

Client Led Information System Creation (CLIC): Navigating the Gap

Donna Champion

The Business School
Loughborough University
Loughborough, LE11 3TU

E Mail: d.champion@lboro.ac.uk

Frank Stowell

Centre for Innovation and Inclusion through Information Systems
University of Portsmouth
Portsmouth, PO1 3AE

E Mail: Frank.Stowell@port.ac.uk

and

Alan O'Callaghan

De Montfort University
The Gateway,
Leicester, LE1 9BH,

E Mail: aoc@dmu.ac.uk

Abstract

In this paper, we offer a new framework to facilitate an interpretive approach to client led information system development, referred to as CLIC (Client Led Information system Creation). The challenge of moving seamlessly through a process of information systems (IS) design is still the subject of much research in the IS field. Attempts to address the difficulties of 'bridging the gap' between a client's business needs and an information system definition have hitherto not provided a coherent and practical approach. Rather than attempting to bridge the gap, this paper describes an approach to managing this gap by facilitating the clients navigating through the information system design process (or inquiry process) in a coherent manner. The framework has been developed through practice and the paper provides an example of navigating through the design phase taken from an Action Research field study in a major UK bank.

Client Led Information System Creation (CLIC): Navigating the Gap

Abstract

In this paper, we offer a new framework to facilitate an interpretive approach to client led information system development, referred to as CLIC (Client Led Information system Creation). The challenge of moving seamlessly through a process of information systems (IS) design is still the subject of much research in the IS field. Attempts to address the difficulties of 'bridging the gap' between a client's business needs and an information system definition have hitherto not provided a coherent and practical approach. Rather than attempting to bridge the gap, this paper describes an approach to managing this gap by facilitating the clients navigating through the information system design process (or inquiry process) in a coherent manner. The framework has been developed through practice and the paper provides an example of navigating through the design phase taken from an Action Research field study in a major UK bank.

Introduction

In this paper we describe a *framework for thinking* developed through practice to facilitate a client led approach to information system design, underpinned by interpretivist principles. We refer to this approach as: Client Led Information system Creation (CLIC). The CLIC framework is a development of the work undertaken by Stowell for the past 18 years (Liang et al, 1998; Stowell, 1985, 1991, 2000; Stowell and West, 1994; Stowell et al, 1997). We suggest that using this framework facilitates a coherent approach to sense making and the creation of a shared appreciation (Vickers, 1965), from the exploration of the problem situation right through to the design for an information system and if appropriate, the application of different methods for designing and creating software. The flexibility facilitated through the CLIC framework is important as it can provide the opportunity to support both the clients and the software developers in using their preferred methods of computer-based information system development.

Various authors have discussed the difficulty of moving from some means of exploring a problem situation to creating a specification for an information system (Avison et al, 1998; Boland, 1985; Dietz et al, 1998; Doyle et al, 1993; Lai, 2000; Liang et al, 1998; Mathiassen and Nielsen, 2000; Miles, 1988; Reijswoud, et al, 1999; Savage and Mingers, 1996; Stowell and West, 1994). Most of the research in this field has been directed at providing some sort of 'bridge' between clients' requirements and a logical specification. The focus on providing a bridge between what the client wants and a specification for a technology-based information system has resulted in a concern to *map*, or *trace*, requirements from one type of model to another. Even within the work of researchers working from an interpretivist

perspective, there is an underpinning concern to translate information system requirements directly from one sort of model to the next. Avison et al, 1998; Guo et al, 2000; Liang et al, 1998; Mathiassen and Nielsen, 2000; Savage and Mingers, 1996 and Stowell and West, 1994 have all offered approaches that suggest different ways of translating requirements directly from one sort of model to another. We suggest that participants in the design process need to be offered support in order to better appreciate the difference (or gap) between the purposeful action they wish to take and the support that will be required in order to implement that action.

The work presented here offers a different approach. We have applied the definition of an information system first suggested by Checkland and Scholes (1990, p. 54), that is: an information system is a system to *serve* purposeful action. This definition is underpinned by the assumption that the information system, (or serving system), is *different* to the 'system to be served', (or the purposeful action that people wish to take). The *system to be served* must be considered and conceptualised in some way before the *serving system* can be considered (Checkland and Scholes, 1990; Winter et al, 1995).

It is important to note that any models created during the design process are regarded as intellectual devices to facilitate participants in first, expressing ideas about what purposeful action might bring improvement to a problem situation, and then to express ideas about what sort of serving system might be needed, if such action were indeed taken. The models are *relevant* to the situation (Checkland, 1995), and are used as devices to facilitate debate and as one means of expressing ideas to help people think through the consequences and implications of design decisions. This is a direct application of Checkland's insight that Systems thinking "*....transfers systemicity from the world to the process of inquiry into the world*" (Checkland, 1983, p. 672, his italics). The focus when using this approach is to use appropriate methods and tools in a coherent and flexible manner to manage the process of sense-making.

The models depicted in Figures 2, 3 and 4 of this paper have been previously published as the starting point for illustrating one means of navigating through the design process, using a computer-aided design tool called *SOMATiK* (Champion and Stowell, 2002). The models are reproduced in this paper to illustrate the CLIC framework for thinking to guide the design process and also to illustrate a way of navigating through the design process using *different* methods and tools to specify a technology-based information system (Unified Modelling Language). The resultant

specification presented here was accepted by the business clients. We believe that the flexibility for using different methods and tools in a coherent manner, achieved through the application of the CLIC framework, is fundamental to its practical value for business and industry.

The CLIC framework for guiding information system design is informed by the work of Vickers (1965, 1983) and is undertaken from an interpretivist perspective. That is, we "...act according to the assumption that social reality is socially constructed, continuously [and we...] use explicit intellectual devices consciously to explore; understand and act in the situation in question" (Holwell, 1997, p. 401). In order to be consistent with these principles of inquiry, we have used an Action Research approach in our work.

Practice-based Research: Action Research

Action Research (AR) has been used as an interpretivist framework to guide collaborative research within the Information Systems field for some years (See Boland, 1985; Stowell and West, 1994; Checkland, 1999). Establishing the validity and rigour of such research is still regarded as being problematic by some (see Davison et al, 2004). Davison et al (2004) argue that Action Research ought to be "...iterative, rigorous and collaborative, involving a focus on both organizational development and the generation of knowledge". We would support this argument. Davison et al (2004) also offer a framework for undertaking Action Research 'with rigour' that they have found useful in their own practice. The Action Research field study described in this paper was undertaken guided by a different, more established framework, the F, M, A model suggested by Checkland (1985), where F is the Framework of ideas to be used, M is the Methodology underpinning the inquiry and A is the Area of concern, all of which are declared in advance to facilitate scrutiny of the learning outcomes by interested individuals. This approach to Action Research is also iterative, rigorous and collaborative and can help to facilitate sense making. It is important to note however, that the frameworks offered by both Checkland and Davison et al, for undertaking Action Research evade a consideration of the *manner* in which the inquiry was conducted (Champion, 2002).

One of the most difficult aspects of undertaking research in real world social situations, is that sometimes, research that is perceived as being rigorous from an

academic perspective is by no means guaranteed to provide practical outcomes that are regarded as being successful by research collaborators from business or industry. Even if participants welcome any practical outcomes of research in the short-term, longer-term effects can be considered to be detrimental (Dash, 1999). Within human social situations, meaning is continually being created and recreated and so nothing can be said to be complete, mapped or finally understood. For a process of social inquiry to maintain its integrity then, there must be the possibility of continuing dialogue and debate. Williams argues that

“honest discourse permits response and continuation; it invites collaboration by showing that it does not claim to be, in and of itself, final. It does not seek to prescribe the tone, the direction, or even the vocabulary of a response. And it does all this by showing in its own working a critical self-perception [...] it makes clear [...] that there are ways in which it may be questioned and criticized”.

(2000, p. 5)

For this reason, when undertaking research within social situations it is also essential to consider not only the academic rigour, but also the manner in which the research has been conducted, or the *character* of the research process (Champion, 2002; Champion and Stowell, 2003, 2004). We have argued elsewhere that the PEARL mnemonic (P- Participants, E –Engagement, A- Authority, r –relationships¹ and L- Learning) can provide both participants and (crucially for research undertaken in social situations) non-participants, with a means of scrutinising the character of the inquiry process and so facilitate a judgement being made on the integrity of the inquiry process (Champion, 2001, 2002; Champion and Stowell, 2001, 2003). One of our main concerns is to facilitate the collaboration of the clients involved in the problem situation, so that the words ‘Client Led’ refer to a practical reality. Many participative approaches to information design and development fail to facilitate active involvement throughout the design phase.

A Criticism of ‘Participatory Approaches’.

In an overview of Participatory Design approaches, Kensing and Blomberg (1998) suggest that practitioners advocating such approaches have three main concerns.

First, to undertake a consideration of the politics of information system design, second, to reflect upon the nature of participation and third, to employ methods, tools and techniques to enable participation (Kensing and Blomberg, 1998). Clement (1994) argues that the prime objective of practitioners using these approaches is to avoid “deskilling” and “dislocation” and to understand the relationships between the people involved in the situation and the technology employed. Within the literature on Participatory Design, there seems an underpinning concern that technology can be used to increase management control (Clement, 1994; Kensing and Blomberg 1998). Some Participatory Design approaches even recommend the exclusion of management personnel from the process of information system design entirely (Bødker, 1996). Early examples of Participatory Design approaches include the DEMOS and UTOPIA projects (Ehn, 1989); both of these projects were concerned with prototyping new information technology tools for participants.

Ethnographic methods of *sense making* are often used in the initial stages of Participatory Design and are also found in the work of Sommerville and Sawyer (1997) on Requirements Engineering. For example, such ethnographic methods might include open-ended contextual interviews, participant observation and analysis of work scenarios (Beyer and Holtzblatt, 1998; Kensing and Munk-Madsen, 1993; Kotonya and Sommerville, 1992; Rodden et al, 2000; Sommerville and Sawyer, 1997). Research on using ethnographic methods in a process of information systems design has resulted in the suggestion that “ethnographic material [can] facilitate the construction of abstract models of work as part of the design process” (Rodden et al, 2000, p. 158). The underpinning assumption here would seem to be that a detailed description of a real world situation is the most useful starting point for information system design. The difficulty with such an approach is that information system design is almost always concerned with future action, not current ways of working and detailed models of current processes do not necessarily provide useful support for those involved in the design process.

Perhaps the most serious criticism of Participatory Design approaches is the continued focus on determining the technological provision that is to be used by the clients. There is little concern voiced in the research on Participatory Design on how to support those involved in the situation of focus in learning about their predicament. Consequently, it often appears that the problem is assumed to be the design of the

¹ We have used a small ‘r’ for relationships deliberately to emphasise the ‘soft’ interpretivist approach

information technology interface and how to involve people in the process of that design, rather than an exploration of the problem situation and the potential for improvement.

Kensing and Blomberg (1998) criticise the field of Participatory Design, as there are few examples of approaches that offer coherent ensembles of tools and techniques. Beyer and Holtzblatt (1998) suggest that their approach, Contextual Design, does offer a complete guide to information system design. They regard the core design problem to be how to enable the users of the IT to decide how they will work in the future (Beyer and Holtzblatt, 1998, p. 3), but again the initial focus is on constructing a description of the current business process, with an emphasis on designing how the human operator will interact with the computer interface. The focus in Participatory design approaches on current processes overlooks the process of problem identification and avoids offering support for creative, innovative thinking that may lead to completely new action being undertaken.

The idea that it may be more useful to support people in actively engaging in thinking about what action they might take to bring improvement *before* creating an information system designed to *serve that desired action* (Checkland and Holwell, 1998; Winter et al, 1995) is not expressed in the so-called participatory approaches to IS design. To facilitate the design of information systems that *serve* purposeful action, it is necessary to enable those involved in the problem situation to first make sense of the situation (Checkland and Scholes, 1990), that is, to undertake an exploration-oriented approach to information system design.

Navigating the Gap

Other more exploration-oriented approaches to information system design and development have been suggested that are underpinned by the principles of Soft Systems Methodology (SSM), and that employ conceptual activity models, or holons, to express ideas for purposeful action. For example, Multiview2 (Avison et al, 1998), and Client Led Design (Stowell and West, 1994) are both IS design and development frameworks developed from the work at Lancaster (Checkland, 1981, 1999; Checkland and Holwell, 1998; Checkland and Scholes, 1990; Vickers, 1983). SSM is widely accepted as being useful as a guide when exploring a problem situation, but there have been problems in linking the learning from this first stage of inquiry with

we take to inquiry within human situations (Champion and Stowell, 2003).

the sorts of models used to express information system requirements. For example, Savage and Mingers (1996) have commented on the awkward transition from ideas for purposeful action expressed using conceptual activity models to other models such as Data Flow Diagrams, or Entity Relationship Diagrams (used in Client Led Design and Multiview respectively). Also, one of the central concerns in information system design methods to date, has been the need to create models that are sufficiently detailed to facilitate the specification of the software components of a technology-based information system (Sommerville and Sawyer, 1997). This need for detailed models by software developers tends to reduce the opportunities for active involvement of non-technical staff in the client group. Savage and Mingers (1996) have argued that any modelling devices used, ought to be easily understandable and openly accessible to all those involved in the design process, and not unduly technical in execution. The framework and modelling devices described in this paper, have been developed to facilitate access and involvement in the design process, and used within practical situations to refine and adapt them further.

The CLIC Framework for Thinking

The main innovation offered by the CLIC framework is the different approach to managing the 'gap' between the clients' business needs and creating the technical specification for an information system. CLIC has been developed through the work undertaken for the UMISD² project and the intellectual devices and approach described in this paper have all been applied in practical situations. The CLIC framework is based on a hermeneutic cycle of learning where theory leads to practice and practice to theory. Hermeneutics is discussed at length by Gadamer (1989) who describes it as the "art of understanding" (p. 164) and he explains this as an attitude that places the mental constructs of those involved in the situation as being prime, rather than a world known through experimental evidence gained from scientific inquiry. Gadamer (1989, p. 301) argues "the very idea of a situation means that we are not standing outside it and hence are unable to have any objective knowledge of it". Following through the logic of an information system being a system to serve purposeful action, and by remaining within an interpretivist philosophical stance, the process of information system design and development can be expressed very simply by the diagram in Figure 1 below:

² The UMISD (Unified Mechanism for Information System Definition) project was an EPSRC funded project directed at developing practical interpretive modelling methods. See Champion, 2001).

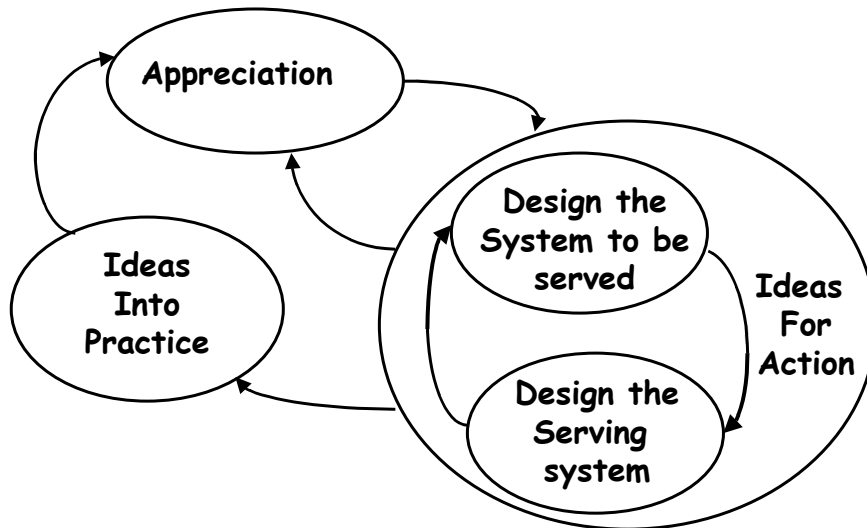


Figure 1: The CLIC Framework for guiding a process of information system design and development.

Maintaining the logical coherence during each phase of inquiry is challenging as each phase of the CLIC framework is a complex undertaking in itself. Following this framework, the first phase of inquiry during a process of IS development will be to undertake an exploration of the problem situation and to create a shared appreciation between the clients, developers and other participants (Vickers, 1965) of the situation of concern. Once some accommodation has been reached about the problems that need to be addressed, participants can move on to consider 'Ideas for Action' that might bring improvement. During a process of information system design using the CLIC framework, this phase of debate is approached by first considering ideas for the *system to be served* and then considering ideas for the *servicing system*. This process is intended to increase the shared appreciation of the potential for improvement amongst those involved. Several iterations of this learning cycle will most likely be necessary before people are ready to move on to implementing the ideas and putting them into practice. A form of conceptual conversation model has been adapted and developed to facilitate the operationalisation of the CLIC framework to facilitate participants undertaking a more detailed consideration of the implications of design decisions.

Conversation Models

The idea that conversations in a social setting can be regarded as enabling co-operative action has been employed by other researchers working in the field of information system design (Ågerfalk, et al, 1999; Dietz et al, 1998; Flores and Ludlow, 1981; Winograd and Flores, 1987). Other researchers have also developed methods for mapping conversations in organizational settings (Goldkuhl and Roslinger, 1999; Harris and Taylor, 1998; Reijswoud, et al, 1999; Schoop, 1998). There is a fundamental difference between our approach and the work undertaken by these researchers. It is important to recognise that the focus in the methods described by Goldkuhl and Roslinger (1999), Harris and Taylor (1998), Reijswoud, et al (1999) and Schoop (1998) attempt to provide accurate maps of conversations that actually occur within a situation. In the approach to information system design described here, the models are employed as tools for facilitating debate and to express ideas about conversations that *might* occur, if any action were to be undertaken. The models created are *relevant* to debate, and not representative of the action (Checkland, 1995). Although Conversation Modelling has been criticised for being difficult to undertake, overly detailed and as offering a very rigid view of a situation (Graham, 1998; Hirschheim et al, 1995), it seemed the explicit and detailed nature of a Conversation Model was useful in this particular instance. When creating a logical specification for a technology-based information system, the modelling devices employed to facilitate debate must fulfil two purposes. First, the devices must facilitate debate concerning the implications of implementing the intended action in the real world situation, enabling those responsible for creating the technical specification to consider how to provide the technical functionality required. Second, any intellectual devices used to support debate in this phase of inquiry must facilitate a sense of coherence being maintained, as the design gap between the *system to be served* and the *servicing system* is navigated. Currently one of the best-known and most useful devices for expressing ideas for purposeful action is a conceptual activity model. By employing a Conversation Model as a debating device for the next phase, an activity-oriented view of the situation is maintained, providing a degree of continuity (Champion, 2001; Champion and Stowell, 2002).

Navigating through Information System Design

The models used to explain the CLIC framework shown below, were created during a field study that took place in the Credit Card Debt Management department of a UK bank (Champion, 2001). The bank had supported the UMISD project from the outset and the bank's workplace had been explicitly offered as an environment in which to

try out any research ideas in the form of a field study. The Director of Information Technology had expressed the view that any ideas that emerged from the research were potentially useful to the bank and that learning from the study would be of interest even if the outcome resulted in ideas being rejected. This is important in an Action Research field study, as it shows that the collaborators in the field study were prepared to relinquish the desire for specific outcomes and participate in the field study simply as a learning experience. Practical learning outcomes did indeed emerge that were of interest and that were useful for all the participants, but it is important to acknowledge that this cannot be guaranteed at the start.

Appreciate

The first stage of the CLIC framework is to undertake an exploration of the situation to create a shared appreciation (Vickers, 1965) amongst those involved about what problem issues are causing concern. For the field study described here, to support the debate, a series of Rich Pictures was created in collaboration with workers from the bank. This process involved many different personnel from different sections and departments in the inquiry process.

Ideas for Action: Designing the System to be Served

Once a shared appreciation of the problem situation had been created, ideas for purposeful action that could be taken that might bring improvement to the situation were considered. Initially this phase of debate focussed on ideas for the *system to be served* and these ideas were expressed as conceptual activity models, or holons. It is important to remember, these models are not regarded as *maps of what will occur*, they are used as “complementary pictures” (Vickers, 1981) to support debate and dialogue amongst the participants in the inquiry. A simplified version of one of the conceptual activity models created in the field study is shown in Figure 2 below. This model expresses the ideas for purposeful action that people working in the bank agreed might bring improvement to their situation.

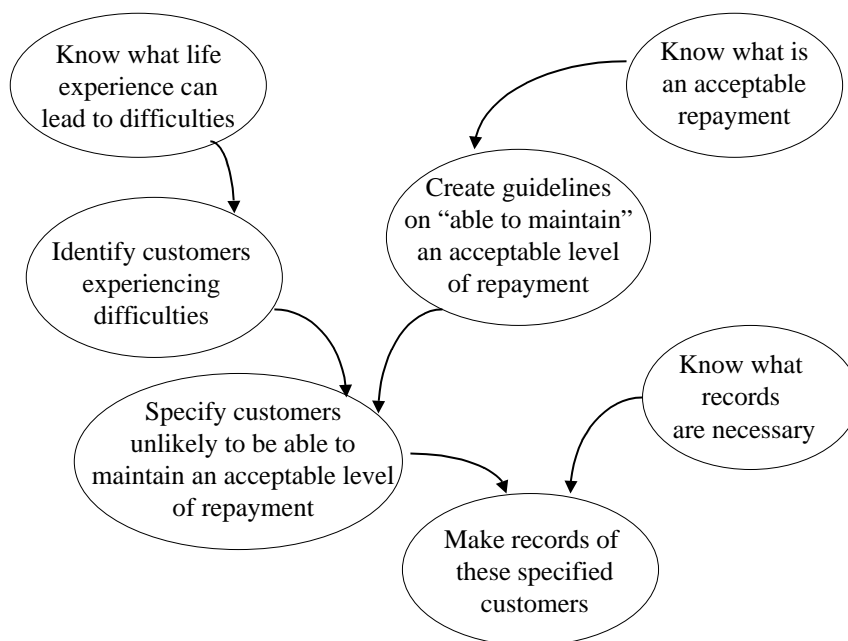


Figure 2: A conceptual model created to express ideas for purposeful action (simplified) (Reproduced with permission from Champion and Stowell, 2002).

Once an accommodation was reached concerning what action might bring improvement, following the CLIC framework, a further stage of debate is initiated to facilitate both the clients and developers in conceptualising how the action might be operationalised in practice. That is, the ideas for action are ‘contextualised’ to create a more detailed “complementary picture” (Vickers, 1981) to facilitate further learning. The intellectual device used to facilitate this phase of debate was a ‘Conversation for Action’ model. Winograd and Flores (1987) argued that a ‘Conversation for Action’ begins with an initial request, which has some underpinning “Conditions of Satisfaction” that must be fulfilled for the conversation to be completed. The models were created through the active participation of the collaborators and to facilitate this participation and to maintain the sense of coherence as the inquiry process progressed, the ‘Conditions of Satisfaction’ were considered first and written directly on to the conceptual activity models, by the participants themselves (Champion, 2001). An example of one of the models from this phase of inquiry is provided below in Figure 3.

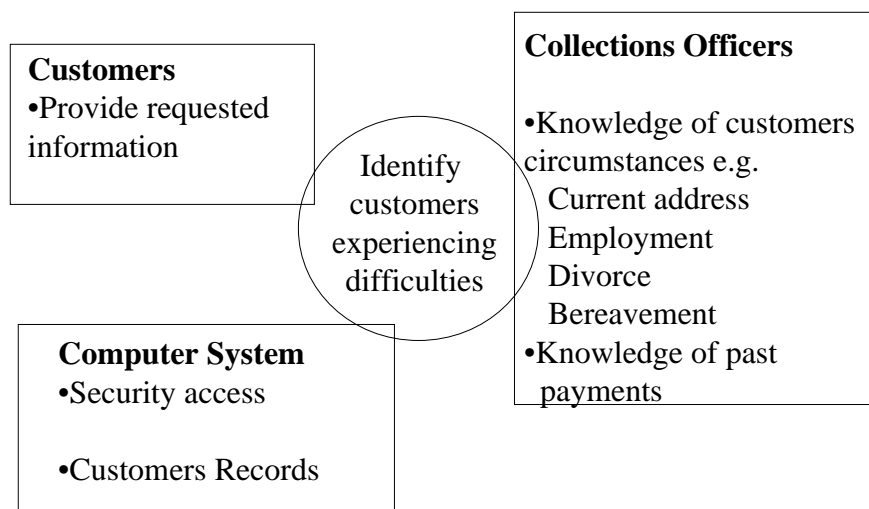


Figure 3: One of the Sub-Systems from the conceptual model in Figure 2, contextualised with some Conditions of Satisfaction. (Reproduced with permission from Champion and Stowell, 2002).

Once all those involved agreed that the Conditions of Satisfaction were sufficiently complete, the next phase was to create some 'Conversations for Action' models that facilitated the clients in thinking through how the action might be operationalised in the real world situation. These models structured the debate about ways that the action *might* unfold in the situation of concern. A simple example of one of the Conversations for Action from the field study is provided in Figure 4 below. The model provides a view of potential 'Conversations for Action' that might be necessary if the 'Collections Officer' in Figure 3 above were to undertake the action described for the activity 'Identify Customers experiencing difficulties'.

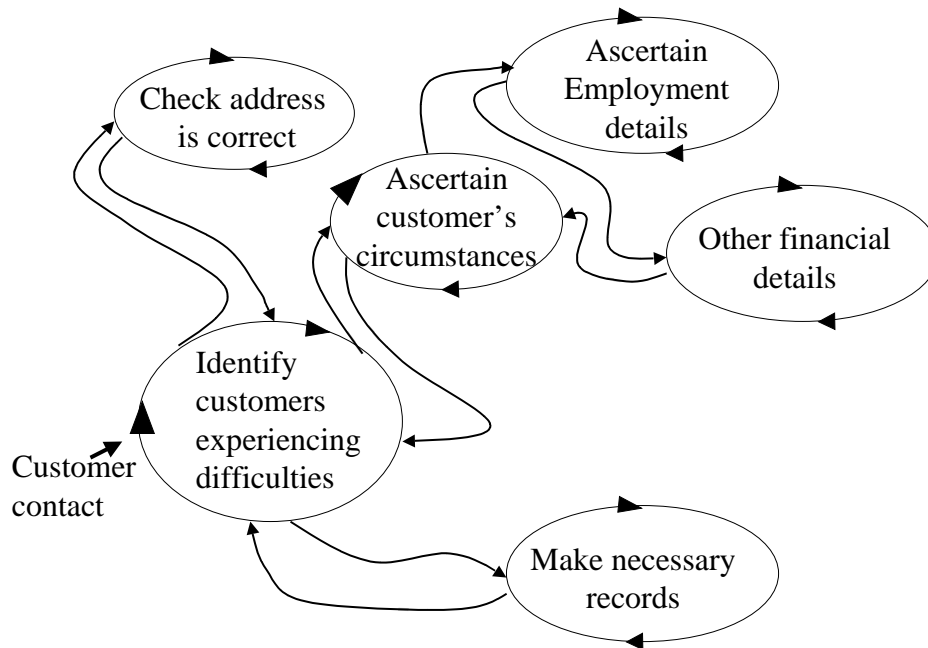


Figure 4: A Conversation for Action model expressing ideas for the Collections Officer's participation in the activity: *identify customers experiencing difficulties*, from the sub-system in Figure 3. (Reproduced with permission from Champion and Stowell, 2002).

Creating a view of potential 'Conversations for Action' acted as a catalyst for clients in thinking through the implications of their ideas for improvement. Such a model also provides a clear view of what action the client wishes to undertake within a particular situation and it is then relevant to begin creating ideas for a *servicing system*, or information system.

Ideas for Action: Designing the Servicing System

In a modern business environment, most information systems will be based on some form of technology. It is important at this stage of the design process that participants (both clients and developers) do not lose sight of the fact that the models do not represent a real world system. The models created at this stage of the design process are conceptualisations. One means of creating ideas for a *servicing system* is to construct models that provide a view of how a person might interact with technology during specific scenarios. Often UML notation (Pooley and Stevens, 1999) is used as a standard means of creating such models. Essentially, a software

developer uses his or her, experience and knowledge of the problem situation to choose certain elements of the requirements to act as unchanging 'objects' within the computer program (See Graham, 1998; Kristen, 1996; Pressman, 1997; Sommerville and Sawyer, 1997). These chosen 'objects' enable a developer to marshal the knowledge created about a situation into a format suited to constructing a computer program. The Class Diagram in Figure 5 below expresses the classes of objects that O' Callaghan, as an experienced software designer, considered would be necessary if building a prototype for a supporting information system for the scenario depicted in Figure 4 above, for the Collections Officers to trial. This model was again created through dialogue and debate, whilst being informed by the ideas expressed during the inquiry process so far.

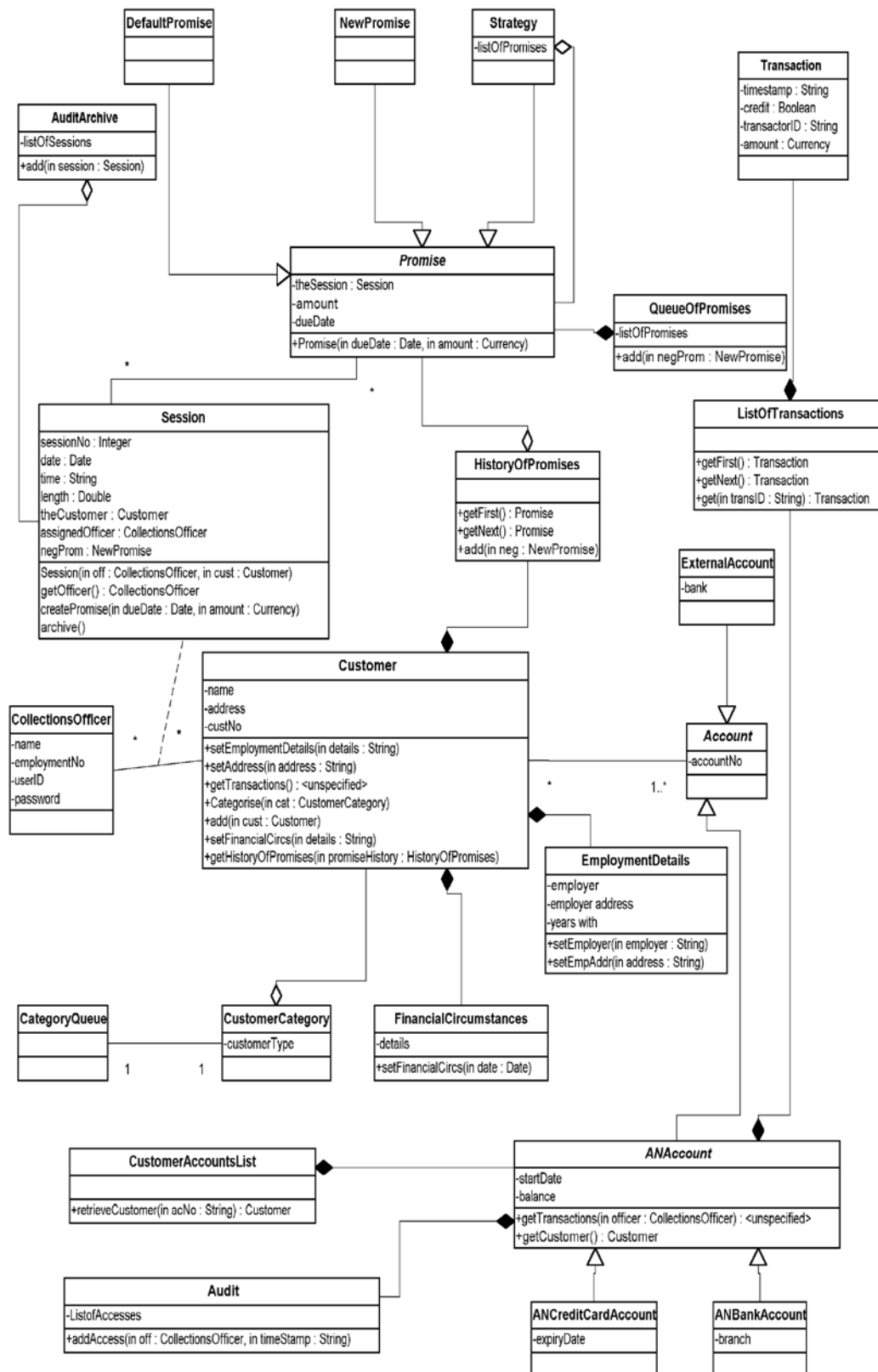


Figure 5: The class diagram created through discussion and debate from the models in Figures 2-4.

Some of the classes of objects apparent in Figure 5 can immediately be recognised from the models in Figures 2-4. For example, *Customer* and *Promises* and *Collections Officer* can all be easily found in the models created by the clients. Other classes of objects arise from the implications of creating a technology-based information system to support up to thirty Collections Officers undertaking the intended action simultaneously, with several hundreds of customers in debt at any point. For example, the class 'QueueOfPromises' arises from the necessity to organise some sort of check on promises that are current and awaiting to be completed or broken. Other classes of objects relate to the ongoing action over time. The class 'HistoryOfPromises' acknowledges that some customers may take some time to clear their debt and make several promises over a period of time and so build up some sort of 'Promise Record'. The difference between the models again reflects the change in focus that becomes necessary as the process of information system design proceeds. Clients tend to focus on their own particular areas of expertise, but the information technology is expected to support many different processes and individuals, simultaneously. Object classes, as expressed in the Class Diagram, provide a structural plan to guide the developers during the building of the software. Object-oriented methods enable the developer to manage the complexity of what is required, as first, key structural elements are identified and second, responsibilities for various operations can then be allocated. The trade-off, however, is that the dynamic behaviour of the system is more difficult to visualize than in other software development methods. The Collaboration Diagram shown below in Figure 6, expresses ideas for how a particular scenario might unfold, and was developed from the Conversation for Action model in Figure 4. The scenario expresses ideas on how the *servicing system* might support a Collections Officer negotiating a promise to pay with a customer experiencing difficulties, and updating address and employment details. The Collaboration Diagram, in providing a description of the interaction between instances of the specified classes, provides a dynamic view that both provides a "complementary picture" (Vickers, 1981) of how the software *might* operate and ensures the structural view will be appropriate. A collection of Collaboration Diagrams and Class Diagrams *together* provide a basic logical specification.

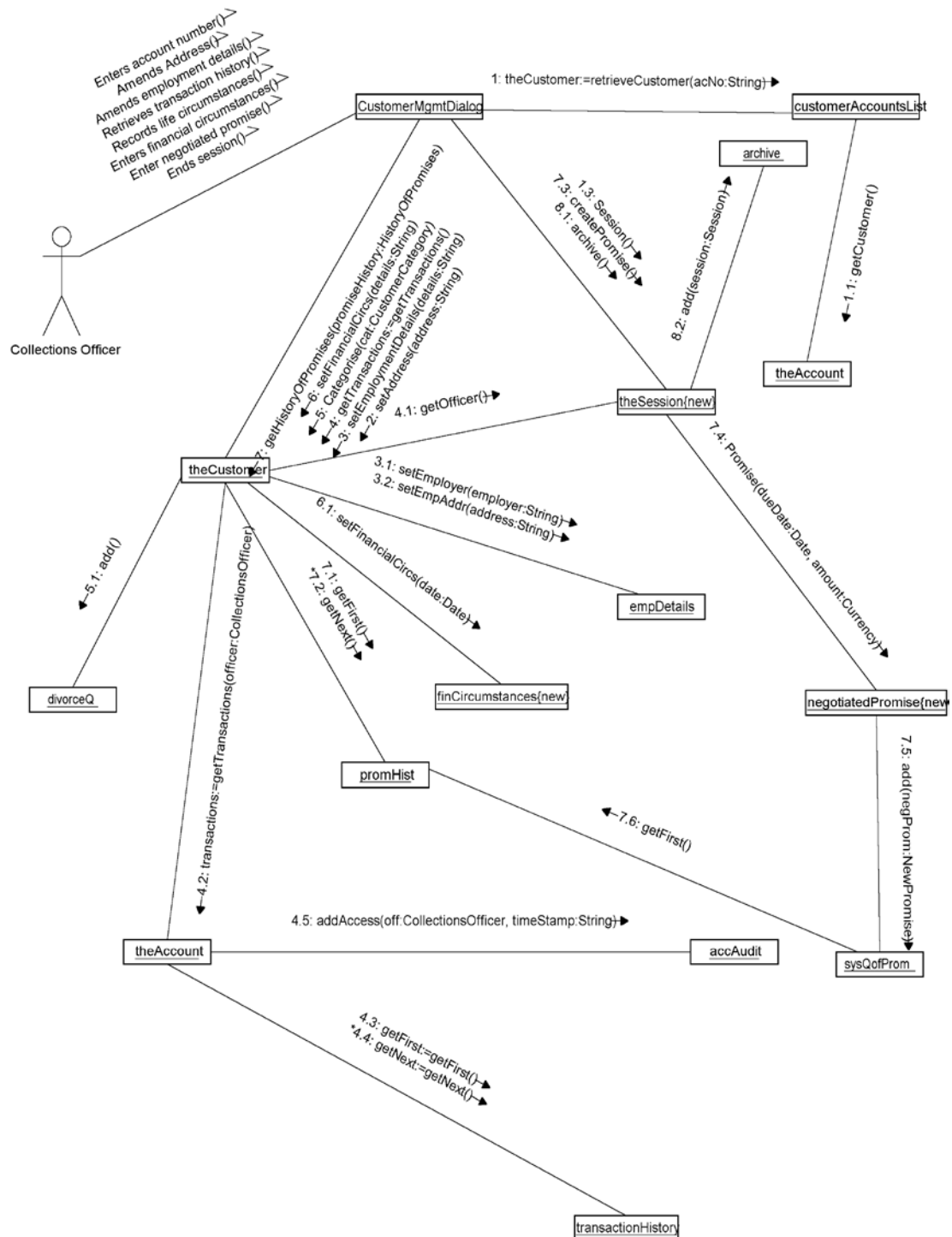


Figure 6: The Collaboration Diagram constructed through discussion and debate during the field study.

Once the devices of Class Diagrams and Collaboration Diagrams are employed in the inquiry, the focus of the inquiry is entirely on ordering the knowledge created about the situation into a suitable format for constructing a computer program. The Collaboration Diagram does provide some sense of the activity being undertaken, and contains notation that will act as guidance for any developer attempting to build a software application. The crucial and often unacknowledged point is that the status of these diagrams is that they too are sense-making devices; these devices inform the design work, they are not maps of what will be built. Even software developers that acknowledge this point (see Graham, 1998) undertake the design work as *if* the models were representative of reality. This is problematic, as once the logical specification is considered to be 'complete', the developers must then use this specification to guide the construction of the various aspects of the software needed to operationalise the technological aspects of the information system. This process is not a seamless progression to a computer program. Using the CLIC framework, up until this point, only ideas expressed by the clients for a potential information system will have been modelled in the Class Diagrams and Collaboration Diagrams. Other object classes such as those needed to provide the human-computer interface, and yet others needed to provide IT infra-structural services such as concurrency and persistence are needed. In addition, when designing the software, the developer will have to take into account a number of possibly conflicting forces such as resource constraints and non-functional requirements, which have meaning only for the virtual machine.

The path from information system specification to working software is a difficult one and, as such, has been the focus of attention of traditional Computer Science since its inception. In terms of object-orientation, UML provides a total of nine diagram types (i.e., another seven beyond those described above) to provide different views in the construction process and each developer has his or her, own preferences in the manner in which they go about such construction (Graham, 1998; Lai, 2000). Maintaining coherence between different models during the construction process is extremely difficult. Software engineering approaches focus on maintaining traceability and this is an ongoing and separate field of research within object-orientation (e.g., D'Souza and Wills, 1998). For example, in the Unified Software Development Process (Jacobson et al, 1999), Class and Collaboration Diagrams are created first as expressions of Use Cases (though in the field these are termed 'realisations' of Use Cases, yet another example of the underpinning functionalist approach to inquiry). All later models created, including the Implementation Model,

can be traced back to the Use Cases. The Use Case construct “ is used to define the behaviour of a system or other semantic entity without revealing the entity’s internal structure. Each Use Case specifies a sequence of actions, including variants, that the entity can perform, interacting with the actors of the entity” (OMG, 2001). When using such a method, for the purposes of design work, ‘Actors’ are idealised users in a role that can be human or non-human (e.g., other machines or even ‘time’) and which trigger responses from the software system. The methods of Computer Science often deliberately model any such ‘Actors’ in a lightweight way, because they are considered to be external to the software. The computer system is the real starting point of the modelling process in the Unified Software Development Process (USDP) and it is assumed from the outset that it will add value to its users. In short, a developer using only USDP would begin by thinking about how to support an action without conceptualising the purposeful action being served. In contrast, Class Diagrams and Collaboration Diagrams that have been derived from a process of navigation as described above, root the logical specification in the context of the business ‘problem’ and the action the clients wish to undertake.

The design features of the software application are logically consistent with the ideas for action expressed by the client. That is, a sense of *coherence* is maintained throughout the design process. The importance of coherence cannot be understated. If attempting to facilitate clients, who are not technical experts, but experts in their own domain, to lead the IS design process, it is important to provide a means of maintaining logical consistency and coherence throughout the inquiry process. Approaches to IS development that suggest the use of a plethora of tools and techniques fail to offer this support (Champion, 2001).

Ideas into Practice

The models used in the design phase are all employed as debating tools. The whole process of information system design and development is dynamic and the CLIC approach is intended to support a flexible and dynamic collaborative inquiry process that is a continuous learning process. The skills that such an approach can foster within an enterprise are much in demand (Willcocks et al, 1997). The research currently underway is aimed at extending the range of navigational tools to improve the flexibility of the approach from both the perspective of the developers and the clients.

The Contribution of the CLIC Framework

A key emphasis in the CLIC approach is the support offered to the clients in creating a shared appreciation of the problem situation at each stage of the design process. Some approaches to information systems design tend to focus on constructing some sort of *description* of the current situation from which the requirements for the technology are 'abstracted' (e.g. see Sommerville and Sawyer, 1997). By offering the CLIC framework, what we are suggesting is that the aim of the inquiry process is not to map requirements between models, but to facilitate coherent sense-making. Moving from conceptualising purposeful action, to conceptualising support for action, will require those involved in the inquiry to think clearly and be certain which design problem they are addressing at any particular moment. The *navigational* devices of Conditions of Satisfaction and Conversations for Action support those concerned in considering any issues associated with operationalising the ideas for purposeful action, so as to enable the movement from conceptualising potentially relevant purposeful action to conceptualising (designing) a serving information system.

The value in *navigating* from ideas for purposeful action to ideas for support is that the gap between the two different design problems is *emphasised*. To facilitate ease of understanding, the models expressing the initial ideas for action, those expressing ideas on how the action might be operationalised, and the models expressing the initial ideas for the serving information system are all activity-based models. This more fluid and coherent movement through the conceptualisations needed for information system design, facilitates those with a non-technical background remaining not only involved, but actively engaged and leading the design process (Champion, 2001). Coherence is maintained in two ways, first, by employing models and tools that offer an activity-based view of expressed ideas, but more importantly, by following the same guiding principles of inquiry throughout.

Ongoing Research

In keeping with our approach to research, we regard this work as part of a continuing dialogue and collaboration with our research hosts and the research community. The next challenge is to increase the flexibility and range of the approach, whilst still maintaining coherence throughout the entire process, so the research continues. An Action Research field study is currently being undertaken in a medium sized engineering company in order to develop other intellectual navigational devices that can be incorporated into the CLIC framework. We have previously published an example of navigation through the design process using an object-oriented case tool to create a specification for a serving information system, called *SOMATiK* which was

developed by Bezant Technologies (Graham, 1998) (See Champion and Stowell, 2002). The contribution of this paper is to offer a coherent 'framework for thinking', CLIC, to guide those involved in a process of information system design and to offer a further example of navigating the gap from client-expressed ideas for action to creating a specification for a technology-based information system, demonstrating the potential flexibility of this approach.

References

Ågerfalk, P.J., Goldkuhl, G. and Cronholm, S. (1999). "Information Systems Actability Engineering: Integrating Analysis of Business Processes and Usability Requirements", in, 4th International Workshop on the Language Action Perspective on Communication Modelling, Copenhagen, 12th-13th September, Download at: <http://www.hj.se/lap99/> (Accessed 10/10/02).

Avison, D.E., Wood-Harper, A.T., Vidgen, R.T. and Wood, J.R.G. (1998). "A Further Exploration into Information Systems Development: The Evolution of Multiview2" *Information Technology and People*. Vol. 11, No. 2, pp. 124-139.

Beyer, H. and Holtzblatt, K. (1998). *Contextual Design: Defining Customer-Centred Systems*, San Francisco, Morgan Kaufmann.

Bødker, S. (1996). "Creating Conditions for Participation: Conflicts and Resources in Systems Development", *Human Computer Interaction*, Vol. 11, pp. 215-236.

Boland, R.J. (1985). "Phenomenology: A Preferred Approach to Research on Information Systems", in *Research Methods in Information Systems*. E. Mumford, R.A. Hirschheim, G. Fitzgerald and A.T. Wood-Harper (eds), pp. 193-201, Amsterdam, North-Holland.

Champion, D. (2000). "A Report on an Action Research Field Study", *Systemist*. Vol. 22, No. 1, pp. 60-85.

Champion, D. (2001). *Navigating the Gap between Purposeful Action and a Serving Information System*, Unpublished Ph.D. Thesis, Department of Computer and Information Sciences, De Montfort University, Milton Keynes.

Champion, D. (2002) "PEArL: Establishing the Authenticity of Action research Field Studies" *Systemist Conference Edition*, G. Ragsdell and J. Wilby (eds), pp. 36-43, *Systemist*, Vol. 24, No. 3. UK.

Champion, D. and Stowell, F.A. (2001). "PEArL: A Systems Approach to Demonstrating Authenticity in Information System Design", *Journal of Information Technology*. Vol. 16, pp. 3-12.

Champion, D. and Stowell, F.A. (2002). "Navigating the Gap between Action and a Serving Information System", *Information Systems Frontiers*. Vol. 4, No. 3, pp. 273-285.

Champion, D. and Stowell, F.A. (2003). "Validating Action Research Field Studies: PEArL", *Systemic Practice and Action Research*. Vol. 16, No. 1, pp. 21-36.

Champion, D. and Stowell, F.A. (2004). "Delivering IT: Managing Change in the NHS", *Systemist*, Vol. 26, No. 1, pp. 57-62.

Checkland, P.B. (1981). *Systems Thinking, Systems Practice*. Chichester, John Wiley.

Checkland, P.B. (1983). "O.R. and the Systems Movement: Mappings and Conflicts", *Journal of Operational Research*, Vol. 34, No. 8, pp. 661-675.

Checkland, P.B. (1985). "From Optimizing to Learning: A Development of Systems Thinking for the 1990's", *Journal of the Operational Research Society*, Vol. 36, No. 9, pp. 757-767.

Checkland, P.B. (1995). "Model Validation in Soft Systems Practice", *Systems Research*. Vol. 12, No. 1, pp. 47-54.

Checkland, P.B. (1999). *Systems Thinking, Systems Practice with a 30 year Retrospective*. Chichester, John Wiley.

Checkland, P.B. and Holwell, S.E. (1998). *Information, Systems and Information Systems: Making Sense of the Field*. Chichester, John Wiley.

Checkland, P.B. and Scholes, J. (1990). *Soft Systems Methodology in Action*. Chichester, John Wiley.

Clement, A. (1994). "Computing at Work: Empowering Action by 'Low-level' Users", *Communication of the ACM*, Vol. 37, No. 1, pp. 52-63.

Dash, D.P. (1999). "Current Debates in Action Research", *Systemic Practice and Action Research*, Vol. 12, No. 5, pp. 457-492.

Davison, R.M, Martinson, M.G. and Kock, N. (2004). "Principles of Canonical Action Research" *Information Systems Journal*, Vol. 14, No. 1, pp. 65-86

Dietz, J.L.G., Goldkuhl, G., Lind, M. and van, Reijswoud, V.E. (1998). "The Communicative Action Paradigm for Business Modelling –A Research Agenda", in, *The Third International Workshop on the Language Action Perspective on Communication Modelling*, Download at: <http://www.hj.se/jibs/vits/lap98/> (Accessed: 10/10/01).

Doyle, K.G., Wood, J.R.G. and Wood-Harper, A.T. (1993). "Soft Systems and Systems Engineering: On the Use of Conceptual Models in Information System Development", *Journal of Information Systems*. Vol. 3, pp. 187-198.

D'Souza D.F. and Wills, A.C. (1998). *Objects, Components and Frameworks with UML: The Catalysis Approach*, Reading, Mass., Addison Wesley.

Ehn, P. (1989). *Work-Oriented Design of Computer Artefacts*. Hillsdale, NJ, Lawrence Erlbaum.

Flores, C.F. and Ludlow, J. (1981). "Doing and Speaking in the Office", in *DSS: Issues and Challenges*. G. Fick and R. Sprague (eds), London, Pergamon Press.

Gadamer, H.G. (1989). *Truth and Method*. (2nd Edition). (Translated by: J. Weinsheimer and D.G. Marshall), London, Sheed and Ward.

Graham, I. (1998). *Requirements Engineering and Rapid Development*. London, Addison Wesley.

Goldkuhl, G. and Roslinger, A. (1999). "Expanding the Scope –From Language Action to Generic Practice", in *The Fourth International Workshop on the Language Action Perspective on Communication Modelling*, Download at: <http://www.hj.se/lap99/> (Accessed: 25/6/00).

Guo, M., Wu, Z., Stowell, F.A. and Cowell, J. (2000). "AM/OO Modelling: Achievements and Problems", in *Information Systems –Research, Teaching and Practice, the 5th UKAIS Conference, 25-28th April, Cardiff*, pp. 312-321.

Harris, G.B. and Taylor, S. (1998). "Organisational Structuration: Interaction and Interrelation", Presented at: 14th. EGOS (European Group for Organizational Studies), Colloquium, Maastricht, The Netherlands. Download at: <http://www.workframe.com/WFI-CORP/OrgStrctHTML.htm> (Accessed 10/6/00).

Hirschheim, R., Klein H.K. and Lyytinen, K. (1995). *Information Systems Development and Data Modelling: Conceptual and Philosophical Foundations*. Cambridge, Cambridge University Press.

Holwell, S.E. (1997). "Soft Systems Methodology and its Role in Information Systems", Unpublished Ph.D Thesis. Lancaster University.

Jacobson I., J.Rumbaugh and Booch, G. (1999). *The Unified Software Development Process*. Reading, Mass, Addison Wesley

Kensing, F. and Blomberg, J. (1998). "Participatory Design: Issues and Concerns", *Computer Supported Cooperative Work*, Vol. 7, pp. 167-185.

Kensing, F. and Munk-Madsen, A. (1993). "Participatory Design: Structure in the Tool Box", *Communications of ACM*, Vol. 36, No. 4, pp. 78-85.

Kotonya, G. and Sommerville, I. (1992). "Viewpoints for Requirements Definition" *Software Engineering Journal*, Vol. 7, No. 6, pp. 375-387.

Kristen, G. (1996). *Object Orientation: The KISS Method: From Information Architecture to Information System*. Wokingham, Addison Wesley.

Lai, L. (2000). "An Integration of Systems Science Methods and Object-Oriented Analysis for Determining Organizational Information Requirements", *Systems Research and Behavioural Science*. Vol. 17, pp. 205-228.

Liang, Y., West, D. and Stowell, F.A. (1998). "An Interpretivist Approach to IS Definition using Object Modelling", *Information Systems Journal*. Vol. 8, pp. 163-180.

Mathiassen, L. and Nielsen, P.A. (2000). "Interaction and Transformation in SSM", *Systems Research and Behavioural Sciences*. Vol. 17. pp. 243-253.

Miles, R.K. (1988). "Combining 'Soft' and 'Hard' Systems Practice: Grafting or Embedding?", *Journal of Applied Systems Analysis*. Vol. 15, pp. 55-65.

OMG (2001) OMG Unified Modeling Language Specification version 1.4. Download at: <http://www.omg.org> (accessed 24/3/02).

Pooley, R. and Stevens, P. (1999). *Using UML: Software Engineering with Objects and Components*. Addison Wesley, Harlow.

Pressman, R.S. (1997). *Software Engineering: A Practitioner's Approach*. London, McGraw-Hill.

Reijswoud, V.E. van, Mulder, H.B.F. and Dietz, J.L. (1999). "Communicative Action-based Business Process and Information Systems Modelling with DEMO", *Journal of Information Systems*. Vol. 9, pp. 117-138.

Rodden, T., Rouncefield, M., Sommerville, I. and Viller, S. (2000). "Social Viewpoints on Legacy Systems" in *Systems Engineering for Business Process Change*, P. Henderson (ed), pp. 151-163, London, Springer.

Savage, A. and Mingers, J. (1996). "A Framework for Linking Soft Systems Methodology (SSM) and Jackson System Development (JSD)", *Information Systems Journal*. Vol. 6, pp. 109-129.

Schoop, M. (1998). "A Language-Action Perspective on Cooperative Documentation Systems", in, *The Third International Workshop on the Language Action Perspective on Communication Modelling*, Download at: <http://www.hj.se/jibs/vits/lap98/> (Accessed: 10/10/00).

Sommerville, I. and Sawyer, P. (1997). *Requirements Engineering: A Good Practice Guide*. Chichester, John Wiley and Sons.

Stowell, F.A. (1985). "Experiences with Soft Systems Methodology and Data Analysis", *Information Technology Training*. May, pp. 48-50.

Stowell, F.A. (1991). "Towards Client-Led Development of Information Systems", *Journal of Information Systems*. Vol. 1, pp. 173-189.

Stowell, F.A. (2000). "Modelling IS Requirements for Complex Systems", in, *Systems Modelling for Business Process Improvement*, D. Bustard, P. Kawalek and M. Norris, (eds), London, Artech, pp. 171-186.

Stowell, F.A. and West, D. (1994). *Client-Led Design: A Systemic Approach to Information System Definition*. London, McGraw-Hill.

Stowell, F.A., West, D. and Stansfield, M. (1997). "Action Research as a Framework for IS Research", in *Information Systems: An Emerging Discipline?*, J. Mingers and F.A. Stowell (eds), pp. 159-200, London, McGraw-Hill.

Vickers, G. (1965). *The Art of Judgement. A Study in Policy Making*. Chapman and Hall, London.

Vickers, G. (1981). "Some Implications of Systems Thinking", in *Systems Behaviour*, (3rd Edition), Open University Systems Group (eds), pp. 19-25, London, Harper and Row in association with the Open University Press.

Willcocks, L. Feeny, D. and Islei, G. (1997). *Managing IT as a Strategic Resource*. London, McGraw-Hill.

Williams, R. (2000). *On Christian Theology (Challenges in Contemporary Theology)*. Oxford. Blackwell.

Winograd, T. and Flores, F. (1987). *Understanding Computers and Cognition: A New Foundation for Design*. Reading, Addison Wesley.

Winter, M.C., Brown, D.H. and Checkland, P.B. (1995). "A Role for Soft Systems Methodology in Information Systems Development", *European Journal of Information Systems*. Vol. 4, pp. 130-142.