Integration of business modelling methods for enterprise information system analysis and user requirements gathering

Hui Shen¹, Brian Wall²*, Michal Zaremba², Yuliu Chen¹, Jim Browne²

¹Department of Automation, Tsinghua University, Beijing, China
²Computer Integrated Manufacturing Research Unit (CIMRU), National University of Ireland, Galway, Ireland

Received 2 May 2002; accepted 23 July 2003
Available online 13 February 2004

Abstract

Business process modelling is an essential part of developing an enterprise information system. There are many modelling methods with software support currently available on the market. Each individual method has its own advantages and disadvantages but always has the limitation of only representing a certain view of enterprise. To describe a system clearly from different perspectives and to provide a complete understanding of the business process both to the developer and to the end-user, it is necessary to adopt more than one kind of modelling technique to establish a set of graphical models describing a system from different views.

The modelling approach described in this paper is composed of three widely used modelling methods: IDEF0 to establish functional models, IDEF3 to capture process descriptions, and DFD to describe information/data flow among the activities. It is a staged approach in which different modelling method is used at different levels of granularity and details of information required. After a careful evaluation and comparison (including respective advantages and disadvantages) of the three adopted modelling methods, a guideline is proposed for using a composite of these three modelling methods to establish a set of business process models from different perspectives. The aim is to combine the advantages of each modelling method and maximise the effect of modelling work. Finally, a case study is presented in order to illustrate the effectiveness of such a modelling framework.

Keywords: Business process; Modelling methods; AS-IS analysis; TO-BE design

1. Introduction

A business process is a set of one or more linked procedures or activities which collectively realise a business objective or policy goal, normally within the context of an organisational structure defining functional roles and relationships [1]. Business process modelling is essential for developing an enterprise information system.

There are many modelling methods with software support currently available on the market. Each method has its own advantages and disadvantages, and each individual method is limited with regard to the view of the enterprise that it can present. To describe a system clearly from different views, and provide a complete understanding of business processes both to the developer and to the end-user, it is necessary to adopt more than one kind of modelling technique to establish a set of graphical models.
describing a system from different views [2]. Even though most experts accept such a notion and have carried out a lot of research work, it lacks practical guidelines in engineering. The aim of this paper is to present such a guideline and provide some suggestions on how to choose and use modelling methods in system analysis and user requirements gathering.

Object-oriented (O-O) modelling techniques are very popular currently because they are programming oriented and can shorten the development life cycle. But at the stage of system analysis and user requirements gathering, a structured methodology is still irreplaceable. The conventional structured modelling method for this stage is IDEF0. IDEF0 has been practised in industrial IT projects for decades. It has proven to be easy to understand by business people acting as a bridge for better communication between technical developers and industrial end-users. Even though there are many newer modelling alternatives to IDEF0, few are as acceptable as IDEF0 for business people. IDEF0 modelling is a strict top-down process. It is initiated from a top-level diagram, and decomposed to several bottom-levels. As with other modelling methods, IDEF0 has the inherent limitation of not being able to describe all aspects of the system. Some complementary modelling methods can be used to work together with IDEF0, such as IDEF3 and DFD, which is suggested for this paper. According to conventional structured approach, if one chose IDEF0, IDEF3 and DFD as the business process modelling methods for system analysis and user requirements gathering, the process would be:

- IDEF0 modelling, from top-level to bottom-level;
- IDEF3 and DFD modelling, at lower levels encompassing specific details, to offset the limitations of IDEF0.

This process follows top-down structured decomposition rules, but in real industrial practice, the authors have found that a combination of top-down and bottom-up modelling approach is more practical, especially for AS-IS analysis. The conventional approach can be changed to; firstly carry out top-level IDEF0 modelling, followed by bottom level IDEF3 and DFD modelling and finally create bottom-level IDEF0 models. These bottom level IDEF0 AS-IS models form a very good basis for the initial stages of the TO-BE design and help to accelerate the requirements process.

In Sections 2 and 3, the authors evaluate and compare modelling methods and a staged modelling approach has been adopted to establish business models for end-users. The modelling approach is composed of three widely used modelling methods: IDEF0 to establish functional models, IDEF3 to capture process descriptions, and DFD to describe information/data flow among activities.

As a basis for discussion, a brief review of modelling methods and tools is made in Section 2. In Section 3, a careful evaluation and comparison of the three adopted modelling methods is made looking at the advantages and disadvantages of each. Then a guideline for using a combination of these three modelling methods to establish a set of business models at different stages and from different views is proposed. The aim is to combine the advantages of each modelling method, thus maximise the effect of modelling work. In order to illustrate the effectiveness of such a modelling idea, a case study is presented in Section 4. Finally, some conclusions and remarks are presented in Section 5.

2. Review of modelling methods and tools for information system analysis and design

A variety of methods and tools can be used to promote enterprise information system development. A classification of the modelling methods and techniques most frequently used is summarised in Fig. 1.

There are three levels shown in Fig. 1. The top level shows the enterprise modelling frameworks, which provide generalised reference architectures and methodologies to guide system analysis and design for the whole life cycle [3]. Among the most widely used of these are CIMOSA, GIM and PERA [4]. CIMOSA, developed by the European AMICE Consortium during the early 1990s, provides a consistent architectural framework for enterprise modelling and integration [5,6]. The CIMOSA Cube illustrates three dimensions; the dimension of instantiation (three levels: generic, partial and particular), the dimension of model derivation (three modelling levels: requirements, design and implementation), and the dimension of view (function, information, resource and organisation). GIM, originally stood for GRAI-IDEF0-Merise [7], now called GRAI Integrated Methodology [8], has its origin in
GRAI, which is a method to model and analyse automated manufacturing systems. It is well known as a method to describe decision centres in an enterprise. PERA, developed by Purdue Laboratory for Applied Industrial Control \[9,10\], is characterised by its layering structure, which covers the full enterprise life cycle. The most significant contribution of PERA is that it is the first architecture that fully considers the human factor.

The middle level of the hierarchy in Fig. 1 represents general system modelling methodologies, which include Structured and O-O methodologies. Structured methodologies such as SADT [11] are a kind of traditional and mature methods, which have been implemented for decades. Even though O-O is regarded as the leading technique nowadays, Structured methodologies still play an important role in system analysis and design. For O-O modelling, Unified Modelling Language (UML) [12] has recently become an important tool. UML is a language, not a method, but it provides several graphical modelling languages (diagrams) to establish the O-O model of enterprise, including Use Cases, Sequence and Collaboration Diagrams (interaction diagrams) and Class Diagrams. UML focuses on a standard modelling language, not a standard process. UML is a relatively young modelling language compared to the IDEF0, IDEF3 and DFD, and in general it is used mainly for software development. The large number of diagram types (of which two are explicitly targeted at software development/deployment) originates from the way UML was created (methods unification) and does not help with non-software oriented modelling. In addition UML models are often more difficult for end users to understand.

The bottom level of the classification diagram represents particular modelling methods for individual views. It has been widely accepted for several years that a system should be modelled from different views. In CIMOSA, there are four views presented as integrated aspects for enterprise modelling. In recent years the number of views has been expanded. Some important aspects, such as decision, economic, dynamic views have aroused the interests of both industry and academic institutions.

---

As detailed in Fig. 1, there are many potential techniques that could be used to design and develop a new information system. The decision on a choice of suitable technique depends on the project requirements, as well as the implementation scenario. As the scenario changes, the technique should also be changed correspondingly. For example, at the beginning of an e-business project during the stage of system analysis and user requirements gathering, the transactions of an enterprise are like a “black-box” to the developer. It is difficult for the developer to gather sufficient information all-at-once, which restricts the use of Class Diagrams in establishing the O-O model. The developer may also find it difficult to adopt ERM or IDEF1X when establishing the complete data model. A structured system analysis technique seems to be the best choice. During the decomposition of the enterprise transactions from the top level to the bottom level, the developer acquires more information and some non-structured methods can be used to describe special parts of the system. Later on, at the stage of detailed system design and pilot system development, with the support of some CASE tools, UML Class Diagrams and ERM become the most suitable techniques for automatic code generation and physical database generation.

UML is more suited to development using O-O methods, but because of its complexity and dependence on very detailed information, it can be inappropriate to use during the stage of conceptual modelling and user requirements gathering. Furthermore, its complexity sometimes makes it hard for end-users to understand and is likely to lead to difficult interaction between system designer and end-user. ERM is a basic modelling method for database design and it includes some revised or enhanced methods such as IDEF1X [25] and extended ER Model (eERM) [26]. ERM can be very useful for system design and development. By using a CASE tool such as S-Designer, Visio, etc. and considering the Data Base Management System used in target system, it could be transformed from a logical data model into a physical data model thereby generating the target database automatically. There is also a relationship between UML Class Diagrams and ER Models in that they can be mapped from one another in certain conditions. ERM is also dependent on very detailed information and this makes models hard to establish during the stage of conceptual modelling and user requirements gathering.

In comparing other modelling methods such as Petri-Nets [16] and RAD [17], with IDEF0/IDEF3/DFD, even though they are also widely use, they are not as understandable as the latter three when discussing with end-users. They can generally be used in the modelling work as complements to the main modelling methods.

In the subsequent stages of detailed system design and prototype development, UML and ERM can play very important roles. But the modelling work of the requirements definition stage will be a basis for further design and development. With IDEF0 models, important information of ICOM code can be extracted, which can form a conceptual data model and push further ER modelling. In DFD models, data stores and information flow among individual functions can be seen. This is useful for understanding the function and information in an enterprise, and thus helps in Class and Database design.

The implementation of improvements to an enterprise are generally for the purpose of increasing long and short term profitability of that enterprise. The end users, referred to in this paper, are business managers and the members of their departments. To ensure a greater chance of success of project after implementation it is generally accepted that a high degree of involvement of the eventual users of the system is recommended.

Business people bring knowledge of the enterprise to the project and generally have a good idea about what needs to change in a business. They need to be able to explain this in a language that is understandable to the developers. Similarly, the developers have knowledge of the available software tools and the limitations of technology but must be able to apply this knowledge to the enterprise-specific domain. This is why a language that is understandable by both the end user and the developer is very important. The developers should not guess what the users need as this may lead to the implementation of technology for technology sake. These initiatives are driven by business and not by technology and for this reason the users and developers must have the same unambiguous understanding of the enterprise. This end is met through the use of an intuitive but structured set of modelling techniques such as IDEF0, IDEF3 and DFD. This is especially crucial at the beginning of the project. As the stage moves to detailed design and
3. Staged business modelling for system analysis and user requirements gathering

3.1. The tasks of system analysis and user requirements gathering

As an initial stage in the System Development Life Cycle (SDLC), system analysis and user requirements gathering will directly affect the ultimate success of an information system project. The tasks in this stage can be summarised as follows [18,19]:

- Make clear the current organisation of enterprise, and the transactions of each organisational unit.
- Make precise descriptions of current business processes.
- Get familiar with current IT environment, including hardware, software and human interaction.
- Gather sufficient information regarding data flows, e.g. order forms, picking notes, invoices, reports, etc.
- Realise the requirements of the problem domain and its boundary.
- Have a clear idea of user requirements.
- Finally, deliver a preliminary system design, which is approved by end-user.
- During this stage, the crucial and substantial work is to provide a set of business process models, including AS-IS models, which are established according to system analysis, and preliminary TO-BE design, which is based on user requirements gathering.

As discussed in Section 2, IDEF0, IDEF3, DFD are suitable modelling methods to apply in these situations. Using the IDEF family of modelling methods also ensure that consistent semantics are applied. This is an important factor when dealing with potentially complex models. They have been in widespread use in information system analysis and design for many years and have proven to be easy to understand by business people. There are also many modelling tools can be found to support all three of them, e.g., BPWIN [20], System Architect 2001 [21].

3.2. Analysis and comparison of the three adopted modelling methods

3.2.1. IDEF0

IDEF0 is a standard modelling method used to establish function models, which has already been accepted by most experts and end-users in this field. It was derived from a well-established graphical language, the Structured Analysis and Design Technique (SADT), and has only two types of graphic notation, the activity box and boundary/interface arrow. Diagrams are formed based on the Inputs-Controls-Outputs-Mechanisms (ICOM) Code and there are strict syntax and semantic rules, which ensure that the model is described precisely. Because of its rigor, it can be integrated seamlessly with other types of models such as IDEFIX [22].

The deficiency of IDEF0 models is that they only describe the functions, the information connection (ICOM) between them and the precedence. The logical and sequential relations among different activity units cannot be described clearly.

3.2.2. IDEF3

IDEF3 is a process description capturing method whose primary goal is to provide a structured method by which a domain expert can describe a situation as an ordered sequence of events, as well as describe any participating objects. It has less strict syntax and semantic rules than IDEF0, which makes modelling work much easier. There are two IDEF3 description modes, process flow and object state transaction network (OSTN), among which process flow is more widely used. A process flow uses Unit Of Work/Behaviour (UOW/UOB) to describe activities, three types of link to represent temporal precedence/object flow/reational. Furthermore, it defines junctions to show logic such as “and”, “or” and “exclusive or”. It can also represent synchronous/asynchronous among concurrent activities. Referents can be used to
describe the participation of an important object in an activity.

The deficiency of IDEF3 models is that they are not as standardised as IDEF0. They are not sufficiently powerful to show clearly the information flow among different activities. Therefore, they are not easy to integrate with data models such as IDEF1X or ER. The authors believe that the deficiency of IDEF3 in representing information flows can be compensated by using Data Flow Diagrams (DFDs).

3.2.3. DFD

DFDs model systems as a network of activities connected to one another by pipelines of objects. DFDs also model holding tanks called data stores, and external entities, which represent interfaces with objects outside the bounds of the system being modelled. The advantage of a DFD is that it can describe information flows clearly, from the source to the destination. The strictness of syntax and semantic rules here is at a level between IDEF0 and IDEF3. The level of difficulty in creating DFD models is also between IDEF0 and IDEF3. The deficiency of DFD is that it lacks the logic expression available using IDEF3.

A detailed comparison of these three modelling methods is presented in Table 1.

3.3. Guideline for a staged business modelling method jointly using IDEF0, IDEF3 and DFD

Based on the analysis in Section 3.2, especially the comparison result presented in Table 1, one can see that an IDEF3 model is easiest to establish and can show clearly the sequential and logical relations among activities. This is important at the beginning of system analysis, especially for the phase of end-user investigation. The DFD models can describe data/information flows clearly, which makes up for the deficiency of IDEF3. Even though IDEF0 has the fewest types of graphic notation, only boxes and arrows, it is the hardest to establish. It is a more structured modelling method, which is more focused on top-down analysis and decomposition. This feature makes it a more abstract and time-consuming piece of work, involving an iterative modelling process. But IDEF0 modelling is in no doubt the most important work in the stage of system analysis and user requirements gathering, especially for preliminary TO-BE design.

<table>
<thead>
<tr>
<th>Structure</th>
<th>Ease of creation</th>
<th>Strictness of syntax/semantic rules</th>
<th>Information expression</th>
<th>Sequential expression</th>
<th>Logical expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDEF0</td>
<td>Can be decomposed to lower levels in a straightforward manner (1)</td>
<td>Detailed models at an early stage are difficult to create (3)</td>
<td>Very strict syntax. Only two types of graphical notation means that strict rules must be applied (1)</td>
<td>Good at representing information but has no data storage notation (2)</td>
<td>Poor sequential representation (3)</td>
</tr>
<tr>
<td>IDEF3</td>
<td>Can be decomposed but not as straightforward as IDEF0 (2)</td>
<td>Intuitive and easy to understand. Can be easier to create (1)</td>
<td>Least strict of the three methods. Four types of notation allowing more flexibility of representation (3)</td>
<td>Not a strength of IDEF3. No data storage notation (3)</td>
<td>Excellent for representing the sequence of a process (1)</td>
</tr>
<tr>
<td>DFD</td>
<td>Can be decomposed but not as straightforward as IDEF0 (2)</td>
<td>Intuitive to a degree. More abstract than IDEF3 (2)</td>
<td>Less strict than IDEF0 but allowing flexibility of expression through variety of graphical notation (2)</td>
<td>This is the strength of DFD. Provides notation for data storage (1)</td>
<td>It is possible to deduce the sequence of a process using some DFDs (2)</td>
</tr>
</tbody>
</table>

1: best, 2: medium, 3: worst.
In order to combine the advantages of each modelling method and make modelling tasks easier to grasp, thus maximise the effect of modelling work, the implementation guideline for a staged modelling method jointly using IDEF0, IDEF3 and DFD is provided here. This approach, which was adopted in the e-business project SMARTISAN (IST-2000-26267) and the BPR project COST-WORTH (IST-2001-52223), is described in the case study, and proved to be very effective.

A framework of such methodology is shown in Fig. 2. The main idea here is to adopt both top-down and bottom-up thinking methods and use each of them in appropriate phases. The detailed guideline to implement such methodology is described as follows.

3.3.1. Step 1—general investigation and main function identification

This is an implicit initial step for carrying out a new IT project in an enterprise. A top-down structured thinking method is used to grasp the top-level information. A very simple IDEF0 model is used in Step 1 that describes the functional blocks of model.

3.3.2. Step 2—detailed investigation

In this stage, the modelling work switches to a bottom-up approach to gather bottom-level descriptions of the processes and data flows, which belong to separate function blocks outlined in Step 1. In order to get detailed description of business processes for each activity (function) and the information exchange, a well-structured and comprehensive interview checklist for system analysis should be carefully created beforehand. This can minimise initial scope omissions and unplanned detours [23]. This investigation can also take the form of informal interviews or observing the relevant operators. Based on the end users’ answers to the checklist and referring to Table 1, the analyst establishes the most appropriate modelling tools to represent the detailed information. The lower-level IDEF3 and DFD models are now established. These methods are suited to this stage as they facilitate detailed discussions between the analysts and the ‘non-management’ personnel. Because these personnel work at a more granular level of detail in the enterprise they possess the appropriate knowledge levels to assist the analysts in generating these detailed models. The analysts now have a clearer understand-

3.3.3. Step 3—analysis and synthesis

Based on bottom-level models generated in Step 2, the top-level diagram of the AS-IS IDEF0 model (A0) can be finished by adding arrows and ICOM code. In the meantime, the bottom-level diagrams are modified, and the hierarchical relationship between each node is added. The decomposed IDEF3/DFD models are attached and referents added to establish the relationships among them. This is an iterative process, which needs repeated analysis and synthesis so that a complete and accurate set of AS-IS models can be finished.

3.3.4. Step 4—user requirements gathering

Use IDEF0 to do a general function TO-BE design, which reflects the real requirements provided by the enterprise. It is another structured top-down process. In order to get accurate and detailed user requirements for the new information system, an interview checklist for user requirements gathering should be carefully designed beforehand. The TO-BE model should reflect the crucial business process changes compared to the AS-IS model.

3.3.5. Step 5—test and confirm TO-BE design

This is another analysis and synthesis process for TO-BE scenario. Show the initial TO-BE IDEF0 diagrams established in Step 4 to the end-user and discuss with them. Get their feedback and modify the diagrams. Decompose important functions into IDEF3 or DFD models. This has the advantage of providing system designers with a head start for further detailed technique-oriented system design, such as workflow integration and database generation. Finally, establish a consensus between modeller and end user and reach a complete set of TO-BE models confirmed by the enterprise.

4. A case study

This case study is based on an e-business project to introduce web-based trading, marketing and logistics
Fig. 2. Framework of a staged modelling methodology.
in small and medium sized enterprises (SMEs). There are several end users taking part in this project. Although their current information systems run well, they are willing to adopt new IT technologies to improve their competitiveness. The project will help them establish a set of new application modules integrating with their legacy information systems, which will strengthen their IT platform and enable them to realise their e-business goals.

At the stage of system analysis and user requirements gathering, the methodology discussed in Section 3 was adopted to establish business process models for each end user. IDEF0 was used as a tool for functional modelling, IDEF3 for process modelling and DFD for information flow modelling. This case study provides a brief view for such a modelling approach.

4.1. Background

One of the end users is a mechanical manufacturing enterprise, which manufactures car exhausts for the replacement market. It shall be referred to as “Sample enterprise” hereafter. They sell their products to specialised retailers who sell to the final customers. The sample enterprise wants to sell their products directly to the final customers. They want to provide information about their products and a list of certified installers. They also want to gather local information from the market about usage life cycle in order to provide feedback to the design teams. Stock availability, delivery and installation date can be provided thus facilitating the decision for order placement and payment [24].

4.2. Modelling approach

4.2.1. Step 1—general investigation and function identification

According to the general background established during the initial investigation, the transactions are divided into 4 top-level function blocks, as follows:

1. process purchasing (materials);
2. process manufacturing-production;
3. manage warehouse operations;
4. process sales (products).

4.2.2. Step 2—detailed investigation and bottom-level modelling

Further investigation for each function block is carried out, during which different business processes/activities are distinguished and a set of bottom-level process flow (IDEF3) models and data flow (DFD) models is established. The list of the models is shown in Table 2.

At this stage, the question arises: how does a modeller choose the appropriate modelling method? It depends on the experience of modellers and the hierarchical level of the person in the target enterprise. The answers are varied. The detailed representations of IDEF3 and DFD models are extremely useful when interacting with the people who actually perform the activity (e.g. the telesales operator who confirms the sales order or the warehouse operative who receives the raw materials). The authors’ suggestion is: when the process is focused on sequential and logical precedence, choose IDEF3 process flow; when the process is focused on information flow interchanging among activities, choose DFD; when it is focused on both, use both methods. For example, the process of “select supplier” is focused on data flows, so DFD is more suitable, as shown in Fig. 3. The process of “receive sales order” is focused on sequential and logical descriptions therefore IDEF3 is used, as shown in Fig. 4. The fulfilment of finished goods as represented by the DFD model in Fig. 5 provides another view of the sales order process but now includes the

<table>
<thead>
<tr>
<th>Business process</th>
<th>Model type</th>
<th>Belong to which function block</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select supplier</td>
<td>DFD</td>
<td>1</td>
</tr>
<tr>
<td>Measure supplier performance</td>
<td>DFD</td>
<td>1</td>
</tr>
<tr>
<td>Make purchase plan</td>
<td>DFD</td>
<td>1</td>
</tr>
<tr>
<td>Handle purchase order</td>
<td>IDEF3</td>
<td>1</td>
</tr>
<tr>
<td>Develop new products</td>
<td>IDEF3</td>
<td>1</td>
</tr>
<tr>
<td>Make production plan</td>
<td>DFD</td>
<td>2</td>
</tr>
<tr>
<td>Manufacture products</td>
<td>IDEF3</td>
<td>2</td>
</tr>
<tr>
<td>Receive raw materials</td>
<td>IDEF3</td>
<td>3</td>
</tr>
<tr>
<td>Manage stock holding</td>
<td>DFD</td>
<td>3</td>
</tr>
<tr>
<td>Fulfil finished goods</td>
<td>IDEF3</td>
<td>3</td>
</tr>
<tr>
<td>Receive sales order</td>
<td>IDEF3</td>
<td>4</td>
</tr>
<tr>
<td>Confirm sales order</td>
<td>DFD</td>
<td>4</td>
</tr>
<tr>
<td>Handle sales order</td>
<td>IDEF3</td>
<td>4</td>
</tr>
<tr>
<td>Route invoice and get payment</td>
<td>DFD</td>
<td>4</td>
</tr>
</tbody>
</table>
information storage element that was missing from the IDEF3 diagram. Table 1 presented in Section 3.2 provides a useful reference when choosing the most suitable modelling method.

The significant interaction that takes place between the analysts and the business personnel during the creation and validation of the IDEF3 and DFD models enhances the level of enterprise knowledge of the analysts. This enhanced knowledge, along with the formal descriptions of the process and information flows, enables the analysts to generate the enhanced IDEF0 for the lower levels. A detailed IDEF0 model at a given level has two primary functions: Firstly it ensures that its parent is modified to contain key interactions. The process of finalising the upper level IDEF0 models is recursive. As key interactions are highlighted in the lower level diagrams they may need to be represented at the higher levels. Secondly and, more importantly, it ascertains which processes at that level need further expansion in IDEF0. The expanded IDEF0 is further analysed using suitable models until acceptable level of detail and granularity

![Fig. 3. DFD representation of process of “select supplier”.](image)

![Fig. 4. IDEF3 representation of process of “receive sales order”.](image)
is obtained. Having established bottom-level process models for each top-level function block, enough information is available to establish the IDEF0 diagrams for each function block (A1, A2, A3 and A4). The AS-IS models that are generated at this stage of the process are used as a basis for generating the TO-BE models and thus there is a one-to-one relationship between the models for a given level of granularity. Fig. 6 illustrates one of the lower level AS-IS IDEF0 diagrams.

4.2.3. Step 3—analysis and synthesis
Based on the bottom-level AS-IS IDEF0 diagrams (A1, A2, A3 and A4), the top-level AS-IS IDEF0
model (A0) drafted in Step 1 is finished, as shown in Fig. 7.

Links are created to reflect relationships among all the models established. Appropriate modifications are also made to describe real transactions as accurately as possible. The context diagram (A0) is finally drawn to show the boundary of the whole system. Table 3 lists the complete model hierarchy at the end of AS-IS analysis.

Table 3
The complete AS-IS model hierarchy

<table>
<thead>
<tr>
<th>Node</th>
<th>Process name</th>
<th>Model type</th>
</tr>
</thead>
<tbody>
<tr>
<td>A-0</td>
<td>Run operations at sample enterprise</td>
<td>IDEFO boundary</td>
</tr>
<tr>
<td></td>
<td>Run operations at sample enterprise</td>
<td>IDEFO0</td>
</tr>
<tr>
<td></td>
<td>Process purchasing (materials)</td>
<td>IDEFO0</td>
</tr>
<tr>
<td>A1</td>
<td>Select supplier</td>
<td>DFD</td>
</tr>
<tr>
<td>A11</td>
<td>Measure supplier performance</td>
<td>DFD</td>
</tr>
<tr>
<td>A12</td>
<td>Make purchase plan</td>
<td>DFD</td>
</tr>
<tr>
<td>A13</td>
<td>Handle purchase order</td>
<td>IDEF3</td>
</tr>
<tr>
<td>A14</td>
<td>Process manufacturing/production</td>
<td>IDEFO0</td>
</tr>
<tr>
<td>A2</td>
<td>Develop new products</td>
<td>IDEF3</td>
</tr>
<tr>
<td>A21</td>
<td>Make production plan</td>
<td>DFD</td>
</tr>
<tr>
<td>A22</td>
<td>Manufacture products</td>
<td>IDEF3</td>
</tr>
<tr>
<td>A3</td>
<td>Manage warehouse operations</td>
<td>IDEFO0</td>
</tr>
<tr>
<td></td>
<td>Receive raw materials</td>
<td>IDEF3</td>
</tr>
<tr>
<td>A31</td>
<td>Manage stock holding</td>
<td>DFD</td>
</tr>
<tr>
<td>A32</td>
<td>Fulfil finished goods</td>
<td>IDEF3</td>
</tr>
<tr>
<td>A33</td>
<td>Process sales (products)</td>
<td>IDEFO0</td>
</tr>
<tr>
<td>A4</td>
<td>Receive sales order</td>
<td>IDEF3</td>
</tr>
<tr>
<td>A41</td>
<td>Confirm sales order</td>
<td>DFD</td>
</tr>
<tr>
<td>A42</td>
<td>Handle sales order</td>
<td>IDEF3</td>
</tr>
<tr>
<td>A43</td>
<td>Route invoice and get payment</td>
<td>DFD</td>
</tr>
</tbody>
</table>
Fig. 8. Top-level TO-BE IDEF0 model (A0).
4.2.4. Step 4—user requirements gathering

When a clear understanding is established of the AS-IS situation in an enterprise, it is possible to identify the functions that need to be implemented to achieve the goals set down by the business strategy. For example, the function “process sales” has evolved to become “process sales & CRM”, which is catering to e-business requirements. Where in the AS-IS model there were the inputs, “request for sales” and “payment from client”, they are still the inputs in the TO-BE model with extended definitions for both traditional mode and web mode. In addition, there are some new inputs added in the TO-BE model, which could reflect the requirements of web-based customer registration and inquiry. These requirements that were identified at the interview stage are now represented clearly in the TO-BE model.

The target system design is described using the IDEF0 method. Fig. 8 shows the top-level TO-BE model. Compared to the correspondent top-level AS-IS model shown in Fig. 7, the changes for enhanced SRM and CRM functions are obviously illustrated. The integration of legacy information system tools is also reflected in TO-BE scenario.

![Diagram](image-url)  
Fig. 9. Process sales order entry (A42, IDEF3 process flow: TO-BE design).
4.2.5. Step 5—test and confirm TO-BE design

By showing the end user the TO-BE IDEF0 model, comments are provided for modifying the design and the users are able to confirm their agreement with the final design. This also facilitates the innovative aspects of key business processes to be identified and clarified. This leads to further decomposition of the function design, so that detailed process and data flow descriptions are modelled, thus providing a useful platform for further detailed system design and pilot system development. For example, Fig. 9 shows an IDEF3 process flow which depicts the future transactions of sales order entry. This process diagram helps in the design of workflow integration.

Fig. 9 illustrates the processing of an order using a combination of the target enterprise’s legacy systems and the Smartisan tool. The Smartisan tool is a web-based catalogue, which can also facilitate interactions with relevant trades persons, or artisans, and which also has the facilities for tracking and tracing. In this process the Smartisan tool facilitates the exchange of information (e.g. order request, estimated ship dates) between the customer and the target enterprise. The order is placed either using fax or email or through the Smartisan tool. The customer’s credit rating is checked. If no credit is available then the order goes no further and the customer is informed. Otherwise the process continues by checking if the customer is satisfied with their account details. These can be changed if requested. The order is checked against the availability of raw material stock required to build it. If stock is available then, through the production planning process, it is ascertained if the estimated availability date falls within the customer’s original request date. If it does then the order is confirmed on the system and the Smartisan tool is updated so that the order confirmation can be sent automatically to the customer.

5. Conclusions and future work

Enterprises are constantly evolving and in many cases one of the consequences of evolution is the need to evolve their information systems. For an enterprise to evolve its information system in a cost effective manner that causes the least disruption to operations, it needs to adopt a structure approach that is understandable by user and developer. Models provide a means for gaining common understanding of the requirements of a new or evolving system. There are many modelling methods that could be used to do business process modelling in the scenarios described in this paper. The combination of IDEF0, IDEF3 and DFD was adopted because of its ease-of-use, understandability and wide acceptability. This paper summarises a guideline for jointly using these methods. The case study provides a means for demonstrating how the requirements stages of an e-Business project were carried out using this approach.

This work is being carried out by the authors in developing a tool to support the modelling procedures described in this paper [27]. The authors are also engaged in further research into techniques for automatic model generation based on answers to an interview checklist and the reuse of existing partial models [28].

Acknowledgements

The Authors wish to acknowledge the EU for funding the projects SMARTISAN (IST-2000-26267) and COST-WORTH (IST-2001-52223) under which the ideas presented in this paper have been developed.

References

Hui Shen received his PhD, MEngSc and BS degrees from Department of Automation, Tsinghua University, Beijing, China in 2003, 2000 and 1998, respectively. He is now a researcher at the IBM China Research Laboratory. From March 2001 to February 2002, he worked on a collaborative research initiative as a visiting student in the Computer Integrated Manufacturing Research Unit, National University of Ireland, Galway, participating in a EU funded e-business project. His research interests lie in enterprise modelling and integration, business process re-engineering (BPR), knowledge management and decision support systems. He has published more than 10 papers in journals and at international conferences.

Brian Wall is a project manager at the Computer Integrated Manufacturing (CIM) Research Unit, National University of Ireland, Galway. He holds a BE in industrial engineering in 1989 and an MEngSc in industrial engineering and information systems in 1991. From 1991 to mid 1994, he worked with Pharm-Allergan in Germany, as a planning manager. In 1994 and took up a position in the area of Supply Chain Management with Microsoft European Operations Centre in Dublin where he worked for 5 years. While there he was involved in implementing a number of supply chain management initiatives with particular focus on the areas of forecasting and distributed requirements. Since joining CIMRU in January 2001 he has been responsible for their participation in a number of EU projects, working closely with the SME participants on user requirements and concentrating his research efforts in the areas of e-business, supply chain management, business process re-engineering and enterprise modelling. He is currently writing up his PhD in area of applying enterprise engineering to e-business realisation at SMEs.

Michal Zaremba is a research engineer at Digital Enterprise Research Institute (DERI) at the National University of Ireland, Galway where he recently undertook postgraduate position after finalising his work for Computer Integrated Manufacturing (CIM) Research Unit at the same University. He holds MSc in computer science and management from the Wroclaw University of Technology in Poland in 2000. He has been a researcher in CIMRU working for number of EU projects, as well as writing his PhD degree. During his work in CIMRU, he has been working closely with end-users, responsible for user requirements gathering, software development and systems testing. He has been also working at the IT department of the National University of Ireland as a Java language tutor. His main interests are semantic web, semantic web services,
electronic business, business protocols for business information exchange and XML and EDI based standards and specifications. He is also a Sun Certified Programmer for the Java 2 Platform.

Yuliu Chen graduated from Department of Electrical Engineering, Tsinghua University, Beijing, China in 1959 and had his advanced study at the University of Strathclyde, UK. Since his graduation, he has been involved in teaching and research at Tsinghua University, Beijing, for over 40 years, and is now a professor, Department of Automation, Tsinghua University. His background is in control engineering, especially large-scale systems theory and its applications. His current research interest lies in Computer Integrated Manufacturing (CIM), especially CIM system architecture and implementation methodology. He worked as a deputy chief designer of Chengdu Aircraft Company for their CIM system design, from 1990 to 1992, and did a lot of experimentation of CIM implementation methodology in the Chinese industry under the collaboration between CEC and China as a Chinese side Project Leader. He has published six books and many papers. Prof. Chen is a senior member of IEEE, a fellow of Hong Kong Institute of Engineers, and an editorial member of international journal of Production Planning and Control.

Jim Browne is registrar and deputy president of the National University of Ireland, Galway, as well as the founder of the Computer Integrated Manufacturing (CIM) Research Unit there. He has many years experience of working in applied research and development, including extensive experience of EU and industrially funded projects. His research interests are in the design, analysis, modelling and operation of Extended and Virtual Enterprises, focusing in particular on Supply Chain Management aspects. He holds Bachelors and Masters degrees from NUI, Galway and PhD and DSc degrees in engineering from the University of Manchester Institute of Science and Technology. He has published extensively, including books such as “Production Management Systems: A CIM Perspective” (Addison-Wesley, 1998) and “Shop Floor Control Systems: From Design to Implementation” (Chapman & Hall, 1991).