1-1-2008

Active exploration of emerging themes in a study of object-oriented requirements engineering: the 'Evolutionary Case' approach

Linda L. Dawson
University of Wollongong, lindad@uow.edu.au

Active exploration of emerging themes in a study of object-oriented requirements engineering: the 'Evolutionary Case' approach

Abstract

The evolutionary case approach provides a framework for qualitative case study research in information systems (IS). It uses revelation, reinforcement, reflection and re-examination to explicitly explore emerging themes in interpretive case study research.

The method is based on the progressive development of a theoretical model grounded initially in the literature and then refined using sequential case studies grounded in practice. The method addresses the gap which often separates data from conclusions in qualitative case study research by documenting the “revealed” and “reinforced” changes in the theoretical model as it evolves from the empirical data.

The paper provides an illustrative study of the use of models in object-oriented requirements engineering to demonstrate the use of the evolutionary case approach.

Keywords

Active, exploration, emerging, themes, study, object, oriented, requirements, engineering, Evolutionary, Case, approach

Disciplines

Physical Sciences and Mathematics

Publication Details

Dawson, L. (2008). Active exploration of emerging themes in a study of object-oriented requirements engineering: the “Evolutionary Case” approach. Electronic Journal of Business Research Methods, 6 (1), 29-42.

Active Exploration of Emerging Themes in a Study of Object-Oriented Requirements Engineering: The "Evolutionary Case" Approach

Linda Dawson
Monash University, Caulfield East, Australia
Linda.Dawson@infotech.monash.edu.au

Abstract: The evolutionary case approach provides a framework for qualitative case study research in information systems (IS). It uses revelation, reinforcement, reflection and re-examination to explicitly explore emerging themes in interpretive case study research.

The method is based on the progressive development of a theoretical model grounded initially in the literature and then refined using sequential case studies grounded in practice. The method addresses the gap which often separates data from conclusions in qualitative case study research by documenting the "revealed" and "reinforced" changes in the theoretical model as it evolves from the empirical data.

The paper provides an illustrative study of the use of models in object-oriented requirements engineering to demonstrate the use of the evolutionary case approach.

Keywords: Case study, action research, qualitative, object-oriented, requirements engineering

1. Introduction

The evolutionary case approach provides a framework for qualitative case study research in information systems (IS). This framework was specifically developed to assist exploratory theory-building research using case studies of IS in practice. It uses revelation, reinforcement, reflection and re-examination to explicitly explore emerging themes in interpretive case study research.

The method is based on the development of a theoretical model grounded initially in the literature and then refined using sequential case studies grounded in practice. The method addresses the gap which often separates data from conclusions in qualitative case study research by documenting the "revealed" and "reinforced" changes in a theoretical model as it evolves from the empirical data.

This approach is different to traditional case study approaches which are usually based on replication of situations or units of analysis with the aim of studying each case in isolation and then using within or cross case analysis to discover common themes.

The paper uses an illustrative study of the use of models in object-oriented requirements engineering practice to demonstrate the use of the evolutionary case approach.

2. The evolutionary case approach

The evolutionary case approach is built on the same foundations (Carroll et al., 1998) as the "structured-case" approach (Carroll and Swatman, 2000). It is similar to structured-case in that it is an iterative, theory-building approach based on refining a conceptual framework or theoretical model. It differs from structured-case in that it is designed to explicitly explore emerging concepts in an evolving theoretical model based on reinforcement, revelation, reflection and re-examination, in order to understand the relationship between theory and practice. It also differs from structured-case in that there is no *"… literature-based scrutiny of the research findings* (p236)" based on Eisenhardt's (1989) "enfolding of the literature" since this can be considered to be theory-testing rather than theory-building.

Case study research aims to examine a phenomenon in its natural context (Yin, 1994, Cavaye, 1996) and aims to contribute to knowledge by relating findings to generalisable theory (Cavaye, 1996). The case study research method is one of the most popular methods in information systems research and is well-suited to understanding the interactions between information technology-related innovations and organisational contexts (Darke et al., 1998). Qualitative interpretive case study research involves more than an examination of qualitative data. It involves looking for a deep understanding of why behaviours occur and why people do what they do. It also attempts to understand the opinions, beliefs, experiences, expectations, etc which influence the behaviour of participants through interviews (Myers and Newman, 2007), participation and/or

Reference this paper as:
Dawson, L. "Active Exploration of Emerging Themes in a Study of Object-Oriented Requirements Engineering: the "Evolutionary Case" Approach." *The Electronic Journal of Business Research Methods* Volume 6 Issue 1 2008, pp 29 - 42, available online at www.ejbrm.com

observation (Walsham, 1995). Researchers need to "let the data tell the story" but when analysing qualitative case study data it is often difficult to demonstrate the links between the data and conclusions.

The action research method involves collaboration between the researcher(s) and the participant(s) and intervention by the researcher(s) in the situation being studied (Susman, 1983, Myers, 1997, Baskerville and Wood-Harper, 1998, Avison et al., 1999). This often involves a cycle of feedback aimed at increasing the understanding of a given social situation (Hult and Lennung, 1980). Galliers (1992) describes action research as an interpretivist subset of the case study approach where the presence of the researcher affects the situation being studied.

The grounded theory approach seeks to develop theory that is grounded in data that has been systematically gathered and analysed (Strauss and Corbin, 1990). A key element of the grounded theory approach is that the theory should be developed with a reflexive 'back and forth' interplay between data collection and analysis (Myers, 1997, Urquhart, 1998) without any preconceptions about the outcome until all the data has been collected and analysed.

The evolutionary case approach incorporates elements of case study research in that it relies on interviews in the field, it is non-interventionary (and non-participatory), and it uses multiple sequential cases. It incorporates elements of action research in that it is iterative (each case has its own cycle), and it relies on learning and reflection on accumulated findings, both within individual case cycles and between cycles of multiple sequential-cases, for refining interview scripts and the theoretical model. And it incorporates elements of grounded theory in that the data is analysed as it is collected and revisited before final conclusions are drawn.

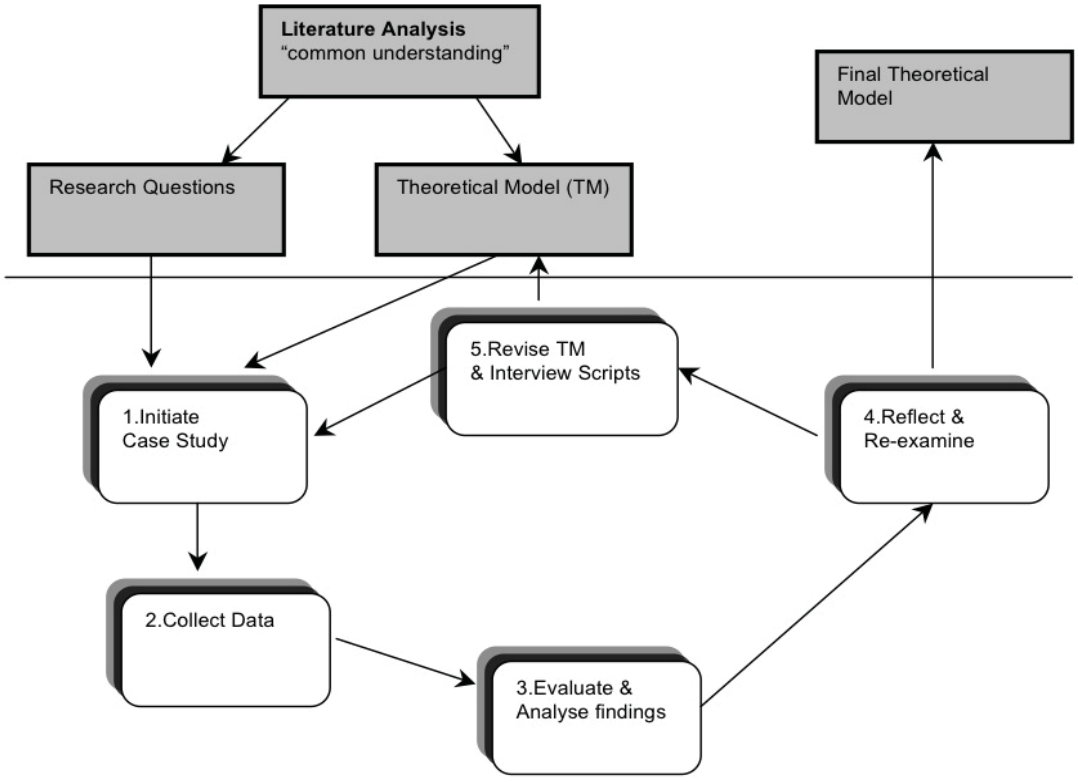

Figure 1: The evolutionary case cycle

Central to the evolutionary case approach is the concept of a *theoretical model* initially grounded in the literature and describing a *common understanding* of "what might be happening" in practice, which is progressively refined based on empirical data grounded in practice. The term "theoretical model" is used here to mean a representation of a theory where the definition of a theory is based on "*A conception or mental scheme of something to be done, or of the method of doing it; a systematic statement of rules or principles to be followed*." (OED, 2004) or "*conjectures, models, frameworks, or bod[ies] of knowledge*" (Gregor, 2006) and contains features of a theoretical model as defined by Dubin (1976) and Bacharach (1988) i.e. the interactions or relations between defined units or concepts within a set of boundaries or constraints depicting a limited portion of the world.

The evolutionary case approach is particularly suited to interview-based data collection and the subsequent discussion in this paper is based on this. In each case cycle the researcher seeks to refine the current version of the theoretical model by:

- looking for **reinforcement** of concepts already contained within the theoretical model
- **revelation –** identifying new areas for exploration and potential reinforcement
- learning and **reflection** on data collected so far
- **re-examining** previous transcripts to find any further reinforcement of an emerging theme

The researcher is active in the data collection. Leading questions are encouraged in order to facilitate reinforcement and semi-structured, open-ended questions are used to facilitate revelation. Exploration of these revelations is incorporated into revised interview scripts which are used in the next cycle. Reinforced concepts are retained in the evolving theoretical model. The process is ongoing but concluded when there has been enough reinforcement for a representative model of the research domain being investigated to stand alone or when theoretical saturation has been reached (Eisenhardt, 1989). Therefore, the outcome of the research method is a revised theoretical model (with several revisions during the process) which represents an theory which has evolved about the area being investigated.

2.1 The research cycle

As shown in Figure 1, the formulation of an initial theoretical model and research questions are based on the definitions and common understanding established from the relevant literature. The theoretical model together with the research questions is used to set up, plan and initiate the subsequent research cycle. This approach is adapted and extended from Miles and Hubermann (1994) who call this "*focussing the collection of data", p 16-25*.

After the evaluation of findings at any iteration, the current accumulated findings and learning are reflected upon. This reflection activity provides two things: possible further refinements of the theoretical model and possible refinements to the interview scripts in order to explore any emerging categories.

As the number of cases increases and the accumulated data increases, an important part of the reflection process involves the re-examination of previous cases. This re-examination allows the potential reinforcement of emerging categories from the previous data.

The next section provides an illustrative example of the use of the evolutionary case approach.

3. Example: the use of object-oriented models in requirements engineering practice

The move towards the use of object-oriented methods for information system development led to a need for the development of object-oriented approaches to requirements engineering. However, little is understood, or reported on the basis of research, of the use of object-oriented methods by practising professionals in the production of requirements specifications for commercial or industrial sized projects (Burton-Jones and Meso, 2006, Dobing and Parsons, 2008). An understanding of what successful professional developers do, how they do it and why they do what they do needs to be investigated (Burton-Jones and Meso, 2006, Dobing and Parsons, 2006, Grant and Reza, 2007).

3.1 Themes from the literature review

This project brings together concepts from requirements engineering and object-oriented methods. Preliminary findings have been reported in Dawson and Swatman (1999) and Dawson and Darke (2002).

3.1.1 Requirements engineering

Requirements engineering refers to the early stage of the systems development process which manages the identification and documentation of system requirements (Loucopoulos and Karakostas, 1995, Macaulay, 1996, Sommerville and Sawyer, 1997, Hull *et al.*, 2005). Various frameworks and models of the requirements engineering process have been suggested in the literature. Highly cited frameworks include Pohl (Pohl, 1994), Loucopoulos and Karakostas (1995) and Macaulay (1996). For this project a review of the literature resulted in the selection of Loucopoulos and Karakostas (1995) framework from which to develop a theoretical model for object-oriented requirements engineering. In this framework the requirements

engineering process can be broken down into three sub-processes, elicitation, specification and validation, which deal with two external entities, the user and the problem domain. See Figure 2.

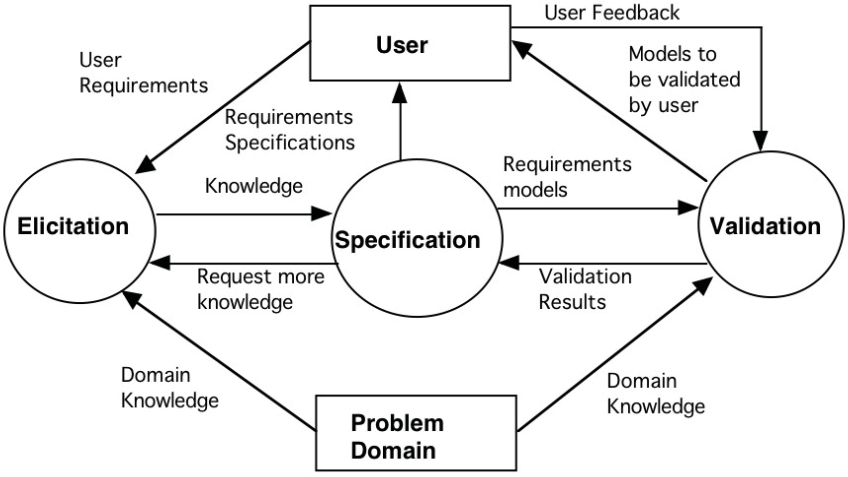

Figure 2: A framework for requirements engineering processes: Loucopoulos and Karakostas (1995) page 21

3.1.2 Object-oriented methods

Object-oriented methods model systems as a set of communicating "objects". Identifying and designing objects is part of the object-oriented systems development process. Object-oriented models include characteristics and attributes of objects modelled as data structures (static properties) and behaviour modelled as methods or operations (dynamic properties), together with concepts of generalization, inheritance and aggregation. For example, a banking system might include "customer" objects and "account" objects and account objects may have various operations including "withdraw", "deposit" and "check balance", etc.

Many claims have been made about the object-oriented paradigm (Coad and Yourdon, 1991, Budd, 1997, Jacobson *et al.*, 1999). These claims include:

- ease of understanding object-oriented models due to a consistent underlying representation throughout the development process
- the ability to model the behaviour of objects (encapsulation of data and process)
- ease of modification and extensibility of object-oriented models.
- ease of reuse of object components from previously designed systems
- superior data abstraction facilities including inheritance and polymorphism

When studying object-oriented methods in requirements engineering, the interest is in how experienced analysts and developers actually use such methods in "real-world" system specification. Researchers need to consider whether the benefits of reuse, abstraction and reduction in complexity outweigh any difficulties in using to object-oriented methods.

3.2 An initial theoretical model

The initial theoretical model (Figure 3) is based on Loucopoulos and Karakostas (1995) and common characteristics of published, well known object-oriented methods (Coad and Yourdon, 1991, Rumbaugh *et al.*, 1991, Jacobson *et al.*, 1992, Graham, 1994, Henderson-Sellers, 1997, Avison and Fitzgerald, 2006, Booch *et al.*, 2007) particularly the concept of separate static models (which describe objects, their characteristics and the relationships between them) and dynamic models (which define states of objects, state transitions, message passing and event handling). This model represents the three processes of the requirements engineering process: elicitation, object-oriented modelling and validation which interact with users and a specific problem domain. The elicitation process clarifies user requirements within some problem domain. These requirements are then modelled within that domain using various techniques and these models are then validated against the original requirements within the problem domain. The initial model explicitly shows the static and dynamic models that are produced during object-oriented modelling.

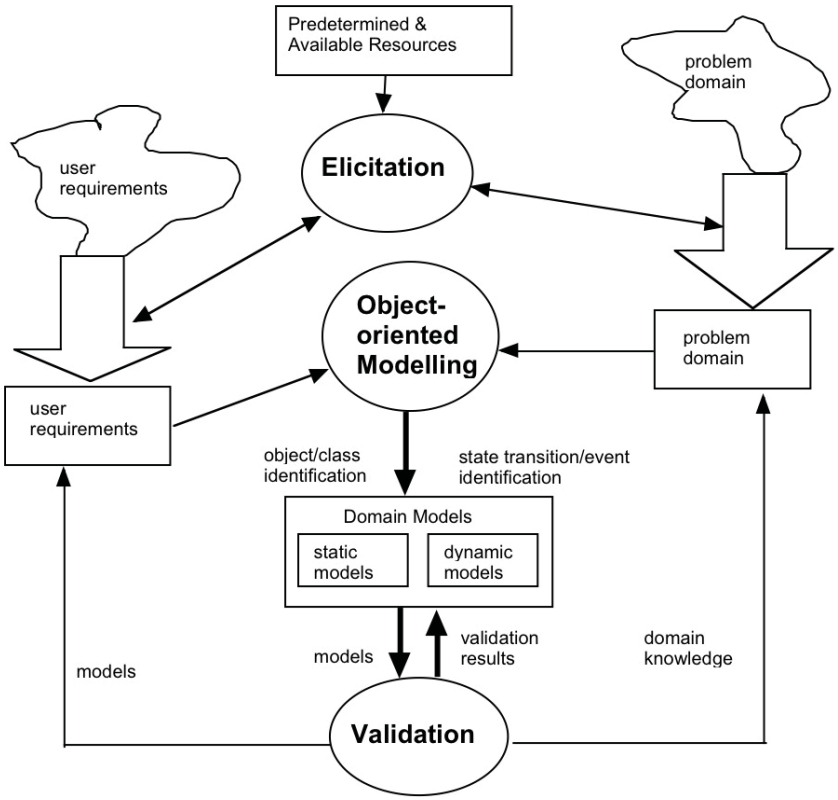

Figure 3: Initial theoretical model

3.3 Research questions

Research questions based the initial theoretical model were:

- Are there three identifiable processes of elicitation, modelling and validation in object-oriented requirements engineering practice?
- Is elicitation an iterative process in object-oriented requirements engineering practice?
- Which static and dynamic models are developed in object-oriented requirements engineering practice and how they are used and by whom?

3.4 The participants

Participants for this study were recruited through industry. All the participants were currently working in the field of object-oriented requirements specification. See Table 1.

Table 1: Background information for each consultant

Case	Title	Years in RE	Client	Project
1	Operations manager	3.5 yrs	Federal Govt	Complex Technical
2	Principal Consultant	15 yrs	State Govt	Web based transactions
3	Senior Consultant	12 yrs	Telecommunications	Fault Mgt System
4	Director & partner	22 yrs	Software developer	Insurance
5	Consultant	14 yrs	Software developer	Life Insurance
6	Technical Manager	12 yrs	Software developer	Stockbroking

4. Findings

Evaluating and analysing rich qualitative data in a rigorous manner is difficult and "…no single qualitative data analysis approach is widely accepted" (Neuman, 1994). The approach in this project was to use an evolving set of categories to structure the qualitative data as it was gathered. Firstly, a set of seed categories (Miles and Huberman, 1984, Fitzgerald, 1997, Wynekoop and Russo, 1997) was formulated based on the initial theoretical model and these were used to formulate the initial structured interview script. In each interview other categories and sub categories emerged and were incorporated into interview scripts for

investigation in the following cases. So the number of categories grew as the case studies continued. A detailed representation of the evolution of a theoretical model is presented in tables showing reinforced and revealed themes and documented revisions of the theoretical model.

The presentation of case data is based on illustrated narrative style, or an oral narrative told in the first person, as described by Miles and Huberman (1994) and Myers (1997) and as used in Fitzgerald (1997) and Urquhart (1998). This approach as described by (Miles and Huberman, 1994) does not resort to explicit coding but looks for *" ... key words, themes, and sequences to find the most characteristic accounts."*

4.1 The reinforcement of the model

4.1.1 Three processes

One of the major concepts embodied in the theoretical model is that there are three identifiable processes in the object-oriented requirements engineering activity. Those processes are elicitation, object-oriented modelling and validation of the models. The specific questions addressing this concept taken from the interview scripts are:

- Is elicitation explicitly undertaken and when does it start? When does it end?
- When does modelling begin? That is, when do you start drawing object models?
- Do you think it is necessary to validate the specification once the models have been produced?
- When does the validation process start?

The three processes of elicitation, modelling and validation were identified in all six cases. In Case 1 there were three stages within the elicitation process. Elicitation started with a "blast-off" meeting and continued with regular interviews with the users and meetings of the project team. There was little traditional modelling although requirements cards were used to represent requirements and their characteristics. Validation was based on walkthrough techniques.

In Case 2 elicitation was initiated by interviews with upper management to "scope the project". Existing documentation and data models were used as a starting point for interviews and setting up use case scenarios. Modelling was based on standard object models and use cases. Validation was based on walkthrough techniques.

In Case 3 elicitation was done by interview and prototyping with a small group of selected users and a specific "subject matter expert". Modelling was based on object/class models and comprehensive textual use cases. Modelling and prototyping were supplemented with a case tool called Software Through Pictures (STP). Validation was based on walkthroughs and revisiting the use cases and the prototype with the user group and subject matter expert.

In Case 4 elicitation was done explicitly where possible but because this project was a generic package the users were not captive to the organisation. For the purposes of elicitation, in-house business analysts and pre-sales people played the role of users in a client organisation. Both entity-relationship modelling and UML models were used to produce requirements models. The consultant also used some use cases together with ad hoc diagrams and rich pictures as models with the users. Validation was done as walkthroughs with the prototype and role-playing based on use cases.

In Case 5 elicitation was done by interviews with an emphasis on building trust and rapport. There was no use of specific modelling techniques or notation. The specification was based on a document which "embodied concepts that could be directly implemented by experienced system developers". This appeared to be based on a "programmer-oriented" approach to system development. Specific validation was not seen as necessary if the team has "built the right thing" although some acceptance testing was seen as useful.

In Case 6 elicitation was explicitly undertaken using interviews and involved gap analysis and the production of a business requirements document. Modelling was based on entity-relationship models and UML notation though not a lot of models were produced. Use cases were not used extensively because there are not a lot of tools to support them. Use cases were seen as useful in the validation process where the team can "verbalise or walk through a scenario". Prototype demonstrations were also considered useful for validation.

The analysis of findings showed that although the methods used in each process in each case were different, the three processes could be identified in all cases and so the existence of three processes was empirically validated.

4.1.2 The explicit use of feedback in elicitation

The explicit use of feedback in the elicitation process of object-oriented requirements engineering as depicted in the initial theoretical model as a double-headed arrow was actively explored with the following question taken from the interview scripts:

- Is knowledge elicitation iterative? That is, do you go back to the users several times?

In Case 1 there was feedback and elicitation was iterative. On-site meetings were held every second day with the clients and these meetings were interspersed with in-house project team meetings.

In Case 2 elicitation was an iterative process. The analyst would go back to the client, on average, three times so that the whole transaction specification took about a week. Information gathering usually took place on an initial half-day with some follow-up over the next two days. A week after the start the specification was given back to the client for review and the analyst walked them through it.

In Case 3 elicitation was considered to be "highly iterative" and included contact with the subject matter expert every couple of days.

The generic nature of the package being developed in Case 4 meant that there were no actual users available and elicitation was done using role-playing. It is not clear whether the role-playing involved iteration.

In Case 5 knowledge elicitation was seen as iterative and based on feedback. The team went back to the users several times where some piece of information triggered the need to explore some new feature or aspect.

In Case 6 users were interviewed several times to clarify points and produce the business requirements document. Different groups of users may be involved in each iteration including large groups and subsets of key users.

The analysis of findings showed that in all but one case the analyst saw the elicitation process as iterative and based on feedback for acquiring and clarifying requirements. The feedback loop in elicitation as shown in the theoretical model was empirically validated.

4.1.3 Identification and use of static and dynamic models

A major concept in the literature regarding object-oriented modelling is that there is a need for both static and dynamic models for the representation of object-oriented concepts. The nature of, and types of, static and dynamic models produced in object-oriented requirements engineering were actively explored using the following questions taken from the interview scripts:

- Which models are produced during specification?
- Do you produce class models, use case models or interaction models?
- Would you categorise models as static or dynamic?

Case 1 did not make extensive use of object-oriented models although these models were available as part of the in-house methodology. The main vehicle for representing requirements was requirements cards and the modelling that was done was based on flow charts. In Case 2 both static object/class models and dynamic interaction models were produced although the dynamic models were put into the appendices of the specification document and were rarely used with the users or for requirements specification.

In Case 3 static object/class models and dynamic interaction models were developed but the interaction models were used later in the system development process rather than in requirements specification. Case 4 included entity-relationship models and static object/class models. The dynamic models included state-transition models and interaction models.

In Case 5 models were viewed as "a set of concepts" and these models were classed as dynamic by the analyst. In Case 6 several types of static and dynamic models were used including entity-relationship

models, object models, state transition models and interaction models " *... the static model, their object model, class diagram, however you want to describe it, is the core. Everything starts there. … And then from that springs forth collaboration diagrams and interaction diagrams*."

Table 2: Reinforcement of initial concepts in the theoretical model

	Three processes	Feedback in the elicitation process	Explicit static models	Explicit dynamic models
Case 1	reinforced	reinforced	reinforced	
Case 2	reinforced	reinforced	reinforced	reinforced
Case 3	reinforced	reinforced	reinforced	reinforced
Case 4	reinforced		reinforced	reinforced
Case 5		reinforced		reinforced
Case 6	reinforced	reinforced	reinforced	reinforced

The analysis of findings showed that in four of the six cases models were produced and that they were specifically defined as static and dynamic models. Overall, the characteristics of the initial theoretical model were empirically validated by the results contained in the data from the six cases (see Table 2). There was no evidence for removing any of the major concepts from the initial theoretical model. The main characteristics analysed in this section were reinforced in five of the six cases where the five cases were different for each concept.

4.2 The evolution of the model

The following section outlines the findings of each of the six case studies with reference to evolution of the theoretical model. The major emergent themes reinforced by subsequent cases were:

- Evidence of the use of use case models as distinct models for requirements representation
- Evidence of mental modelling by analysts during elicitation before any models were committed to paper
- Evidence of the use of separate formal and informal models where informal models were the only models shown to users by analysts when discussing requirements.

4.2.1 Theme 1: Use cases

Jacobson (1992) defined a use case as:

> *"When a user uses the system, she or he will perform a behaviourally related set of transactions in a dialogue with the system. We call such a special sequence a* ***use case.*** *p 129" Jacobson, 1992)*

Simply, a use case can be thought of as a description of a transaction within the system. It can take a graphical or textual form. The use of use case models (both textual and graphic) as requirements models distinct from the static and dynamic models emerged as a category worth exploring in Case 2.

Although use cases or scenarios are described as ways of modelling requirements and transactions for object-oriented systems in several methodologies, they are difficult to classify as specifically static or dynamic models. Use cases can come in textual forms, graphical forms or both. Use case models could be classified as static models because of their textual characteristics or they could be classified as dynamic models based on the graphical, process-oriented representation. As a result use case models were not explicitly shown in the initial theoretical model. The use of use case models (both textual and graphic) as requirements models distinct from the static and dynamic models emerged as a category worth exploring in Case 2.

After Case 2 the questions incorporated in subsequent interview scripts exploring the use of use case models were:

- Do you use use case models at this stage [elicitation]? What form do they take, textual or graphical?
- Could you comment on the role and/or importance of use cases in the specification process?
- Could you comment on the relationship between use cases and static and dynamic models?

On reflection and re-examination Case 1 used a methodology that potentially included use cases but the analyst did not use them in the project under study. In Case 2 the use of simplified use cases as models in requirements specification emerged as a significant technique. "*The generic object model is a standard*

object model and the rest of the methodology is built around Jacobson use cases. The focus is on use cases. There is a standard process modelled as a use case flow diagram and a use case [structured dialog] and the client has to tick the box if there is a good fit."

Subsequently, Case 3 and to a lesser extent Cases 4 and 6 gave evidence of the use of use cases for requirements modelling. "We saw the need for a static type of model and a dynamic type of model and use cases…the use case model was more stand alone and really became the functional statement that went behind or reinforced the UR [user requirements] prototype … really most of the first phase of the work developed our use case model and a prototype to go with it." [Case 3]

> *"Well, that's … according to the theory they [use cases] are the backbone – you start with them and they go right through to the testing and so forth and I think that's reasonable. Generally speaking I think it's a good way to use them." [Case 4]*
>
> *"We do [use use cases] a little … when we have got a prototype and some rules and requirements [and] the users now start saying well, 'what if?' … We verbalise or walk through a scenario and then say well how does this system handle this. " [Case 6]*

Use cases were used to model requirements in four of the six cases and were explicitly added to the theoretical model – version 2 (see Table 3 and Figure 4).

4.2.2 Theme 2: Mental modelling

The concept of mental modelling during elicitation by an analyst emerged in Case 3 as the following selection of quotes from the transcript illustrates: *"…there were fragments of that [business] model getting developed in a couple of people's heads for probably three months…I would say that it was being done largely privately and it was not written down until the last minute when it was just a dump"*.

> *"You will be listening very carefully [in project meetings] and collecting and cataloguing constraints and refining the abstractions in your mind."*

This concept was explored in subsequent case study interviews with the following question:

- Do you start developing mental models during elicitation?

Overall four of the six analysts believed that they were continually "modelling in the mind" during the elicitation process and that these mental models were further refined in the mind before they were communicated to others (users or fellow analysis team members) or before they were committed to paper.

In Case 4 creating mental models involving key objects was seen as a natural part of requirements elicitation and modelling "I think that I do immediately start thinking of key objects during requirements gathering, not in any formal way, they just pop into one's head. I don't agree with the implication … that identifying objects and 'building mental models of the system' are mutually exclusive. One can help the other."

In Case 5 mental modelling was perceived as an integral part of abstraction "This is really about how people think. Some of us more conscious of the models, others not."

In Case 6 producing mental models during elicitation was seen as a natural way of thinking for that particular analyst although he thought it was a personal thing and that other analysts might not work that way "*I do that. ...That's not true of some of the others. You can tell when you look at their work that they've not actually thought about any form of underlying structure at all. All they've really done is try to gather the business requirements*."

Mental modelling emerged as significant in Case 3 and was subsequently reinforced in Cases 4, 5 and 6 and was explicitly added to the theoretical model – version 3 (see Table 3 and Figure 4).

4.2.3 Theme 3: "Formal" and "informal" models

As described above, when asked when the modelling process began, most analysts said they were building models in their heads long before any formal models were written down.

Evidence of the use of "informal" models (simple use case models, ad hoc pictures, diagrams, animations etc) instead of "formal" notated models (object/class diagrams) for communicating the specification to clients and users emerged in Case 4. Subsequent reflection and re-examination of the transcripts of Cases 1, 2 and 3 revealed further evidence of this "separation of models". For the purpose of discussion formal models and informal models are distinguished in the following way. Formal models are considered to be those models

that require training in order to be understood or explained. That is, models that contain specific, often graphical notations such as UML models, interaction models or state models. Informal models are considered to be models that can be understood and explained without specific training. In this category are natural language models including text descriptions, use case scripts, ad hoc diagrams and interactive demonstration models as often produced for prototypes.

Questions regarding types of models which were part of the original interview script were:

- Which (how many) models are produced during specification?
- Who uses them? Who are they produced for?
- Which models, if any, are shown to the user? Which models are used internally by the development team?

Questions regarding types of models asked in interviews subsequent to Case 4 were:

- Which (how many) models are produced during specification? Do you produce class models, use case models or interaction models?
- Do you use informal models (pictures etc) to communicate with the users?
- Do you use use case models at this stage? What form do they take?

In the case studies there seemed to be two different kinds of modelling taking place. Firstly, there were the informal models that were used to communicate with the users. Secondly, more formal models were developed which were not shown to the users because it was believed that the users would not understand them. The formal models were developed primarily for design purposes and were private to the analyst or team of analysts. In effect these formal models were the analysts' internal version of informal models.

In Case 1 requirements cards were used directly with the users to represent or "model" requirements and it was not made clear which models, if any were used in the design and implementation process "*The card is pretty much self documenting …straight into the actual requirement spec. So once you have the cards complete, a lot of the hard work is done ... we cannot write the requirements, they must tell us the requirements ... we work with them on the cards.*"

In Case 2 only simplified use case diagram and dialogues were shown to the users when describing requirements. Models based on formal notation were considered too complex for users to understand "*We tell them [the users] that the model is technical mumbo jumbo ... you know I wouldn't show them a data model either … the closest I've gotten is working with this type of flow diagram (use case flow diagram)…they can follow that pretty well but they don't usually have the patience to really work through the interaction diagrams or the model. It just takes too much explanation*."

In Case 3 use case scripts and a prototype were used to develop the requirements with users and formal object models were not shown to users " *… unless the 'users' were IT-literate people, which most aren't. (Do you believe it is not necessary to show them?) I believe it is not only not necessary, but potentially dangerous. It is the analyst's job to perform the use case to business object model translation*."

In Case 4 the analyst was explicit about using various ad hoc diagrams, pictures, PowerPoint simulations and some use cases to communicate the requirements to users "*I mean if you draw a picture and that doesn't make any sense to them then you draw another one ... A requirements specification has to be in terms that they understand and those three mechanisms we've already mentioned are the way: the use case, the ad hoc diagrams and the dynamic screen simulations … I am not saying one should do away with the formalisms. They are a powerful aid to one's own understanding and analysis but they are not a good tool for feeding requirements back to the users. It's much better to bend the formalism to the user than the user to the formalism*."

Table 3: The evolution of the theoretical model

	use case models	Version 2 of theoretical model	mental models	Version 3 of theoretical model	formal and informal models	Version 4 of theoretical model
Case 1					informal models for users reinforced by reflection and re-examination	
Case 2	revealed				reinforced by reflection and re-examination	
Case 3	reinforced		revealed		reinforced by reflection and re-examination	
Case 4	Weakly reinforced		reinforced		revealed	
Case 5	reinforced		reinforced		weakly reinforced	
Case 6	Weakly reinforced		reinforced		reinforced	

The consultant in Case 5 did not use models or explicit diagrams. Most of the specification was based on text-only documents including a formal specification document.

In Case 6 the most important objective was to get the users or clients to sign off on the specification "…we would have a formal walkthrough with the users where we go through the requirements, … and probably the prototype and we would get formal sign off". In this case the prototype was the most used tool for validation and use cases were only used for exceptions or special cases "…as we were looking at the requirements document we had the prototype running and projected up on a big screen and we actually walked through the prototype in relation to the requirements".

Table 3 shows the evolution of the theoretical model in terms of revelation, reinforcement reflection and re-examination. Figure 4 shows version 4 of the theoretical model incorporating, use case models (version 2), mental modelling (version 3) and the separation of formal and informal models in the object-oriented requirements engineering process. The feedback from the validation process for formal modelling in the problem domain is via the ad hoc mapping of informal models to formal models for use in design and implementation.

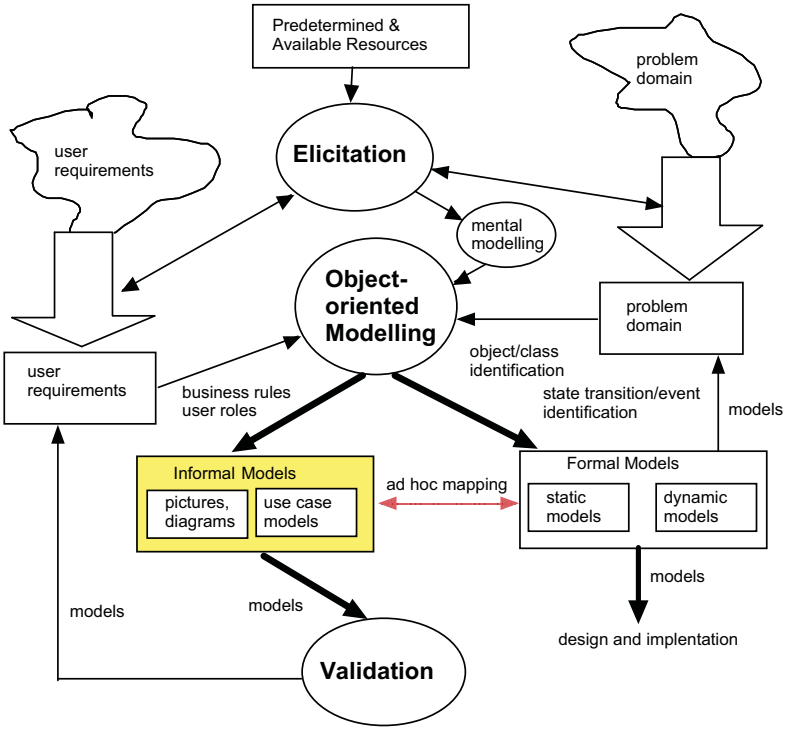

Figure 4: Final theoretical model

The final theoretical model (Figure 4) now embodies all the concepts of the initial theoretical model which were proposed at the start of the evolutionary cycle based on the literature regarding object-oriented requirements engineering, together with the concepts that emerged from the sequential case studies. This revised theoretical model provides a theoretical representation of the object-oriented requirements engineering process grounded both in the literature and in professional requirements engineering practice.

5. Conclusion

The evolutionary case approach provides a framework for qualitative case study research which actively explores emerging themes. Emerging themes that are contained in the current version of the theoretical model are actively explored in interviews and themes that are reinforced by empirical data become part of the next version of the theoretical model. An ongoing process of reflection and re-examination further refines the data and the model.

This framework was specifically developed to assist exploratory theory-building research using case studies of IS in practice. The theoretical model and the associated tables documenting the emergence and reinforcement of themes provides structure and support for theory building within the specific constraints of interpretive, qualitative, interview-based data collection.

This structure and support for the research process makes contributions in two ways. It provides productivity gains by providing a specific framework and process based on the development of a theoretical model for representing the current state of the area under study and it contributes to quality outcomes by providing clear documentation of the process and results.

The illustrative example of the use of models in object-oriented requirements engineering demonstrates that the evolutionary case approach provides a research environment for truly explorative theory building in an evolutionary and active way.

References

Avison, D. & Fitzgerald, G. (2006) Information Systems Development: methodologies, techniques and tools. McGraw-Hill, Maidenhead.

Avison, D. et al. (1999) "Action Research". Communications of the ACM, Vol 42, No. 1, pp94-97.

Bacharach, S. (1988) "Organizational Theories: Some Criteria for Evaluation". Academy of Management Review, Vol 14, No. 4, pp496-515.

Baskerville, R. & Wood-Harper, A.T. (1998) "Diversity in information systems action research methods". European Journal of Information Systems, Vol 7, pp90-107.

Booch, G. et al. (2007) Object-Oriented Analysis and Design with Applications, Addison-Wesley, Reading MA.

Budd, T. (1997) An Introduction to Object-Oriented Programming. Addison-Wesley, Reading MA.

Burton-Jones, A. & Meso, P.N. (2006) "Conceptualizing Systems for Understanding: An Empirical Test of Decomposition Principles in Object-Oriented Analysis". Information Systems Research, Vol 17, No. 1, pp38-60.

Carroll, J.M., Dawson, L.L. & Swatman, P.A. (1998) "Using Case Studies to Build Theory: Structure and Rigour". Ninth Australian Conference on Information Systems, University of New South Wales, Sydney, Australia, pp64-76.

Carroll, J.M. & Swatman, P.A. (2000) "Structured-case: a methodological framework for building theory in information systems research". European Journal of Information Systems, Vol 9, pp235-242.

Cavaye, A. (1996) "Case study research: a multi-faceted research approach for IS". Information Systems Journal, Vol 6, No. 3, pp227-242.

Checkland, P. & Scholes, J. (1990) Soft Systems Methodology in Practice. Wiley, Chichester.

Coad, P. & Yourdon, E. (1991) Object-Oriented Analysis. Yourdon Press/Prentice-Hall, Englewood Cliffs, NJ.

Darke, P., Shanks, G. & Broadbent, M. (1998) "Successfully completing case study research: combining rigour, relevance and pragmatism". Information Systems Journal, Vol 8, No. 4, pp273-289.

Dawson, L. & Darke, P. (2002) "The Adoption and Adaptation of Object-Oriented Methodologies in Requirements Engineering Practice". Tenth European Conference on Information Systems, Wrycza, S., Gdansk, Poland, pp406-415.

Dawson, L. & Swatman, P. (1999) "The use of Object-oriented Models in Requirements Engineering: a Field Study". Twentieth International Conference on Information Systems, De, P. & DeGross, J.I., Charlotte, NC, pp260-273.

Dobing, B. & Parsons, J. (2006) "How UML is Used". Communications of the ACM, Vol 49, No. 5, pp109-113.

Dobing, B. & Parsons, J. (2008) "Dimensions of UML diagram use: a survey of practitioners". Journal of Database Management, Vol 19, No. 1, pp1-18.

Dubin, R. (1976) Theory Building in Applied Areas. In Handbook of Industrial and Organisational Psychology, (Ed, Dunnette, M.) Rand McNally College Publications, Chicago, pp. 17-39.

Eisenhardt, K.M. (1989) "Building Theories from Case Study Research". Academy of Management Review, Vol 14, No. 4, pp532-550.

Ericsson, K.A. & Simon, H.A. (1980) "Verbal Reports as Data". Psychological Review, Vol 87, No. 3, pp215-251.

Fitzgerald, B. (1997) "The use of systems development methodologies in practice: a field study". Information Systems Journal, Vol 7, No. pp201-212.

Galliers, R. (1992) Choosing information systems research approaches. In Information Systems Research: Issues, Methods and Practical Guidelines, (Ed, Galliers, R.) Blackwell Scientific, Oxford, pp. 144-162.

Graham, I. (1994) Object Oriented Methods. Addison-Wesley, Wokingham, UK.

Grant, E.S. & Reza, H. (2007) "Towards the development of a rigorous model-driven domain-specific software engineering environment". Third IASTED International Conference Advances in Computer Science and Technology, Phuket, Thailand,
Gregor, S. (2006) "The nature of theory in information systems". MIS Quarterly, Vol 30, No. 3, pp611-642.
Henderson-Sellers, B. (1997) A Book of Object-Oriented Knowledge. Prentice-Hall, Upper Saddle River, NJ.
Hull, E., Jackson, K. & Dick, J. (2005) Requirements Engineering. Springer, London.
Hult, M. & Lennung, S.-A. (1980) "Towards a definition of action research: a note and bibliography". Journal of Management Studies, Vol 17, No. May, pp241-250.
Jacobson, I., Booch, G. & Rumbaugh, J. (1999) The Unified Software Development Process. Addison Wesley Longman, Reading, MA.
Jacobson, I. et al. (1992) Object-Oriented Software Engineering: A Use Case Driven Approach. Addison-Wesley/ACM, New York.
Loucopoulos, P. & Karakostas, V. (1995) Systems Requirements Engineering. McGraw-Hill, London, UK.
Macaulay, L. (1996) Requirements Engineering. Springer-Verlag, London.
Miles, M.B. & Huberman, A.M. (1994) Qualitative Data Analysis: An Expanded Sourcebook. Sage Publications Inc, Thousand Oaks, CA.
Myers, M.D. (1997) "Qualitative research in information systems," MISQ Quarterly, Vol. 21, No. 2, pp. 241-242, http:/www.qual.auckland.ac.nz, access date January 15 2008.
Myers, M.D. & Newman, M. (2007) "The qualitative interview in IS research: Examining the craft". Information and Organization, Vol 17, No. 1, pp2-26.
OED (Oxford English Dictionary Online), http://dictionary.oed.com (current Aug. 10, 2004).
Pohl, K. (1994) "The Three Dimensions of Requirements Engineering: A framework and its applications". Information Systems, Vol 19, No. 3, pp243-258.
Rumbaugh, J. et al. (1991) Object-Oriented Modeling and Design. Prentice-Hall, Englewood Cliffs, NJ.
Sommerville, I. (2006) Software Engineering. Addison-Wesley,
Sommerville, I. & Sawyer, P. (1997) Requirements Engineering: A good practice guide. John Wiley and Sons, Chichester.
Strauss, A. & Corbin, J. (1990) Basics of Qualitative Research: Grounded Theory Procedures and Techniques. Sage Publications, Newbury Park, CA.
Susman, G.I. (1983) “Action Research. A Sociotechnical Systems Perspective”. In Beyond Method: Strategies for Social Research, (Ed, Morgan, G.) Sage, Newbury Park, pp. 95-113.
Sutcliffe, A.G. & Maiden, N.A.M. (1992) "Analysing the novice analyst: cognitive models in software engineering". International Journal of Man-Machine Studies, Vol 36, No. pp719-740.
Urquhart, C. (1998) "Analysts and Clients in Conversation: Cases in Early Requirements Gathering". Nineteenth International Conference on Information Systems, Hirschheim, R., Newman, M. & DeGross, J.I., Helsinki, Finland, pp115-127.
Walsham, G. (1995) "Interpretive case studies in IS research: nature and method". European Journal of Information Systems, Vol 4, pp74-81.
Yin, R.K. (1994) Case Study Research: Design and Methods. Sage Publications Inc, Thousand Oaks, CA.

