Object Oriented Principles in Information Systems Alignment with Enterprise Modelling

Lecturer Cătălin STRÎMBEI, linus@uaic.ro
Faculty of Economics and Business Administration, "Al. I. Cuza" Univ. of Iaşi, România

This paper begins with an analysis of how some business or enterprise modelling approaches intend to reflect corporate goals into business process models by extensively using object oriented principles. Then we try to emphasize the object-oriented principles' potential in the strategic orientation of information systems, starting this investigation with these business models previously presented. Thus we discuss some methodological aspects of information systems development, supporting the implementation of organizational goals and strategies. Finally, we try to particularize the data organization and design aspects within the information system context approach stated above.
Keywords: *enterprise/business modelling, object oriented modelling principles, information systems development, data and database organization, and design.*

1 Introduction

In general, the term "business model" (BM) is accepted as an abstraction of a firm's strategy (Porter, 1996), thus outlining the basic details of a corporate *value* proposition from its different stockholders' perspectives.

The idea of structuring a business model is to find ways to better position the company (firm, organization) against its competition, or, in other words, to assure a competitive positioning by gaining competitive advantage.

1.1 Strategies, business models, corporate goals and competitiveness

Seldom' article (Seldom et. al., 2004) focuses on the relationship between business model and strategy as distinct concepts. The authors try to demonstrate that there is a difference between these concepts *at an abstract level*. From their perspective, business models are abstraction of strategy. Thus, business models are more inward looking than strategy: they take into consideration *the details* of *the activity-system of how a firm creates economic value*. On the other hand, strategy is more outward looking: first of all it takes into consideration competitive positioning (how to obtain a competitive advantage). So, we can conclude that business model is about generalization and strategy is about differentiation or, in other words, strategy focuses on those aspects of a business model that can lead to competitive advantage.

Business model formalisation is a key factor for the coherence of a strategy: the business model will outline and interconnect the essential "elements" of *value chain* activities and the organization will invest *in those activities that provide a competitive advantage*, and will outsource the activities irrelevant to its competitive position (Buchanan, 2003).

The key concept to understand the role of a business model is "value". In fact, value implies two interacting concepts: financial efficiency and business effectiveness (Buchanan, 2003). The former, financial efficiency is more concrete and easier to be quantified because it results from reducing costs or enhancing the financial yield from investment. The other concept, business effectiveness, is more abstract and more difficult to quantify due to its mainly qualitative side; it emerges when the company increases its market share, defeats competitors, improves quality or achieves a closer relationship with customers

1.2 Business engineering and business modelling disciplines

Modelling business processes in the larger context of an organization is the main subject of some research field and disciplines such as business engineering, enterprise or business modelling. Giaglis defines the field of Business Engineering as the *integral, concurrent*

design of organizational processes and the information systems to support those (Giaglis, 1999). Reference disciplines of business engineering include: *process-based organizational design, information system development, and information system evaluation*. The main concern of business engineering is to develop the methods, techniques and tools that help enterprises achieve change (change management approach).

In the larger context of an organization, Adigun and Biyela present a definition of enterprise engineering as *an integrated set of disciplines for building or changing an enterprise, its processes and systems. It guides management to the understanding of how to build a value stream for either a customer and/or market through effective, efficient and flexible business processes* (Adigun&Biyela, 2003: 155).

The 1990's witnessed the emergence of business process *re*-engineering (BPR) as a management area which, combined with some advanced information system technology, focused on the need to reformulate the enterprise's vision in order to find new strategic ways for better competitive positioning. BPR supposes four sub-processes: goal definition – re-engineering goals, information acquisition, modelling and evaluation (Tsalgatidou, 1995). So, business modelling is viewed as a constituent part of a larger process – business process re-engineering. In this context, Blyth outlined several research directions and issues, among which *business system modelling and analysis* (Blyth, 1998), that is concerned with the creation and the analysis of a quantitative and qualitative business model.

1.3 Main principles of business processes modelling

Tsalgatidou sees business process *modelling* (in the larger context of BPR) as consisting of three tasks: *Choose Modelling Philosophy, Choose a Modelling Formalism* and *Apply Formalism to the selected Business Process* (Tsalgatidou, 1995). In his paper, business process modelling approaches are classified under two modelling paradigms, namely task-oriented paradigm and business policy-oriented paradigm.

Two very important issues in process modelling are the *information to be modelled* and the *modelling formalism*. The information is modelled mainly as **business Process Objects**: a business model should cover all process-related information at two levels: at *re-engineering level* and *at implementation level*. So, there are two levels of abstraction, and the intersection of them is *called core business process information.* The objects comprised in this intersection area contain information about: *activities* [which are the basic elements on which a business process is built up], *control* [of a business process which describes when and which activity is executed], *resources* [which are assigned to activities. These are objects that are necessary for the execution of activities], *organisational structure* [which can consist of organisational units, people, roles, competence]. The resource flow, which shows how resources are exchanged between activities, is represented using a **has in-/output** relationship between activities and resources, and the assignment of an object of the organisational structure to activities is represented using a **has actor** relationship between activities and actors from the organisational structure. Also, Tsalgatidou presents two approaches concerning modelling formalism: a task-oriented approach and a business policy-oriented approach. The latter is more appropriate to object oriented principles. Thus, the main idea of the *business policy-oriented approach* is to explicitly represent the business rules of an organisation. A modelled business process is defined as a tuple <P, A, I, RG, RS, RE> where: *P* is a set of business process classes = {P1, P2, …}, *A* a set of activity classes = {A1, A2, …}, *I* a set of information classes = {I1, I2, …}, *RG* a set of generation rules = {RG1, RG2, …}, *RS* a set of sequencing rules = {RS1, RS2, …}, and *RE* a set of execution rules = {RE1, RE2, …}. The rules describe the guidelines of an organisation and are represented by production rules. Rules are usually used in more than one business process, and are collected to rule classes. *Generation rules* determine the activities the business process consists of (by

setting specific **goals** in the attribute of business processes), *sequencing rules* describe the **control** of the business process and thus determine the order of the activity instances during execution of the business process, *execution rules* **select** the **actors** of activities and **determine** which **information** objects are needed for the execution of activity instances. **Activities** are structured in an object hierarchy. They can be system activities, which are executed without any user interaction or activities that are being activated and executed by one or multiple persons. **Information objects** correspond to resource objects and can be seen as documents or data. They are used for describing the states of the business process when it is executed. They are also described as classes that are instantiated at execution time. **Process instances** describe global information about the business process such as information objects, start time, current activity, and goals (Tsalgatidou, 1995:21-23).

Regarding the same issue, but highlighting more the object oriented potential for business process modelling, Erikkson and Penker, on the one side, and Marshall, on the other side, propose two frameworks based on concepts similar to the business-policy approach presented by Tsalgatidou. The model proposed by Marshall abstracts the following elements as basic concepts of the UML meta-model: *policy, objective, process, step, value, artifact, party, and situation*.

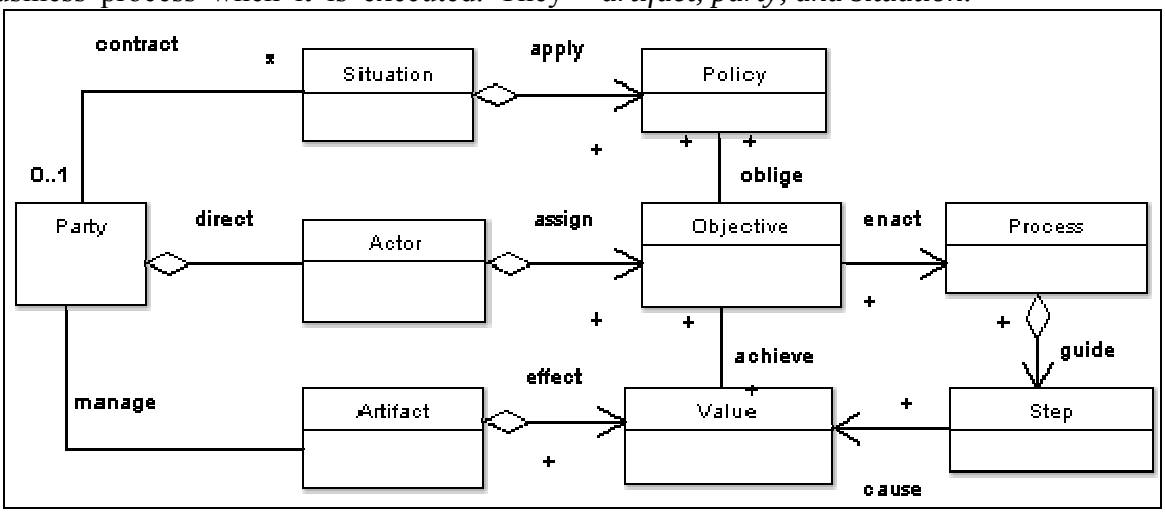

Fig.1. Marshall's business-modeling framework

Source: Marshall, 2000: 120.

A brief description of this meta-model can be reproduced as follows: the organization purpose describes the organization's value to its shareholders, customers, suppliers, employees, and other stakeholders to define why it exists. **Purpose** includes vision, missions, *goals*, and *objectives* of an enterprise, in which high level vision and missions are abstract and difficult to quantify. **Resources**, used by actors to achieve their assigned objectives, are reflected by the artifacts that induce value produced by the **processes** activated by the objectives defined by the policies applied to various situations.

Eriksson and Penker also start their reasoning from the premise that a business or enterprise represents a complex system that has a specific purpose or goal. All business functions interact to achieve this goal (Eriksson&Penker, 2000). Organizations differ in scope and internal architecture, but similar concepts can be used to describe their structures and operations. A business model represents a particular description using these concepts. The proposed meta-model is structured around concepts such as resources, processes, goals and rules. **Processes** manipulate (use, consume, refine, or produce) **resources** (materials, information, products, people). Also, processes represent **activities** performed inside an organization. They modify the state of resources and are governed by **rules**. **Goals** reflect the desired states of resources and are achieved by processes. Also,

goals can be expressed as one or more rules. These rules govern the business and they influence how resources are structured in order to achieve goals. There is an obvious relation between the two meta-models presented above in terms of how they conceive modelling. Thus rules are roughly equivalent to policies, resources to artifacts, problems to situations, and processes have about the same meaning in both contexts.

Consequently, before deciding on processes and the ways to use resources, the meta-models above emphasize, fist of all, the interdependencies between *situation-policy-objective* and *problem-goal-rule*. Furthermore, the goal or objective determines essentially the manner in which processes will run and the manner that resources will be used, this approach being a basic principle of strategic management. After have been established their basic principles, these modeling frameworks approach *the purpose, goals, and strategies modeling* in a distinct manner, as an essential theme.

Dwelling on the same issue of modelling for business purposes, Kavakli and Loucopoulos propose an interesting framework based on a modern organizational approach. The objective of Kavakli's *Enterprise Knowledge Development* (EKD) *framework* is to provide a systematic approach to developing and documenting enterprise knowledge, thereby helping organisations to conscientiously develop models to implement changes (Kavakli&Loucopoulos, 1999). This framework consists of *four different enterprise knowledge models:* (1) knowledge about the *current enterprise goals* and how they are achieved through the current enterprise behaviour (*As-is Model*); (2) knowledge about the *stakeholders' change goals* and how they can be satisfied (Change model); (3) knowledge about the desired enterprise situation – *future enterprise goals* (To-be model); (4) knowledge about the *stakeholders' evaluation goals* (Evaluation model). These four model types correspond to four distinct states that could form the "life-cycle" of organisational changes. EKD is achieved through the use of: a *common set of concepts* to describe enterprise knowledge regarding organisational change (EKD enterprise *ontology*), and a methodology roadmap plus its associated guidelines for navigation within the space of the possible routes connecting the four knowledge states. Thus, the EKD ***ontology*** provides the *conceptual modelling framework* to describe knowledge about the four knowledge states by integrating two complementary views: *enterprise goal view* and *enterprise process view*. EKD *goal submodel* uses a 'network' of goals to express the casual structure of an enterprise, in terms of ends-means relations from 'intentional' objectives that control and govern the system operation to the actual 'physical' enterprise processes available to achieve these objectives. The EKD *process submodel* represents the behavioural aspects of an organisation in terms of *roles* that are played collaboratively by *enterprise actors* to operationalize the enterprise *goals*, and dependencies between these roles. Using EKD ends-means links, changes in enterprise goals will propagate top-down as reasons or requirements to reorganise enterprise processes. Changes in the physical basis of the enterprise will propagate bottom-up indicating how new operational conditions affect the organisational objectives (Kavakli&Loucopoulos,1999:8-9).

2. Information system design and development alignment with business process design

A general literature survey reveals the strategic importance of the design coordination relationship between information systems and business processes; we have found some pertinent argumentations regarding this problem, and also several methodological approaches.

2.1 Reasons for information system design strategic coordination with BP design

The opportunity of using information systems for business process change is outlined by Giaglis, who states that IS should be introduced for business process transformation: *business goals can be supported by redesigning the existing process while at the same time considering how IS support new processes* (Giaglis,1999: 3). One-way to bring BP

design and IS design together is *to incorporate high-level IS design into business process design projects*. Thus, this author emphasizes the importance of IS as an enabler of organizational change, and, due to their recursive relationship, IS and business processes need to be considered together.

Also, Buchanan specifies the need of a systematic approach to align IT projects with corporate goals and priorities, because *the business architecture is constantly being refined or revised as strategists and corporate planners identify environmental trends that require new corporate responses or priorities, and as corporate responses are modified, the information and technical architectures must be modified. That, in turn, leads to new application and infrastructure development priorities* (Buchanan, 2002).

In the same line of thought, Kavakli and Loucopoulos notice the need for *integration of both business processes and support systems* and *externalization of business practices*. These authors outline the fact that a new role of IS has emerged (besides serving traditional business): *the potential for such systems to adopt a supervisory and strategic support role* (Kavakli & Loucopoulos, 1999:3).

Teufel observes that the role of IT in organizations has shifted beyond its initial use as back office support towards an **integral part of the competitive strategy** of the organization (Teufel,1995). In 2003, Raul J. Paul, before fully elaborating his research assuming a layered relationship between business processes and information technology, observes that IT should be seen as an enabler of organizational change rather than as a tool to implement business processes, and that the success of IT in enabling business process reengineering lies in *information systems strategy integration* (Paul, 2003).

2.2 Basic principles of information system design within the larger context of defining business architecture for competitiveness

In what follows, we will present several approaches to the integration of business processes and information systems design. These approaches will serve to formulate several assertions and guidelines of how to build information systems in order to better support corporate goals and strategies.

Giaglis' approach is based on a strategy where IS design is treated along two dimensions: the former is concerned with *the organizational impact of IS*, whereas the latter is concerned *with technical implementation* details. The declared goal of this research is to study IS *evaluation* in the context of business engineering. Giaglis declares that IS evaluation *should abstract away from technical details and focus on justifying the need for, and the cost and benefits associated with, the development of a system in terms of its impact on business processes and organizational performance* (Giaglis,1999:12). The author notes that classical and accounting techniques may be inappropriate to assess IS investments, and he argues that from the many IS evaluation methods that exist, only the "experimental" ones (i.e. prototyping and evaluation) can address the issue of estimating the expected impacts of a proposed IS on business performance. In addition to this, Ray J. Paul proposes the *BPISSS framework* (BP and IS simulation) that attempts to portray the behaviour of both IS and BP using *discrete-simulation techniques*. The major objective of the BPISS it to provide guidelines to develop a simulation model that provides stochastic measurements of the way business process and information system behave, thus assessing the impact that IS may have on BP. To achieve this objective, the BPISS framework is divided into a number of tiers, as follows:

- T1: develops BPS (Business Process System) model.
- T2: identifies BP limitations and possible IS solutions
- T3: Identifies and captures IS NFR (non-functional requirements)
- T4: describes current IS functionality. The aim is to capture three major aspects of the current IS. First, to identify the overall workflow of the IS activities that are related to BP; second to identify and understand how the data manipulated by IS may affect BP en-

tities, thus BP performance; third, to identify the IS operations performed during the process flow.

• T5: maps BP and IS Entities. A *framework* that is proposed assumes two different types of entities: the former is named *Record Entities* (RE) which are those entities found at the business level and usually represent objects that contain information and are used in the BP model to represent process behavior; the latter is named *Field Entities* (FE) and it consists of a collection of entities that represent the information contained in a RE. IS uses the information conveyed in these entities to perform part of its functionality.

• T6: develops the As-Is BPISS model using the information from previous tiers.

• T7 and rest are concerned especially with verifying and validating of the BPISS model.

Reflecting on how to "bridge the gap" between the concerns of corporate strategists and IT project managers, or, in others words, between business strategy and the implementation via technical infrastructure and application systems, Buchanan proposes a framework to support *Enterprise Architecture (EA)* processes. Enterprise Architecture describes *the logical linkage between enterprise business, information and technical architectures and the enterprise solutions architecture* (Buchanan,2002:2). Thus, EA is extended so that the hierarchy of architectures extends from the level of business strategy and links to the IT implementation level. The extension process is based on the concept of *value*, or, more accurately, on the concept of *value chain* proposed by strategic management discipline[*]. Thus, Buchanan proposes the idea of *extended value chain* of core business processes taking into account that processes or activities are enabled by the flow of information. Therefore, the value chain concept is converted into an *extended information value chain*. In this context, Enterprise Architecture is a top-down **business strategic** driven process that **coordinates** the parallel, internally consistent development of: *enterprise business architecture* (EBA), *enterprise information architecture* (EIA), *enterprise-wide technology architecture* (EWTA), as well as *the enterprise application portfolio* (EAP). This approach represents a **holistic** expression of the enterprise's key business, information, application, and technology strategies and *their impact on business functions and processes*. The EA framework artefacts consist of a *Common Requirements Vision* (CRV), *Conceptual Architecture* (CA), as well as current and future state *models*. These elements are instantiated and interrelated within all four key architectures mentioned above. So, at the EBA level they reflect business strategies, the assets and processes that articulate enterprise value chain; at the EIA level they reflect information strategies and mirror enterprise value chain into extended information value chain; and, at the EAP and EWTA level, they reflect the details of the enterprise's technology strategies, and its extended technology linkage. Hence, the artefacts at one level mirror those of the previous (superior) level so that, ultimately, the conceptual models of consecutive levels form an extended chain by derivation. To support this type of derivation from business architecture to information systems implementation, Buchanan suggests that the OMG's MDA approach can be used.

Using MDA, OMG has defined a *model-based approach* to application development. UML, because of its flexibility and ability to extend across multiple business processes, can be used to model functional business patterns *and provide a foundation for deeper application design,* by *integrating the two approaches.* What the OMG proposes (through the MDA approach) is that companies create high-level UML models of how application will be structured and integrated. These descriptions will be independent of any actual implementation details. From such a high-level UML model (Platform-Independent-Model or PIM), a more constrained UML design (Platform-Specific Model or PSM) can be generated. A PSM design can then be converted into a language

[*] Michael E. Paper, in his papers *Competitive Advantage* and, later, *Competitive Strategy*, shows that a value chain includes all of the business processes from the customer order to delivery of the final product.

code designed for a specific platform. So MDA offers companies a standard way to specify an *enterprise business architecture model*. Changes in the business architecture can be quickly incorporated into the MDA model of the IT organization which can then regenerate the interfaces needed to support changes in the current environment.

Buchanan offers an abstract or general vision to start a coherent common design of business (enterprise) and information systems (applications and technical infrastructure). Focusing on the same issue, Eriksson and Penker present a more detailed systematic approach. In one of the previous paragraphs we have already presented the framework proposed by Eriksson and Penker for business modelling. In fact, these authors have a more holistic approach using a business architectural system that consists of four views: business vision view (goal structure), business processes (activities and value created through them), business structure (resources and their relationships), business behavior (individual behavior of each important resource and process) (Eriksson&Penker, 2000). In this approach, business vision view, which depicts corporate goals, must be modelled before process modelling. Along with the goal model the business conceptual model will be developed. With clarified goals and business concepts (meaning resources: things and information) business processes view modelling can take place. The context of a business processes consists of:

- goal objects - a goal object (from goal model) has to be allocated to a processes (the connection between the processes and its goal is modelled as a UML dependency relationship);
- input objects – these objects come from the conceptual model (resources) and are either consumed or refined in processes;
- output objects – these object are also presented first in the conceptual model, but they are produced by processes;
- supplying objects – these resources participate in the process but they are not refined or consumed;
- controlling objects – these resources control or run the process.

In this business modelling context, Eriksson and Penker propose their own business extension (UML has the advantage of adaptability by using extension mechanisms): *assembly line diagram*. The assembly line diagrams are meant to model information systems that support processes. These diagrams consist of process diagrams (placed on the top) and a number of horizontal packages (placed on the bottom). These horizontal packages are stereotyped as <<assembly line>> and represent groups of information objects from one specific class or from different classes. The purpose of this kind of diagram is *to demonstrate how the processes in the upper part of the diagram* (business model) *write and read objects in the assembly line* (information model) (Eriksson&Penker,2000:114). Reading and writing represents connection links or references modelled vertically as dependency relationships and they are laid out so that interpreting the diagram must be made from left to right. A set of these references typically identifies a use case in the information systems, and information objects from the assembly lines represent types from the information system conceptual model (known in information system theory as business model). An assembly line package could *represent an entire information system, a subsystem in an information system, a special category of classes in an information system, or a specific type or group of resources*. Thus, what is interesting in the Eriksson and Penker approach is that they use object-oriented modelling possibilities from business architecture to information systems, and not from software modelling, to business architecture.

3. Database design in the context of information systems design and business modelling

As all database practitioners know, the literature concerning database design stages reveals three modelling levels: *conceptual level, logical level and physical level* (Date,2000), (O'Neil,2001), (Teorey,1999).

Tackling the issue of integrating database modelling and design with object oriented methods and methodologies used at a large scale, Naiburg and Maksimchuck try to structure database design processes with respect to the modelling level, but using UML object oriented methodology tools (Naiburg&Maksimchuck,2001).

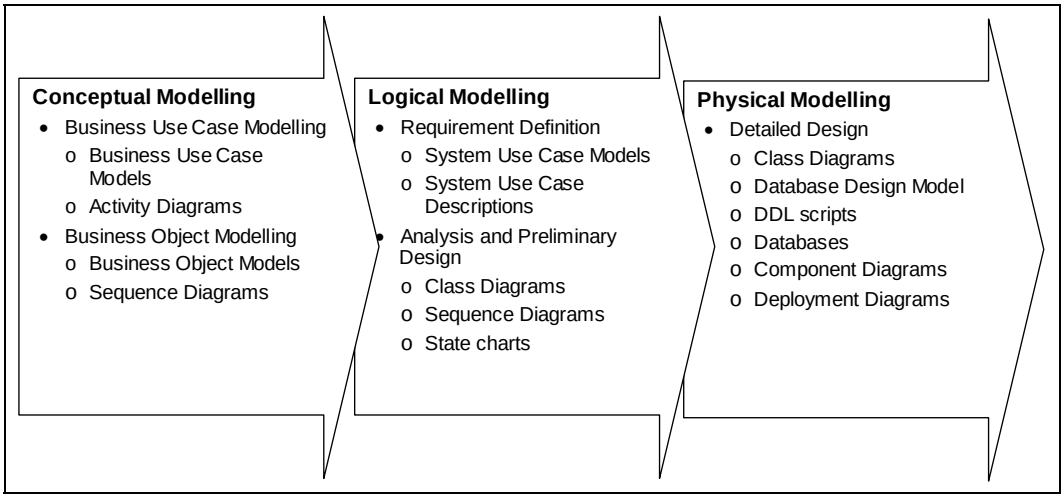

Fig. 2. Using UML in database design process
Source: Naiburg and Maksimchuck: 2001: 284

In the following table we resume the reasons of using object oriented methodology to formalize conceptual and logical models implied in database design.

Table 1. UML artifacts useful in database design process

Diagrams	**Description**
Use cases	Reflect business function or processes driving conceptual database design
Activities and Interactions	Reflect business processes flow and access requests, as queries and writing operations, that will affect the security model and even the physical access structures (like indexes)
Classes	Reflect business entities structures and relationships that will ultimately determine the physical database design structure

The main advantage of this approach is the integration of traditional database modelling principles with the intimate object oriented structures used in the larger context of information system design process. This integration will make the connection between the business or enterprise vision (goal model) and the database design models.

First of all, we can identify which business objects are reflected in software, from which business processes and with which business goals attached. Eriksson and Penker propose (as we have shown previously) process and assembly line diagrams to indicate corresponding software classes from software system and subsystems. They claim that resources used in these diagrams are frequently reflected in software, and the processes can also be represented in software as *processes supporting objects*. Ultimately, they propose a meta-model showing category of specification classes in the business model and categories of implementation classes in the software model.

As we can see, this meta-model divides resource objects into *things* and *information*. This subtle distinction is motivated by the fact that *the information of the business resource concepts is implemented to the information system* and *the actual resources cannot usually be implemented in software*; failure to make this separation of information by the object itself is *one of the reasons that translating business models into information systems has proven to be so difficult* (Eriksson&Penker, 2000: 380-381). The business

objects involved in the collaborations revealed by assembly line diagrams, are categorized as *active (process supporting objects), reactive (business event objects)* or *passive*. Passive objects are referred to as *entity objects* in UML class diagrams. They are the main interest for database design because they hold information that will persist in a database. So, resource diagrams (including information and organization diagrams) that are part of the business structural view represent a valid basis for identifying entity objects because information and state of resource (which can reflect the accomplishment of goals) are stored in the information system databases.

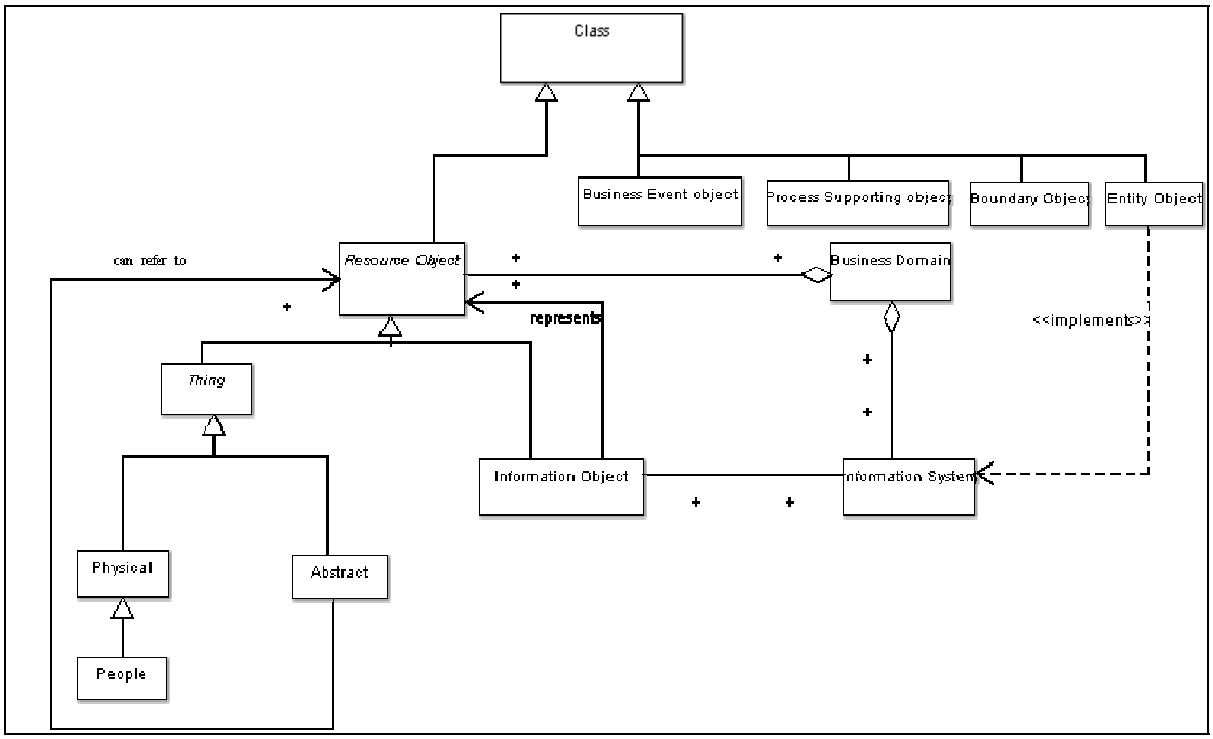

Fig. 3. Mapping objects from business models with software objects from information system models

Source: Eriksson and Penker: 2000: 380

With respect to these entity objects, Tsalgatidou and Junginger state that **information objects** correspond to resource objects and can be seen as documents or data (Tsalgatidou&Junginger, 1995). These objects are reflected also in the BPISSS framework proposed by Ray J. Paul as Record Entities (RE): *which are those entities found at the business level and usually represent objects that contain information and are used in the BP model to represent process behavior* (Paul, 2003:1789). Tier 5 of the BPISSS framework aims to identify all RE used in the business process level and to map them to their corresponding FE, to map the FE to the corresponding process/activity in the BP model and to identify the operations performed during these activities and that affect RE or FE using a table scheme.

4. Conclusions

In this paper we have tried to outline the importance of the integration of business process and information systems, so that traditional IS and database design methodologies could be extended towards a more architecturally holistic approach. Starting with the production of business models is a valid approach to better understanding the intimate business mechanism to better clarify the context of implementing competitiveness strategies. Business modelling is aimed first to understand and find new ways to improve business structure and operations. Building a "bridge" between business processes and in-

formation systems, so that large areas from business models can be mapped directly onto software objects (process supporting objects and entity object that persist in database structures) will be the agent that will integrate information systems into business. As a result, information systems will be more easily updated and evaluated from a corporate strategy point of view. To use a common modelling approach at all enterprise architecture levels (business, information and technical architecture) is relatively a difficult task, but to use UML object-oriented structures and extension mechanisms can provide a reasonable and feasible solution. This solution has several advantages and implications at database design level so that business modelling and database modelling have common ground to build a solid relationship.

Bibliography

Adigun, Matthew O., Biyela, Diligent P. *Modelling an Enterprise for Re-engineering: A Case Study*, Proceedings of SAICSIT 2003, pages 153-164

Blyth, Andrew *Research directions and emerging issues in business process re-engineering*, SIGGROUP Bulletin, August 1998/Vol. 19, No.2

Buchanan, Richard D., Soley, Richard Mark *Aligning enterprise architecture and IT investments with corporate goals*, OMG and Meta Goup Whitepaper, 2002

Date, C.J. *An introduction to Database Systems 7nd Edition*, Addison-Wesley 2000

Erikkson , Hans-Erik, Penker , Magnus *Business modelling with UML: business pattern at work*, John Wiley & Sons, Inc. 2000

Giaglis, George M (1999), *On the Integrated Design and Evaluation of Business Processes and Information Systems*, Communications of the Association for Information Systems, Volume 2, Article 5, July 1999.

Kavakli, V., Loucopoulos P., *Modelling of Organisational Change Using EKD Framework*, CAIS, Volume2, Article 6, July 1999

Marshall, Chris *Enterprise modelling with UML: designing successful software through business analysis*, Addison Wesley Longman, Inc. 2000

Naiburg, Eric J., Maksimchuck, Robert A. *UML for database design*, Addison-Wesley 2001

O'Adigun , Mattew, Biyela , Diligent P. *Modelling an Enterprise for Re-engineering: A Case Study, Proceedings* of SAICSIT 2003

O'Neil, Patrick, O'Neil, Elizabeth *Databases—principles, programing, and performance 2nd Edition*, Morgan Kaufmann Publishers, USA –San Francisco 2001

Paul, Ray J. *Simulation for Business Processes and Information Systems Design*, Proceedings of the 2003 Winter Simulation Conference S. Chick, P.J. Sanchez, D. Ferrin, D.J. Morrice, eds.

Porter, M.E. *What is Strategy?* Harvard Business Review 74(6), 1996

Seldom, Peter B., Lewis, Geoffrey P., Freeman, Phil, and Shanks, Graeme *The case for viewing business models as abstractions of strategy*, CAIS, volume 13, 2004

Teorey, Toby J. *Database modeling & design 3nd Edtion*, Morgan Kaufmann Publishers, 1999

Teufel, Stephanie , Teufel, *Bernrd Bridging Information Technology and Business – Some Modelling Aspects*, SIGOIS Bulletin, August 1995/Vol. 16, No. 1

Tsalgatidou, Afrodite, Junginger, Stefan *Modelling in the Re-engineering Process*, SIGOIS Bulletin, August 1995/Vol. 16, No. 1