to ensure that new technical solutions should surpass the legacy systems. The old requirements can be used as a basis for validation of the new requirements, and also enable the developers to learn from the experience and lessons of the old systems. Since its publication, the AMBOLS approach (Liu *et al.* 1999a; Liu *et al.* 1999b) has attracted interest from the research community and industrial users, and has been applied in systems re-engineering projects (Xie & Liu 2002; Xie *et al.* 2003).

This paper will briefly examine the related work in the next section, where a quick comparison sets up a landscape of legacy systems re-engineering. This will help readers better appreciate the features of our approach. Section 3 discusses Organisational Semiotics, the theory on which the AMBOLS is based and the two key semiotic techniques for business and technical systems analysis. Section 4 presents an overall architecture of the AMBOLS approach and introduces the major techniques employed in different stages of requirements recovery. Section 5 illustrates the application of the AMBOLS approach through a case study of a distributed university library. The last section concludes the paper by summarising the major aspects involved in the requirements recovery and the future direction of the research.

2 Related Work

In 1996, an EPSRC funded research programme of Systems Engineering for Business Process Change (SEBPC) was launched in the United Kingdom to address the business and technological issues of legacy systems. The research programme aims to combine the skills of systems engineering and business process communities to understand better how to map business process change, economically and incrementally, onto large, installed IT systems (Henderson 2000). The work of the thirty funded projects within the programme has brought a significant impact to the research and industry in the UK. AMBOLS as one of the projects in this programme efficiently completed in 2001 but the application of the approach has continued in research and industry (Xie & Liu 2002).

Much work has been done to modernise the legacy systems by investigating the available documents and source codes. Major industrialised methods include screen scrapping, database gateway, XML integration, database replication, CGI integration, object-orientation wrapping, and componentisation of legacy systems (Comella-Dorda *et al.* 2000). Strategic issues have been raised (Rolland *et al.* 1998; Tilley 1998) and methodologies and practices reported (Ashish & Knoblock 1997; Diaper

1989; Graham 1994; Kim 1995; Vidal *et al.* 1998; Weiderman *et al.* 1997b; Yan *et al.* 1997), most of which can be categorised as bottom-up approaches. Object-oriented methods provide another approach to the management and use of data resources from legacy systems. Great effort has been put into the integration of heterogeneous databases, client/server architecture and distributed databases to achieve interoperability between legacy systems (e.g. Bukhres & Elmagarmid 1996; Comella-Dorda 2000; Kolkman 1995; Seacord 2001). Data warehousing and object wrappers are also found in practical applications as well as in research for dealing with legacy systems (Henderson 2000). Different task analysis techniques (El-Ramly 2002) are also extensively used to analyse existing systems often with a view for redesign. The Rational Unified Process (Booch *et al.* 1999), an influential approach to enterprise modelling and information systems engineering, offers several modelling techniques for systems integration; however, none of these deals with recovery of legacy systems requirements. The AMBOLS approach recognises the experience and need of the traditional reverse engineering and object-oriented approaches, but also investigates the benefits of the top-down approaches.

The top-down black-box approach is relatively less explored. El-Ramly *et al*. (2001) proposes a method to recover requirements from systems-user interaction, based on similar assumptions to ours. Yu (1997) proposes the i* framework for process reengineering based on strategic relationships, which has a strong AI and knowledge management flavour.

Deriving original intentions embedded in the legacy systems must be based on a number of assumptions. Even if we allow us to assume that people are able to state what they want, which is often subject to many constraints, fundamental challenging questions remain such as how people perceive, understand and express the world, and how the organisational, cultural and contextual rules govern people's behaviour.
Some branches of philosophy, linguistics and cognitive psychology have formed systematic answers to these questions and therefore can be used as theoretical foundations for this work. In the AMBOLS approach, we adopt organisational semiotics as our major theoretical foundation.

3 Organisational Semiotics

C. S. Peirce founded semiotics as the "formal doctrine of signs" (Peirce 1960). A sign is defined as something that stands to someone for something else in some respect or capacity. Organisational semiotics and the analytical methods (Liu 2000; Liu 2004; Liu *et al.* 1999a; Liu *et al*. 2002; Liu *et al*. 2001; Stamper 1997; Stamper 2001) offer a theory to understand business organisations, with or without the computerised information systems. Organisations are deemed as systems where signs are created, transmitted, and consumed for business purposes. For example, although the business of a retail store involves the substantive behaviours like ordering, display, and selling goods, it can also be deemed as an organisation that creates, transmits, and consumes signs like bar codes, stock records, display records, prices, cashes, credits, and receipts, etc.

Ronald Stamper's school of organisational semiotics argues that in contrast to the concept of information, signs offer a more rigorous and solid foundation for understanding information systems. For example, within a business context, a bank note is much more than a piece of coloured paper with digits on. It stands for the bank note holders' wealth and the purchasing power, as well as the issuing bank's authority, obligation and much more. It would be oversimplifying and even misleading to abstract the business concepts into pure digits, overlooking the attached social relationships and behavioural possibilities.

Organisational semiotics adopts a subjectivist philosophical stance and an agent-in-action ontology. This philosophical position states that, for all practical purposes, nothing exists without a perceiving agent and the agent engaging in actions. Any knowledge must be represented in the following form:

Agent - *behaviour*

The classical distinction between entity, attribute and relationship disappears to be replaced by the concepts of agents (a responsible person, a group of people, or an organisation), affordances (the ability, pattern of actions or behaviours of the agents) and norms (the socially defined patterns of behaviour) related to their antecedents to indicate the ontological dependency. Ontological dependency denotes that the existence of one thing (antecedent) is a necessary precondition for the existence of the other (dependent). The antecedent must exist during the whole period of the existence of the dependent, not just at the start or the finish.

The agent-in-action ontology has important implications for requirements engineering (Liu 2000). In particular, it requires the relevant agent to be specified in every component of the requirements definition. The benefits of this constraint are that (a) we know the originators responsible for production of every part of the requirements, (b) we know exactly whom to consult over details of design, (c) terms are clearly defined within a context of responsible agents, their behaviours and lifecycle, (d) we can organise norms (system functionality) according to the various group agents involved. Clearly a philosophical position is not a peripheral issue but has fundamental practical relevance.

Organisational semiotics makes use of some important psychology concepts such as affordances to tackle the problem of how people perceive and interpret the business concepts, and thus deals with semantics in a more profound manner. Based on a rigorous semantic model, the behavioural patterns of the responsible people and organisations appear to be an effective way to represent requirements.

3.1 Semantic Analysis

Semantic Analysis (Liu 2000; Stamper 1997) is one of the semiotic techniques we used in the reverse requirements engineering. In developing these techniques for requirements elicitation and representation, Ronald Stamper adopts the concept of affordance from the perceptual psychologist James Gibson, who defined "the affordances of the environment" as "what it offers the animal, what it provides or furnishes, either for good or ill…" (Gibson 1979). Based on the theory, the focus of an analysis is on the patterns of behaviour of an organisation or its members. Such entities are seen as agents that perform actions and interact with the social and cultural environment. The effective approach to requirements engineering is to capture the "invariant repertories of behaviours, either substantive affordances or social norms" (Stamper 2001) that are available to the responsible agents. For example, in the context of a university library, what happens to a book from various stakeholders' perspectives determines the functional requirements on the book.

A book to be possibly borrowed by someone offers a potential capability (i.e. an affordance), which may or may not be realised in the business context. However, once it is realised, new possibilities may emerge. For example, a borrowed book may be retained or returned to the library by the user. Under certain circumstances, the library may also call it back. This shows that affordances have dependency relationships among them. In organisational semiotics this relationship is called ontological dependency.

We may schematically show this relationship as following, with the antecedents on the left side and the dependencies on the right, and the solid line denotes the ontological dependency:

book — borrow — return

Ontological dependency does not just show a dependency between the concepts. It shows the dependencies between patterns of behaviour: the book which is possible to borrow can afford the action of returning. Therefore, between the antecedents on the left and dependents on the right, there exists an ontological dependency – a precondition for the pattern of behaviour on the right side of the line to be meaningful. The existence of the dependencies may not exceed the existence of the antecedents; the precise semantics of the dependents is bounded within the context defined by the antecedents. This ontologically bounded, behaviourist paradigm allows rigours modelling of an application domain with precise representation of business semantics.

3.2 Norm Analysis

Another technique used in the requirements modelling is the Norm Analysis (Liu & Ong 1999; Stamper & Liu 1994; Stamper *et al*. 2000). Norms, which define the dynamic conditions of the pattern of behaviour, exist in a community and govern how members behave, think, make judgements and perceive the world. The shared norms, formal or informal, explicit or implicit, define a culture or subculture. A subculture, for example, can exist in a team so that the members of the team can behave like a team and act in a coordinated manner. Norms include formal and informal rules and regulations. Their presence enables the individuals and organised group to exhibit normative patterns to be more or less “predictable”. Otherwise, a static semantic model may represent the patterns of behaviour, but no one would be able to talk about realisation or instantiation of a particular pattern which may be triggered by meeting certain conditions.

Four types of norms exist, namely evaluative norms, perceptual norms, cognitive norms and behavioural norms; each governing human behaviour from different aspects. In business process modelling, most rules and regulations fall into the category of behavioural norms. These norms prescribe what people must, may, and must not do, which are equivalent to three deontic operators “of obligation”, “of permission”, and “of prohibition”. Hence, the following format is considered suitable for specification of behavioural norms:

whenever <condition>
if <state>
then <agent>
is <deontic operator>
to <action>

With the introduction of deontic operators, norms are broader than the normally recognised business rules; therefore provide more expressiveness. For those actions that are "permitted", whether the agent will take an action or not is seldom deterministic. This elasticity characterises the business processes, and therefore is of particularly value to understand the organisations.

If we know the various norm-subjects who are the agents in the organisation and we know the specific norms they should obey, then we can deduce what information individual or collective agents in the organisation will require at what time for what purposes.

This paper will only focus on the semiotic techniques, although the reverse requirements engineering also deploys other techniques which can be found in the mainstream literature. Detailed discussions of the semiotic techniques, moreover, will be offered in further discussions with the help of examples.

4 AMBOLS Requirements Recovery Approach

Based on organisational semiotics, an IT system is an integral part of the organisation and the system functions are governed by the norms. Requirements recovery for an IT system must start from the analysis of the business domain and operations, as reflected in the AMBOLS approach.

The whole process of requirement recovery is organised into a series of activities in three stages: Behaviour Capture, Dynamic Behaviour Modelling, and Requirements Derivation.

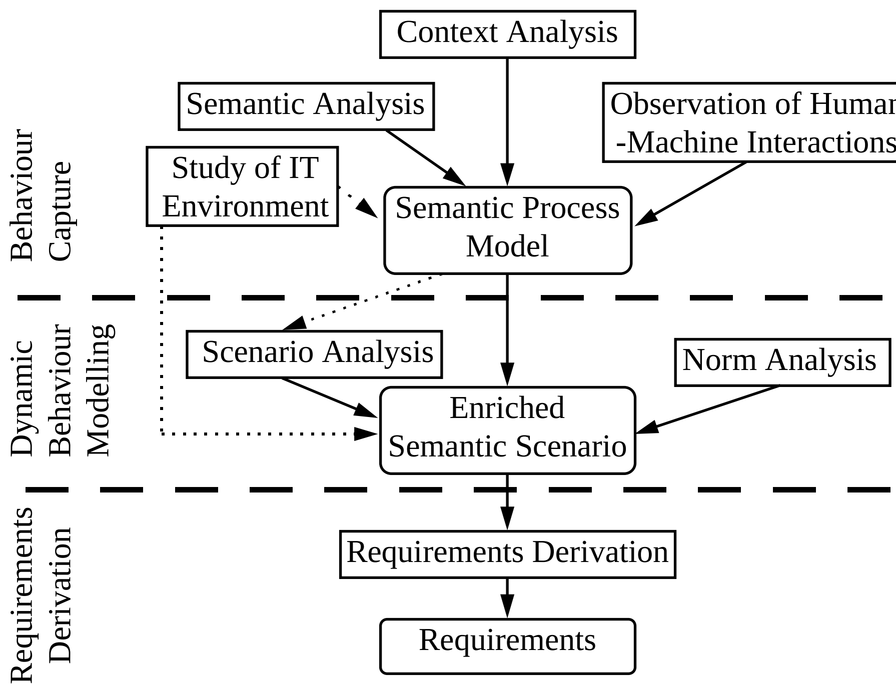


Figure 1: The AMBOLS Approach.

AMBOLS uses a number of techniques available from the fields of software engineering, requirements studies and information systems design, with extensions and alterations when appropriate. These techniques are described and then the derivation of requirements is addressed.

4.1 Stage One: Behaviour Capture

This stage of analysis in AMBOLS focuses on external behaviour of the system and its context in which the system operates, which is achieved mainly by observation and interviews. The first task in this stage, the context analysis, is to examine the organisational and business environment wherein the IT system operates. The analysis covers the domain, organisational structure and business processes. The purpose is to understand the nature of the business and major business functions, and to identify stakeholders. A 'soft systems method' (Kolkman 1995) provides a basis for developing a checklist for stakeholders, which includes suppliers, customers, competitors, governing bodies and users of the system. The use case method (Booch *et al*. 1999) is used to represent the relationships between stakeholders and major clusters of system functions. This investigation is done through interviews and collection of any relevant literature.

The observation of human-machine interactions is carried out by observing and studying how the system is used by all user groups. A similar method has also been proposed by El-Ramly *et al*. (2002). In this step, detailed notes are taken and, often, additional explanations from the users are required regarding the meanings, purposes of and reason for certain operations. The human-machine interactions not only include the screenshots and keyboard inputs, but also other kinds of interactions, e.g., e-mail. Figure 2 shows an email received from the university email system recalling a borrowed book.

Date/Address
User Name: *********** ID: *********
Department: *********

Dear ****:

URGENT RECALL NOTICE

The following item has been requested by another user.
Please return it as soon as possible and no later than the new due date printed on this notice.
Thank you.

Book Name: **********/ Author(s): ***********
Call number:********** ID:***********
original date due: 18/2/2003,23:59
NEW date due: 21/1/2003,23:59

Figure 2: An email to recall a book.

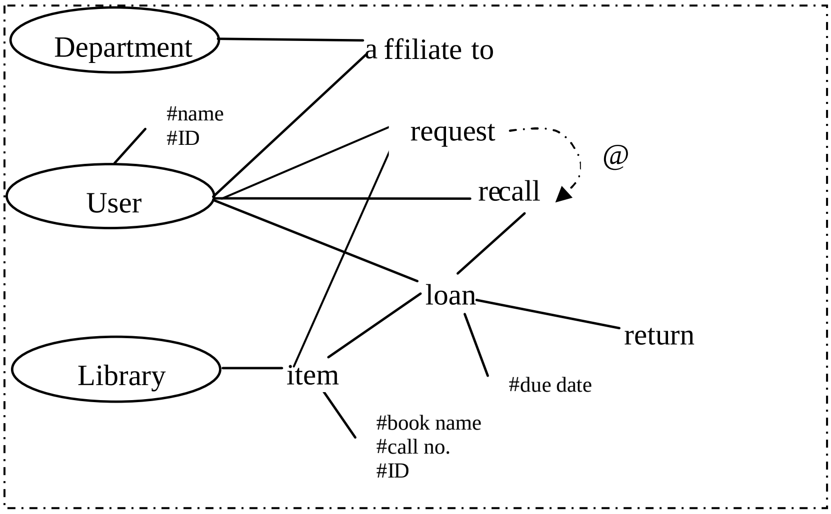


Figure 3: Semantic Analysis on the library recall notice.

Semantic analysis clarifies and captures the meanings of concepts, terms and functions. Using semantic analysis, we can organise these concepts and terms into a semantic model, which is called an Ontology Chart. An ontology chart describes the patterns of behaviour of human actors and IT systems, and defines the relationships of concepts and terms that correspond to the patterns of behaviour. Figure 3 shows an example on how the terms used in the recall notice can be organised using the method of semantic analysis.

The ontology chart in the figure can be interpret as following: A user has attributes like name and ID; a user is affiliated to a department; an item exists within the context of the library, and has attributes like book name, call number and ID; a user can loan an item, and the existence of the user and the item is the pre-requisite for the behaviour "loan" to be meaningful. Once the loaning is realised, an attribute "due date" is attached to the loan. A user can request an item and if that item is on loan, the system can generate a "recall" on the "loan". The ontological dependency between "loan" and "recall" is important because "loan" forms the context for the behaviour "recall". For an item which is at the point of time not on loan, such behaviour is meaningless. The dotted line with a sign "@" represents a semiological relationship (as opposed to an ontological one), which means a request may or may not lead to a recall, depending on some authority's (e.g. a librarian's) decision. It is important to know who the actor is for each behaviour. Semantic analysis meets these conditions by specifying 1) who does what, e.g., a user loans an item, a user returns the loan, the library recalls an item, and 2) under what circumstances the above affordances could come into existence, e.g., an item must be on loan to a user before it can be returned, and a recall can be made during the loan. These contexts may seem straightforward at first sight. But when the systems grow larger and more complex, such clearly defined contexts will be necessary as they build a rigorous representation of understanding of the information systems.

The information collected from the context analysis, human-machine interaction and semantic analysis can be incorporated into a semantic process model. In other words, the interactions between the user and the system can be described in a process model, while the concepts and terms used in the process models are defined in semantic models. Role-activity diagrams (Ould 1995) are chosen as a base technique for representing processes, as they can be easily enriched by the semantic analysis method to clarify meanings.

4.2 Stage Two: Dynamic Behaviour Modelling

Analysis at this stage takes a process perspective, which places an emphasis on the sequences of actions and interactions between human and machine. Information items relating to business concepts and operations, actions performed by the user and the system, and governing norms for these actions are the objects of study at this stage (as seen in Figure 1).

Scenarios are widely used in the software engineering community as a means to represent the problem domain for requirements elicitation, representation and analysis. There are at least four recognised views of the contents and the use of scenarios (Filippidou 1998):

- Process view: scenarios describe the sequences of ‘actions’ or ‘events’, not individual acts.
- Situation view: a scenario represents a situation or an ‘episode’ with a temporal component.
- Choice view: scenarios represent alternative options for selection for further design or implementation.
- Use view: scenarios enable users (e.g. an end-user or designer) to focus on the target artefact from outside the boundary of the system.

We adopt the process view, with a supplement of a temporal component. Scenarios represent the stakeholders, actions and events in a process oriented model within an explicit time-based framework.

Norm analysis (Stamper *et al.* 2000) enables us to elicit and represent business rules that have an effect on the system. Norm analysis recognises various types of rules in a broad sense, which may be either embedded in the system or in the procedures of using the system:

- Rules that can be specified explicitly as conditions and constraints for system operations;
- Rules defining responsibilities and authorities;
- Rules for reasoning and decision-making specified by possible conditions and resulting consequences;
- Ethical values, socially acceptable practices and business standards.

Norms governing the business give us access to requirements. Norms are associated with purpose in an organisation. Using this method, norms can be specified to identify the controlling relationships with certain system actions. Take the loan recall process as an example. Norms can be expressed such as:

whenever <an item is on loan>
if <a user who does not loan this item needs this item>
then <this user>
is <permitted to>
to <place a request to the item>

Modelling dynamic behaviour of the system has been widely used in reverse engineering and requirements capturing. However, as argued by Systa (1999), the static model is also very useful and complements the dynamic model. Although Systa refers the static model mainly to class/object structure, the same principle applies to requirements recovery, where the static model refers to the semantics and the dynamic model refers to the scenarios. As well as describing the systems from both views, combining them into a single view will check consistency and enrich each other. The outcome, the enriched semantic scenarios, consists of the process models, sharing clear connections to the norms that govern the change of states and the control of the actions. Details of information items, data fields and formats and screen layouts may be relevant in some cases.

4.3 Stage Three: Requirements Derivation

Having created descriptions of the system, the task is now to derive the requirements. This may require a number of iterations through the activities of deriving requirements on the basis of the technical evidence about the system, synthesis of the derived requirements with requirements postulated by the users, and validation of the requirements against the running system in a form of user acceptance testing.

The first two stages of the AMBOLS approach create a technical representation of the running system in a variety of complementary representations. This final requirements derivation stage creates requirements which explain the various properties of the running system.

The explanation of system capabilities by user requirements is bi-directional. We conjecture the existence of a particular requirement because of some function. We also consider the functions which might arise from some conjectured requirement. Such requirements derive from knowledge of the domain or domains of the system. These domains may be deduced from the terms (ontology) used by users to describe the system. We find these condensed in the ontology charts created through the semantic analysis related to the organisational semiotics approach. So we reason both from capabilities to requirements and from requirements to capabilities. This requires the use of mixed reasoning such as induction, deduction and abduction.

4.3.1 Plausible Users Requirements

We cannot claim to derive the actual user requirements for a current system. We cannot (by our assumption that there is no documentation extant for the system) prove that any user requirements we derive were original user requirements. In reality, we can only derive plausible user requirements from existing systems. They will be plausible in the sense that the existing system is a solution to the problem posed by those user requirements. Those user requirements give rise to everything in the system that is visible. We do not give ourselves the luxury of studying the code to determine if there is anything that is not visible. So the plausible user requirements should be a necessary and sufficient set to explain the existence of the operational system as we observe it.

We can undertake verification of the plausible user requirements through the task of user acceptance. Here the operational system is verified against (a set of tests derived from) the user requirements to demonstrate compliance with user needs. In our context, a successful user acceptance test would verify that our derived user requirements were plausible.

We need to be precise about user requirements. Systems have an operational (business) purpose expressed through the objectives or needs of the acquiring organisation. Dardenne *et al*. (1991) describe the needs as goals. These goals are generally too imprecise to form the basis of a development. They must be expressed in a more detailed form which includes the means of assessing the degree to which the goal is met. This detailed form we call the user requirements, each requirement expressing a particular aspect of the need and its verification criteria. Our interest is not confined to user requirements; we also wish to derive goals and purposes.

"There is a well known axiom in anthropology that what people say and do are not the same" (Blomberg *et al*. 1990). This and our preceding analysis indicate the need to utilise both elicitation and observation, combined with verification through the usual user acceptance techniques. Our process must then take into account the need to elicit ideas of purpose and goals from those with intimate knowledge of the domain under study. It is not sufficient to ask the stakeholders of the system what it does (in their opinion) but also what it is for (in their opinion). The capabilities can be elicited from users and operators and derived from observation. These give an assessment of capability from which plausible purposes, goals and requirements can be deduced. Validation of user requirements against capability by various means, including techniques of user acceptance testing, will indicate the degree of plausibility. We may have to return to elicitation and observation to identify misinterpretation, oversight or other causes of the mismatch.

4.3.2 Requirements Management and Validation

Management of requirements is important given the need for iteration, traceability and visibility. The project has chosen to use the DOORS requirements management tool (QSS 1999). The DOORS meta-model is flexible enabling tailoring for the specific AMBOLS data structures to be represented (Qureshi & Liu 1999). The data model represents the requirements and the constraints (non-functional requirements) pertaining to the trace of requirements over the past period of time, together with the higher level purposes and needs from which they derive.

Validation is intended to uncover any discrepancy between the recovered requirements and the running system. Traceability links between the user requirements and the technical models of the system provide an initial check. Every element of the technical models must be explicable in terms of user requirements, and a requirement is only valid if it explains an element of the technical models. We may also consider user acceptance testing, where the system is tested for conformance against validations derived from the requirements. We may need to reiterate earlier analysis stages if mismatches are uncovered. With the use of DOORS as the assisting tool, the process of the management and validation of requirements can be specified and implemented.

5 Case Study: The Library

One of the case studies we conducted is a multi-site University Library system. It is a relatively new computer system which was bought in 1995, but there have been ongoing alterations of the system performed by a system team of the University library. The library system distributes across several geographic locations to serve the customers on the University's campuses and in those federated institutions. Different requirements in the federated institutions have to be allowed. In addition, local customisation of some system functions has to be accommodated. All these make the University Library system not as simple as it seems.

5.1 Behaviour Capture

As shown in figure 1, this stage of the AMBOLS approach focuses on the observation and analysis of system behaviour, starting from the context and interfaces and moving into system processes.

Context analysis. There are multiple stakeholders who take interest in and benefit from the library system, as identified in the table 1.

Table 1: Library system's stakeholders and responsibilities.

Stakeholder	Category	Responsibility
The University	Principal	Ensure the library provides sufficient facilities for education and research
Librarian, Deputy librarian	Actor	Responsible for library system, media services, acquisition, cataloguing, video unit, counter services
System manager	Actor	Ensure system works properly; respond to problems. Communicate with the system supplier
Cataloguing librarian	Actor	Maintain catalogue enquiry facility OPAC
Senior technician	Actor	Manage the video/media unit of the library
Acquisition manager	Actor	Order new books for library
Library administrator	Actor	Manage finance and resources
Principal technician	Actor	Manage media services
Senior tutor librarian	Actor	Provide consultation, teaching, and liaison with schools, assess books for purchasing
Library assistant, information service assistant	Actor	Issue and renew library items, and deal with queries and inter-library loan requests
Library user (University staff, student, library member from other organisations)	Client	Use facilities offered by the library including borrowing library materials, inter-library loan, photocopy, fax, reservation and renewal of library materials.

Members of the public	Client	Access newspapers, journals, books – no right of borrowing
Publisher	Provider	Supply library materials such as books and journals
System suppliers	Provider	Supply software for upgrading and carry out major maintenance.
Other libraries in the UK	Collabo-rator	Lend and borrow materials
Local and central government	Legisla-tor	Make policies and regulations

The method adopted here is Problem Articulation Methodology (Kolkman 1995), which categorises stakeholders based on the strength of their influence on (or 'distance' from) the core system. The stakeholders are identified in each category according to this distance:

- The principals and actors who have direct involvement in the use of, and the power to make decisions about the system.
- The clients and providers who are either directly affected by or have direct impact on the system.
- The collaborators and competitors.
- Finally, most distant but still relevant, bystanders and legislators.

Stakeholders have roles to play, which are described in our analysis as responsibilities. Understanding their roles and responsibilities helps capture their different views and requirements of the retail store system. The analysis of stakeholders also helps us identify the major categories of business activities supported, and relate the system functions to the stakeholders. Use cases (Booch *et al.* 1999; Jacobson 1992) are used to identify the major scenarios, such as use, administrating, lending, return, cataloguing, and so on.

The context analysis gives us an understanding of the different views of the system and its major functional areas.

Human-machine interactions. The analysis at this stage is carried out by observing how the system is used in daily operations. Screen shots are captured and organised in the same order as they are displayed on the running system. After initial investigation, the main categories of system functions are documented. In this case study the functions such as borrowing (or check out), return (or check in), user administration and cataloguing are most readily visible.

So far, we have conducted our investigation by interviews and observations of system screens and its behaviour in operation. The analysis in the next step uncovers the information required for these operations, without being concerned too much with the syntactic structure of the information.

Semantic analysis. The focus of a semantic analysis is to identify concepts and terms involved in system operations. This analysis aims at revealing the information items that constitute the information model for the system. The semantic analysis involves the analyst and other stakeholders of the system in generating and verification of the model.

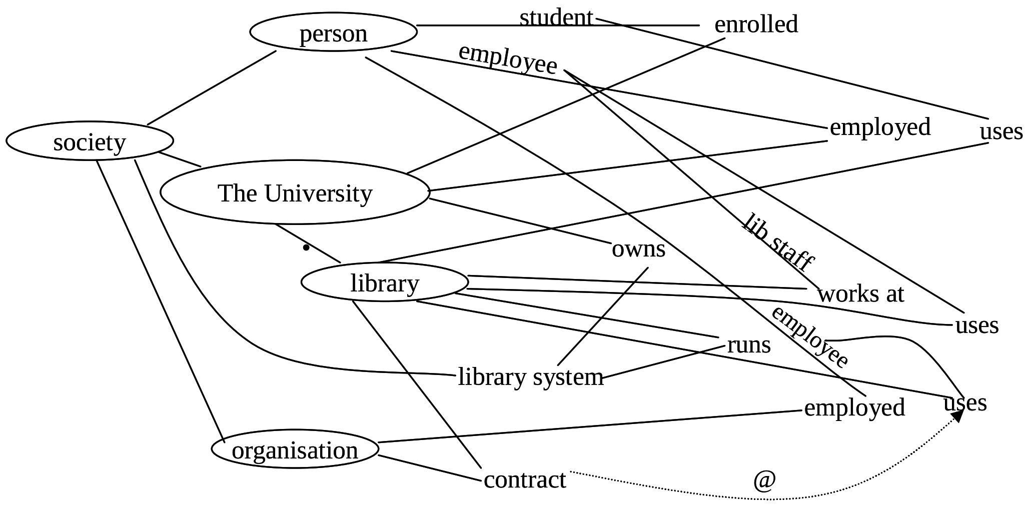
The semantic model (ontology chart) (as shown in figure 4) defines the behaviour of the system by describing the roles and responsibilities of human actors involved. The relative left-right position of entities linked by a line is significant. An entity on the left of a line is the antecedent of the entity on the right. For example, the society is the antecedent for a person, the University and an organisation. All these three entities can only exist and be meaningful within the context of the society. The entities in ellipses are agents who can take actions and can be responsible for their actions: society, person and organisation. The student and employee are role names (placed along a line), determined by the activities they are involved (i.e. enrolled at the University and employed by the University). The entities within a box are specifics of a certain type: e.g. members and the public are specifics of the user. The dotted line with an "@" sign denotes an authorisation: e.g. a contract governs the employees' use of the library. The information captured in this ontology chart defines the semantics of the concepts and terms used in the problem domain. It also serves as a foundation for describing the behaviour of the information system.

The three University libraries use one computer library system called Horizon, which handles all on-line services. This system is connected with libraries at affiliated institutes, though services provided to them are limited. The library system communicates with affiliated colleges and national libraries through the Internet. The library system is also connected to other university libraries and national libraries governed by agreements, such as for inter-library loans.

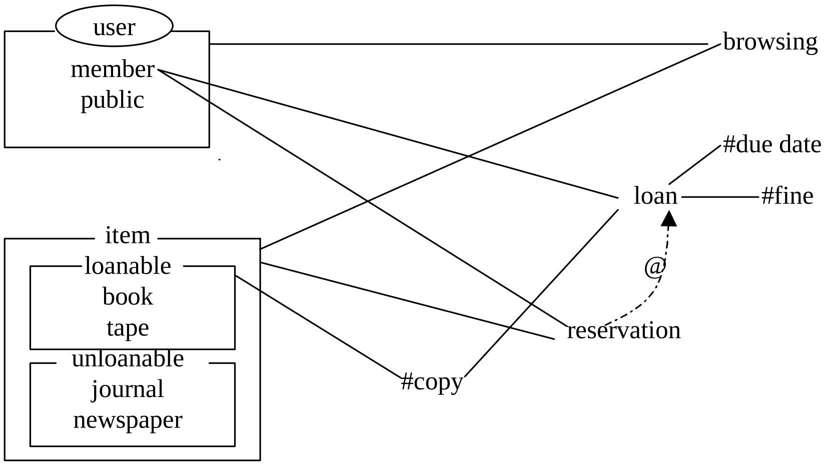
The visible system components are electronic tills, bar code readers, displays on the screen and input from the keyboard. Transactions at different tills can clearly interact with each other, so they are connected in some way. It is not plausible to suppose that every till is connected to every other. It is reasonable to suppose some central system (within one library or elsewhere) which mediates the communication. We see that we can reason about the observation to deduce plausible actors, functions and components. Blocking a library assistant from concurrent use of two tills is another indication of central control.

The information on the hardware and software platform is relevant for us to understand the non-functional and other requirements from the business such as the service level, access level, system performance and security issues. It aids us in reasoning about requirements derivation and validation from another perspective in the requirements recovery.

Semantic process models. A business process, which involves actors and a computer system, is described with a role-activity diagram. Figure 5a shows the process of checking out an item such as a book. This process model and the semantic model complement each other, as the former concentrates on the processes while the latter defines the roles, responsibilities, and meaning of the terms used in the process model.



(a) Actors involved in library operations.



(b) Lending and return of a library item.

Figure 4: Semantic model of the library system.

5.2 Dynamic Behaviour Modelling

The investigation at this stage focuses on the behaviour of system and the interaction between human users and the system.

Scenarios. Scenarios have been widely used for capturing requirements and for software engineering. There are many varieties of methods and techniques, each has advantages and suits specific purposes (Rolland *et al.* 1998). We use scenarios to organise information on actors and activities in a time-based framework. As shown in figure 5b it describes the dynamic aspect of the checking-out process of a loan item. This provides an overview of actions, with supplementary details from other models:

context, interaction and semantic models. Figure 5a shows activitives conducted by the two tipical roles (i.e. the library assistant and the borrower). A dark box indicates an activity by the single role, while a grey box an activity that has a connection with another role. The while box represents a dummy activity (or a passive activity). Figure 5c uses the State Diagram technique to describe the sequence of the system interfaces and actions (indicated along the lines). As can be seen in the figure, an action is possible when an interface is displayed and it will lead to a new interface. These figures document the observation of the system behaviour. As discussed earlier, a context model identifies stakeholders, an interaction model describes human-machine interfaces, and a semantic model defines the concepts and terminology used in modelling.

Norm analysis. In system operation, there are conditions to be satisfied for invocation of a process and for events to take place. These conditions are described in norms. The norms determine whether and when a certain activity takes place; they also determine the behaviour of the system in performing certain tasks. The business rules must be collected and specified. The norm analysis assists this task.

The norm analysis uses the results of the semantic analysis. Each concept in a semantic model represents an object (which may be an entity, an action or an event). Each has a lifespan marked by a start and a finish controlled by a set of norms. For example, the acceptability of a library card is determined by satisfying certain conditions including an expiry date. A variety of norms can be identified, of which two types are most relevant in the library application:

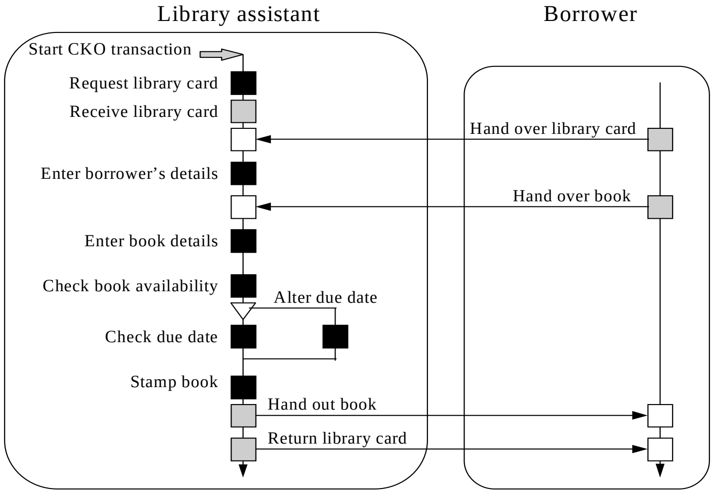
- Definitional norms define meanings, boundaries, responsibilities, authorities and status.
- Action norms indicate when and which actions are taken.

These are examples of norms derived from the semantic model and the library regulations:

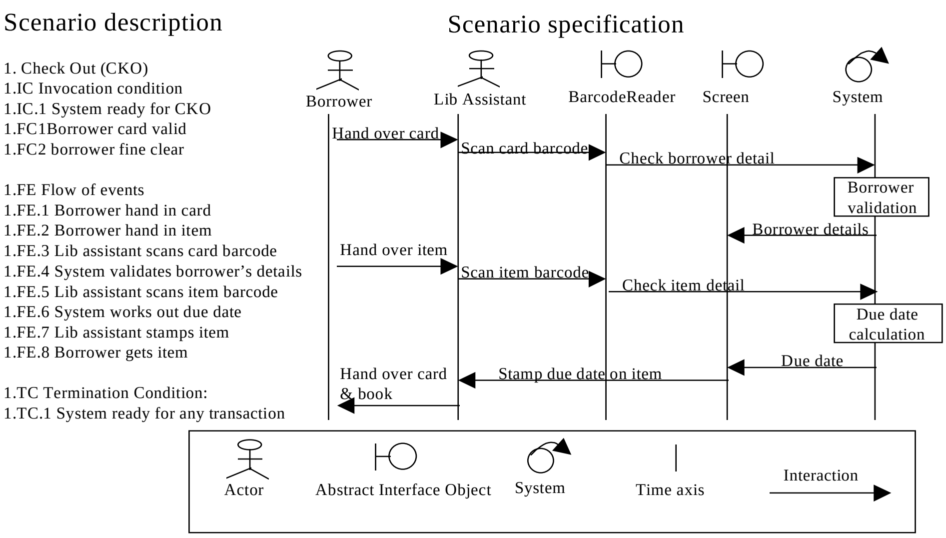
- Borrowed items must be returned by the date due.
- Borrowers must not exceed their allowance of items on loan at any one time.
- No items will be issued to a borrower who has items overdue or an unpaid fine.

As mentioned in Section 3.2, a norm specification covers the following elements: condition, state, actor, modality and action, as shown below:

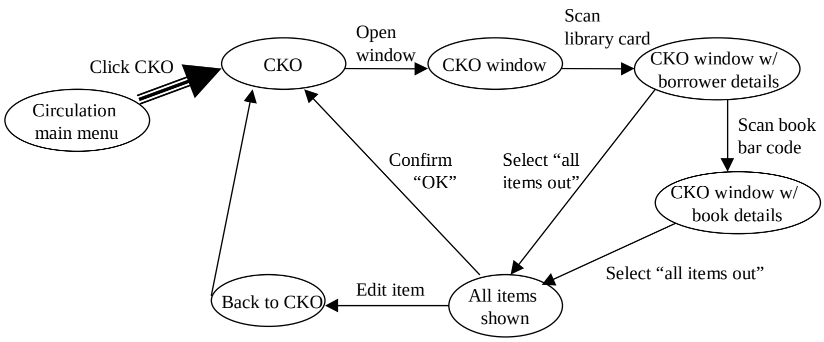
(condition)	When an item is issued to a borrower
(state)	If there is an overdue item OR unpaid fine
(actor)	The library assistant
(modality)	Is obliged
(action)	To stop the issuing process.



(a) Role-activity diagram.



(b) Scenario



(c) State diagram

Figure5: Checking out a book.

This norm requires a series of procedures (such as library security) and facilities (such as tills, bar-codes, library cards) to implement it. There can also be a norm to define how to calculate payment, which can be applied in conjunction with this norm.

Enriched semantic scenarios – integrating norms. Each business process is described by a scenario, which captures actors, actions and behaviour of the system. Semantic analysis defines the meanings of the concepts and terms in the scenario; and norm analysis specifies rules and conditions which control the actions and events. An abstract interface object in a scenario may suggest the use of one or many data objects in the system. Norms and system interactions with the users represent the system behaviour, which is an aggregation of the behaviour of all subsystems and program components. The integration of these analyses represents the system behaviour from different perspectives, each complementing and validating the entire understanding of the system. The final view of system behaviour and major data objects can be obtained through several iterations in which hypotheses are set, discarded or altered, and validated.

5.3 Requirements Derivation

Deductions from domains involved. When we know the domain is the library, the domain knowledge we have helps in deriving the missing parts of the requirements models, as well as in validating the results. If, however, an investigation is in an unfamiliar domain, the study of the domain and operating context becomes crucial before any requirements recovery is attempted.

In our case, we can deduce the meaning of "borrow", "loan item", "due date" and "fine”. Our understanding of the connotation of these concepts enables us to derive the associated actions and processes taking place though human users and within the computer systems. ‘Log on’ and ‘denied access’ imply security; "checking out" means a certain standard process; and so on. We can now look at each domain or context from knowledge of what it implies from a broader view. Library service entails the law of intellectual properties; use of library cards entails the law of individual's rights, and so on. Admittance of the public and employment of staff imply health and safety requirements. We can see the effects of these requirements in the observation.

Deductions from system components. We note the use of library card readers and bar code wands. These give accuracy and speed of operation. Accuracy of recording is important to the library and the customer. Speed of operation enables queue sizes to be minimised during peak hours, an important system attribute when customers can be discouraged by queuing time. Transactions times must be low. Bar code readers increase transaction speed. Electronic tills, printers and library card readers are all technologies to increase transaction speed.

We can infer that the tills are linked to the library system. We know that the tills are linked to the library membership database and catalogue; and there are means of interacting with these databases and sub-systems at different sites.

Deduction from artefacts. The existence of a library card and bar code of a loan item shows what sorts of information is needed by the operation and stored in the system. By examining the library card and various forms (e.g. the membership application form and book purchase form), we know more about the items of information and even possible formats of data kept in the system. The library has security gates which sense any item taken through them and set off an alarm. These indicate a requirement to prevent theft.

Deductions from norms. Staff must log-on to a till to make it operable. One assistant may be logged-on to at most one till at a time. Each till can deal with only one transaction at a time. It cannot start the next until the previous is finished either by normal completion of the transaction or by voiding. These norms relate to avoiding fraud and theft, on the part of staff and customers. Transactions may not be completed manually. This indicates automated ledger activities. Only correctly labelled items can be lent. Using correct codes indicates computer assisted catalogue control.

Figure 6 gives a summary of some derived plausible goals and requirements that covers both business and system sides.

Goal: Operate properly to provide library services.

Sub-goal 1: Operate correctly in loan operations.

- **Sub-goal 1.1:** Check members' legitimacy (identity, membership grade, fine statues, etc.).
- **Sub-goal 1.2:** Check statues of items (loanability, not reserved by others, etc.)
- **Sub-goal 1.3:** Lend items.
- **Sub-goal 1.4:** Process transaction correctly and quickly.

Sub-goal 2: Operate correctly in return operations.

- **Sub-goal 2.1:** Record returned items.
- **Sub-goal 2.2:** Check whether overdue and calculate fine, if appropriate
- **Sub-goal 2.3:** Update loan record of the members

Sub-goal 3: Maintain membership information corrently.

Sub-goal 4: Maintain the catalogue of all items.

- **Sub-goal 4.1:** Keep catalogue up to date.

Sub-goal 5: Avoid loss due to theft, fraud, and so on.

Sub-goal 6: Maintain library information integrity.

Figure 6: Some plausible goals and requirements.

5.4 Validation of Requirements

Validation of the requirements is important. This is achieved in several ways. First of all, the semiotic approach adopted in this project offers us a way to employ abduction throughout the exercise of requirements recovery. Together with induction and deduction, abduction allows the research team to formulate a hypothesis and verify it with further observation and investigation. As explained by Liu *et al.* (1999), abduction departs from some surprising facts in observation and formulates a hypothesis; the hypothesis is further tested for adjustment or refinement. We are clearly aware that the requirements recovery is never a one-run activity but requires a series of iteration in which the hypothesis is constantly tested and adjusted. For example, initially we had failed to observe system artefacts related to health and safety. However, when our analysis indicated the presence of the public, our knowledge of the existence of the related health and safety domain prompted us to return to the library. As expected we found fire exits, fire hydrants, signs, safety notices and other artefacts related to that domain.

The second means for validation is that we conduct the analysis from multiple angles. This results in a redundancy of information, which is useful in validating the analysis. For example, although the process of checking out a book is modelled in figure 5a, using a role-activity diagram; the same process is studied by observing operations at the till in the library, represented as a state diagram (figure 5c). The state diagram is constructed independently without using the information on the role-activity model, which allows cross checking between the two models.

5.5 Management of Requirements

The requirements recovery involves large amount of information collected from various sources and produced at different stages. We record the interview transcripts and we record our observations as observational notes. These two documents are our main sources of information. The information produced involves Use Case diagrams, role activity diagrams, scenario analysis, norm analysis, HCI analysis and semantic analysis. The documents generated in the AMBOLS project are in different formats such as graphics and text. The results of analyses from the different stages must be managed and cross-referencing is made between different analysis.

The requirements management tool DOORS (QSS 1999; Qureshi & Liu 1999) provides the support required. It provides a high level of customisability, version control features, import and export links, and flexible reporting features. In DOORS, files and requirements models are defined "objects", which are grouped into modules within a project. It defines three kinds of modules each having different purposes: 1) a descriptive module represents an unstructured, read-only input documents: 2) a formal module represents a structured documents with sections and sub-sections; and 3) a link module represents any number of relationships of a specific type. Like most requirements tools, all objects can be created, copied, deleted, edited, and moved as required. In addition, DOORS maintains module and requirements histories for auditing purposes; therefore, users can create baseline versions of entire modules for tracking module revisions. This has enormously enhanced the usability of the AMBOLS approach.

6 Conclusions

To recover requirements is an important activity in the modernisation and replacement of a legacy system. Successful engineering and reengineering requires good user requirements. To make decisions about a system's future we must understand its intended purpose.

The impact of the AMBOLS approach continues in research and industry as legacy systems re-engineering and integration is still a high resource-demanding activity. Documentation of the system is often unavailable. Even where design documents and source code are available they are usually out-of-date or otherwise inaccurate. Well documented user requirements, kept in line with changes, are rare even in current practice. In many circumstances, technical expertise about system internals has long disappeared. The system itself and those who use it are the only reliable sources of information about what it does and why.

Organisational semiotics theory and methods provides the theoretical foundation for the proposed top-down black-box approach, which organises the observations of the operational systems into structures of agents and affordances interlinked by ontological dependencies and governed by norms. User requirements are deduced from the enriched semantic scenarios.

Currently we focus on what may be deduced from the functionality offered by the system. Generally the prime motivations for a system are to provide automated capability to assist processes of the business. Our emphasis on descriptive scenarios arises from this observation. The next most telling category of information contains the limitations or restrictions on the system functions. We capture these as norms. The norms are indicative of business policy which the system encapsulates, so again they will suggest user requirements. Other constraints on the system, such as performance, the use of particular software and hardware, can be captured and expressed as user requirements with varying degrees of difficulty and of value. We need to distinguish between constraints imposed by the technical capabilities at the time of development and those truly expressive of business need.

An aphorism frequently quoted in reverse engineering is ‘You can make sausages from pigs, but not pigs from sausages’. In our scenario ‘only the sausage skins can be inspected’. However, there are experts who can tell a great deal about the origins of a sausage from its size, colour, smell and taste. This gives inspiration to our venture where we have no desire to reconstruct pigs, but dearly wish to know the requirements for sausages.

Further researches will be undertaken to combine the proposed approach with other legacy systems modernisation methods, especially the bottom-up white-box methods, to form an integrated tool package for industrial users. We will also consider the integration of information about the system available from its users, both in terms of its capabilities and other technical attributes, and their understanding of the user requirements for the system.

Acknowledgements

This work is partly funded by UK EPSRC (GR/L43473 and GR/S04840/01). The author would like to thank the AMBOLS team members: Alan Dix, Hanifa Shah, Bernadette Sharp and Zubair Qureshi; especially Albert Alderson and Zhiwu Xie for detailed discussions and follow-on research on the application of the AMBOLS methods in industrial cases. Part of the work was first published in (Liu *et al.* 1999).

References

Ashish, N. and Knoblock, C. (1997) Wrapper Generation for Semi-structured Internet Sources, *ACM SIGMOD*, 26, 4, 8-15.

Blomberg, J., Giacomi, J., Mosher, A. and Swenton-Wall, P. (1990) Ethnographic Field Methods and the Relation to Design, *Proc. 1st Participatory Design Conf.*, Seattle

Booch, G., Rumbaugh, J. and Jacobson, I. (1999) *The Unified Modelling Language User Guide*, Addison-Wesley, MA, 233 -241.

Bukhres, O.A. and Elmagarmid, A.K. (1996*) Object-Oriented Multidatabase Systems: a solution for advance applications*. Prentice Hall, London.

Comella-Dorda, S., Wallnau, K., Seacord, R. and Robert, J. (2000) *A Survey of Legacy System Modernization Approaches*, Technical Report CMU/SEI-2000-TN-003, Software Eng. Institute, Carnegie Mellon University.

Dardenne, A., Fickas, S. and van Lamsweerde, A. (1991) Goal Directed Concept Acquisition in Requirements Elicitation, *Proc. 6th Int'l Workshop Software Specification and Design*, Como, Italy, 14-21.

Diaper, D. (1989) *Task Analysis for Human Computer Interaction*, Ellis Horwood, Chichester.

El-Ramly, M., Stroulia, E. and Sorenson, P. (2002) Recovering Software Requirements from System-user Interaction Traces, *Proc. 14th Int'l Conf. Software Eng. and Knowledge Eng.*, Ischia, Italy, 447-454.

El-Ramly, M., Iglinski, P., Stroulia, E., Sorenson, P. and Matichuk, B. (2001) Modeling the System-User Dialog Using Interaction Traces, *Proc. 8th Working Conf. Reverse Eng.*, Stuttgart.

Filippidou, D. (1998) Designing with Scenarios: A Critical Review of Current Research and Practice, *Requirements Eng. Journal*, 3, 1, 1-22.

Gibson, J. J. (1979) *The Ecological Approach to Visual Perception*, Houghton Mifflin, Boston.

Graham, I. (1994) *Object Oriented Methods*, 2nd edition, Addison-Wesley, Wokingham.

Henderson, P. (2000) *Systems Engineering for Business Process*, Springer, London.

Hruschka, P. (1997) Detailing and Deriving System Requirements, *Proc. Int'l Conf. and Workshop Eng. Computer Based Systems (ECBS 97)*, Monterey, CA, 25-32.

Jacobson, I., Christerson, M., Jonsson, P. and Overgaard, G. (1992) *Object-Oriented Software Engineering: A Use Case Driven Approach*, Addison Wesley, Reading, MA.

Kim, W. (1995) *Modern Database Systems*, ACM Press, New York.

Kolkman, M. (1995). Managing Ambiguity and Change with the Problem Articulation Methodology, in Pettigrew, A. and Hines, D. (eds), *Globalisation and Procedural Analysis of the Organisation*, Sage, London.

Liu, K. (2000) *Semiotics in Information Systems Engineering*, Cambridge University Press, Cambridge, UK

Liu, K. (2004) (ed) Virtual, Distributed and Flexible Organisations - Studies in Organisational Semiotics, Kluwer Academic Publishers, Dordrecht

Liu, K and Ong, T. (1999) A Modelling Approach for Handling Business Rules and Exceptions, The Computer Journal, 42(3), 221-231.

Liu, K., Alderson, A. and Qureshi, Z. (1999a) Requirements Recovery of Legacy Systems by Analysing and Modelling Behaviour, *Proc. Int'l Conf. Software Maintenance*, Oxford, England, 3-12.

Liu, K., Alderson, A., Shah, H., Sharp B. and Dix, A. (1999b) Applying Semiotic Methods to Requirements Recovery, in Jayaratna, N. (ed.), *Methodologies for Developing and Managing Emerging Technology-Based Information Systems*, Springer, London, 142-152.

Liu, K., Clarke, R., Anderson, P., Stamper, R. (editors) with Abou-Zeid, E. (2002). Organizational Semiotics: evolving a science of information systems, Proceedings of IFIP WG8.1 Working Conference, Kluwer Academic Publishers, Boston.

Liu, K., Sun, L., Dix A. and Narasipuram, M. (2001) Norm Based Agency for Designing Collaborative Information Systems, Information Systems Journal, 11, 229-247.

Liu, K., Sun, L. and Bennett, K. (2002) Co-Design of Business and IT Systems, J. of Information Systems Frontiers, 4, 3, 251-256.

Ould, M.A., (1995) *Business Processes: Modelling and Analysis for Re-engineering and Improvement*, Wiley, New York.

Peirce, C. S. (1960) Collected Papers of C. S. Peirce 1931-58, In Hartshorne, C. and Weiss, P. (eds.), Harvard University Press Cambridge, Mass.

QSS (1999) http://www.qss.co.uk (accessed February 1999).

Qureshi, Z. and Liu. K. (1999) Requirements Management and Traceability in AMBOLS - A case study using DOORS, in Proceedings of BIT'99 conference on Business Information Technology Management, Manchester Metropolitan University, Manchester.

Reddy, M.P., B.E. Prasad, P.G. Reddy & A. Gupta (1994) A Methodology for Integration of Heterogeneous Databases. IEEE Trasc. on Knowledge and Data Engineering, 6(6) pp920-933

Rolland, C., Achour, C.B., Cauvet, C., Ralyte, J., Sutcliffe, A., Maiden, N., Jarke, M., Haumer, P., Pohl, K., Dubois E. and Heymans, P. (1998) A Proposal for a Scenario Classification Framework, *Requirements Eng. Journal*, 3,1, 23-47,

Tilley, S. (1998) *A Reverse-Engineering Environment Framework*, Technical Report CMU/SEI-98-TR-005, Software Engineering Institute, Carnegie Mellon University.

Seacord, R. (2001) *Legacy System Modernization Strategies*, Technical Report CMU/SEI-2001-TR-025, Software Engineering Institute, Carnegie Mellon University.

Stamper, R. (1997) Organisational Semiotics, in *Information Systems: An Emerging Discipline?*, Mingers, J. and Stowell, F. (ed.), McGraw-Hill, London, 267-283.

Stamper, R. (2001) Organisational Semiotics - Information without the Computer?, in Liu, K.,. Clarke, R.J., Andersen, P.B. and Stamper, R.K. (ed.), *Information, Organisation and Technology, Studies in Organisational Semiotics*, Kluwer, Boston.

Stamper, R. and Liu, K. (1994) Organisational Dynamics, Social Norms and Information Systems, Proc. 27th Hawaii Int'l Conf. System Sciences, vol IV, 645-654.

Stamper, R., Liu, K., Hafkamp, M. and Ades, Y. (2000) Understanding the Role of Signs and Norms in Organisations, - A Semiotic Approach to Information Systems Design, *J. Behaviour and Information Technology*, 19, 1, 15-27,

Systä, T. (1999), On the Relationships between Static and Dynamic Models in Reverse Engineering Java Software, *Proc. 6th Working Conf. Reverse Eng.*, Atlanta, 304-313.

Tilley, S. (1998) *A Reverse-Engineering Environment Framework*, Technical Report CMU/SEI-98-TR-005, Software Engineering Institute, Carnegie Mellon University.

Weiderman, N., Bergey, J., Smith, D., Dennis, B. and Tilley, S. (1997a) *Approaches to Legacy System Evolution*, Technical Report CMU/SEI-97-TR-014, Software Engineering Institute, Carnegie Mellon University.

Weiderman, N., Northrop, L., Smith, D., Tilley, S. and Wallnau, K. (1997b) *Implications of Distributed Object Technology for Reengineering*, Tehnical Report CMU/SEI-97-TR-005, Software Engineering Institute, Carnegie Mellon University.

Vidal, M.E., Raschid, L. and Gruster, J.R. (1998) A Meta-Wrapper for Scaling up to Multiple Autonomous Distributed Information Sources, *Proc. 3rd IFCIS Int'l Conf. on Cooperative Information Systems*, New York, New York, 188-197.

Xie, Z. and Liu, K. (2002) Improving Business Modelling with Organisational Semiotics, *Proc. 5th Int'l Organisational Semitics Workshop*, Delft, The Netherlands, 119-128.

Xie, Z., Liu, K. and Emmitt, D. (2003) Improving business modelling with organisational semiotics, in Gazendam, H., Jorna, R. and Cijsouw, R. (eds), Dynamics and Changes in Organisations – Studies in organisational semiotics, Kluwer Academic Publishers, Dordrecht, 89-102.

Yan, L.L., Özsu, M.T. and Liu, L. (1997) Accessing Heterogeneous Data through Homogenization and Integration Mediators, *Proc. 2nd IFCIS Int'l Conf. on Cooperative Information Systems*, Kiawah Island, North Carolina, 130-139.

Yu, E. (1997) Towards Modelling and Reasoning Support for Early-Phase Requirements Engineering, *Proc. 3rd IEEE Int. Symp. Requirements Eng.*, Washington D.C., 226-235.

About the Author

Kecheng Liu, Fellow of British Computer Society, is a full professor and holds a chair of informatics and e-business at the University of Reading, UK. He is the Director of the Informatics Research Centre, and a recipient of several awards from the UK and international institutes. He is a visiting professor at Beijing Institute of Technology, Fudan University, and the Graduate School of the Chinese Academy of Sciences. Professor Liu's research work can be found in requirements studies, information systems analysis and design, object methods for systems engineering, normative modelling of software agents, e-commerce systems, agent-based systems for collaborative work, HCI and CSCW.