Chapter 1.3
Exploring Enterprise Information Systems

Malihe Tabatabaie
University of York, UK

Richard Paige
University of York, UK

Chris Kimble
Euromed Marseille École de Management, France

ABSTRACT

The concept of an Enterprise Information System (EIS) has arisen from the need to deal with the increasingly volatile requirements of modern large-scale organisations. An EIS is a platform capable of supporting and integrating a wide range of activities across an organisation. In principle, the concept is useful and applicable to any large and SMEs, international or national business organisation. However, the range of applications for EIS is growing and they are now being used to support e-government, health care, and non-profit / non-governmental organisations. This chapter reviews research and development efforts related to EIS, and as a result attempts to precisely define the boundaries for the concept of EIS, i.e., identifying what is and what is not an EIS. Based on this domain analysis, a proposal for using goal-oriented modelling techniques for building EIS is constructed; the proposal is made more concrete through illustration via an example.

DOI: 10.4018/978-1-60566-856-7.ch021

INTRODUCTION

This chapter focuses on a grand challenge for organisations: dealing with their evolving requirements and goals, and the impact of these changes on their Information Technology (IT). In particular, we are interested in large-scale organisations such as multi-national companies, or public-sector organisations, which are sometimes called *enterprises* in the literature.

Organisations use IT in many different ways: to facilitate communication, to support commercial

transactions, to advertise, etc. In order to understand the effect of organisational and enterprise changes on use of IT, we start by defining the nature of an organisation. The current literature defines that an organisation is thus about a group of elements (human, automated system, structure, policy etc) that are arranged in a specific manner to accomplish a particular purpose (Buck, 2000; Laudon & Laudon, 2007; Terry, 1975). This definition applies to small, medium, and large-scale organisations.

As we said earlier, a large-scale organisation can sometimes be designated by the word *enterprise*. However, we find it helpful to be more precise in defining enterprise; in our view, an enterprise is a large-scale organisation that is involved in, and must orchestrate, more than one independent business processes. We come to this definition by observing that many organisations, such as small IT houses, engage in a single business process. Identically some large organisations, such as online retailers, have a single business process. Organisations that have many different business processes, that must be coordinated in some way, such as Mitsubishi, have different requirements and different characteristics. Such organisations are often very large scale (e.g., public health organisations) and multi-national. In our view, the need to coordinate different business processes is a key characteristic in distinguishing an enterprise from another organisation.

This paper investigates the validity of an assumption regarding the root of complexity of IT systems in complex organisations, where the IT systems support business processes directly. The assumption is that complexity is due to the following factors:

- Increasing size of IT systems and the organisation itself;
- The interactions between different IT systems;
- The involvement of many different organisations in the constructions and use of these IT systems; and,
- The increasing rate of organisational and social change.

By investigating the validity of this assumption, and the importance of these factors, this chapter aims to contribute a better understanding of Enterprise Information Systems (EIS), their dimensions, their boundaries, and the challenges that arise in their construction and development.

As part of this investigation, and as a result of the analysis of the literature that commences in the next section, we propose one key challenge for understanding and building EIS:

- *Understanding diverse and volatile stakeholder requirements.*

To aid in understanding these constructs, we propose the use of goal-oriented modelling techniques; this is discussed in the last section of this chapter.

The rest of the chapter is organised as follow: The *background* section outlines the challenges in large-scale organisations as a motivation for discussing the systems that can address these challenges. A specific instance of large-scale organisations is an enterprise; hence, section 2 also discusses the requirements of IT systems for enterprises. One of the main difficulties in this area is the imprecise definition for EIS, and how an EIS differs from a general purpose IT system. Hence, we provide a working definition for EIS in this section.

The *Enterprise Information System* section describes EIS in more detail by discussing state-of-the-art definitions and effective elements, such as business and organisation, based on a literature review. The *future trend* section describes goal-oriented modelling techniques as a promising approach for attacking one of the main challenges of building an EIS by making the system more clear for its stakeholders. Section 4 also provides an example to clarify this idea.

BACKGROUND

A brief review of the history of enterprises and software systems helped us to construct a working definition for EIS. This working definition is our basis for presenting an argument about what is and what is not an EIS, and for refining our understanding of the objectives for this type of systems. This section therefore discusses some examples of EIS to shape the argument.

Challenges of Large Scale Software System

Since the 1950s organisations have been developing computer-based information systems to support their business processes. Through improvements to IT, computer based systems have become more complex and yet more reliable; therefore increasing functional requirements have been placed upon these systems (Edwards, Ward, & Bytheway, 1993). However, building this kind of system has many challenges, including fundamental challenges regarding the construction of such systems, and the challenges of evolving systems to accommodate new requirements. Understanding the challenges of building such IT systems is essential for planning, designing, and development in order to provide as early as possible risk understanding, as well as understanding of the potential means for mitigation.

The challenges of understanding and building large-scale software systems can be observed in both the public and private sectors. In the public sector, understanding the challenges, and reflecting based on these challenges during the development process, is important because failure (whether financial or otherwise) can result in significant damage to the reputation of the government.

The National Audit Office/Office of Government Commerce lists the common causes of the project failure as follow (Projects, 2004):

1. Lack of clear connections between the project and the organisation's key priorities, including agreed measures of success
2. Lack of clear senior management and Ministerial ownership
3. Lack of effective engagement with stakeholders
4. Lack of skills and proven approach to project management
5. Lack of understanding of and contact with the supply industry at senior levels in the organisation
6. Evaluation of proposals driven by initial price rather than long term value for money (especially securing delivery of business benefits)
7. Too little attention to breaking development and implementation into manageable steps
8. Inadequate resources and skills to deliver the total portfolio

The first item in this list refers to the conceptual gap between project priorities and those of organisations; later in this chapter, more discussions address this challenge. In addition to these causes, hidden challenges threaten the IT projects; in particular the large-scale ones. For example, stakeholders should understand the conditions and limitations of the system. Having unreliable expectations from the system can move the domain of the project out of its limits and cause failure.

Another important and hidden challenge is the lack of visualisation in the software systems. Software is not visible and tangible for the stakeholders; therefore, stakeholders cannot picture the functionality of the software before it actually built, which can cause unrealistic expectations and other undefined problems. For example, in the case of constructing a building, stakeholders can visualise the building by looking at its mock-up; in the case of software there is no such a clear and easy to understand mock-up.

Flexibility and supporting changes are other challenges that software systems should deal with.

It is important to note that software systems can improve the speed of the processes in organisations and deal with the complex and well-defined processes. However, they are not intelligent enough to improve the business model; hence, software systems are not the solution for the ill-defined business model. This challenge can be seen mainly in large-scale software systems that deal with businesses in organisations, such as EIS. The term Enterprise Information System is a common term in today's industry, which suffers from misinterpretation and an imprecise definition. The rest of this chapter discuss this type of systems in more detail.

Large Scale Software System: Enterprise Information System

A specific kind of large scale IT system is those that support enterprises. We call these software systems, EIS. The business aspect of organisations motivates engineers to develop systems that satisfy real requirements of organisations, particularly requirements associated with business processes. As a result, technologies such as Service Oriented Architecture (SOA) are currently popular in design and implementation of systems for businesses. However, the term business often implies a process that focuses on delivering financial value; but in practice, large-scale processes, and their associated IT systems, i.e. EIS, can support delivery of different kinds of outcome, which are not always directly linked to financial value. In fact, today's businesses include both financial organisations as well as public organisations, which deliver services to the public. The success or value of these types of services is not always evaluated by the financial results they deliver.

To commence our main discussion on EIS, we first discuss enterprises; in our view, an EIS supports business processes of an enterprise. It is important to have an understanding of an enterprise to understand what an EIS is.

What Is an Enterprise?

The literature is not rich on the history of enterprises; however, Fruin (1992) is one of the researchers that explained the history of enterprises briefly and with an eye on the Japanese revolution in industry and business. According to this book:

The enterprise system appeared around the turn of the twentieth century when the factory system was effectively joined with a managerial hierarchy in production and distribution. It is the emerging coordination of previously independent organizations for production, management, and distribution-shop-floor, front office, and sales office- that generates the organizational innovation known as the Japanese enterprise system. (Fruin, 1992, p. 89).

According to Fruin (1992), the notion of an enterprise system was established after the First World War, when new industries came to the market and many industries combined and amalgamated. Three types of enterprises were identified: National, Urban and Rural; which all have some common elements such as inter-firm relations, Marketing, Mode of Competition, Finance, Ownership, Management, Administrative Coordination, Government Relations.

Mitsubishi is an example of an enterprise dating back to 1926; it integrates distinct yet affiliated companies, particularly Mitsubishi Heavy Industry, Mitsubishi Warehousing, Mitsubishi trading, Mitsubishi Mining, Mitsubishi Bank, Mitsubishi Electric, Mitsubishi Trust, Mitsubishi property, Mitsubishi steel, Mitsubishi Oil, Nippon Industrial Chemicals, and Mitsubishi Insurance (Fruin, 1992). There are many other examples of enterprises including Boeing, General Electric, Kodak, IBM, Norwich Union, Samsung, and Philips. From a consumer or client's point of view, these enterprises are often perceived as involving only one single organisation (e.g., Mitsubishi's car division).

Another example in this area is General Electric, which has independent divisions focusing on healthcare, aviation, oil and gas, energy, electrical distribution, security, and many others (GeneralElectric, 2008).

History shows that enterprises have existed from the turn of the twentieth century; nevertheless, the concept still suffers from an unclear definition.

Conclusion

Today's large-scale IT systems increasingly provide support for the business processes of organisations. The aim of using information systems is to increase the automation of the processes within organisations. Enterprises integrate organisations, departments, and even entire businesses to achieve shared goals. Processes within enterprises can benefit from IT infrastructure; in this section, we have argued for calling such IT infrastructure an EIS.

Working Definition

From the discussion on the history of enterprises and challenges of large scale software systems, we see that EIS are computer-based systems that satisfy the requirements of enterprises. EIS are designed and developed for enterprises rather than a single business unit or department. They can deal with the problems of a large organisation (which includes different SMEs or different partners), and they can deal with the problems of a medium or small enterprise (which is an organisation that includes different departments).

This working definition will be refined in later sections. After providing a brief background for EIS, we will discuss the definition in more detail.

ENTERPRISE INFORMATION SYSTEMS

Introduction

Based on the working definition developed in the last section, in this section we focus on refining the definition to include additional detail, particularly in the organisational and business context. As a result, this section proposes a concrete definition for EIS. To help explain the definition further, and partly to validate it, we relate it to well-known examples of organisations.

Challenges

The notion of enterprise is a widely used term for instance in the case of Mitsubishi. However, a precise definition of what constitutes an enterprise – and hence, what precisely constitutes an EIS – is still missing. One of the main difficulties in defining what is an EIS is in distinguishing it from any other large-scale software system. For example, perceived challenges in designing and developing an EIS will arise in the form of having to meet fixed costs of development, in dealing with volatile requirements, and in managing the complexity of the EIS. However, these are also challenges all kinds of large-scale software systems. Therefore, we do not aim to enumerate all of the design and development challenges of EIS; instead, this section will address one of the essential challenges, which is unclear definition for EIS; hence, we aim to propose a definition for this term.

To define EIS, this study reviewed the current definitions found in the literature; the next section will cover some of them.

State of the Art Definition

Organisations continue to find that they need systems that span their entire organisation and tie various functionalities together. As a result, an

understanding of enterprise systems is critical to succeed in today's competitive and ever changing world (Jessup & Valacich, 2006).

A good definition for EIS introduced it as a software system with the specific ability to integrate other software systems of an organisation.

Enterprise systems integrate the key business processes of a firm into a single software system so that information can flow seamlessly through the organization, improve coordination, efficiency, and decision making. Enterprise software is based on a suite of integrated software modules and a common central database. The database collects data from and feeds the data into numerous applications that can support nearly all of an organization's internal business activities. When new information is entered by one process, the information is made available immediately to other business processes. Organization, which implements enterprise software, would have to adopt the business processes embedded in the software and, if necessary, change their business processes to conform to those in the software. Enterprise systems support organizational centralization by enforcing uniform data standards and business processes throughout the company and a single unified technology platform. (Laudon & Laudon, 2007, p. 382)

This definition seems very specific on what is an EIS; however, there are points that are ignored by this definition. For example, the argument that mentioned when new information is entered by one process, the information is made available immediately to all other business processes. However, it can be argued that the information should be available to the other processes depending on their access domain. By this, we mean the level of access to the information should be different from process to process. It is not reasonable to expose information to the processes, which do not require it. Therefore, based on the access level of processes, only the suitable and updated information should be visible. This security policy does not have any contrast with the idea of enterprise processes, which their goal is to let the information flow seamlessly.

Moreover, (Strong & Volkoff, 2004, p. 22) defines an ES as a system which its task is to support and *"integrate a full range of business processes, uniting functional islands and making their data visible across the organization in real time"*. This definition adds to the previous definition, the fact that the data and information entailed by the system should be understandable by all its business processes.

Another definition for enterprise systems is based on legacy systems; a legacy system is an existing computer system or application program, which continues to be used because the company does not want to replace or redesign it (Robertson, 1997). Most established companies, who have been using a system for long time, are in this group. Legacy systems mainly suffer from deficiency of documentation, slow hardware and difficulties in improvement, maintenance and expansion. However, there is evidence that overtime EIS replaces the stand alone applications and the functionality of legacy systems (Strong & Volkoff, 2004). In contrast to enterprise systems, legacy systems are not designed to communicate with other applications beyond departmental boundaries (Jessup & Valacich, 2006) even if middleware offers a potential solution to adapt the novel parts with the legacy system. Nevertheless, regarding the price of developing a middleware, the following question comes to mind: can middleware alone solve the problem of integrating new subsystems with a legacy system?

In short, the common idea in the existing definitions illustrates that an EIS is about various businesses, business processes, organisations, information systems, and information that circulates across the enterprise. In other words, EIS is about the businesses model in the organisation. Therefore, the two main elements of EIS are organisation and business. The two following sections cover these points.

Organisation

The EIS definitions that we extracted from the literature linked the EIS to organisations (Laudon & Laudon, 2007; Strong & Volkoff, 2004) or large companies (Jessup & Valacich, 2006) and we assume that in both cases the definitions refer to the same concept: organisation. Based on this assumption, it is vital to review the different types of organisations that can influence the different types of EIS. Therefore, this section discusses categorise of organisations based on their goals. Elizabeth Buck categorises organisations in three groups (Buck, 2000):

- Public Organisations
- Private Organisations
- Not for Profit Organisations

The *public organisations* include central or local government, where elected members (e.g., minister) will decide on the goals of organisations, and may influence how goals are achieved. The aim of this type of organisation is to supply services to or for the public, considering a 'value for money' rule. Examples of this type of organisation can be health service, prison, police, social security, environmental protection, the armed forces, etc.

Individuals or other private organisations own *private sectors* organisations. This group of organisations can have the following goals:

- Satisfy their customer
- Satisfy their staff
- Satisfy their owners

All the above goals focus on increasing the market demands for products or services.

Examples of *not for the profit* organisations could be charities, mutual societies, etc, which provide some services for the society. The customers are also the member of the mutual society; therefore, they are the owner of the business. The value for money rule exists in this group too. The usual way to evaluate the success of this group of organisations is to measure how well they achieve their goals considering the available resources. Table 1 illustrates some of the characteristics of organisations that were described; it also summarises the different type of organisations.

By understanding the categories of organisations, we can focus on understanding their goals. By knowing the goals of organisations we can design and develop an EIS that satisfy the defined requirements and goals; but there are other questions in this area: what are the EIS' goals? Are the goals of EIS similar to the goals of organisations? It seems that EIS' goals could be a sub set of the organisations' goals. When the EIS' goals get closer to the goals of organisations it could become a better EIS. The final and optimistic goal for an EIS is to improve the goals of the organisation it services. However, defining the goals of an EIS is the path for analysing and developing the organisation's business model and thus the next section will explore the role of business in the definition of EIS.

Table 1. Organisations' categories

Type of Organisation	Decision Makers	Value for Money	Owner	Goal(s)	Example
Public	Elected members	Yes	Public	Supply Services to or for the Public	UK central Government
Private	Share holders	No	Share holders	Satisfy customers/ Satisfy staff/ Satisfy owners	Mitsubishi
Not for Profit	Elected Manager	Yes	Members/ Customers	Provide some services for the society or members	NCH (Children Charity)

Business

Another main factor that influences the architecture and functionality of an EIS is the business model (Figure 1). Supporting the strong relationship between business processes is the aim of ES. In fact, the ability to define various business processes in enterprise systems is the element that distinguishes them from normal systems for a company or a department; for example BMW involves in a diversity of businesses to produce cars or engines for other car brands (e.g. Rolls-Royce), in addition to building bicycles and boats. A normal system in a company contains components and subsystems that belong to one specific business and satisfy its requirements. A normal company may need to contact other companies to continue its business but involving partners or suppliers is not their main concern. In contrast to normal company where the focus is on one particular business, an enterprise focuses on a collection of business processes which could be relevant to each other or not but all of them are under the arch of the main principals of the enterprise. Indeed, making profit is not one of the essences of business model. There are non-profit governmental or non-governmental organisations such as healthcare organisations that can have their own business model which deals with the process of treating patients.

The presentation of Enterprise System in this chapter is not about detailed implementation of business functions; its focus is mainly about a very top-level view on the whole business model of an organisation as defined by Clifton (2000):

business involves a complex mix of people, policy and technology, and exist within the constraints of economics and society (Clifton, Ince, & Sutcliffe, 2000, p. 1).

Figure 1. Business model [based on (Kaisler, Armoiur, & Valivullah, 2005)]

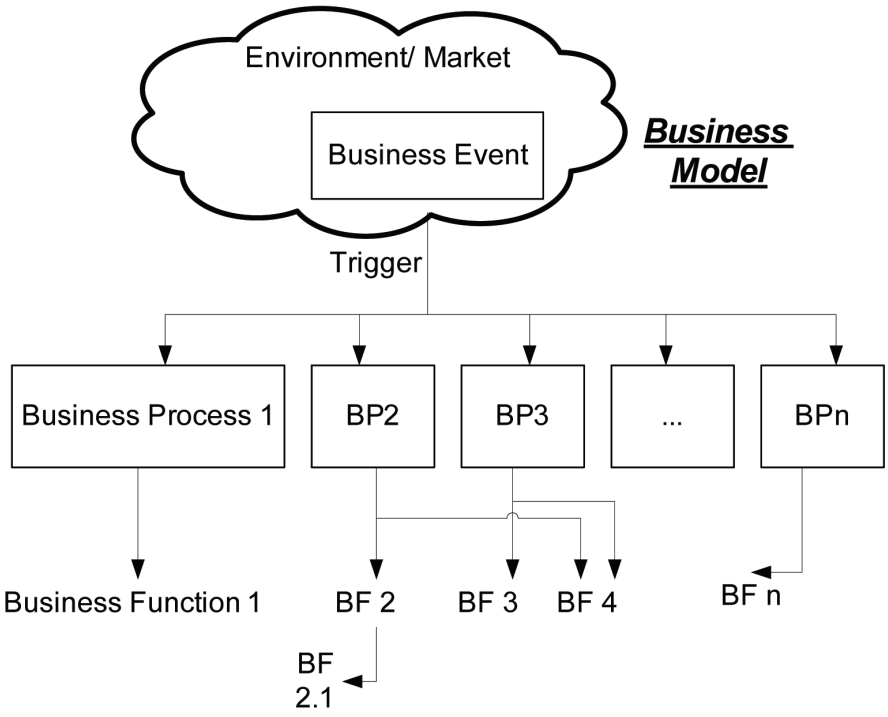


Figure 1 illustrates the general structure of a business model where the business model includes business processes and business functions. Business processes are "*a set of logically related tasks performed to achieve a defined business outcome*" (Davenport & Short, 1990, p. 100). For example, in the case of BMW, the business processes is putting new orders for part suppliers. When there is a new demand for specific car (e.g. model Z5), this new market request creates a business event that triggers a set of business processes such as increasing the amount of resources for producing the Z5 (e.g. BP2 in Figure 1), and putting new orders for parts suppliers. Each of these business processes is subdivided into different business functions (e.g. BF2 and BF3 in Figure 1). Examples include the functions required for inputting new orders such as checking the parts suppliers' ability for new demands, organising the time that is needed for each part to arrive to assembly line, etc.

According to (Kaisler, Armoiur, & Valivullah, 2005, p. 2) "*business processes must be modelled and aligned across the function, data and information systems that implement the processes*". Therefore, the term business function in our research refers to the functionality that is required for implementing a business process. Figure 1 is a simple explanation for business process model. Each of these business functions can trigger a business process too. Moreover, the business processes can breakdown to other business processes, which is not shown in this diagram to keep it simple to understand. The aim of this diagram is mainly to explain business processes and functions in a general business model.

Understanding business models is helpful for developing EIS because their role is to integrate a full range of business processes (Strong & Volkoff, 2004). Before defining the concept of EIS, Legacy systems were the type of systems that were developed to handle the requirements of organisations (Robertson, 1997). However, legacy systems are not designed to communicate with other applications beyond departmental boundaries (Jessup & Valacich, 2006); hence the concept of EIS has grown to fill this gap.

In short, the common idea in existing definitions illustrates that an EIS amalgamates concerns from various businesses, business processes, organisations, information systems, and information that circulate across an enterprise. In other words, it is about the business models of the organisation. However, a definition for EIS that just emphasises the financial profit side of businesses for organisations is out of date. In the next section, a definition that considers other aspect of organisations, the domain and objectives of EIS is proposed.

Enterprise Information System Definition

This section proposes a definition for EIS, which is the result of our analysis of the state-of-the-art definitions and of industrial case studies. The definition that considers business and organisational aspects of EIS is as follow:

An Enterprise Information System is a software system that integrates the business processes of organisation(s) to improve their functioning.

Integration of business processes plays an important role in this definition. Integration could be accomplished by providing standards for data and business processes. These standards will be applied to various part of the system such as a database or clusters of databases. As the result of such integration, information may flow seamlessly.

Another point in this definition is the software characteristics of EIS. At this stage, we consider EIS as a type of Information System; therefore, this software system includes both humans and hardware.

The next term, used in the definition is organisation. Different types of organisations are discussed earlier in this chapter. Organisations may include an organisation with its partners, or a group of organisations. Table 2 refines the above definition

Table 2. EIS boundaries, objectives and challenges

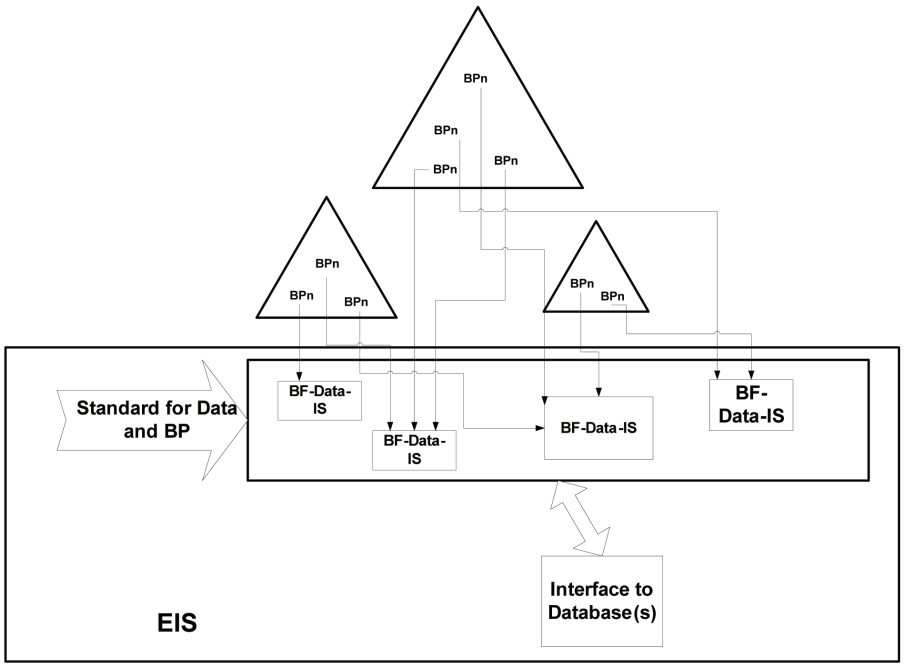
Objective	Integrity of the organisation and collaborators
	Seamless Information flow
	Suitable access to data and information for various stakeholders
	Matching the software system structure with organisation structure
Goal	Improving coordination, efficiency, and decision-making of business process in an organisation
Domain	Covers the internal and external business activities of organisation
Challenge	Security challenges that should be considered carefully for organisations' processes. Otherwise, mixing the required information of one business process with another one can cause problem for the organisation
	Improve flexibility in organisation processes

and describes what we propose as the objectives, goals, domain, and challenges of EIS.

In addition, Figure 2 describes the definition of EIS graphically. Note that BP in this figure are business processes. As it can be seen in this figure, each organisation contains various business processes. Moreover, in Figure 2, the database could be a cluster of databases; however, it is highly likely that there would be a single interface to exchange data with the database without having a concern about where the data is and what are the various resources. As can be seen in this figure, the bigger rectangle describes the boundaries of EIS, which is flexible.

The two following sections aim at continuing the discussion about EIS by presenting some examples in this area. The results of reviewing these examples lead to a better clarification of what is an EIS and what is not.

Figure 2. An enterprise information system



Examples of Enterprise Information System

The review of the industrial cases of what might be considered as an EIS moves our discussion toward the example of Mitsubishi. As was mentioned earlier, Mitsubishi with more than 400 companies all around the word is an example of enterprises (Mitsubishi, 2007). Thirty top-level managers manage all the individual Mitsubishi's companies. This does not mean that each company does not have enough freedom to make their own decisions; it means that this group of thirty managers will make some of the top-level decisions and they provide the high-level standards that all these companies should consider. In this case, if there is a computer based system that links various parts of the Mitsubishi organisation (including high-level managers) together and makes information flow seamlessly between them, then we view this system as an EIS. Developing such a system is a large and complex problem; hence, there is a need for powerful, reusable solutions to develop this type of system in a manner that can benefit all of the enterprise.

Another example in this area is the infrastructure being developed to support the National Health Service (NHS) in the United Kingdom where the information systems being developed to support management of patients' records and prescriptions can be considered as an EIS, because such IT infrastructure aims to connect independent departments within and outside of the NHS. While we are looking at the NHS, which is a public sector organisation, we can raise e-Government as another example of public sector organisation that may be supported by and hence benefit from, EIS infrastructure because it connects various governmental organisations or departments together to let information flow seamlessly between them.

What Cannot Be an Enterprise Information System

As we were discussing the public, private, and governmental examples for an EIS, the next step is introducing some examples of Information Systems that are not EIS according to our definition.

eBay is one of the well known international Information Systems that focuses in the auction industry. This online market which involves around 147 million people (Gopalkrishnan & Gupta, 2007) provides a platform for individuals or companies to trade their products or services; but it does not connect the business processes of organisations together. Therefore, according to our definition, an EIS connects different business processes of organisations or departments of organisation together to make the information flow seamlessly and thus it seems that based on this characteristic of EIS, eBay is not an EIS. The information system is the element that processes data and put them online, there is no evidence of connection between business processes because it is not a requirement in this Information System. The same argument can be followed in the case of Amazon, therefore even though it is large-scale and international online shop but it is not an EIS.

Conclusion

In short, this section described EIS in more detail by providing the definition for EIS. Defining any kind of system is essential for defining its domain and objectives. Without this basic information, the researches on the similar area will not be consistent. However, there is no claim that the given definition is the only definition for EIS. This definition is based on our studies, observations, interviews, and comparisons on the current theoretical and practical definitions and case studies. Part of this ongoing work is presented in this chapter.

To make the results of our study on the definition of EIS more clear, two examples are discussed in this section. The earlier examples describe the case that can be an EIS and the case that cannot be an EIS. This categorisation is based on this chapter's criteria, which is discussed in the given definition. Hence, each of these cases could be the objective for more discussions on the possibility of being an EIS or not. Considering different point of views and the context of arguments, one Information System can be an EIS or not. Therefore, it is crucial to consider the writers' point of view and given definition, in the preceding examples.

After discussing what can be an EIS, the next section will focus on an approach for developing this type of system.

FUTURE TRENDS

Goal-based and goal-oriented thinking is used to plan for the future or to solve problems (Kim, Park, & Sugumaran, 2006). The concept of using goal-oriented techniques has been proposed as one possible way to manage some of the difficulties associated with developing large-scale complex systems (Kavakli, Loucopoulos, & Filippidou, 1996), particularly the challenge of clearly identifying and specifying requirements. As we discussed in the previous section, an EIS is an instance of large-scale complex system. This section promotes the idea of using goal-oriented modelling techniques for developing EIS by briefly discussing them and their roles in defining EIS system requirements. We will summarise our discussion on goal-oriented techniques by presenting an example of goal graph.

Goal Oriented Techniques

Goal oriented techniques have been widely discussed in the requirement engineering domain (T. P. Kelly, McDermid, Murdoch, & Wilson, 1998; Axel van Lamsweerde, 2001; A. V. Lamsweerde, 2004). Goals are also used in the safety and security research community – for example, to present safety cases and safety arguments (T. Kelly, 2004; T. Kelly & Weaver, 2004) - and in software assessment (Weiss, Bennett, Payseur, Tendick, & Zhang, 2002).

Kelly (1998) defined a goal as 'requirements statement', Lamsweerde (2003) used goals as criteria for deriving software architecture. Kim et al (2006) defined goal model from (Axel van Lamsweerde, 2003) point of view as a criteria for designing the architecture for systems; therefore, the aim of software architect is to implement a system based on the architecture to accomplish goals (Kim, Park, & Sugumaran, 2006). Logically goals are the motivation for developing a system; therefore, all the stakeholders should have a clear understanding about the goals of the system. In addition, the goals of the system should be realistically defined before continuing any other step of the development. There are attempts to show the goals in graphical notations such as GSN (Timothy Patrick Kelly, 1998), Kaos (Axel van Lamsweerde, 2001), and (Kim, Park, & Sugumaran, 2006). Moreover, Kaos defines the formal textual notation to describe the goals in addition to informal text. This attempt is respectful because it considers the larger group of audience to understand and benefit from the goal model.

Different stakehold ers require different forms of presentation for their goals. For example, the high-level manager may not require seeing a formal explanation of the goals because they may not understand it; however, they can better understand an informal explanation in a simple diagram. On the other hand, there is a good possibility that a programmer's team, requires the formal explanation for the goals in detail to understand and implement the system in the correct and expected way. It is important to bear in mind that goal diagrams are aimed at making the system more clear to different stakeholders, therefore goal-oriented ideas should prevent adding more confusions for different stakeholders. Any approach that makes

the goals of the system more clear for stakeholders should be considered; it can be different goal models for different stakeholders.

The next section explains an approach for designing a goal model. This approach is very high level without explaining the details. The aim is to introduce a possible approach for developing a goal model to readers. This approach benefits from the information in similar studies in this area such as (Timothy Patrick Kelly, 1998; Kim, Park, & Sugumaran, 2006; Axel van Lamsweerde, 2001).

Designing a Goal Model

One of the main reasons for developing unsuccessful software systems is unrealistic planning and design. Hence, the aim of goal-oriented approaches, as discussed in the previous section, is to provide an environment such that different stakeholders can understand the goals at different levels of abstraction and decomposition. One way to accomplish this is to use a graphical modelling language, such as GSN; another way is by documenting the requirements and design precisely and accurately using a textual format. It is also possible to present the prototype of the system and discuss it with various stakeholders. All these approaches and other similar ones could be beneficial for different type of systems. The approach that is discussed in this section is a simple approach for developing a goal model. The aim is to develop a goal model that can present the system's high-level goals clearly. Furthermore, it does not involve the details of the goals or their descriptions; this can help to provide an understandable top down model for high-level goals of the system for non-technical decision makers.

The basics of this approach for designing a goal model is to create a list of goals, a list of actions, and a list of occurred problems. Goals were defined earlier; actions according to (Kim, Park, & Sugumaran, 2006, p. 543) "*are the atomic unit for representing software behaviour, which can be observed at runtime and has a start and end point of execution*". This paper also argues that most methods in the class diagram can be action, but because the runtime of actions should be observable, the size of the action should be restricted in a manner that makes it possible to be observed in the software model.

The issues in this case are the challenges and difficulties that occur when developers consider the implementation and the execution of a system. These challenges can be a technical difficulties, or goal conflicts, etc.

After producing the goals, actions, and problems lists, the relation between these elements should be created. The notation here is similar to the notation in (Kim, Park, & Sugumaran, 2006), which is as follows:

$$(G_z, A_n) \rightarrow P_x$$

A represents an action, G represents a Goal, and z and n are the symbolized identification of random variables that present the ID of the goal, for instance, it can be G 1.1.2, which means goal with ID 1.1.2. An example of action could be 1.1.2/1, which presents the required action that can be done to achieve this goal.

The next notation illustrates the relationship between a goal, action, and problem. Following is an example of this notation:

$$P_x \rightarrow (G_z, A_n)$$

The above notation means, the Action with ID n which is required for satisfying Goal with ID z can cause the problem with ID x. This notation describes the case where action that belongs to a goal causes a problem or problems. The next notation describes the case that a problem can be solved using a specific action:

$$P_x \rightarrow (G_z, A_n)$$

The above notation means to solve the problem P with ID x, the Goal G with ID z is required, and to satisfy this goal Action with ID n should be done.

In the case that the developer team does not know the required action yet, action n (An) can be replaced with '?'. Before starting to implement the system, all the question marks (?) should be filled with actions as solutions to satisfy goals. Nevertheless, in the case that as the result of limitations in the technology, resources, etc. one or more question mark (?) cannot be replaced by solution, there could be a bottom-up check to see if the system is still worth implementing; considering the unsolved problem(s), the functionality of the system should not rely on the non available solutions.

However as it was discussed before, a goal model should have different levels of abstractions for different users. Hence, designers should avoid destroying the purpose of the goal model, which is to make the systems goals clear for stakeholders by mixing and presenting all the information to the ones who does not require it. The next section will provides an example of goal model for stroke care.

Example

The aim of the case study is to design the goal model for a system that collects the data of treatment for a specific serious condition. The data can be collected from different sources such as doctors, researchers, nurses, emergency staff, etc. Moreover, each of these stakeholders can have a different way of communicating with the database, for instance, laptop, paper, phone, etc. The role of this system is to collect the data from various sources, analyse them and provide some data as an output for different purposes. Based on the case study in (Bobrow.D.G, 2002) we can call this system a knowledge sharing system.

Figure 3 illustrates the described system. In this figure the boundaries of the described system is shown as a box surrounding it. The big arrow in the left hand side of the box illustrates the fact that this system is one of the information systems in the defined enterprise. The enterprise in this figure is shown using a pyramid, which is mainly a symbol of organisation.

To make this figure simple and clear we did not include the option that this EIS can be shared and used with other enterprises around the globe. Note that by having a design for EIS, we try to have a big picture of enterprise that includes the possible changes and extension in the future. The EIS does not have a local design that cannot be used when changes occur. The current solution for extending a system or merging systems is mainly developing middleware, which enterprise architect should avoid relying just on middleware. Considering that in some cases middleware can be so expensive that the organisation's decision makers may decide to use manual paper based system instead.

After drawing an overall view of the requested system, the goal of the system should be defined. Each goal should have its own action, which acts as a solution for the system and the possible problems. Figure 4 illustrates the goal diagram for this system. This diagram is very high-level, which targeted non-technical decision makers. This diagram is the starting point for creating a complete goal model for this system. As can be seen in this diagram, goals have their unit identity, which in this case is shown by numbers. These numbers makes the traceability of goals possible within this model. In addition, it is possible to implement it in tools for drawing diagrams. This is a AND-OR graph and it means the parent goal with OR child can be satisfied when at least one of the child goal reach to the solution. It is similar to AND-OR in logic mathematic. Furthermore, goal graph is a weight graph; hence, the goals in the same level can be prioritised over other goals.

Prioritizing goals is helpful in different context. For example, in allocating resources or in some cases, when satisfying a lower priority goal is

Figure 3. Example of knowledge sharing system

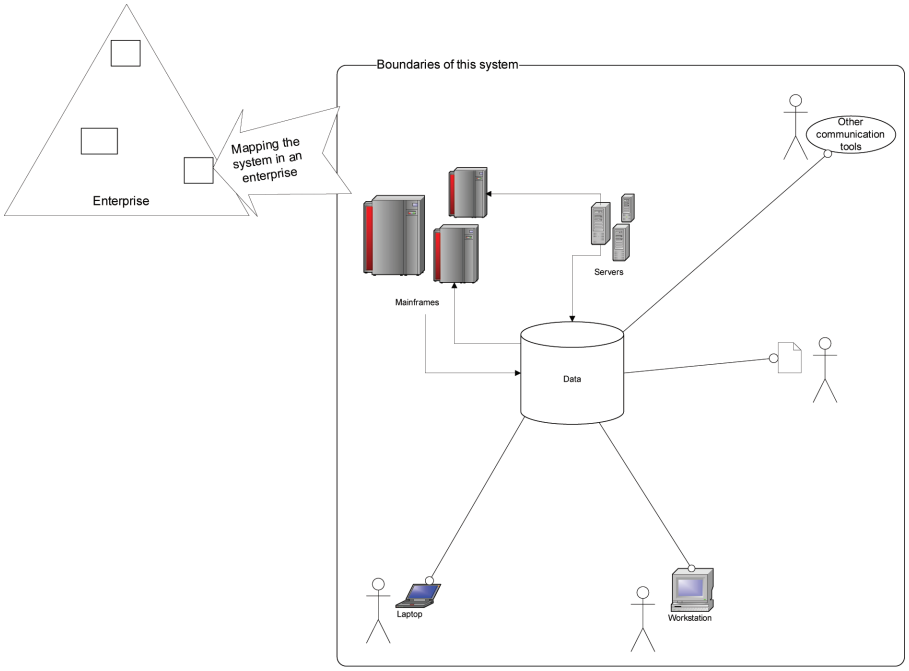
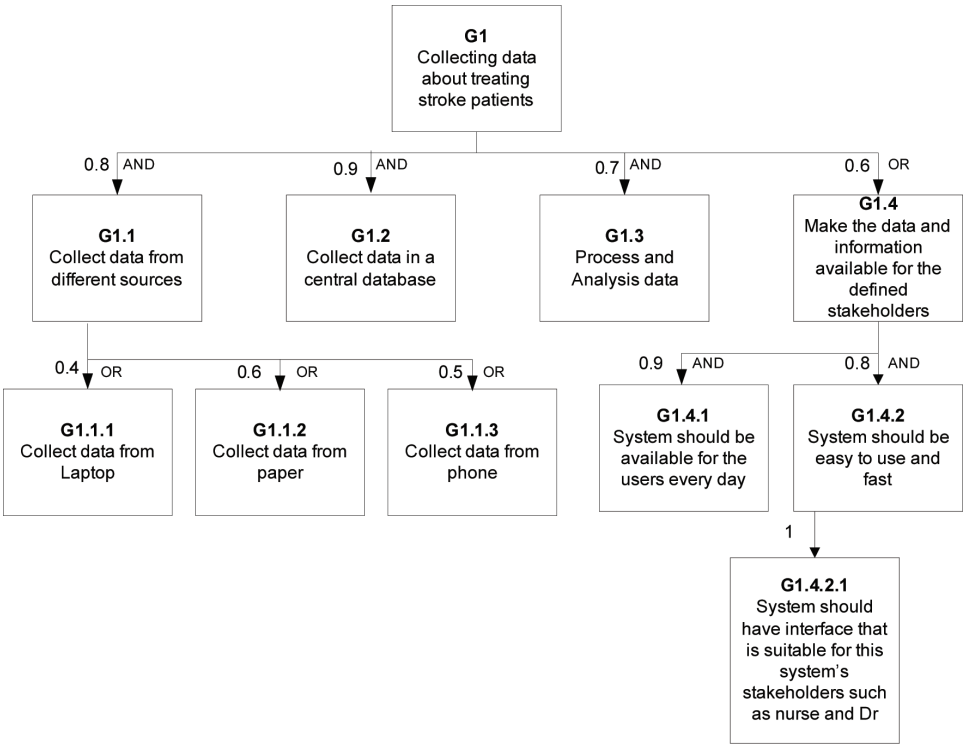


Figure 4. Goal graph



depend on satisfying the higher priority goal. In general, Figure 4 shows the basic requirements for the goal graph. We emphasise that the aim of this graph is to provide the high-level clear image of the system's goals and present it to the stakeholders to be used for brainstorming for example.

Conclusion

In conclusion, this section proposed the future work for the study on how to develop EIS. The fact that developing required components for EIS can be similar to other large-scale complex system makes this field of work valuable; because finding better solutions for different challenges of Information Systems provides a platform for developing various kinds of suitable systems. This effort and study on EIS provides an easier and safer life for individuals and organisations that benefit from this type of IT products. It influences the government's performance, it provides better innovative platform for industries. All these reasons bestow enough motivations for us to continue improving this study.

CONCLUSION

By looking at various ways that the word 'enterprise' is used, it becomes clear that there is an ambiguity in this term. Yet this term and others such as 'Enterprise Architecture', 'Enterprise Information System', etc. are increasingly used. This fact encourages us to look at these terms and clarify them for future use in our research and other relevant ones. The simple definition for enterprise is an entity engaged in economic activities. This definition does not cover requirements for defining an EIS. The argument in this chapter illustrates that an EIS covering the requirements of any entity engaged in economy activity is simply an IS and they can hardly be categorized as a separate group with the name of EIS. The fact that the number of people employed by an organisation can increase the complexity of the software system in some cases is hardly the leading factor in developing an EIS. The basic requirement for research on how to improve the development of EIS is to achieve more knowledge on what an EIS is.

Consequently, the main objective of this chapter was to explore the boundaries of EIS; this was achieved by developing a definition for EIS. This definition captured what we believe are the important characteristics that should be considered while we attempt to build an EIS; characteristics such as organisations, their goals, business processes, and the business model. None of these characteristics is based on the size of the organisation; therefore, it can cover different sizes of enterprises, small, medium, or large. Accordingly, this chapter did not use a specific term such as SME, Small and Medium Enterprises, to define EIS.

Any discussion of EIS encompasses a number of facets, including general IT system development, requirements, organisational theory, and distributed systems technology. Our aim is to more precisely define what an EIS is, and what it is not, to assist in providing better methodologies and techniques for building such increasingly important software systems. We believe that it is clear that the volatile requirements of modern organisations require special business processes, and these business processes cannot be fully achieved without IT systems and in some cases without an EIS. A high-quality EIS can provide a connection between the different, independent business processes in an enterprise.

As discussed, we argue that goal-oriented modelling techniques are important for helping to understand what is required for a business or organisation, and for understanding what an EIS should provide. Thus, we argue that a first step for developing a system for an enterprise is to find and justify the enterprise's goals. When all the stakeholders have a clear idea about the goals of the enterprise, their expectations will be realistic in principle; the desired system's boundary should be more precisely defined, and in principle building the system should be possible. We do not

claim that following this approach will provide a full guarantee for developing a suitable EIS: such systems are always challenging to build, and goal-oriented techniques only tackle an important part of a large problem. Additional research and experiments are needed to identify what further techniques are needed to supplement goal-oriented modelling for designing, implementing, deploying, and maintaining Enterprise Information Systems.

ACKNOWLEDGMENT

We would like to thank Dr. Fiona Polack for her valuable suggestions.

REFERENCES

Bobrow.D.G, W. J. (2002). Community knowledge sharing in practice: The Eureka story. *Society for organizational learning and Massachusetts Institute of Technology, 4*, 47-59.

Buck, E. (2000). Different Types of organisation. *NEBS Management/QMD Ltd.* Retrieved September 1, 2008 from http://www.teamsthatwork.co.uk/Organise%20&%20improve%20team%20work%201.pdf

Clifton, H., Ince, D. C., & Sutcliffe, A. G. (2000). *Business information systems* (6th ed.). Essex, England: Pearson Education Limited.

Davenport, T. H., & Short, J. E. (1990). The New industrial engineering, information technology and business redesign. In M. Lewis & N. Slack (Eds.), *Operations management: Critical perspectives on business and management* (pp. 97-123). London and New York: Routledge.

Edwards, C., Ward, J., & Bytheway, A. (1993). *The essence of information systems* (2nd ed.). London: Prentice Hall.

Fruin, M. W. (1992). *The Japanese enterprise system*. New York: Oxford University Press.

GeneralElectric. (2008). Product and services. Retrieved September 1, 2008 from http://www.ge.com/products_services/index.html

Gopalkrishnan, J., & Gupta, V. K. (2007). eBay: "The world's largest online marketplace" - A Case Study. *Conference on Global Competition and Competitiveness of Indian Corporate* (pp. 543-549).

Jessup, L., & Valacich, J. (2006). *Information systems today, why is matters* (2nd ed.). NJ: Pearson Education, Inc.

Kaisler, S., Armoiur, F., & Valivullah, M. (2005). *Enterprise architecting: Critical problems.* Paper presented at the 38th Annual Hawaii International Conference on System Sciences, Island of Hawaii, HI.

Kavakli, E. V., Loucopoulos, P., & Filippidou, D. (1996). *Using scenarios to systematically support goal-directed elaboration for information system requirements.* Paper presented at the IEEE Symposium and Workshop on Engineering of Computer Based Systems(ECBS '96), Friedrichshafen, Germany.

Kelly, T. (2004). *A Systematic approach to safety case management.* Paper presented at the SAE 2004 World Congress, Detroit, MI.

Kelly, T., & Weaver, R. A. (2004). *The goal structuring notation - A safety argument notation.* Paper presented at the 2004 International Conference on Dependable Systems and Networks (DSN 2004), Florence, Italy.

Kelly, T. P. (1998). *Arguing Safety- A systematic approach to managing safety cases.* University of York, York.

Kelly, T. P., McDermid, J., Murdoch, J., & Wilson, S. (1998). The goal structuring notation: A means for capturing requirements, rationale and evidence. In A. J. Vickers & L. S. Brooks (Eds.), *Requirements engineering at the University of York*: University of York.

Kim, J. S., Park, S., & Sugumaran, V. (2006). Contextual problem detection and management during software execution. *Industrial Management & Data Systems*, *106*, 540–561. doi:10.1108/02635570610661615

Lamsweerde, A. v. (2001). *Goal-oriented requirements engineering: A guided tour.* Paper presented at the 5th IEEE International Symposium on Requirements Engineering (RE'01), Toronto, Canada.

Lamsweerde, A. v. (2003). *From system goals to software architecture. Formal methods for software architectures* (. *LNCS*, *2804*, 25–43.

Lamsweerde, A. V. (2004). *Goal-oriented requirements engineering: A roundtrip from research to practice.* Paper presented at the 12th IEEE Joint International Requirements Engineering Conference(RE'04,), Kyoto, Japan.

Laudon, J. P., & Laudon, K. C. (2007). *Management information systems: Managing the digital firm* (10th ed.). Prentice Hall.

Mitsubishi. (2007). *About Mitsubishi*. Retrieved September 1 2008, from http://www.mitsubishi.com/e/group/about.html

Projects, T. C. C. I. (2004). The challenges of complex IT projects. Retrieved September 1, 2008, from http://www.bcs.org/server_process.php?show=conWebDoc.1167

Robertson, P. (1997). Integrating legacy systems with modern corporate applications. *Communications of the ACM*, *40*(5), 39–46. doi:10.1145/253769.253785

Strong, D. M., & Volkoff, O. (2004). A roadmap for Enterprise system implementation. *IEEE Computer Society*, *37*, 22–29.

Terry, P. (1975). Organisation behaviour. *Industrial & Commercial Training*, *7*(11), 462–466. doi:10.1108/eb003504

Weiss, D. M., Bennett, D., Payseur, J. Y., Tendick, P., & Zhang, P. (2002). *Goal-oriented software assessment.* Paper presented at the 24th International Conference on Software Engineering (ICSE '02), Orlando, FL.

This work was previously published in Social, Managerial, and Organizational Dimensions of Enterprise Information Systems, edited by Maria Manuela Cruz-Cunha, pp. 415-432, copyright 2010 by Information Science Reference (an imprint of IGI Global).