Published work
Fernandes, K. (2005) *Interactive situation modelling in knowledge intensive domains.* Working Paper. Department of Management Studies, University of York, York.

University of York
Department of Management Studies
Working Paper 14
December 2005

ISSN Number: 1743-4041

Interactive Situation Modelling in Knowledge Intensive Domains

Kieran Fernandes
Department of Management Studies

This paper is circulated for discussion purposes only and its contents should be considered preliminary.

Abstract

Interactive Situation Modelling (ISM) method, a semi-methodological approach, is proposed to tackle issues associated with modelling complex knowledge intensive domains, which cannot be easily modelled using traditional approaches. This paper presents the background and implementation of ISM within a complex domain, where synthesizing knowledge from various sources is critical, and is based on the principles of ethnography within a constructivist framework. Although the motivation for the reported work comes from the application presented in the paper, the actual scope of the paper covers a wide range of issues related to modelling complex systems. The author firstly reviews approaches used for modelling knowledge intensive domains, preceded by a brief discussion about two main issues: symmetry of ignorance and system behaviour, which are often confronted when applying modelling approaches to business domains. The ISM process is then characterized and critiqued with lessons from an exemplar presented to illustrate its effectiveness.

Interactive Situation Modelling in Knowledge Intensive Domains

Introduction

The major challenge to modelling complex, knowledge intensive domains is synthesizing knowledge from various sources; as such domains involve people from different backgrounds, with different knowledge and expertise, and work in different organizational roles (Oswald, 1996). This fact is compounded by problems, where organizational roles are not properly defined or are ill-structured. In most cases, the modellers are unfamiliar with working practices within the organization, and the process owners are not sure about what they want, or even what benefits they could obtain, from the domain being modelled. This phenomenon is termed as 'symmetry of ignorance', and clearly identifies the gap that exists between the modellers and the process owners (Rittel, 1984; Fischer, 1999). There exists literature ranging from traditional waterfall methods to the more recent ontology-based modelling, which attempt to bridge this symmetry of ignorance gap (Sun, 1999). Most of these approaches are process-based, and provide rigid and formal protocols, which attempt to guide the modelling process in a systematic and cost-effective manner. As these approaches regard the modelling activity as deterministic, mechanical and rational, they are not designed to cope with complex changing environments within today's businesses. In addition to this, most of these 'recipe-based' and methodological solutions assume that both the modeller and the process owner are capable of understanding each other's domains without loss of knowledge (Fernandes et. al, 2002). Given this knowledge gap, the main purpose of this paper is to address the basic problem faced while modelling such domains, "Is it possible to understand and analyze a complex knowledge intensive domain using a modelling process, which aims to bridge the symmetry of ignorance gap?" The paper thus has a much wider scope than just presenting a new modelling process within a complex environment. As the paper aims to tackle issues related to both the modeller and the process owner, it includes an overview of complex business domains and their characteristics. The paper is organized in the following manner: Section 2 summarizes the role of current approaches to modelling, which is preceded by a brief introduction to the concept on 'symmetry of ignorance' and 'system behaviour'. Section 3 presents the details of the

Interactive Situation Modelling process, followed by its implementation in Section 4. Discussion and conclusions about ISM is presented in sections 5 and 6.

Background

There is much ambiguity in literature about the use of the term 'systems modelling'. Some refer to this as purely a system understanding phenomenon, while others refer to it as a modelling activity. It is argued in this paper, that these two terms are inseparable and cannot be treated as two processes. Understanding and modelling, argues Oswald, are at once the product as well as the producer (Ostwald, 1996). He argues that only by modelling a domain can one understand the system, and by this recognition make the model is enriched, due to the modeller's understanding. Researchers have proposed constructing artefacts as one possible method for a co-evolutionary approach, where understanding and modelling activities can be conducted simultaneously. Given this nature of 'systems modelling', the process can be referred to as a semi-formalization of the development activity without reference to the particular characteristics of the product to be developed. The term semi-formalization indicates the author's bias towards non methodological approach due to the unpredictability within a business domain, and is termed as semi-methodological in this paper.

Unpredictability within such business domains has increased over the past 20 years, mainly due to increased globalization, which has made businesses enhance their productivity, have shorted product life cycles, increase product customization, and improved responsiveness to remain viable in the market (Deshmukh, 1993). In order to achieve these seemingly conflicting objectives, the trend in business domains has been towards integrating all domain functions (Stonier, 1997). Such integrated systems, also referred to as agile, lean or flexible depending on their level of integration, possess one common characteristic, viz. the ability to react to disturbances or changes (Stratton et. al, 2003; Naylor et. al, 1999). These disturbances could occur due to the introduction of new products (Calantone et. al, 2002), volatile consumer demands (Adamson, 2003), changing management objectives, or operating uncertainties such as process failures or in recent times industrial strikes (Ostwald, 1996). Successful implementation of these integrated systems is dependent on the correct reaction being taken to any disturbance. However, due to an incomplete understanding of the interactions among system components and their inability to exchange meaningful information between components, it has been difficult to make decisions about the design of these systems and prescribe effective operating strategies once the designs have been implemented.

Given these facts, it is imperative that knowledge intensive domains or complex systems should be modelled for a thorough understanding of the system with all its 'knowledge richness' incorporated in the model. It is quite interesting to note that the author has used the terms 'knowledge intensive domains' and 'complex systems' interchangeably, because in both expressions 'symmetry of ignorance' is an implicit part. This means that no part or entity of a system can provide sufficient information to actually or statistically predict the properties of other parts, which again relates to difficulties associated with modelling complex systems.

Symmetry of Ignorance

Typically users are domain experts and the modellers are technology experts. This concept in fact was first stated by Jordan (1969) and advocated by Ostwald (1996), where they state that a system is an interaction between what is 'out there' and how we can organize it 'in here'. The 'out there' concept is analogous to the 'domain experts' and the 'in here' is similar to the 'technology experts'. Here the domain experts know and understand the elements 'out there' in the system, while the technology experts know the technology application 'in here'. This phenomenon is termed as 'symmetry of ignorance' and shown in table 1 below.

Table 1. Symmetry of Ignorance

	Domain Knowledge	Systems Knowledge
Domain Experts	1	0
Systems Experts	0	1

It is interesting to note that Rittel bases his theory on the fact that 'knowledge is the basic foundation' block for a system. Checkland (1997; 1999), further states that this exhibited behaviour is the "coherency principle" which makes a system behave as a system. The author agrees with Rittel and Oswald's fact that "distributed expertise leads to symmetry of ignorance”. In most systems stakeholders have ‘part of the knowledge’ required to complete a task, which creates pockets of expertise within the system under observation. For example, in a simple machining workshop, the lathe operator possesses the knowledge to machine accurate and acceptable parts, while the design engineer possesses the knowledge to create appropriate CAD drawings based on the requirements of the customer and finally the computer system administrator possesses the knowledge to provide and maintain the computer infrastructure that is required to perform any general computing activity. This harmonization of activities within this system can result in the output – in this case the finished parts. However due to this distribution of knowledge, the system invariantly introduces the concept of distributed knowledge or expertise. This concept is also true for systems where modellers and users possess different sets of expertise.

System Behaviour and Systems Modelling

The term ‘system’ or ‘business domain’ is commonly used in an informal sense, as an abstraction from collections of real world objects or human activities in the world, for example, the ‘Enterprise Resource Planning (ERP) system’ or ‘accounting system’. Advents in technology and interaction within a business domain, has forced researchers to define system not in terms of machines, but in terms of complex entities and interaction. A common accepted definition of systems is “a group of units so combined as to form a whole and to operate in unison" (Blanchard et. al, 1998). However there are dozens of such similar definitions of systems in the literature that reflect a wide range of philosophical perspectives. Due to the brevity of this paper, details on this are not discussed, and the reader is referred to the work of Joslyn & Turchin for details (Joslyn et. al, 2005). The author defines a system or business domain as, *a collection of unprescribed independent entities within an observable or measurable boundary, displaying necessary operational behaviour or interaction to form a relationship, which can be observed by an observer, who may or may not be part of the domain*. The terms unprescribed and independent are used to define the fact that a business domain is a collection of unforced and independent entities. This implies that entities within such business domains are necessary and sufficient for the operation of the ‘whole’, but do not necessarily produce each other.

This type of system or business domain is referred to be synetairistikopoiesis in nature, a term deriving from the Greek works ‘synetairistiko’ (cooperative) and ‘poiesis’ (production). The process of modelling such complex knowledge intensive domains requires the modeller to synthesize knowledge from a synetairistikopoiesis domain. In an attempt to synthesize knowledge, method-based approaches have suggested the use of rigid and formal methods, which are designed to guide the modeller ‘synthesize’ knowledge in a systematic and cost-effective manner. For example, processes like the waterfall and spiral forces the modeller to adopt a defined sequence of steps to model and analyze a system (Royce, 1970). Using sequential activities like 'analyze', 'design', 'code' and 'test', a modeller focuses on gathering knowledge from a process manager, with the assumption that the entire project can be scoped at the very start. The spiral method on the other hand is designed for projects with undefined scope, giving the modeller an opportunity to redefine and perfect the ‘plan’ throughout the course of the project lifecycle. One thing obvious from both the waterfall and spiral methods is that they do not specify the way an actual artefact is designed and built. However its popularity has mainly been due to the fact that it gives the modeller the illusion of being orderly and well-defined. Overall these two approaches have proven to be quite successful in domains where complexity is quite low and the requirements of the system are extremely clear (Floyd, 1984; Buddle et. al, 1992), and hence termed as evolvement methods.

The main shortcoming in evolvement approaches is that it puts the ‘process owners’ outside the ‘systems modelling’ process. As an outsider, the real process owners seldom have any authority to contribute to the development activity. As the users’ requirements are formalized by non-experts, information and knowledge about this system is unavoidably lost, as this is presented to the modellers in the form of technical reports and formal documents. This symmetry of ignorance creates models that do not represent the system. In addition to this, no social aspect associated with a system has been considered, with process owners being the ‘outsiders’ and developers the ‘insiders’.

Improvements to such evolvement methods were proposed as part of a ‘user-centred’ philosophy, referred to as ‘contributory methods’ in this paper. Here modellers took on the role of a tutor while the process owners took on the role of tutees. These ‘teacher-student’ concepts claimed to include the process owners in the decision making process, but still considered them to be ‘outside’ the system. The contributory methods considered prioritization, project monitoring and risk assessment core to system modelling. The PICTIVE (Plastic Interface for Collaborative Technology Initiatives through Video Exploration) method used video films, as a method for exploration, allowing the modeller to design system interfaces and judge its behaviour with 'limited contribution' from the process owner (Muller 1991, Muller et. al, 1991). The PICTIVE method due to its nature of operation, did not consider system development time to be a major factor. This concept of 'deadline pressure' was later on termed as 'timebox', by Scott Shultz of DuPont, as a key component of Rapid Application Development (RAD) method (Kerr et. al, 1994). The concept of timeboxing was aimed to "grab" system problems "by the horns" and wrestle it to the ground (Zahniser, 2005), thus adding elements of scheduling to the contributory methods. RAD and DSDM (Dynamic Systems Development Methodology) methods consider schedule and scalability as the most important aspect of a project (Martin, 1991), with focus on timeboxing and time to market as the key driving motivation (Stapleton, 1997). This view on time and scalability, limited their focus on the process owner as more concentration was put in managing the system requirements to time. The Joint Application Development (JAD) method overcomes some of this limitation, by selling itself as a 'management process' rather than a technique like RAD and DSDM. The JAD approach requires the modeller to define the roles of all the people involved in the system, with specific emphasis on the process manager (Jackson et. al, 1997), thus attempting to incorporate some of the social factors.

As can be seen, the contributory methods are focused to 'break' the barriers between the modellers and the process owners. In short these methods do recognize the existence of the symmetry of ignorance phenomenon, and attempt to eliminate this by allowing the process owner to be part of the decision making process. The focus is on encouraging dialogue between the users and the developers, and expecting the outcome of this to be a formal system specification. The intention of contributory methods is to distribute the knowledge between the process owners and the developers using mock-ups, design scenarios, prototypes etc. Contributory methods as can be seen are cooperative but lack social focus.

Soft System Method (SSM) is one of the first approaches to consider systems modelling as a social process, rather than a pure technical activity, using 'holons' (holos = whole) and notations, to model and describe human activities in a system.

The author catalogues methods that consider modelling as a social process, to be 'social methods'. SSM focuses on 'real world modelling', but does not consider aspects of computer artefacts as a possible method to create models. Ostwald's Evolutionary Artefact Approach (EAA) (1996), like SSM not only focuses on cross functional teams, expert judgment and observation analysis, but also considers prototypes and artefacts central to systems understanding. The EAA provides the modeller with computer based support for modelling and process understanding. This approach argues that knowledge acquisition is an evolutionary approach, and hence requires prototypes that will give non process experts, a tool to elicit knowledge. There are several limitations to this approach, firstly it did not explicitly state the behaviour of the 'system under consideration' and secondly, it did not prescribe a process for constructing such artefacts.

One observation that can be made from the SSM and EAA approaches is that they do not prescribe any method to represent the system behaviour or process observations. To overcome this limitation Jacobson (1994, 1997) proposed precise notations based on object oriented concepts. The philosophy of Object Oriented Software Engineering (OOSE) considers behaviour within systems as deterministic and predictable, where knowledge within systems can be expressed in terms of data (entity) and tasks (operations). Knowledge capture is conducted using a sequence of standardized notations, called Unified Modelling Language (UML) (Booch et. al, 1998, Tsai et. al, 2004, Koo et. al, 2003). OOSE has proved to be perhaps one of the most successful methods for system analysis, especially within the software development community.

This shift in paradigm to have more participation has empowered the process owners to contribute to the modelling activity. These social methods rely on different philosophies, principles and concepts, but fail to facilitate interaction between developers and users. These methods, however, provide better interaction than evolvement and contributing methods, where 'fixed activity pattern' was the main agenda. The social methods tackle the issues of team working but do not explicitly state a proposal to achieve this.

This gap has been filled by two specific approaches, which the author terms as team methods. The CSCW (Computer-Supported Cooperative Work) (Greif, 1988; Sigchi 94, Baecker, 1992) and FAOR (Functional Analysis of Office Requirements) (Schaefer et. al, 1988) methods, consider the participation of team members and team working critical to understanding complex systems. CSCW clearly emphasizes on the need to conduct cooperative work, but does not prescribe a model or methodology for bringing together people of different backgrounds to define a common view.

One of the major limitations of these approaches is that, they do not define the role of actors within the system. Understanding the behaviour of the process owner is critical to modelling a system, as this dictates the behaviour of the system. Empirical Modelling (EM) (Beynon, 1985, Rungrattanaubol, 2002) focuses on cognition and human interaction as central to modelling and is hence termed by the author as interaction methods. In the case of interaction methods, both the users and developers have a mutual learning experience and this symbiosis bridges the symmetry of ignorance. In short, the developers have to understand the working methods and procedures of the process owners and vice versa, the users should at least understand the possible feasible technical alternatives they can pursue. In addition to this the EM approach considers circumscription and construction from deduction (construal) fundamental to its philosophy. These systems clearly reflect the change in attitude of SSD from a practical process to a more social process. One of the major limitations of this approach however, is the inability of these methods to distinguish between 'relevant' and 'non-relevant' entities within the system.

Observation from the interaction and other approaches is the need for business domains to react to the business surroundings or environments they are placed in. This means that there is a need to understand the type of system they operate in. In short, future modelling approaches must consider holistic approaches to 'system under observations' rather than just 'general system analysis,' like approaches followed by social, team and interactive methods. There is an also a clear need for an approach that is capable of defining the type of system before any analysis can proceed. Also modelling techniques should be flexible enough to model knowledge intensive domains based on instances of situations rather than just precise notations as proposed by OOSE. The main element of this new technique should be based on 'observation of system' rather than 'pure documentation or partial participants of process owners and managers'. The modelling technique should also be able to translate observations into multiple view points with all the knowledge richness embedded into the computer artefact. This will only be possible if methodologies adopt a more 'observation and experimentation approach' to modelling rather than fixed or open ended modelling techniques. This means that modelling techniques should focus on semi-methodological solutions as proposed in this paper, rather than formal, deterministic modelling methods, or for that matter, open-ended approaches like EM.

Interactive Situation Modelling in Knowledge Intensive Domains

Knowledge intensive domains or complex systems, like a manufacturing unit, are rich in detail and information, where process owners have the ability to interact and add knowledge to the system. In such systems process owners and other agents exhibit a remarkable amount of versatility (Newell, 1990). The author, in section 2 has defined such systems to be synetairistikopoiesis in nature. The interaction between the unprescribed and independent entities within such a system gives rise to the system boundary. For example, consider a manufacturing paint shop. The painters and the associated painting guns and paint form the unprescribed and independent entities. These entities are individual units, e.g. human being, painting gun, paint, etc and hence are independently identifiable. As these units are unique parts of the system and are necessary and sufficient for the operation of the system, they are not prearranged and hence are termed as unprescribed in this paper. The motivating concern for developing these system concepts was that researchers draw boundaries around systems, as a means of both ecological and political identification, even when systems lack spatial and temporal boundaries. Rather than forcing identities on such systems to make analysis, understanding, and planning easier, the author advocates developing new concepts that force the reader to recognize the complexities, contradictions, and uncertainties that exist.

From this definition of system, proposed in this paper, the key questions that arise are: How are relations established and what creates the pattern of organization that describes such a system? Synetairistikopoietic systems provide a new interpretation of a systems boundary (it could be a manufacturing unit). It states that a firm's boundary consists of boundary elements that include various roles and functions. The roles and functions can be embedded in individual persons, groups, or information and communication systems. For example, painting teams, client groups or a project manager in the painting unit described above. In his role, the 'paint shop project manager' interacts with the clients, acquires experiences and accumulates new knowledge and best practices into the painting unit. The role is continually produced by the synetairistiko (cooperative) nature of the production (poietic) system. Part of the role can be embedded in an information system that helps plan and control the painting project and communicate it with the client.

People cannot constitute the boundaries in a synetairistikopoietic system. They act as physical agents in the manufacturing units. For example, the painter cannot form the boundary of the paining unit. Hence the interpretation of boundaries as roles and functions enable firms to learn and evolve as non-physical synetairistikopoietic systems. The boundary is not a passive result of other external powers, such as line of inclusion and exclusion between firms and its environment. Instead it consists of elements that can be identified and whose behaviour can be observed or measured. The entities can be permanent (like the paint gun) or temporary (like a shift painter), which means that it is possible to create new kinds of boundary elements that supplement each other and increase the absorption capability of the boundary. The boundary of the synetairistikopoietic system has a two-fold role. It separates the entities from the environment and it connects and exposes the entities to the environment and thus enables interaction. In addition to this synetairistikopoietic system have self-determination as an intrinsic property and exhibit distributed control.

Information in a synetairistikopoietic system is distributed and decentralized. This means that information resides as an integral part of the entity and is distributed. However this information can be gathered for analysis purposes. The system, therefore can maintain organizational openness or closure by controlling the information flow. In addition to this synetairistikopoietic systems are evolutionary in nature and have potentially infinite temporal trajectories. This means that the focus of synetairistikopoietic systems is evolutionary based, i.e. the system is continually changing. This does not mean that the system is consistently changing. Characteristic Tendencies of Synetairistikopoietic Systems is illustrated in Figure 1.

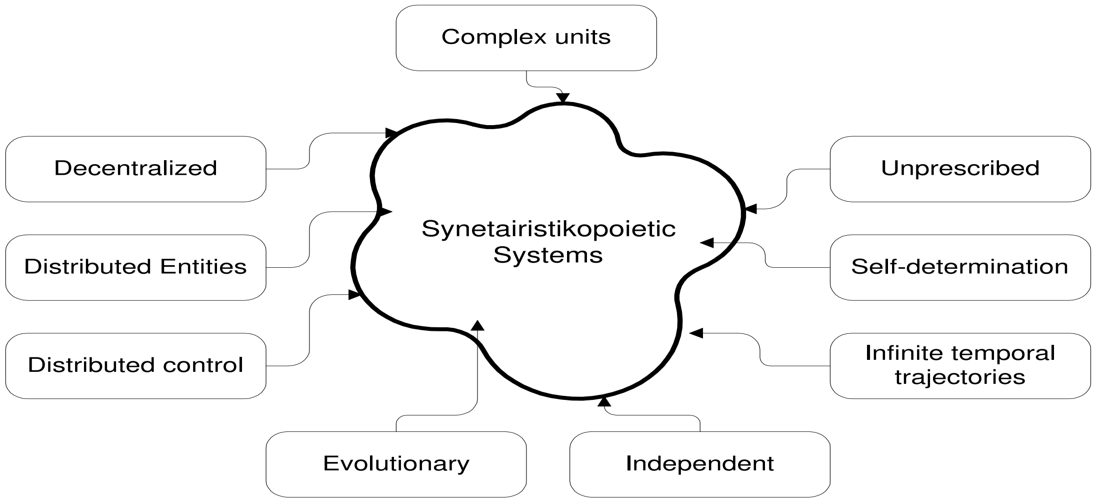

Figure 1. Characteristic Tendencies of Synetairistikopoietic Systems

One question that emerges from this discussion is "How do I determine if a system is Synetairistikopoietic?" The author uses a set of simple guidelines using if-then or se então logic to determine the nature of a system. This has now been computerized and includes a series of simple if-then questions that the modeller (or any other interested person) can use to determine if a system is synetairistikopoietic.

The se então is an integral part of the ISM process. As stated in earlier sections, ISM is a semi-methodological approach, as it does not suggest a fixed method of process, but relies on a series of concepts. As outputs from concepts contribute to different phases, this technique can also be referred to as a seeding technique. Even though this method is not a rigid method, there are certain rules the ISM viewpoint follows:

- An initial artefact can be developed by the modeller based on his non-expert observations or based on his/her previous experience(s);
- Observation can be regarded as a virtual experiment where a corresponding response can be a notification to the computer artefact;
- Flux between experimentation and observation can be used to determine the congruence between the two models;
- The aim of the experimentation is to replicate the behaviour of the real world with the computer artefact;

- The computer artefact can be open-ended till a point that the number of seeding occurred refines the artefact to behave congruently like the real-world with an obvious time gap;
- The time gap between observations can be discarded till the modeller either replicates and verifies the process owner's perspective or constitutes the self-emergence of a process owner.

These 6 conventions of ISM are part of the ISM semi-methodological process and shown in Figure 2 below:

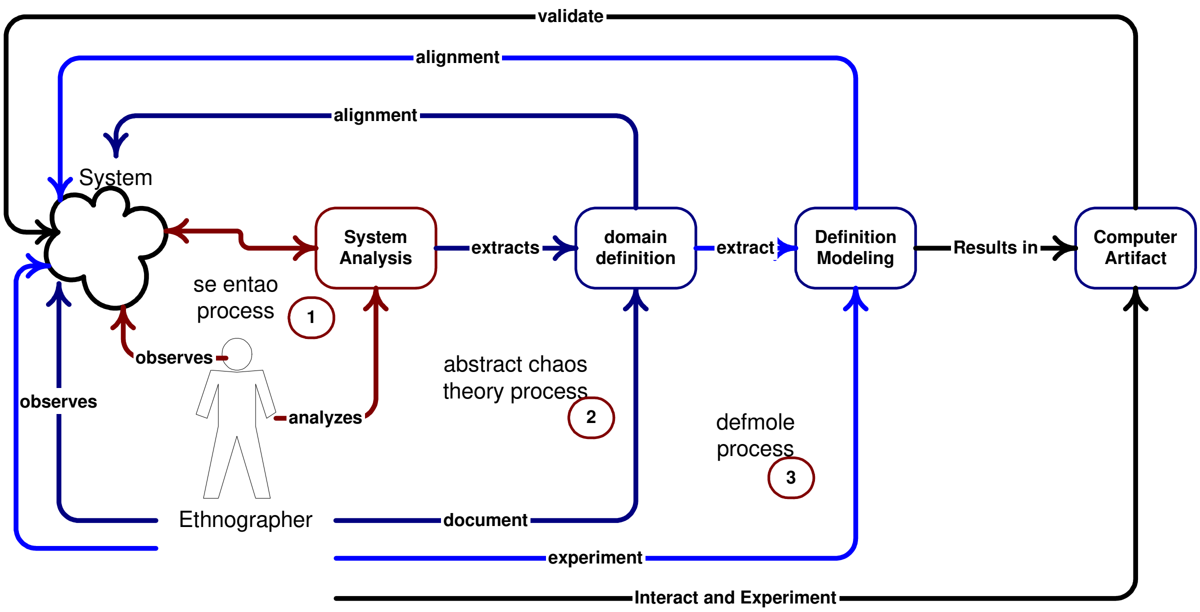

Figure 2. ISM Semi-Methodological Process

Se entao Process

The main philosophical foundation of ISM is based on the modeller's understanding and interaction with the business domain. This means that observation and interaction of situations within the domain is central to the ISM modelling concept. Scenarios involve different situations and serve different goals, but have one important thing in common: a coherent sequence of actions, called situated activities (Suchman, 1987). Situated activities are 'observable or measurable' interactions. Situated activities are essential as they are constructed by interaction between the actor and their surrounding. It is important to note that an action is only situated if it involves conscious reference to the context and choice of action. This means that an action is not situated if it takes the form of a prescribed response or if it is an unconscious automatic response. Within a situated activity, as described by Suchman, each situated activity is a dynamic interaction with the actor's external environment and is very difficult to prescribe in advance (Suchman, 1987).

This in fact is the main reason why formal algorithms find it hard to prescribe a situated activity through which a human agent can interact with a specific environment through a preconceived, fixed and well-defined method or rules. From this perspective the concept of situated activity is necessary in supporting systems modelling and forms the basis of the term Interactive Situation Modelling. It is obvious from this discussion that situated activity is more versatile than a formalized activity for solving problems in the real world. The most significant reason is because it is centred on human agents rather than on strict laws, algorithms or so called 'plans' arising from a particular account of the world. In fact Suchman clearly states that most human agents simply use such 'plans' as a resource rather than a source of control in everyday life. In any situated activity, it is most appropriate to give human agents sovereignty to solve problems, rather than binding them by some stated rules. By reflecting on the surrounding resources, such as known information, individual experience and knowledge, and the current state of the environment, human agents can conduct reasoning in their minds to 'preview' proper results, and can consequently undertake corresponding actions towards a new expected or unexpected situation. Each action undertaken promptly and tacitly affects both the internal mind and the external environment, which leads to a new situation and concurrently, enables the situated activity to progress. This means situated activity instead of prescribing activities and specifying the stimulus-response relations between human agents and their environment, highlighting the importance of human agents coping with diverse situation in the real world by taking the context into account.

Abstract Chaos Theory Process

As can be seen from Figure 1, the ethnographer observes the system and determines whether the complex system is synetairistikopoietic using the se entao process. Observing and understanding complex systems is a challenging task. For example an ethnographer observing a busy highway observes a number of cars passing on the lanes. He also observes the cars stopping when the signals turn red and moving when they turn green. In addition to this he observes pedestrians walking on the pavement and talking or performing some sort of action. The ethnographer is unable to comprehend all this knowledge and analyze it. However one thing is obvious: that change in one observable causes an effect on other observables. For example the traffic light turning red causes the motorists to stop. This cause and effect phenomenon in this paper is termed as cause-and-effect relationship or abstract chaos theory process. In ISM the connection between the computer artefact and the referent is established through the realization of the similarity between the cause-and-effect relationship in the artefact and the referent. This realization is only possible if the association between change and consequence is reliable and timely – if not 'immediate'. In order to relate interaction with the artefact and interaction with the referent in this way, it is also important to know which of the observables is sensible to change directly.

In abstract chaos theory process the ethnographer observers the synetairistikopoiestic system and uses a semi-structural notation to represent his initial analysis, as shown below:

```
{
actor actor_name;                                          Actor Information
        variables                                          Variable Information
                {       referred list of referred variables;
                        fixed list of fixed variables;
                        derived list of derived variables;
}
observations                                               Observations
                {
                        O1  |  P
                        O2  |  Q
                        O3  |  R
                        .   |  .
                        .   |  .
                        .   |  .
                            |  S
        }
}
```

As can be seen from the abstract chaos notation there are three sections to the notation. The first is the actor; the second is the variables and finally the observations. An actor instance is identified by its name and parameter list, in the same way that an entity instance is identified by the name of the generic entity description and the parameter list. As can be seen this notation details:

- The observables whose values can act as stimuli for an actor (referred);
- Those whose existence is intrinsically associated with the actor (fixed);
- Those indivisible relationships between observables that are characteristic of the interface between the actor and its system (derived);

In the interest of brevity, the author is unable to present additional details about referred, fixed and derived variables, and is referred to the works of Fernandes (2005) and Beynon (1995) for details. The observations are represented using the 'Fitch-Style' system, details of which can be referred from literature (Fitch 1952; Fitch 1975). Firstly, it can be seen that the ISM concept is very actor oriented, as observation is central to this modelling concept. The above notation can be explained informally as observed by the ethnographer. As a notation on its own merit this is still under evolution and can provide potential for future research activities. The main concept in the abstract chaos theory process is to record what the ethnographer regards as the relevant features (observables) of the synetairistikopoiesis system. In the course of identifying these observables and from experience of the subject domain, the ethnographer also needs to identify the 'source of change', the so-called state-changing actor of the system. The observables are then grouped around the actor to which they are related. The notation is termed as abstract chaos notation. As can be seen the technique represents instances of observation and is also capable of representing instantiated behaviour.

Dependency Modelling Process (Defmole)

A dependency is a special kind of relationship between two elements, specifying that a change in one element may affect another element that uses it, but not necessarily the reverse (Rungrattanaubol, 2002; Cartwright, 1994). Dependency in a synetairistikopoiesis system is motivated by trying to use the computer to help problem solving. The key concept observed at the abstract chaos theory process, was that ethnographers were good in identifying global constraints to were to be achieved, but were unable to efficiently carry out any computation and searching to find a solution. ISM motivates users to explore these constraints. This exploration can be carried out prior to any knowledge of global constraints, and with no specific problem in mind. The formulation of problems and identification of constraints relies on the experimentation done by users to identify patterns of dependencies. Assume a general representation of a = b + c. This means that the value of a is dependent on b and c. Any change in values of either b or c will make a change in the value or a. This concept is termed in general as dependency. It can be seen that there are three distinct structures within this representation, viz.: dependency constitution, position constitution and contextual constitution. Dependency constitution is the pattern of which observables are related to each other. Position constitution refers to the physical organization and arrangement of definitions in the defmole, and contextual constitution is the grouping of definitions according to different contexts for observations and interpretation. From the abstract chaos notation the modeller (not necessarily the ethnographer) has to consider all three of these configurations before deriving with defmole. The dependency structure in ISM is determined by definitions of observations derived from abstract chaos notations. The location constitution is determined by organization of definitions and contextual structure using dependency diagrams, as shown in Figure 2.

a = b + c;
b = e;
c = b + d;

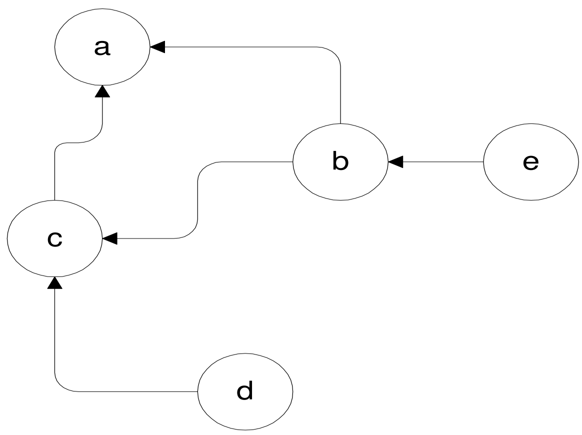

Figure 2. Example for Defmole constitution

This is a pattern of which observables are related to each other, e.g. a is dependent on values of b and c; c is dependent on values of b and d. Here in dependency constitution, the modeller represents the relationship of the observables and their relations. Most of these are in fact derived from the abstract chaos notation. From figure 2 we can see that the relationship is represented as a then b and finally c. This position constitution gives the modeller a better understanding of the instances of observation. Using graphical representation of the definitions gives the defmole proper contextual constitution as can be seen from the line diagram in the right hand of the figure 2.

Finally the outcomes of steps 3.2 and 3.3 were modelled using a modified version of the EDEN toolkit called ADAM. EDEN is the Engine for DEfinitive Notations and was designed to model and build computer artefacts (Sun, 1999). The technical working of the ADAM and EDEN toolkit it not within the scope of this paper, and the reader is referred to the works of Fernandes (2005) and Sun (1999) for details on the two tools respectively.

Lessons from the Application of ISM – Case Study

The ISM process was implemented in a company called Biddle Air Systems. Biddle Air Systems is a well recognized business, specializing in climate control and air-curtain technologies, with a manufacturing facility in the UK and sales offices in France, Belgium and Germany. Document Management Process (DMP) at BAS is the practice, where a major procedural or policy change is required to be done in compliance with ISO 9000 protocols. This process at BAS was a complex knowledge intensive domain, which had two process managers interacting with several process owners. Both managers had to respond to 'change requests' in a short period of time and had to operate under very tight budgets, compounded with the fact that one of the managers was a part-time employee. Some of the problems observed by the ethnographer (the author) were loss of information, lack of adopting procedures, slow response times, high numbers of non-compliance errors and duplication of activities. The problems within this domain were so severe that one of the directors stated "the lost-time due to delay getting correct documents for the job would be the downfall of the company". This fact could be overvalued; however the importance of solving this problem could not be overlooked. The author modelled the domain using ISM principles and the developed software tool – adam. Some of the key observations of the developed artefacts were:

- Artefacts in ISM had Setting: Entities within the domain could explicitly describe their state.
- Artefacts in ISM were actor based.
- Artefacts in ISM displayed goals.
- Artefact in ISM had plots as these included sequence of actions and events, things that actors did, things that happened to them, changes in the circumstances of the setting and so forth.

From its application at BAS, it is clear that the ethnographer was able to capture and model complex knowledge intensive domains like the DMP using ISM, thus bridging the 'symmetry of ignorance' gap. The resultant artefact showed a concrete hypothesis about what the people will do, think and experience, thus evoking the concept of reflection-in-action. The semi-methodological ethnographic nature of ISM, ensured the ethnographer experiences, i.e. the 'felt-path' of the process owners.

Discussion

Interactive Situation Modelling is a new approach to model complex systems using computers. The concept of a semi-methodological solution reflects the fact that its construction proceeds incrementally in association with interaction both with the model and with the external state or situation to which it refers. The description 'ethnography' in ISM reflects the emphasis that is placed upon experiment and observation in the process. As was seen in sections 2 and 3, ISM is best conceived as a situated activity, in which the construction of the model proceeds in parallel with the development of insight into the relevant domain. The novelty of the approach rests upon using representations that are experiential rather than logical in character. That is to say, the ISM is a source of experiences that serve to represent experiences of its referent. Experimental scientists employ similar principles for knowledge representation when they create physical artefacts to express their understanding of a phenomenon and by designers when they construct an engineering prototype. In ISM, beliefs about behaviour are represented in a characteristic and distinctive manner, via abstract chaos notations and defmole. In comprehending and developing complex systems, the emphasis is upon ethnography and establishing how agents and their behaviour is shown especially in knowledge intensive domains, as explained in the DMP process at BAS. The use of se entao process for hypothesizing agents and then establishing how these are involved in framing behaviour is another significant knowledge contribution.

The construction of an artefact based on abstract chaos notations and the defmole using patterns is closely linked to comprehension, in that the interactive activity of making sense of the business domain and the development and validation of the ISM are interleaved and interdependent. Once a reliable correspondence between states and transitions of the system under observation is established, the user (process managers, reengineers, etc) can enhance their domain understanding. ISM and conventional modelling of complex systems are sharply distinguished by the character of their underlying beliefs about behaviour. In ISM, the modeller (ethnographer) is not making indisputable assertions about objective facts, but attempting to imitate their own experience in a computer model. This activity is bold in character than formal specification, and is better suited to synetairistikopoiesis systems. Conventional computer-based modelling trades in abstractions and conceptions of systems that are far too sophisticated in experiential terms to respect the nuances of belief. To adopt such non-logicist perspective for an enormous body, empirical evidence is needed to support beliefs about the objectivity of different agent perceptions, or endorse such a sophisticated conception as autopoiesis (Mingers, 1995). Even objects and processes are richly informed by experience: objects through familiarity built up through acquaintance with many occurrences, and processes through repeated sequences of events in which such issues as commonality of perception, extending to an awareness of “location in an abstract process” on the part of some of the participants, play an essential part. In conception and application, ISM is unlike a methodology, and - like the so-called ‘scientific method’ - is primarily concerned with exploratory activity rather than systematic problem-solving procedures. Its basic presumption is that there is always experiential substance in the subject of which the modeller is unaware, that experience of the subject is a key element in acquiring insight, and that there can be no guarantee that such insight will lead to the attainment of preconceived goals. This attitude calls into question how far re-engineering can be addressed by proposing methodologies, standards and ontologies. It also engages with issues that conventional approaches to modelling typically fail to address. Where it can be successfully applied, ISM leads to the construction of models that can still be useful in singular conditions, and can serve as a way to disclose where such singularity may arise. It can also be used to investigate matters of agency and observation, such as what knowledge, perceptions and skills are exercised in a role.

Conclusion

ISM is a new and radically different approach to complex systems modelling. Its primary focus is on comprehension, and upon the use of computer-based interactive situation models that represent the way in which aspects of system behaviour are constructed. Such use of the computer to create experiential representations of knowledge is associated with both logicist and non-logicist stance with far-reaching implications for the conceptual framework surrounding complex systems design. In this process, concepts such as propositions, processes and objects lose their primitive status and the primary focus of attention shifts from methodologies, procedures and behaviours to situated activity that engages with state-as-experienced through experiment and observation.

Acknowledgement: The author is particularly indebted to Biddle Air Systems, for supporting this research. The author's work has been strongly influenced by valuable input from Prof. Vinesh Raja, Dr. Steve Russ, Dr. Meurig Benyon and Dr. Ashley Ward.

References

- Adamson K. An examination of consumer demand in the secondary niche market for fuel cell vehicles in Europe. International Journal of Hydrogen Energy, Vol. 28, No. 7, pp. 771-780, 2003.
- Baecker R. Readings in Groupware and Computer Supported Cooperative Work: Assisting Human-Human Collaboration. Morgan Kaufmann, 1992.
- Beynon W. Definitive notations for interaction. Proceedings of Human Computer Interaction Conference, ed. Johnson & Cook. Cambridge University Press, pp. 23-34, 1985.
- Blanchard B, & Fabrycky W. Systems Engineering and Analysis. Prentice Hall, 1998.
- Booch G, Rumbaugh J, & Jacobson I. The Unified Modelling Language Guide. Addison-Wesley, 1998.
- Buddle R, Kautz K, & Kuhlenkamp. Prototyping: An Approach to Evolutionary System Development. Springer- Verlag, 1992.
- Calantone R, Droge C, & Vickery S. Investigating the manufacturing–marketing interface in new product development: does context affect the strength of relationships? Journal of Operations Management, Vol. 20, No. 3, pp. 273-287, 2002.
- Cartwright A. Application of Definitive Scripts to Computer Aided Conceptual Design. PhD thesis, University of Warwick, 1994.
- Checkland P, & Holwell S. Information, Systems and Information Systems, John Wiley & Sons, 1997.
- Checkland P. Soft Systems Methodology in Action. John Wiley & Sons, 1999.
- Deshmukh A. Complexity and Chaos in Manufacturing Systems. PhD Thesis, Purdue University, 1993.
- Fernandes K, Raja V. A Structured Modelling Approach to Business Domains in Aerospace Industries. International Journal of Aerospace Management, Vol. 1, No. 4, pp. 353-366, 2002.
- Fernandes K. Interactive Situation Modelling in Knowledge Intensive Domains. Doctor of Philosophy Thesis, University of Warwick, 2005.

- Fischer, G. Symmetry of Ignorance, Social Creativity, and Meta-Design. Proceedings of the third Creativity & Cognition Conference, ACM Press, New York, pp. 116 – 123, 1999.
- Fitch F. Symbolic Logic: An Introduction. Ronald Press, 1952.
- Fitch F. The Logical Enterprise. Yale University Press, 1975.
- Floyd C. A systematic look at prototyping. Approaches to Prototyping, ed. Buddle R, et al. Springer, 1984.
- Greif I. Computer – Supported Cooperative Work: A Book of Readings. Morgan Kaufmann, 1988.
- Jackson R, Embley D. Using joint application designs to develop readable formal specifications. Information and Software Technology, Vol. 38, No. 10, pp. 615-631, 1996.
- Jacobson I, Booch G and Rumbaugh, J. The Unified Software Development Process. Addison-Wesley Pub Company, 1999.
- Jacobson I, Ericson M, Jacobson A. The Object Advantage: Business Process Reengineering with Object Technology. Addison-Wesley Pub Company, 1994.
- Jordan, N. Some Thinking about 'System' in Themes in Speculative Psychology, London, Tavistock, 1969.
- Joslyn C, & Turchin V. Heylighen, C. Joslyn and V. Turchin (editors): Principia Cybernetica Web (Principia Cybernetica, Brussels), URL: http://pespmc1.vub.ac.be/REFERPCP.html, verified on Tuesday, 23 August 2005.
- Kerr J, & Hunter R. Inside RAD: How to build fully functional computer systems in 90 days or less. McGraw Hill, 1994.
- Koo S, Son H, & Seong P. A method of formal requirement analysis for NPP I&C systems based on UML modelling with software cost reduction. Journal of Systems and Software, Vol. 67, No. 3, pp. 213-224, 2003.
- Martin J. Rapid Application Development. MacMillian Publishing, 1991.
- Mingers J. Self-Producing Systems: Implications and Applications of Autopoiesis. Plenum Press, 1995.
- Muller M, Blomberg J, Carter K, Dykstra E, Madsen K, Greenbaum J. Participatory Design in Britain and North America: Responses to the ‘Scandinavian Challenge’. Proceedings of the SIGCHI conference on Human factors in computing systems: Reaching through technology, 1991.

- Muller M. PICTIVE- an exploration in participatory design. Proceedings of the SIGCHI conference on Human factors in computing systems: Reaching through technology, New Orleans, Louisiana, pp. 225-231, 1991.
- Naylor J, Naim N, & Berry D. Legality: Integrating the lean and agile manufacturing paradigms in the total supply chain. International Journal of Production Economics, Vol. 62, No. 1-2, pp. 107-118, 1999.
- Newell A. The knowledge level. Artificial Intelligence, Vol. 18, No. 1, pp. 87-127, 1990.
- Ostwald J. Knowledge Construction in Software Development. PhD Thesis, University of Colorado, 1996.
- Rittel H. Development in Design Methodology, Second Generation Design Methods, in N. Cross (Ed.). John Wiley & Sons, NY, pp. 317-327, 1984.
- Roe C, & Beynon W. Empirical Modelling principles to support learning in a cultural context. Proceedings of 1st International Conference on Educational Technology in Cultural Context, University of Joensuu, Finland, pp. 151-172, 2002.
- Royce W. Managing the development of large software systems. Proceeding of WESTCON, San Francisco, 1970.
- Rungrattanaubol J. A treatise on Modelling with definitive scripts. PhD thesis, University of Warwick, 2002.
- Schaefer G, & Schaefer G. Functional Analysis of Office Requirements: A Multiple Approach. John Wiley & Sons, 1988.
- Sigchi. CSCW 94: Computer Supported Cooperative Work Proceedings held in Chapel Hill, NC. Association of Computing Machinery, ISBN 0897916891, 1994.
- Stapleton J. DSDM, Dynamic Systems Development Method: The Method in Practice. Addison-Wesley Pub Co, 1997.
- Stonier T. Information and the Internal Structure of the Universe. Springer-Verlag, 1997.
- Stratton R, & Warburton R. The strategic integration of agile and lean supply. International Journal of Production Economics, Vol. 85, No. 2, pp. 183-198, 2003.
- Suchman L. Plans and situated actions. The problem of human-machine communication. Cambridge: Cambridge University Press, 1987.

- Sun P. Distributed Empirical Modelling and its Application to Software System Development. PhD thesis, University of Warwick, 1999.
- Tsai T, & Sato R. A UML model of agile production planning and control system. Computers in Industry, Vol. 53, No. 2, pp. 133-152, 2004.
- Zahniser R. Timeboxing for Top Team Performance. CASElab report, URL: http://www.belizenorth.com/articles/TIMEBOX.htm, verified on: Tuesday, 23 August 2005.