RJ10488 (A1108-010) August 15, 2011
Other

IBM Research Report

Entity-Centric Operations Modeling for Business Process Management Part 1: A Multidisciplinary Review of the State-of-the-Art

Jorge L. C. Sanz
IBM Research Division
Almaden Research Center
650 Harry Road
San Jose, CA 95120-6099
USA

Research Division
Almaden - Austin - Beijing - Cambridge - Haifa - India - T. J. Watson - Tokyo - Zurich

Entity-Centric Operations Modeling for Business Process Management Part 1: A Multidisciplinary Review of the State-of-the-Art

Jorge L. C. Sanz
IBM Almaden Research Center

'An entity is actually in a state of flux. It is the subject of a process'.
K. A. Robinson, 1979

'Ten years before "objects" became a buzzword in information technology, information system designers have promoted the idea of describing a process by the entities, relationships and attributes that were manipulated. The natural next step for them has been to describe integrity constraints, expressing prohibited states for entities, relationships or attributes, and more recently, to describe their state-transitions'.
H. Tardieu, 1992

Abstract

The entity-based approach for operations modeling was published for the first time three decades ago. Specifically, the notion of entities as the main subjects of processes and entity life-cycle as a technique for dynamic modeling of operations were introduced independently by K. Robinson in 1979, C. Rosenquist in 1982 and M. Jackson in 1983. This modeling work emerged in clear contrast with static entity-relationship modeling found in the data-base tradition. These three pioneer contributions and other substantial research done at the realm of information engineering, structured systems analysis and social sciences in the 80's and 90's have established an important foundation for business operations modeling.

On the other hand, Business Process Management (BPM) has continued to receive great attention from practitioners and scholars. In spite of its steady growth, the industry side of BPM seems to have evolved somewhat unaware of related progress in the above sister disciplines. Specifically, recent claims on the need to integrate information and activities in process modeling and some rediscoveries of core ideas from entity-based dynamic modeling offer some examples of the disconnection. These and other findings suggest that the BPM field may not have yet fully benefited from the work done in the tradition of structured analysis, information engineering and process theory schools. Furthermore, the possibility of using entity life-cycle for modeling operations addressed by Case Management is an important byproduct. Entity-based life cycle offers a conceptual framework to integrate different types of enterprise operations whose modeling has not yet been reconciled in the BPM tradition.

Part 1 of this series presents an in-depth, multidisciplinary review of the state-of-the-art on entity life cycle modeling. The focus of this review is exclusively on modeling concepts and methodology while tools, programming models and other aspects of entity-life life cycle implementation will be addressed in companion papers. This will help pave more holistic approaches to business process modeling by benefiting from the work in different disciplines. Part 2 of this series builds further by presenting new horizons in entity-based modeling. Process theory and industrial organization show that processes have different structure and dynamics. However, most of the processes used in BPM are of a *factory* or *production* type. Part 2 will show that not all operations that matter can be addressed through the *factory approach* to entities. This will be illustrated with a wide family of enterprise operations called *oversight processes.* Furthermore, not all operations may be adequately explained by a life-cycle theory of subject evolution. Part 2 addresses a different developmental foundation that brings to bear related seminal work from Social Sciences. An abstract entity, called *driver*, will be used in Part 2 to show the value of entity-based modeling for business analysis beyond processes. *Drivers* provide a suitable *subject* that enables a more rigorous approach to so-called *value-streams*, an extensively used business term also needing formalization.

1. Introduction and Overview

Business processes have attracted a great deal of attention from different communities of practice and scholars for several decades. Business Process Management (BPM), i.e., the analysis, design, implementation, optimization and monitoring of business processes [1] has emerged as an area of study from different sciences. The BPM market builds upon the need from enterprises to reduce operating costs through process automation and optimization. Being one of the main hinges between theory and practice of business operations, BPM enjoys contributions from several domains of research such as economics, social sciences, engineering and computing.

The evolution of BPM has not happened without significant struggles, divergence and to some extent, also confusion. The state-of-the-art is plagued by language chasms, cultural silos and idiosyncratic viewpoints. Some of these challenges were documented recently in [**Indulska et al**, 2009], [**De Man**, 2009a], [**Recker et al,** 2009], [**Reijers et al,** 2010] and others. In [**Reijers et al,** 2010], the authors state the BPM challenge in clear terms: "*Considerable confusion exists about what Business Process Management entails …".* The authors in [**Indulska et al,** 2009] present a list of main issues faced in business process modeling. The wide number of distinct subjects and priorities raised by three types of experts who participated in a Delphi study (academics, vendors and practitioners) provides evidence about the struggle in BPM. The need for aligning the research agenda to the main challenges faced by industry is also called out in the closing recommendations from the study in [**Indulska et al,** 2009]: " *… despite being an actively researched field, anecdotal evidence and experiences suggest that the focus of the research community is not always aligned with the needs of industry".* The authors in [**Reijers et al,** 2010] also addressed the importance of rooting BPM activities in industrial practice and correctly questioned the understanding of the actual adoption of BPM by organizations: *"… it may come as a surprise that contemporary insights are missing into which categories of organizations are adopting BPM and which type of BPM projects they are carrying out".*

While the goal of this paper is not to provide a survey of BPM, it will be useful to illustrate some of its confounding aspects. This discussion will also help position the main subjects of this series within a context that should be valuable to practitioners and scholars. The first BPM challenge is the very meaning of business process. In spite of several decades of work in this area, the definition of business process is still troubled by ambiguity. A review of the literature shows that there is not a single, formal definition. Following [**Davenport**, 1993], a business process is *"a structured, measured set of activities designed to produce a specific output for a particular customer or market*". According to [**Weske,** 2007], *"a business process consists of a set of activities that are performed in coordination in an organizational environment. These activities jointly realize a business goal*." On the other hand, [**Hammer & Champy,** 1993] state "*a business process is a collection of activities that takes one or more kinds of input and creates an output that is of value to the customer*' and in [**Smith,** 2006], a process is defined as *"the complete and dynamically coordinated set of collaborative and transactional activities that deliver value to customers."*

[1] In [**Van der Aalst et al,** 2003b] the authors state that BPM ' … *supports business processes using methods, techniques, and software to design, enact, control and analyze operational processes involving humans, organizations, applications, documents and other sources of information".*

Many other definitions using similar but not identical terms also exist, such as those in [**Debevoise,** 2005], **[Guha et al,** 1997]**,** [**Johansson et al,** 1993], [**Ould,** 1995], and the list goes on [2].

Object Management Group (OMG) recognized the foundational problem with the definition of process. In [**Siegel,** 2008], the author states: "*there is no agreed-upon industry definition of Business Process. Instead, there are multiple definitions, each looking at the field from its own unique point of view, concentrating on its own set of concerns*". Certainly, it is not a matter of one definition being right and the others being wrong, as also stated in [**Siegel,** 2008]. But the varying points of view used in these definitions cannot be attributed only to differences between rigorous specifications and practical concepts from consultants or analysts: the foundations under differing views are not identical [3]. As a consequence, the main efforts in process modeling standardization have not yet yielded the expected outcomes, as discussed in [**Recker,** 2010] and more broadly exposed in [**Indulska et al,** 2009].

If taken *verbatim,* some of the above definitions of process may be interpreted in such a general form that almost everything an organization does could be included within BPM. A consequence of this ambiguity is the difficulty to draw a distinction between the organizational concerns addressed by business process management from those which are not. Unquestionably, most people do have a similar and informal notion of "business process". But this intuitive agreement does not mean a convergence across viewpoints. In fact, the variations in the definition of process suggest that the term may be a *boundary object* across disciplines, individuals from different units of an organization or communities of practice. In other words, the notion of business process may be a common denominator linking different organizational phenomena, a reification of a non-existing artifact or a convenient term to denote a wide and general class of organizational behavior.

Other disciplines in Social Sciences have also focused extensively on the concept of process and its definition. In a seminal paper [**Van de Ven & Poole,** 1995], the authors proposed to "*reduce the confusion*" by distinguishing between three meanings of process, namely, (1) a logic that explains a causal relationship between independent and dependent variables, (2) a category of concepts of variables that refers to actions of individuals or organizations, and (3) a sequence of events that describes how things change over time. In fact, Van de Ven's work addressed these concepts in the context of one of the most complex and critical types of processes in organizations, i.e., the strategy process. The depth of Van de Ven's classification reveals the foundations underlying most business process definitions in BPM. In spite of having been published two decades ago, this work has gone unnoticed in most of the BPM literature [**Toussaint et al,** 1997], [**Aguilar-Saven,** 2004], [**Lu,** 2007], [**Klein,** 2007], [**Ould,** 2005], [**Ko et al,** 2009], [**Recker et al,** 2009], [**Trkman,** 2010].

---

[2] Interestingly, Van der Aalst excludes in his definition certain families of processes such as strategy processes.

[3] The comments from a Masters student posted late in 2010 show that the ambiguity of the BPM definitions is a topic of struggle beyond academics, analysts and standards organizations: "*When I started writing my master thesis I came across loads of BPM definitions* […] *it seems that businesses like Wikipedia better than academic materials and actually I have to admit that we also use this definition* [text extracted from Wikipedia] *in our company because it is more self explaining than the other definitions*" [**Veldhuizen**, 2010].

Another challenge faced in BPM comes from the restrictions imposed on the type of business processes through which the above ambiguities may be resolved. The issue is that these restrictions are not always established clearly or maintained distinctly in the course of the work reported, as illustrated by work done on 'Workflows'. With some notable exceptions like the contributions from Van der Aalst et al. [**Van der Aalst & Berens,** 2001; **Van der Aalst et al**, 2000, 2002, 2003a, 2003b, 2004], restrictions made on the class of processes being modeled as 'workflows' get blurred and consequently, business process and workflow are often used interchangeably. For instance, it is common to find books and papers talk indistinctly about processes and workflows even if their target is only the latter [4]. Furthermore, most tools known in BPM provide the capability to design 'flows of tasks'. In spite of the fact that not all processes can be modeled as flows, these tools are said to represent 'business processes' without revealing much about the underlying restrictions on the type of processes actually supported.

The third example is concerned with language chasms across different schools of thought or communities of practice. In particular, there is an unclear relationship between the concept of business process and that of *organizational routine*. Rich literature is available on the study of routines [**Becker,** 2004], the significance of routines as a unit of analysis for organizations [**Pentland & Feldman,** 2008], [**Levin,** 2002], the collectivist meaning of routines and the need for establishing solid micro-foundations [**Felin & Foss**, 2004], and others. It is very likely that business process and routine address identical concerns in organization theory; however, in spite of the prolific technical production in the two subjects during decades, their formal relationship and the reasons for keeping two different terms remain unclear.

The last example is the relationship between BPM and *Case Management* (more recently also called *Adaptive Case Management* by the authors in [**Swenson**, 2010] and *Dynamic Case Management* by analysts in [**Forrester,** 2011]). The need for Case Management has been illustrated with different enterprise operations such as claim processing in insurance and account opening in banking. However, the reasons why these operations are amenable to case modeling and not fit for process modeling are rather obscure. Specifically, it is not clear whether the need for Case Management is the result of the limitations of BPM software systems or due to the focus of the BPM discipline on specific processes (i.e., those that can be described by using the BPMN standard).

Van der Aalst and others [**Van der Aalst et al,** 2003c] presented *Case Handling* as a new paradigm for supporting *flexible* and *knowledge intensive* business processes. Henk de Man [**De Man,** 2009a] states that 'workflow' is an adequate representation for factory-type, highly predictable behavior admitting for little or no deviation from pre-established models. De Man also recognizes that this type of 'factory' behavior is not all what organizations do. In recent literature ([**Khoyi,** 2010]), the argument in support of the need for Case Management hinges around the fact that "*Case Management allows the business to be described in known terms rather than artificially fitting it into a process diagram*".

[4] In some cases, the term "workflow process" is also used (see, for example, [**Berry,** 1998] and other more recent documents from the Workflow Management Coalition (WfMC)).

This statement seems to corroborate that conventional BPM modeling is adequate only for certain types of processes while the scope of operational behavior found in organizations is much wider. This limitation of BPM software, foundationally based upon the Business Process Management Notation (BPMN), renders Case Management software systems amenable to a large family of commonly encountered operations in organizations [**Rooze,** 2007], [**De Man**, 2009a]. However, whether the advent of Case Management is explained by software limitations of BPM products or by a limitation of modeling approaches in BPM (including the BPMN standard) remains unclear. This subject is definitely important and will be revisited later in this paper.

Unquestionably, the BPM industry has continued to struggle with a variety of issues. In part, this is due to its fast pace of growth in the last couple of decades and also to the multidisciplinary nature of the technical field. As important knowledge relevant to the evolution of BPM may be found across different sciences and disciplines, there is always a risk of reinvention. There are some signs of this disconnect in BPM communities: recent propositions for modeling processes with entities, entity life cycle for describing operations, a need for connecting data and process in tools and so on (see [**Ebpml1,** 2010], [**Ebpml**2, 2010], [**Ramesh**, 2011], among others).

It is clear that BPM will definitely benefit from an in-depth review of contributions that other disciplines bring to bear to process modeling, particularly in the context of entity-based and life cycle modeling of processes. This paper addresses the state-of-the-art of *entity-centric approaches to operations modeling*. The content is limited to techniques for modeling and methods for discovering the elements that are within the model boundaries. In Section 2, the foundational concepts of entity and life cycle will be reviewed from systems design and information engineering. These concepts, well-known and documented for several decades in the tradition of these disciplines, provide some important unification of concepts for BPM. These capabilities will be discussed in the context of families of processes not conventionally covered by BPM, including Case Management-amenable processes. Conclusions are also included in Section 3 with pointers to upcoming work and other activities related to the content of this paper. In particular, Part 2 of this series will explore new horizons by revisiting the conventional concept of entity and linking its possible generalizations to different families of enterprise operations. In spite of their frequency and criticality for organizations, some of these operations have not been addressed in BPM modeling, methods or tools.

The review of the state-of-the-art in this paper and the new horizons introduced in Part 2 of this series suggest that entity-based dynamic operations modeling is a very important instrument in the repertoire of capabilities for process modeling and business analysis. While the roots of this approach come from several decades of work and different schools of thoughts, not all BPM practitioners seem familiar with these concepts and their literature sources. On the other hand, entity-based and life cycle modeling applied to BPM is vast in several aspects not covered in this review. For example, software tools, formal specifications and programming models are also important. These dimensions have received substantial attention recently through work that goes well-beyond the foundational roadmap paved along three decades. Due to the extension of the present review, these important dimensions are left for a future paper.

2. A Review of the Concepts of Entity and Life Cycle

In spite of notorious ambiguities in the definition of business process, the concept of process takes mainly a historical developmental perspective and focuses on the sequences of incidents, activities and stages that unfold over the duration of a central subject's existence. Van de Ven proposed this definition as one of the main three usages of process [**Van de Ven**, 1992; **Van De Ven & Poole** 1995]. It is important to remark that this notion of process is essentially a Social Science construct. Being organizations social systems, it should not be surprising that many important ideas that support the formal study of enterprise operations may be found at the realm of Sociology. In fact, the description of the real-world finds its roots into process metaphysics and basic philosophy. *Processes* and *Things* represent an important duality of principles coming from different schools of thought since the times of ancient Greek philosophers. The issue of whether *substance* has priority or primacy over *processes* or instead, *processes* have priority or primacy over *things* has paved the journey of several doctrines and internal variations.

Computer scientists noticed how the dilemma of seeing the world through nouns or verbs has landed also in Business Process Management. For example, in [**Liu et al.,** 2007] the authors documented: "… *business stakeholders describe their business by stating 'first we do A, then B, then C, and while doing C, we also do D'. We propose to focus on what is acted upon, thus describing business operations by first identifying things that matter to their business (e.g. Purchase Order, Insurance Claim), and second how these things are processed to achieve a certain goal"*. Several decades earlier, Robinson introduced the critical role of subjects (i.e., *things that matter*) in the description of processes [**Robinson,** 1979]. The same principle was pursued in parallel at least by two other authors, specifically Rosenquist [**Rosenquist,** 1982] and Jackson [**Jackson,** 1983]. The integration of these two views of the organization, i.e., things and processes, represents a phenomenal task in terms of depth and significance of the endeavor. Computer Scientists and Information Engineers just face the visible signs of a deep dissent coming from the times of Heraclitus and Democritus, later also involving notable thinkers like Leibnitz, Hegel, Peirce, James and several other more contemporary philosophers [**Rescher,** 1996]. As organizations are less vast than the entire world, the reconciliation of process and things is a more tractable problem in this more limited context as was demonstrated through solid progress in entity-based life-cycle modeling since 1970's.

As stated by Social Scientists the general meaning of process in organizations encompasses explicitly two main dimensions, namely, a *central subject* and a *family of activities* that the subject goes through during the course of its *development* [5]. These two dimensions and their linkages have been addressed through decades of research and practice also in other disciplines such as structured analysis, information engineering

[5] More precisely, in describing the progression of an entity, the work in [Van de Ven, 1992] embraces all core concepts that are also found in the tradition of information systems schools. Specifically, Van de Ven refers to the verbs that support an entity development (i.e., *activities*); the incidents that affect its evolution (i.e., *events*) and the stages the entity goes through its progression reaching well-defined states (i.e., *stages* or *milestones*). While the context of Van de Ven's work was strategy process, the concepts are directly applicable to all processes.

and systems design. The jargon used is not identical in all cases as, for example, Information Engineers have preferred to call the central subject an *entity* and the incidents *events*; some Computer Scientists have occasionally used the terms *artifacts* and *tasks. S*ocial Scientists have used terms such as *things*, *actors* and *activities*; and so on. Beyond these language differences, there is an identical concern addressed in the individual disciplines which also explains the remarkable similitude of the main concepts. Process and entity have traditionally been two different ontological views that have greatly influenced the study of organizations along differing journeys. These views have emerged from different foundations and in fact, this distinction has not only permeated through organizational theories but also other disciplines including Art History [**MacDonald,** 1993]. The distinction between viewing reality either as things or processes has deeply influenced the study of organizations as nouns (i.e., social entities) and verbs (social processes). Exemplary cases of this dichotomy are the works of [**Heath**, 2001], [**Tsoukas,** 2001; 2002; 2005] and [**Whetten**, 2006]. In [**Van de Ven,** 2005], an excellent bibliographic account and research summary of the ontological view of organizations as things and organizing as processes have been presented.

On the other hand, the Information Engineering and Structured Analysis foundations linking processes to entities and their developments can be found in several seminal works such as [**Jackson,** 1975], [**Robinson,** 1979], [**Rosenquist,** 1982], [**Jackson,** 1983], [**Martin**, 1989], [**Ould**, 1995], [**CCTA,** 2000], among several others. Interestingly, these key contributions from Information Sciences have taken place independently from others in Social Sciences. This Section will reconcile some of the contributions made from different disciplines and will shed light on the value for business process management. One of the main objectives of this Section is to provide a thorough review of the work done on the fundamental connection between 'things' (entities) and 'behavior' (processes).

Summarizing, Process and Entity are both critical concepts for organizational design and analysis and have been cultivated for a long time in both Social and Information Science. This interdisciplinary connection also provides a richer insight than any one approach provides by itself. Furthermore, the linkages between processes and entities developed mostly at the realm of the European Information Science schools constitute a body of essential, practical and inspirational outcomes. These topics will be addressed in more depth in the following Section.

Process Structures as Entity Life Cycle Models

The history of entity-based approaches to process modeling is rich and deeply rooted in the tradition of different schools. The notion of entity was introduced first as a static concept and its origins go back in time, at least, to foundational work done on data-bases in Computer Science. In the data-base tradition, entity-relationship models incorporate some of the important semantic information about the real world [**Chen**, 1976]. Chen's *highest level view* of the model is about significant *chunking* of information, done accordingly to business operations semantics and not fine-grain data-level representation concerns. In Chen's own words "*Level 1 consists of information concerning entities that exists in our minds"*. Chen goes further in explaining and illustrating the *coarse-grained* notion of entity: "*… An entity is a 'thing' which can be distinctly identified. A specific person, company or event is an example of an entity*". Potential issues in using Chen's notion of entity for dynamic models do not necessarily reside in granularity considerations but in the static nature of the derived model. This means that a potentially large number of entities will be made into the model without being necessarily needed for dynamics. Granularity or *significant chunking* can always be adjusted from a data-base modeling view to a more business-centric need by selecting things that business people care for their operations. Even though the data-base approach to entity modeling leads to a static time-slice of reality rather than reality itself ([**Robinson**, 1979], [**Jackson**, 1983]), the value of this entity-discovery approach for dynamic models was recognized in [**Rosenquist,** 1982], [**Ould,** 1995; 2005], [**Sharp & McDermott,** 2001], [**Cohn & Hull,** 2009] among others.

The independent works of Robinson, Rosenquist and Jackson documented the main seminal ideas and foundations of entity life cycle modeling ([**Robinson,** 1979], [**Rosenquist,** 1982] and [**Jackson,** 1983]). These authors introduced the central concepts and developed simple techniques to guide the selection of entities that are essential to dynamic models. According to the work of these authors, there are three main key characteristics that entities involved in the modeling of dynamic operations should satisfy:

- The concept of entity includes both the case of things described by a *type* as well as *individual* things in the world being modeled. Depending on the operations being modeled, individual instances and classes of things existing in the enterprise may be needed inside the model boundaries. This requirement means that the concept of entity includes both. Jackson addressed and illustrated the importance of this concept phenomenally well in the Entity-Action step of his method (see, for instance, pages 64 and on in [**Jackson,** 1983]). This characteristic is important and it has transpired in practical methods for identifying candidate entities. For example, Ould suggests to test potential candidate entities by using the word 'a' or 'the' in front of selected nouns that are of essence to the business being modeled [**Ould,** 1995; 2005]. On the other hand, as observed by Rosenquist, end-users are not familiar with the distinction between types and instances so it is important that the difference between the two concepts is made visible in the model only for the right stakeholders [**Rosenquist,** 1982].
- Each entity is *unique* in the sense that it has a unique identity or individuality that distinguishes it from any other entity in the world. This means that instances from a type or individual entities not deriving from a type are given unique identifiers and are kept distinct throughout their lives. This requirement allows different behavior in an organization to be modeled around the same data. The concept was introduced and illustrated with different examples in [**Jackson,** 1983], such as the same person (including related personal data) being represented as two different entities in a dynamic model (for example, subscriber and reader in an information services company). In particular, this characteristic implies that exactly the same information could exist in repeated form and still be modeled as different and unique *things* inside the model boundaries. In other words, it is not the intrinsic content of the data (identical records) what makes an

entity unique but the behavior modeled around such information. This has been also highlighted by different authors [**Jackson & Twaddle,** 1997], [**Nigam**, 2003], [**Bhattacharya, Hull & Su,** 2009] among many others.

- Entities that are selected within the model boundaries are required to never be *inert things*. Indeed, the defining characteristic is that an entity must have a set of actions that it performs or suffers. This characteristic stresses the goal of modeling dynamics of operations and thus, it establishes the notion of entity distinctly from the data-base tradition. In particular, entities that represent significant operational concerns in an organization are high-level in the categorization from Chen, i.e., they involve "significant chunks of information". The latter property will become clearer as life cycle is introduced. This "significant chunking" yields typically a smaller number of entities than those used in static models.

As the entities chosen should not be 'inert things', their significant changes through the operations have to be captured in the model. This dynamic aspect can be addressed in different ways depending on the kind of evolution an entity shows. Many known operations in organizations are amenable to being modeled through the notion of *Entity Life Cycle* (also called *Entity Life History*). Some of the characteristics commonly found in the literature are as follows:

- An Entity Life Cycle charts all the events and activities that may cause a particular entity occurrence to change in some way from its birth to its death.
- The *stages* that the entity may go through during its life are represented as a set of *states*. Some authors called *Milestone* to a related concept, i.e., the specific conditions and events that need to be met by an entity at a state for such a state to persist or for a transition to another state to take place [**Sharp & McDermott,** 2001], [**Rooze**, 2007], [**De Man**, 2009] [6].
- A valid Entity Life Cycle has an associated graph in which its nodes are the main *states of the entity* and the arcs denote the *transitions* that the events/activities may generate during the entity's existence.

The above concepts were introduced by different authors in the 80's by using slightly different but closely related ideas. Strictly speaking, [**Rosenquist**, 1982] was the first to present a complete Entity Life Cycle Modeling approach by following the above three characteristics. Rosenquist also presented a method for identifying candidate entities, defining the interconnection between different entity life cycles and representing life cycles through finite state machines. Different authors and institutions have followed his work, including the concept of *access modes* to manipulate an entity in its different states by using the Create Read Update Delete permits (CRUD) [**Martin,** 1990], [**CCTA,** 2000] and others. Figure 1 shows the role of Entity Life Cycle as a representation for process structure and as a mechanism to integrate both data and process into the same modeling approach [**Rosenquist**, 1982].

[6] In [**Van der Aalst,** 2001; 2002], *Milestone* was introduced as a workflow construct for deferred control in the evolution of a process.

On the other hand, Figure 2 shows a significant entity and its life cycle from the work of Rosenquist. The entity in Figure 2 and others in [**Rosenquist,** 1982] such as ‘Company’, ‘Bank’, ‘Fluid Hydrocarbon Type’, ‘Storage Facility’, ‘Loading Facility’ and ‘Vessel’ are excellent examples of Chen’s Level 1 entities [**Chen,** 1976]. The business relevance of these entities is self-explanatory. Indeed, ‘Client Contract’ represents a sizeable and complex fragment of operations encapsulated under the name of this entity, even more so considering that Oil Industry is the context used by Rosenquist. In particular, this size explains also the rich number of events and states present in the analysis of this entity, i.e., 20 and 10 respectively.

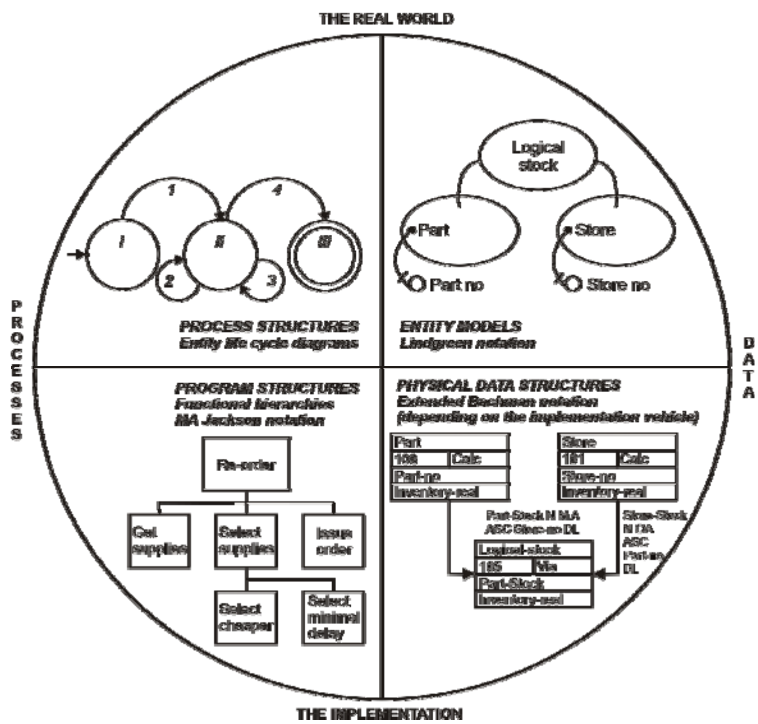

Figure 1: The four quadrants of processes, data, “real-world” and “implementation world” according to the seminal work from Rosenquist in 1982.

The examples from Rosenquist, among many others available throughout three decades of work in Entity Life Cycle, corroborate the fact that the granularity of the entities in the model may be coarse or fine depending on the actual problem at hand. These examples disprove that available entity models from data-base centric approaches serve only static modeling needs. While data-base modeling is static, the discovery of entities for dynamic operations begins with the identification of those *things that matter to the business*. This list of *candidate things* plus the additional requirements reviewed in this Section provide an excellent starting point and prepare the list for further refinement in the life cycle phase of modeling. Rosenquist also presented a method for identifying candidate entities, defining the interconnection between different entity life cycles and representing life cycles through finite state machines. Different authors and institutions have followed his work and that of Jackson in 1983, including the concept of *access modes* to manipulate an entity in its different states by using the Create Read Update Delete permits (CRUD) [**Martin,** 1990], [**CCTA,** 2000].

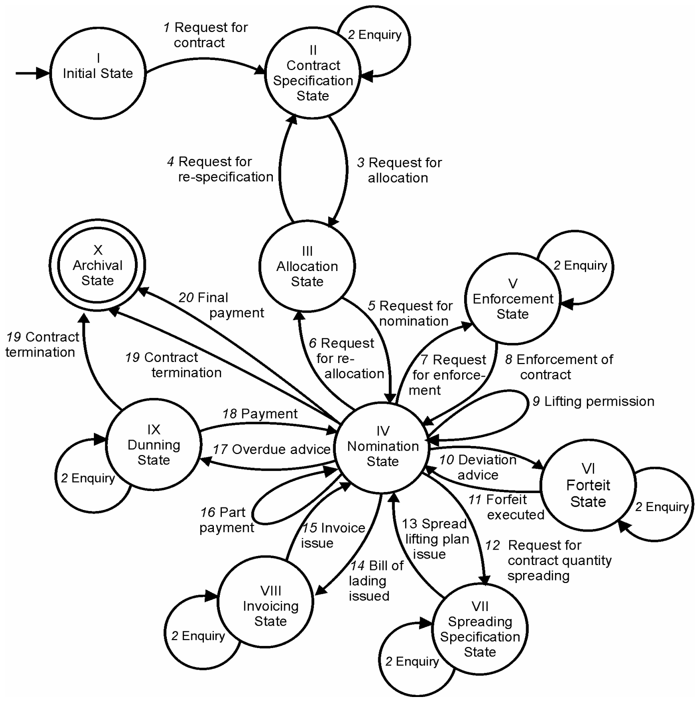

Figure 2: Entity Life Cycle for the "Client Contract" entity [Rosenquist, 1982]

The information contained into the detailed description of an entity is inherent to Entity Life Cycle modeling and also a consequence of the specific implementation chosen. The representation of an entity in an information system will have to carry all the data needed to support and realize its life cycle. This means that the aggregation of information encapsulated by an entity makes the data model quite localized in the sense of providing all needed information about the life of the entity in 'one place'. This characteristic of entities for operational modeling has been long-investigated, particularly at the realm of work in several European methods [**Tardieu,** 1992]. More recently, this feature has been highlighted as an important characteristic of entities that matters for business purposes in [**Nigam,** 2003], [**Cohn & Hull,** 2009] and other authors.

As Entity Life Cycle modeling requires taking into account the causes for transitions across states in the model then, the detailed information held in an entity aims at supporting the conceptual description of these transitions, collecting the value-attributes involved during the entire life of the entity, etc. Some of these conceptual rules have been discussed extensively in [**Tardieu,** 1992]:

- For each state transition, the cause that triggered the state transition must be established.

- For each state transition, the transformations of data implied by the state transition must be described.
- Once the data transformation has taken place, the causes that triggered the transitions do not need to be necessarily traced.

The above concepts have also been more recently used under other names different from entity such as *business entities, artifacts, business artifacts* and *adaptive documents.* In [**Bhattacharya et al,** 2009], Bhattacharya, Hull and Su stated the convergence of all these related concepts explicitly: "[this paper addresses] … *a design methodology for business processes and workflows that focuses first on 'business artifacts' which represents key (real or conceptual) business entities, including both the business-relevant data about them and their macro-level lifecycles*". In addition, the same convergence is also explained and documented in [**Kumaran et al,** 2008]: "*This modeling paradigm, which models business processes as intersecting life cycles of information entities, has been proposed. Appropriately, this approach is called information-centric process modeling. The information entities that are used to describe business processes in this manner have been called various names, including adaptive documents (ADoc)* [7]*, Adaptive Business Objects (ABO), business artifacts. And lately, Business Entities".*

Another proof of convergence on the concept of entity beyond identical semantics across independent interpretations is given by actual applications. The same entities have been invariably used across most of the available literature in dynamic modeling. This convergence is very important because it signals that *entities provide a good foundation to create a more stable description of operations and thus, establish the basis for a process architecture.* Indeed, a number of authors show the same set of typical entities such as Purchase Order, Claim Form, Insurance Claim, Customer Check, Drug Compound, Rental, Customer Complaint, Shipping Notice, Repair Order and the list goes on. At the beginning of the modeling stages, these entities are all simplified, i.e., they are described by using a short narration. As the dynamic modeling task progresses, these entities are endowed with the necessary detail to capture the entire information involved during their development or evolution but implementation details aside, the examples show that when describing what a business operations is about, entities provide a very appealing common ground.

Robinson in 1979 and Jackson 1983 used a different notation for modeling the dynamics of an entity, yielding a variation of the above technique. These methods are included in different text books, including those from Jackson in [**Jackson**, 1975; 1983]. In [**Robinson**, 1979], the author proposed the specification of entity behavior by describing the set of events that drives that behavior. He built upon the work of Jackson by using JSA/JSD diagrams [**Jackson,** 1975] to represent an entity's development. Robinson considered a number of entities and events as dual elements in his models which were based on one *life-history* diagram per significant entity used. The core element representing a Life Cycle of a single entity in the tradition of JSA/JSD diagrams is an

[7] An *Adaptive Document* models a "domain artifact" that displays state-dependent behavior. The domain artifacts are essentially the concepts by which the problems and their solutions in the domain are described [**Kumaran, Nandi et al,** 2003].

event / activity. The structure of JSA/JSD diagrams follows certain syntactical conditions, essentially constraining the layout to be a combination of *sequence*, *selection* and *iteration* of events / activities. The resulting graph is intimately connected to *State Transition Diagrams* (STD). STD and its sibling, *Finite State Machine* (FSM), were reviewed and illustrated with several examples in the book by J. Martin on diagramming standards whose applicability goes beyond entity models [**Martin,** 1987; pp. 297]. There is a strong duality between Jackson's diagrams and Rosenquist's state-based representation. This duality will be addressed in the next Section.

The identification of critical entities, life cycles and their *interconnections* are the main challenges in the analysis of business operations. In Rosenquist's work, the latter challenge is addressed through the following central idea: *"… the interrelationships between the corresponding life cycles are identifiable by establishing the entry priority sequence in which the entities have to exist within the information systems universe in order to satisfy the structure of the real world".* Life cycles for different entities are also typically connected to each other because the complexity of the operations being modeled requires more than an isolated entity. In particular, activities or events in a given Entity Life Cycle can start a new life cycle for other entities. These relationships are very important in describing the dynamics of a complete operating model. Simple examples of such relationships or interdependencies are the ability to start or to halt a life cycle from the life cycle of another entity. While this concept also comes from the early work of Rosenquist, it has been elaborated into great detail by different authors such as Jackson and Twaddle [**Jackson**, 1997; pp. 69-160]. This problem has been studied more recently by Kumaran, Liu and Wu through a deep characterization of duality of two representations, namely a family of workflows and the corresponding entity-centric life cycle model of operations [**Kumaran et al,** 2008]. Quite interestingly, the hierarchy of entities derived heuristically by Rosenquist in [**Rosenquist,** 1982; page 311] appears intimately related to the *"dominance"* property relating different entities in [**Kumaran et al,** 2008]. Rosenquist's notion of *"entry priority sequence"* is a form of temporal order that requires less knowledge than well-defined workflows, i.e., complete description of input entities and transformed or created output entities for every task in the operations model of Kumaran et al.

The need for detecting 'substance that matters' -paraphrasing process metaphysics studies- is not exclusive to life cycle as a model for dynamics. Indeed, this need is also addressed in other modeling methods that start by finding a list of potential things that matter to the business such as the work of [**Ould, 1995]**. Many techniques have been published since the 80's for the discovery of these "significant entities", i.e., chunks of operations relevant for different modeling purposes. These "subjects" or "things that matter" have also become an essential instrument in process architecture and related work on process taxonomies carried out since the early 90's [**Ould**, 1995], as will be reviewed later in this paper and also in Part 2 of this series [8]. In short, these essential entities are most stable constructs to architect enterprise operations than the actual tasks

[8] In other sciences, this concept has received attention much earlier. For example, the notion of "evolution of a subject" as a characterization of process [**Van de Ven**, 1992] had been used in the study of narration styles in literature since the beginning of the 20th century [**Propp,** 1929]. Also, the study of subject and the distinction from object have been topics of active contribution by sociologists in the context of actor-network theory [**Latour**, 2005].

used to instrument the related behavior or said in simpler terms: *the what is more permanent than the how*.

The seminal work from Rosenquist, Robinson and Jackson was continued with two decades of contributions on methodology for practical systems definition and implementation. For example, Learmonth and Burchett Structured Development Method (LSDM) and its successor Structured Systems Analysis and Design Method (SSADM) have incorporated the entity life-cycle analysis technique from [**Rosenquist**, 1982]. This fact was explicitly acknowledged in [**Hawryszkiewycz**, 1988; pp. 301-302] as part of a summary of methodology efforts. Independently, the Business System Development (BSD) series on best-practices and standard approaches established originally by the Central Computer and Telecommunications Agency in the United Kingdom has published a volume dedicated to *behavior and process models* in which Entity Life-Cycle Analysis is the main method [**CCTA**, 2000] [9]. The schools of thought represented by the work of Robinson and Rosenquist include many other individuals and institutional contributors along three decades. The language used by Robinson and Rosenquist shows clear traces imprinted by the Information Engineering School ([**Martin,** 1987, 1989], [**Finkelstein,** 1989]) (particularly, in connection to concepts such as *"encyclopedia'*, '*finite state machine transition diagram',* etc.) and the Structured Analysis and Systems Design School ([**Jackson**, 1975; 1983], [**DeMarco**, 1978], [**Yourdon**, 1979], [**Hawryszkiewycz**, 1988; 1994] among others).

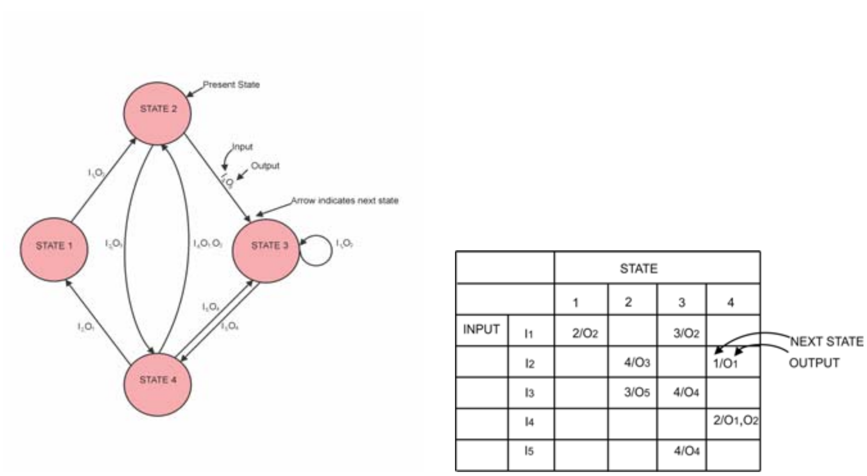

Figure 3. State Models from Martin, 1990

The integration of a number of business dimensions such as process models, rules, data models, strategies, goals, specifications, etc. lies at the realm of the Information Engineering School [**Martin,** 1989]. These individual dimensions and their coordination through well-established techniques are among the core of the Information Engineering

[9] These facts imply that SSADM has evolved well-beyond the relatively "unstructured" modeling of business activity from its early versions. More recently, CCTA and its standardization work in information technology has been made a part of the Cabinet Office in the UK.

deliverables [10]. More precisely, in the methodological edifice of this school, *Business Area Analysis* is a key phase in which processes needed to run the business are identified, as well as how these processes are interrelated and what data is needed. Martin states: '… *in business systems, an entity type has a life cycle. It is created, various processes are applied to it, and eventually is deleted. A state transition diagram can help clarify the states that comprise the life cycle. This analysis tool can help give a completeness and consistency check on the analyst's representation of processes and their dependencies'* [**Martin**, 1987; pp. 306]. This and other references show that entity life cycle analysis is not only a very useful capability pattern adopted for operations modeling but also confirm that STD is a useful tool for analysts and system architects. Figure 3 shows a simple example. There are different records in the literature of both the methodology and diagramming variety for Entity Life Cycle modeling. Figure 3 shows a State Diagram structure and a companion transition table proposed by Martin in [**Martin,** 1990] as diagramming standards [11]. In [**Hawryszkiewycz**, 1988], a very similar notation was used, as shown in Figure 4. Obviously, this diagramming and notation resembles very closely the work by Rosenquist in 1982 reviewed before in this paper.

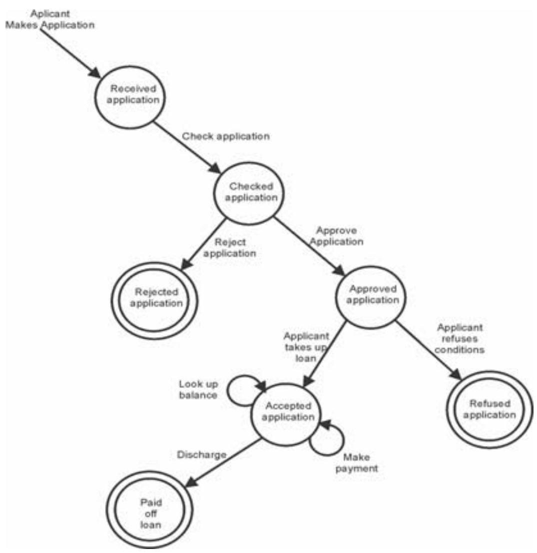

Figure 4. Life Cycle Diagram [**Hawryszkiewycz**, **1988**]

The contributions from Robinson, Rosenquist and Jackson made it as an integral part of the Information Engineering methods being adopted by many public organizations, consulting companies and educators [**Hares**, 1992]. In particular, the linkage between processes and entities has been addressed at different levels and in different phases of the Information Engineering method. First, at a conceptual level, *process-entity matrices* are one of the key artifacts provided by the method [**Martin**, 1990; pp. 152 and 272]. Moreover, these matrices are also equipped with an indication as to whether the type of action by the process is to create, retrieve, update, delete the entity, or a combination of

[10] The Enterprise Architecture field has emerged from, and could also be considered an evolution of, Information Engineering.

[11] Variation is found in the language used by different authors. For example, Jackson and Twaddle have not used "state" and "stage" as synonyms in their life cycle language [**Jackson,** 1997]. However, these language differences are truly not substantial and they do not lead to diverging theoretical foundations. This means that "stage" and "state" can be used interchangeably without any risk of confusion.

them (i.e., the well-known CRUD acronym) [12]. This matrix construct is one of the four main types of diagrams used in the Business Area Analysis [**Martin**, 1990]. Admittedly, the conceptual linkage provided by the process-entity matrix does not yield a technique to model processes based on entities. But in further detailed phases of the Information Engineering and System Design methods, entity life cycles are intensively used for operations modeling ([**Hawryszkiewycz**, 1988], [**Martin,** 1990], [**CCTA**, 2000] among others).

Finally, the same book [**Martin,** 1990] contains some illustrative cases of state transitions used to model typical operations such as payments for course registrations in an education business. This example is shown in Figure 5 where the entity is Payment. The states and transitions here are represented in a different type of diagram capturing the same concept (the same one was used in Oracle data-base software by [**Ensor**, 1997]). The depth of the model or fit with the actual business operations is not the subject addressed here and thus, none of these examples should be analyzed in terms of their comprehensiveness.

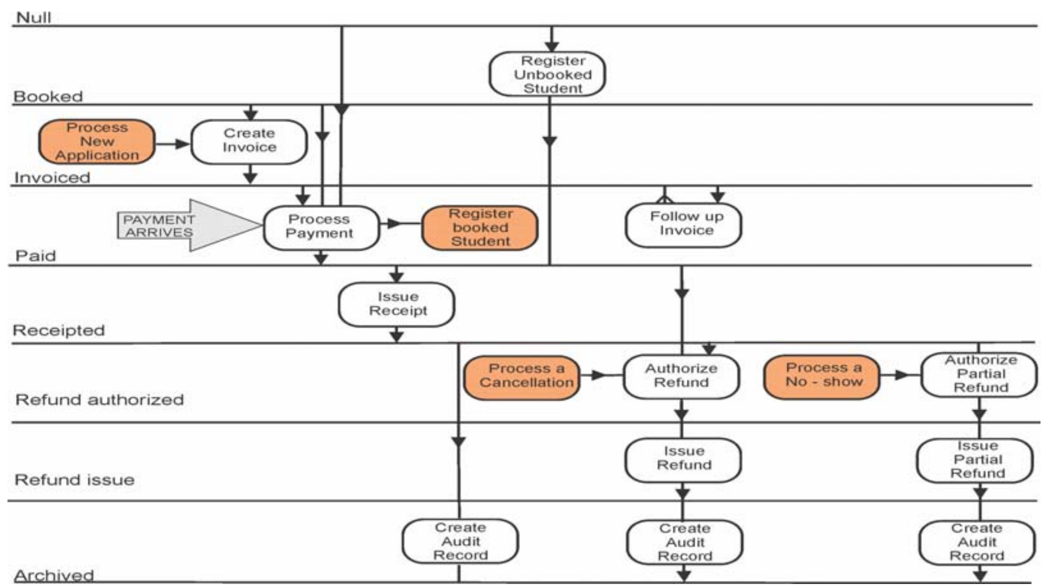

Figure 5. A different representation of an entity life cycle-based model [Martin, 1990]

As said above, the Structured Analysis School has also incorporated entity life cycle techniques in its core methodology. This can be clearly seen in the early work by Jackson whose JSD method includes the so-called "*Entity Action Step*" and "*Entity Structure Step*" [**Jackson,** 1983; pp. 64-120]. These methodologies represent significant technical progress

[12] The methodology goes further by also linking *Entities* and *Functions* in the organization and using CRUD relationships to cluster Functions into Business Areas [**Martin**, 1990; pp. 171-180]. These concepts could be regarded as one of the origins of information-centric business componentization and related techniques. This subject goes beyond the scope of this paper and will be addressed elsewhere.

and adoption efforts beyond the original works by Robinson and Rosenquist. Many organizations around the world, including governments and consulting companies, have grown significant activities around the standardization and deployment of the techniques reviewed in this Section. In fact, SSADM has become an important piece of intellectual capital for systems development [**Weaver,** 2002], particularly disseminated and adopted across the European Community. In particular, H. Tardieu from the French school in Information Systems has provided a survey of work done at the realm of different European projects [**Tardieu**, 1992]. Under the title "*Interaction between Data and Process Models*", he strongly praised the value of the entity-centric modeling approach: *'Ten years before 'objects' became a buzzword in information technology, information system designers have promoted the idea of describing a process by the entities, relationships and attributes that were manipulated. The natural next step for them has been to describe integrity constraints, expressing prohibited states for entities, relationships or attributes, and more recently, to describe their state-transitions. Practices have now put in common use Entity Life History (SSADM, Merise / 2) which describes the sequence of states which are authorized for a given entity'*. These remarks are enlightening and very important to appreciate the volume of intellectual capital and the degree of convergence reached across different schools of thought in information systems. These foundations and applications are widely available for conceptual advancement and practical value of the BPM industry.

Recent practical applications of these concepts reported in [**Bhattacharya et al,** 2005; 2007a], [**Chao et al,** 2009], [**Cohn & Hull,** 2009] and others corroborated the importance of the seminal ideas on entity life cycle modeling coming from the traditional schools reviewed in this Section. In [**Nandi et al.,** 2010], the authors use an example of entity shown in Figure 6. As is easily seen, the concepts of information, life cycle and access policies define jointly the notion of Courier Shipment Entity. In this case, the entity is a type, i.e., it represents many such things coming from the actual operations.

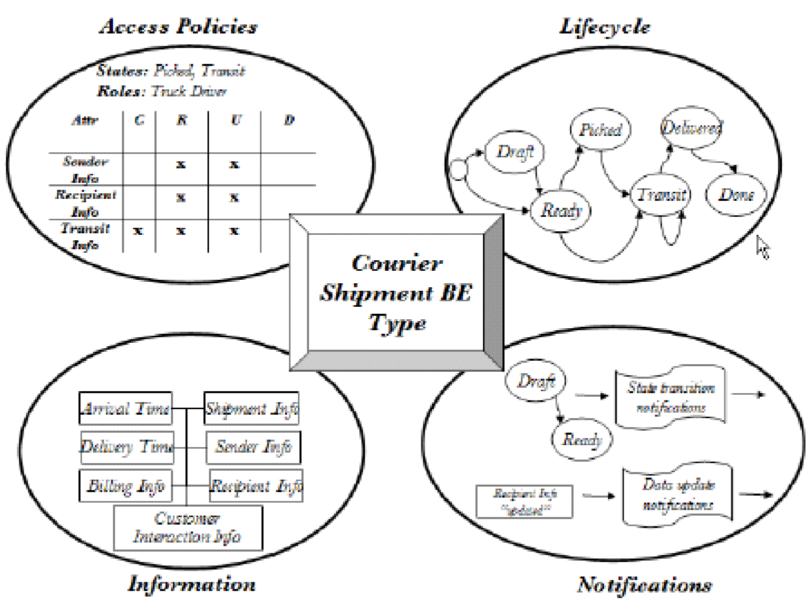

Figure 6: A diagram from Nandi et al [Nandi et al, 2010] showing an entity as defined in the tradition of Entity Life Cycle modeling

In 2009, a thorough study of a Line of Business in finance and accounting was conducted and reported. The operations were modeled by using three essential entities, described in [**Cohn and Hull,** 2009] as follows:

- **Deal**: The activity around evaluating a client request, negotiating terms and conditions, signing the contract, issuing invoices for the assets to be financed, monitoring payments, etc.
- **Supplier Invoice**: The purchase and shipping of the asset(s) to the client location(s).
- **Asset**: The individual hardware asset(s) when accepted from the supplier, titled to financial services company, delivered to the client, used by the client, and finally sold or disposed.

Figure 7 shows a representation of the three corresponding life cycles and their interconnections. Notice the number of states per entity involved, i.e., ten, three and five respectively. This high-level diagram represents what the business does in a very compact and simple way, thus facilitating communication across stakeholders in the organization and role-players involved in the detailed definition of the operations which are subsequent to this stage of analysis.

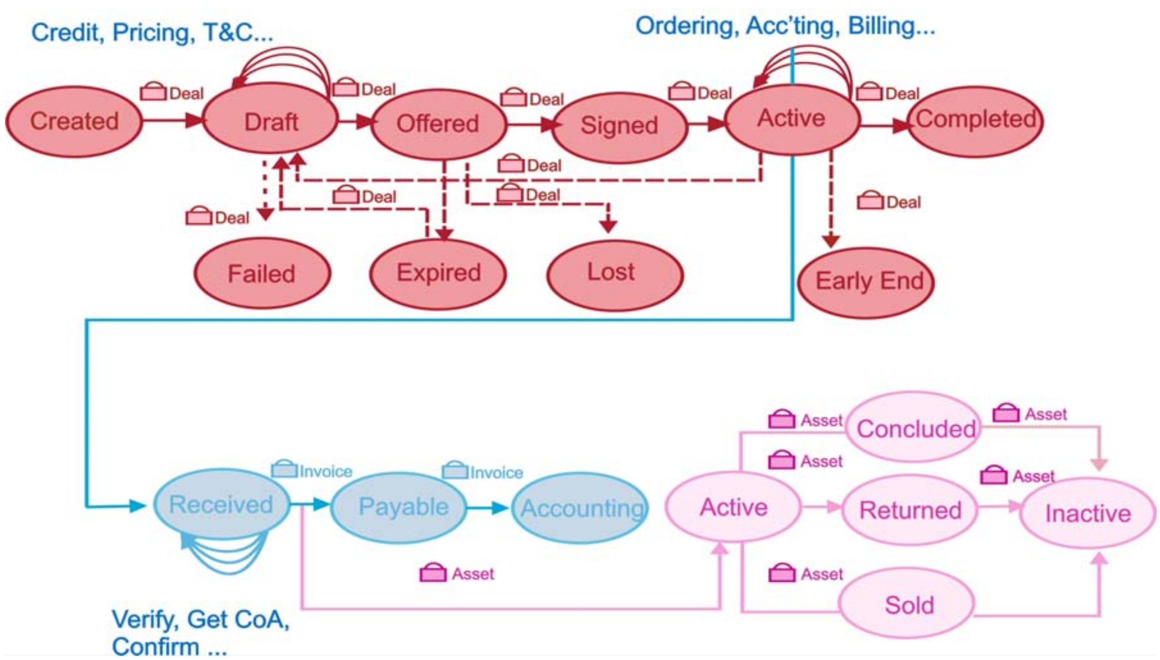

Figure 7: Life Cycles for the three entities presented in [Cohn and Hull, 2009]

Another reason for reviewing these contributions by going back to the classical concepts and original sources of the ideas is that recent Business Process Management fora, including the software industry side of BPM, highlighted a need to connect data and process modeling. Several active blogs in different communities of practice and papers published show a remarkable lack of awareness about the state-of-the-art of entity life cycle modeling [**Ebpml1**, **Ebpml2**, 2010]. Motivated by the claim that data is an "afterthought" in business process modeling tools [**Nandi et al,** 2010], some practitioners from the BPM industry have begun to discover entity-based modeling [13]. In the light of the intellectual capital and practical legacy coming from the Information Engineering, Structured Analysis and the System Design Schools, no gaps between data and process exist today that may explain why BPM tools have not benefited from entity life cycle modeling. However, this inconsistency is not the only one in BPM software: as highlighted in Section 1 the advent of Case Management systems shows that new "case tools" are necessary to model processes beyond traditional BPM systems.

Methodologies for Selecting Significant Entities and States

Defining the boundaries of a process and the main phases that define its progression are key activities in operations modeling. There are many techniques documented in the BPM literature to accomplish these objectives, involving a wide variety of methods [**Recker,** 2009] and standards [**Ko,** 2009], such as those based on goals [**Kavakli**, 1999], functional and activity-centric [**Sharp & McDermott,** 2001], Role-Activity Diagrams [**Ould,** 1995], communication-based [**Harrison-Broninski**, 2005], workflow-centric [**Wang**, 2005], based on Petri Nets [**Van der Aalst** , 2000], case-based [**De Man,** 2009a; 2009b], Event Process Chains [**Green,** 1999], [**Van der Aalst,** 1998], [**Scheer,** 1998], and so on. This subsection focuses on entity-centric methods. As reviewed in the previous Section, entity-based process modeling presents some foundational differences with respect to the rest of the modeling approaches: it focuses on the identification of significant *entities*, their *individual evolution* or behavior through a *life cycle* description and the *interconnection* of these entities through establishing needed dependencies through their life cycles.

Entities as reviewed in the previous Section help identify and model processes. These entities are *things that matter* to the operations being modeled. Sharp correctly points it out in [**Sharp & McDermott,** 2001] that the word entity has been used as a substitute because *"thing modeling would not cut it as a discipline"*. Interestingly, Robison had introduced earlier the notion of *subject* as the main *thing* a process deals with and not coincidentally, other authors have followed similar language [14]. The value of detecting the *main* or *significant entities* as a vehicle to model core processes in an organization stands by itself, i.e., irrespectively of the specific technique used to model the entity evolution or its behavior. Selecting *significant entities* and identifying its main *States* bear a strong connection with each other, as will be reviewed later. There have been different terms used in the literature for these '*significant entities*', such as *principal entities*

---

[13] http://www.ebpml.org/blog2/index.php/2010/04/21/mde-bpm-ibm-s-bedl-analysis.
In another blog http://www.ebpml.org/blog2/index.php/2010/09/14/soa-bpm-rest-lifecycle-example, an author recently stated: "*I built a simple business entity lifecycle example* … [because] … *this type of concept is still foreign to most people in the industry"* [sic].

[14] Ould, for example, states that significant entities are the things the business has as its *subject matter* [**Ould**, 1995], [**Ould**, 1997]. This language has also been followed by individual consultants in analyst firms [**Miers,** 2009].

[**Rosenquist**, 1982], *subjects of process* [**Robinson**, 1979], *real-world entities* [**Jackson**, 1983], *essential business entities* [**Ould,** 1995], *artifacts* [**Nigam,** 2003], among others. All this language variation addresses the same entity-centric concepts reviewed in the previous Section.

It is noteworthy that all main techniques known for identification of *Significant Entities* in process modeling have been published in the 80's and 90's. In part, the reason for all this long-standing activity is that data modeling is a well-established domain and indeed, it has some common methodological roots with entity-based process modeling. While data-base modeling does not focus on BPM concerns, the need for modeling reality in both fields offer essential cross-pollination [15]. In other words, a well-accomplished data model always reflects important aspects of the real world. Sharp and Dermott back in 2001 clearly illustrated the methodological value of the liaison between data models and process modeling: *"… data models are a powerful tool early in a process-oriented project –they improve communication and consistency, because the entity names and definitions provide a standard vocabulary for the things the process deals with"* [**Sharp & McDermott**, 2001]. As a consequence, thousands of excellent examples have been available where entities discovered through information modeling techniques represent *Significant Entities* for operations and process modeling.

As reviewed in previous Sections, *significant entities* have distinguished characteristics that make them stand out among those detected by pure data-modeling approaches. But entities are essentially *nouns* and represent some aspect of the reality of the organization being modeled. Techniques from information modeling have proven useful to detect *Significant Entities* for process models. This remark has been stressed in several books beyond Jackson's in 1983, such as [**Ould,** 1995] and [**Sharp & McDermott,** 2001]. The latter authors state that traditional entity-relationship modeling techniques are a good starting point for the detection of process modeling-centric entities. This deep connection that links the traditions of two large domains of research and practice is surprisingly not cited in most of the BPM literature. Object Orientation adds a similar story in the sense that available methods for finding 'objects that matter' should be considered for dynamic entity modeling [**Ryndina et al.,** 2007], [**Kuester et al,** 2007]. Hay stated that there is a remarkable tendency in Information Systems to ignore years of methodology progress as soon as a new method becomes fashionable (see [**Hay,** 1997] and the author's web site located at www.essentialstrategies.com for several related and more recent documents). By following analogous principles to Hay's, potential arguments such as 'objects are not entities' or 'entities, unlike objects, are not a software concern' should be considered oversimplifications. Dismissing the value of entity identification methods coming from information-centric modeling due to supposedly too small of a granularity or static nature of the models is incorrect and inadequately leaves the modeling of processes disconnected from several decades of progress in sister disciplines.

[15] Admittedly, the information modeling job is way too often approached as a technology exercise. This point has been stressed very emphatically by Sharp & McDermott: *"… technically oriented "modelers" jump straight into excruciating detail, dense jargon and complex graphics, incomprehensible for process participants."* [**Sharp & McDermott**, 2001; pp. 275].

As mentioned above, Jackson developed very early two steps called "entity-action" and "entity structure" as part of the JSD method [**Jackson,** 1983]. In the entity-action step, Jackson recommends to start by building lists of candidate entities accompanying each with a verb that describes a possible action that the entity suffers or carries out. Ould has been the first to recognize the fact that the concept of *Significant Entities,* and consequently all its siblings, fall within the realm of Jackson's work in 1983: *"… the older among us will hear echoes of Jackson's JSD"* [**Ould,** 2005; pp. 172] [16]. The method by Jackson also signals an important touch-point between the static nature of "*things*" and the dynamics revealed by companion verbs. In fact the identification of these verbs is one of the possible hints in helping with the development aspects of the process.

In JSD, the dynamics is assembled in the form of event/activity diagrams in the "entity structure" step. An example of these diagrams is offered in Figure 8a, borrowed from a car rental business example published in [**CCTA,** 2000]. This entity structure captures some form of ordering of the main actions affecting the development of the entity. Notice that Jackson builds this dynamics by considering only one *Significant Entit*y at a time, and thus, the behavior described focuses exclusively on the entity whose evolution is being studied, i.e., the *"subject"* is kept throughout the diagram [17].

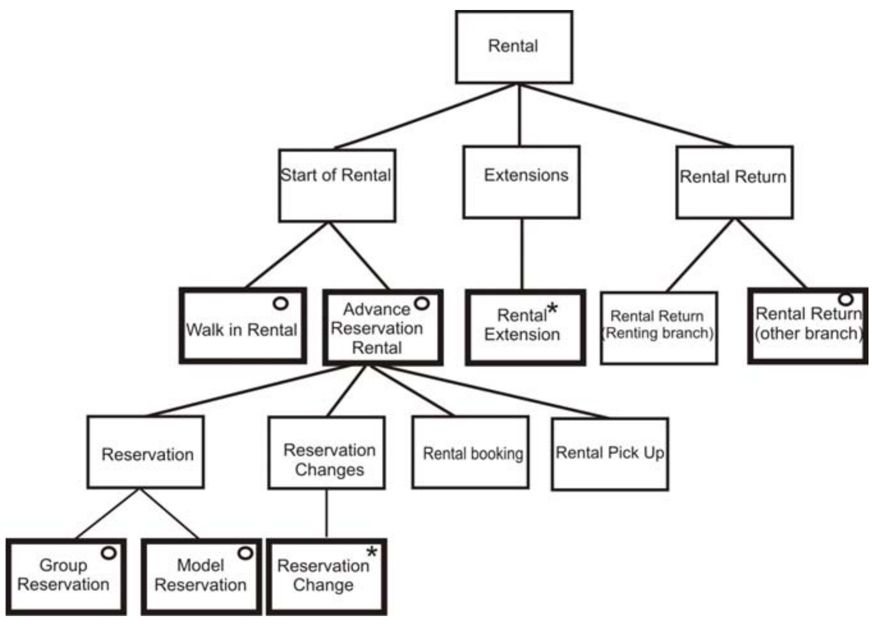

Figure 8a. An example of entity-action diagram in Jackson's tradition and notation. In this diagram, the circle symbol in a box means *selection* (i.e., option) and the star means *repetition*.

---

[16] Interestingly, Ould does not cite Robinson's or Rosenquist's seminal works ([**Ould,** 1995], [**Ould,** 1997], [**Ould,** 2005], [**Ould,** 2006]) but this is partly explained by the fact that Ould's approach does not rely upon the life cycle of Essential Business Entities as the method for modeling the behavior of processes. On the other hand, Ould acknowledges the previous work of the Structured Analysis School and the influence that it has exerted on his EBE-based process modeling method.

[17] In the JSD notation, the circle symbol in a box means *selection* (i.e., option) and the star means *repetition*.

It should also be stressed that the JSD structure of the entity (i.e., the terminal boxes in the diagram) are actions and events involved the lifetime of the entity. Then, this diagram does not aim to represent any functional decomposition, hierarchy or a program-oriented model at all [18]. Moreover, as mentioned earlier in this paper, this entity structure can be made into a life cycle diagram following Rosenquist's tradition and others by means of a graph duality transformation in which a selected number of *states* represent the stages through which the entity goes as a consequence of *transitions* (i.e., the events / activities from the JSD representation) across different states represented by arrows. This translation of one representation into another should be no surprise since Jackson himself stated: *"A fundamental requirement in specifying entity structures is to span the whole lifetime of the real world entity…"* [**Jackson,** 1983; pp. 85]. In fact, Jackson's diagrams allow for repetition of events and activities (individually or in groups), thus confirming that JSDs do not capture any "hierarchical decomposition" of the system but a form of ordered behavior affecting the entity. In Figure 8b, an example of such a transformation is shown, showing the life cycle of rental [19].

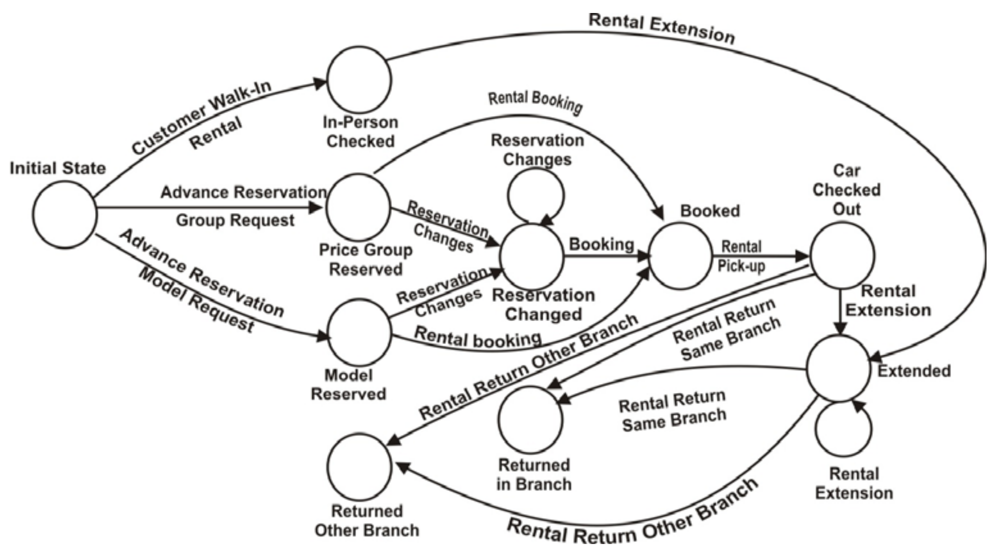

Figure 8b. State Life Cycle equivalent representation of Figure 8a by following [Rosenquist, 1982], [Hawryszkiewycz, 1988], [Martin, 1990]

More generally, Figure 8c shows a typical skeleton of a JSD structure and Figure 8d shows the life cycle version in the tradition of Rosenquist and several other authors. The intent of this example is to illustrate readability of both representations. The actual family of events and activities involved in the lifetime of the entity in a JSD are only those with a highlighted frame (Figure 8c). The rest of the structure in the diagram is needed for grouping purposes. Given the semantics of this representation, the additional

[18] The methodology work in [**Jackson,** 1983] contains further steps to translate the models into computer programs.
[19] Notice that the intent of the example is to illustrate the nature of the JSD diagrams and its duality with a life cycle concept and not to exhibit a good model of a 'car rental' business.

boxes help introduce the desired partial ordering across the events and activities. Thus, the notation has some redundancies as can be seen from Figure 8c, since 7 boxes out of 17 are meaningful while the rest is overhead. This should be compared to Figure 7d, in which a life cycle of states is used instead. In the latter case, there are 7 different states only.

The above examples conclusively illustrate that the System Development, Structured Analysis and Information Engineering schools have all introduced the subject of entity life cycle for modeling dynamics of operations and contributed to its adoption through different methodologies. The concept behind the notation used by Jackson at the realm of systems building techniques is essentially identical to state-based representations standardized by Martin and introduced by others.

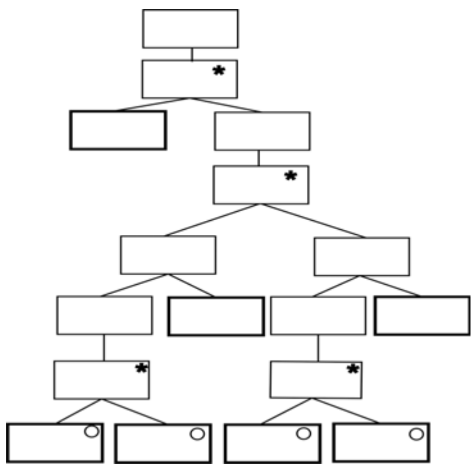

Figure 8c. A skeleton of a JSD

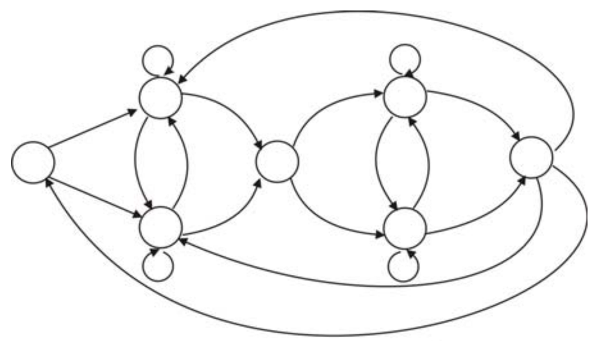

Figure 8d. State Life Cycle representation

From a methodology perspective, Ould has developed selected aspects of behavioral modeling of processes to a significant level of depth, going well-beyond the tradition of the System Development, Structured System Analysis and Information Engineering Schools reviewed in this paper. Ould works are illuminating and include two books published ten years apart and several papers on process taxonomy and architecture. The process modeling methodology proposed by Ould is called *Riva* and its origins go back to [**Ould,** 1995] and its evolution published a couple of years later in a paper [**Ould,** 1997]. *Riva* is based on *Significant Entities* [**Ould,** 1995] and a combination of this concept with the so-called *"Units of Work"* (UOW). *Riva*'s *Significant Entities* are called "Essential Business Entities" (EBE). Candidates EBEs are detected by following similar principles to those reviewed in this Section and then, the list of candidates is refined through the notion of UOWs [**Ould,** 1997].

In addition to technical contributions, Ould's work has brought substantial field evidence from his consultative BPM practice, thus making his propositions around *Riva* grounded into actual client adoption [**Ould,** 2006]. Ould's methodology includes important best-practices through workshops dedicated to discovering EBEs and UOWs, particularly in the form of workshops. The discovery of EBEs follows a number of steps to be pursued together with operations people in the organization, yielding anywhere from forty to one hundred candidate EBEs. These EBEs come from questions guiding the brainstorming with the client, shown in Table 1.

After coming up with the list of candidate EBEs, Ould continues through a number of litmus tests and filters to determine whether each candidate is truly an EBE. Tests suggested include things like putting the word "a" or "the" in front of each candidate EBE and keeping those whose meaning makes sense. Other tests aim to remove candidate EBEs that are roles, including LOBs and individuals; or EBEs that are contained inside another EBEs without having a lifetime of its own. The next significant step in Ould's methodology brings UOW to play, i.e., only EBEs whose life the organization has to look after are kept. Ould's methodology does not involve life cycle analysis of the entities but instead, a combination of concepts around Role-Activity theory is used. The details go beyond the scope of this paper.

What do we make?
What do we sell?
What product lines do we have?
What services do we offer?
What services lines do we have?
What things can we simply not get away from?
What are our external customers?
What are our internal customers?
Are there things our customers have, or want, or do, that might be our EBEs?
What thinks do we think differentiate our organizations from others?
What sorts of things do we deal with day in, day out?
What business entities are listed in our corporate data model?
What things do our information systems keep information about?
What events in outside / inside world do we need to respond to?

Table 1: A list of questions guiding the discovery of Essential Business Elements [Ould, 1995]

Several important applications of the entity life cycle modeling approach have also been shown for different industries and related patent activity. For example, the authors in [**Bhattacharya et al,** 2005] address a pharmaceutical research process area; in [**Nigam & Caswell,** 2003], the industry at play is food service and restaurant chain segment; in [**Chao et al,** 2009], the authors addressed financing services to clients in a large corporation; additional related works are documented in [**Bhattacharya et al**, 2007a; 2007b], [**Liu et al**, 2007], [**Fisher et al,** 2007], , [**Hull**, 2008], [**Strosnider,** 2008], [**Cohn**, 2009], [**Nandi et al,** 2010] and [**Della Bordella et al,** 2010] among others. The benefit of dynamic modeling of operations through entity life cycle is that both outcomes and processes are integrated, thus facilitating communication in the organization and the understanding of the models by different stake holders. This contrasts with other approaches to design processes as flows of activities or event chains that struggle to capture the actual intentions of stakeholders involved in the modeling work. The above and other industrial applications in the last decade conclusively prove the appealing nature of entity based life cycle modeling for operations. The BPM industry still has to fully benefit from this value and related field-based experience available.

Entities as a Modeling Paradigm for Case Management

Several schools and even different scientific disciplines have addressed the connection between process and entities in a deep way, yielding seminal methodology for the dynamic modeling of operations. As shown in previous Sections, the state-of-the-art in sister disciplines is rich and applies directly to Business Process Management. However, the issue of whether Business Process Management tradition includes all types of processes from organizations is still unclear, particularly considering the BPM ambiguities highlighted in Section 1 of this paper. The advent of Case Management provides more evidence to the importance of reviewing this topic in more depth, specifically: '*Is Case Management included in BPM?*, and the closely related question: '*Why do Case Handling Systems exist in distinct separation from BPM Systems?'* This Subsection presents and elaborates the point-of-view that the concepts of entity and life cycle modeling offer a unifying approach across a significant class of processes in organizations, including all those that traditional BPMS and Case Management systems address.

In [**Van der Aalst et al,** 2003b], the authors drew a clear distinction between two types of processes, namely, 'straight through processing' and 'case handling'. The former means that the processes are completely automated without any human intervention. The latter is about processes that need to accommodate much variation or exhibit excessive complexity to be captured in a process diagram (understood as a flow), requiring also human intervention. Interestingly, 'case handling' was addressed even earlier in what it appears to be the first track of BPM literature where workflows are recognized to be insufficient for a comprehensive process representations [**Van der Aalst et al,** 2001]. It is also noteworthy that the authors in [**Van der Aalst et al,** 2003c] characterized BPM as an extension of workflow management to support a wider class of operational processes. Support for processes requiring human intervention has been also the target of some products from the BPM software industry in the last decade (even when these products are not necessarily called "Case Management" or "Case Handling") [20]. This means that Case Management has not quite emerged from the rediscovery of human-based decision making in processes but from a different need. This need finds its actual justification in the complexity of processes with multiple variations imposed by business operations and the inadequacy of flows to represent such complexity. Admittedly, different human roles affecting the process lifetime during execution brings an additional requirement for adaptation and thus, more sources of complexity.

A key remark made in [**De Man,** 2009a] is worth quoting: "*… as most state-of-the-art BPM Systems adopt BPMN for process modeling, case management processes cannot yet benefit from the power of these systems"*. This comment openly implies that the Business Process

[20] Websphere Lombardi Edition (www.ibm.com) is an example where abundant support for processes requiring human participation is provided while the product is not called "case handling" or "case management".

Modeling Notation (BPMN) standardization work does not provide the adequate foundation to represent large classes of processes like those Case Management systems are aimed to support. The more recent paper published by the same author appears to reconfirm the thought [**De Man,** 2009b]. This means that the main modeling instrument for BPM, i.e., BPMN, is not a complete edifice yet to host all processes that organizations need. Considering that Case Handling was introduced at least in 2001, it is noteworthy that the standardization work in BPMN has continued to focus on workflows (whether fully automated or also admitting human participation) and has not included the significant type of processes that Case Management has come to address.

In the light of the above discussion, the question of whether Case Management is really included in BPM or not is still left unanswered. The advent of Case Management in early 2000's as a new paradigm after 10 years of BPM work is noteworthy, particularly when many significant organizational processes have shown a sustained need for flexibility and human intervention over several decades. These characteristics seemed to have been 'rediscovered' by some BPM scientists, practitioners and software companies [21]. The question should be posed both from a conceptual standpoint and an industry perspective. Unfortunately, the answer is not entirely clear in either context and depends on the point of view from the expert, company or analyst firm. For example, some analysts show distinct tracks to address each of these two domains of practice. For example, the authors in [**Forrester**, 2011] state: "*In speaking with more than 30 case management references across the US and Europe … we found a clear recognition that older process automation approaches based on traditional mass production concepts no longer fit an era of people-driven processes*". This course of reasoning is also found in earlier work [**Van der Aalst et al,** 2003c] where the authors stated: *"Claim handling is a new paradigm for supporting flexible and knowledge intensive business processes … The knowledge worker in charge of a particular case actively decides on how the goal of that case is reached, and the role of a case handling system is assisting rather than guiding him in doing so."* In [**Van der Aalst et al,** 2001] the authors maintained that BPM includes both workflow and case handling, while some companies clearly differentiate Case Management offerings from BPM products, even when they have both in their portfolio[22].

All published literature signals that Case Management emerged from the need to model processes where human intervention brings substantial variability to the course and progress of a process [23]. Indirectly, this statement implies that BPM systems have reflected poorly the automation-related requirements from those families of processes

[21] This is not entirely new as related phenomena have been observed and documented. For example, Van der Aalst et al. emphasize in [**Van der Aalst et al,** 2002]: "*Most computer scientists seem to have a frame of mind, typically derived from programming, where the notion of "state" is interpreted in a narrower fashion and essentially reduced to the concept of data*". This narrow interpretation of state in Van der Aalst's language is probably a key to the delay in the BPM industry to adopt Entity Life Cycle-based modeling into products.

[22] IBM Case Manager offers an example as the following concepts clearly illustrate the point: "*Traditional Enterprise Content Management (ECM) systems and the structured control of Business Process Management (BPM) are insufficient to satisfy the requirements of these applications* [Case Management business applications] …" [**IBM Case Management**, 2011].

[23] Forrester's definition of Case Management: *It is a semi-structured, but also collaborative, dynamic and information-intensive process that is driven by outside events and requires incremental and progressive responses from the business domain handling the case. Examples of case folders include a patient record, a lawsuit, an insurance claim, or a contract, and the case folder would include all the documents, data, collaboration artifacts, policies, rules, analytics, and other information needed to process and manage the case* [**Forrester,** 2010].

and operations Case Management fits in. These requirements have largely come from the focus of the BPM industry on certain families of processes which several authors call '*factory*' or '*mass*', clearly alluding to the repetitive nature of the underlying processes.

The ongoing focus of the BPMN standard has led Case Management advocates to claim that BPMN is insufficient or inadequate to represent those processes that are knowledge-intensive. Some recent literature goes further by separating "case" from "process" such as [**IBM Case Management,** 2011], where Case Management is said to have unique characteristics that differ from those of BPM (see, for instance, page 5 in the referenced book). Indeed, if the main BPM standard (i.e., BPMN) does not fit Case Management-amenable processes, then either BPM does not intend to include Case Management or a '*case*' is not specific type of '*process*'. While extreme positions are not advisable, particularly given that evidence abounds in opposite directions, the most likely explanation for this phenomenon is that BPM has grown with a focus on a restricted type of business processes. Consequently, both BPM standards and software industry have worked on their main subject of interest which has excluded many significant enterprise operations. Case Management is a significant example of this oversight but there are other important families that BPM has also ignored which are not 'cases' either, as will be seen in Part 2 of this series.

The central concept for Case Management is the *case* and not the activities or the routing [**Van der Aalst et al,** 2003]. Interestingly, this concept implies that the development of the process, i.e., the evolution of the main subject encompassed, is not conveniently described by a structured set of activities where all possible options for routing are laid out in a flow. As Case Management is relatively new (at least, in the context of the BPM software industry) and business process communities have enthusiastically embraced case handling as a way to cope with the restrictive nature of workflows, the possibility of using entity life cycle modeling deserves to be explored.

It is noteworthy that the parallelism between case handling and entity life cycle modeling finds its roots in past work done at the realm of Case Management software companies. Specifically, in [**Cordys,** *on-line presentation*], the diagram shown in Figure 9 has been presented. As is seen, this diagram is a metamodel defining the general structure of a '*case*'. Among other constructs shown on the figure (in fact, the original presentation contains substantial additional detail), the '*case*' is defined as a family of organized states along with a life cycle characterized by a number of transitions between pairs of states [24].

Unquestionably, the core of the structure of the '*case*' in this metamodel is identical to the concept of entity life cycle. The '*case*' is the entity, its life cycle is an integral part of the case definition and it is characterized by states and transitions whose activations are signaled by events, milestones and other conditions determined by contextual information. In other words, the '*case*' is equivalent to the concept of 'subject' (i.e., the central element of the process) and its 'evolution' can be defined by a family of states

[24] Notice the introduction in the Figure of the concept of Milestone Event as the definition of key conditions affecting the different state transitions in the case model.

along with a wide set of conditions and activities (including decisions and tasks made by humans) that make the case progress toward its completion.

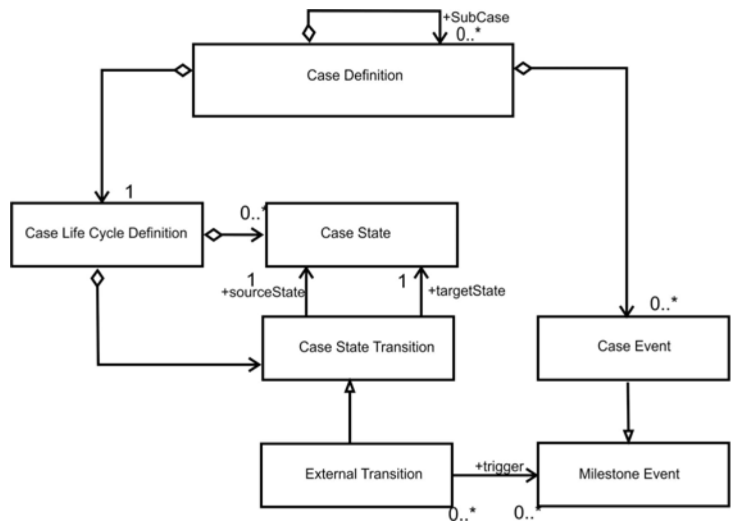

Figure 9. Case metamodel slightly adapted from [Cordys, *on-line presentation*] [25]

As was seen earlier in this Section, entity life cycle-based modeling does not have any explicit routing of activities structured into a pre-established or rigid flow, thus accommodating complex processes. Thus, this approach yields a flexible design for Case Management that supports a large variation of the conditions and activities that define the development of the process.

Interestingly, the adequacy of using entity life cycle modeling for Case Management promotes a significant convergence of conceptual techniques across different families of processes and potentially, of their corresponding supportive technologies as well. Unfortunately, these two process families have been kept separated, most likely as a consequence of the underlying Information Technology concepts involved. This point has been stressed quite explicitly in a remarkable passage by the authors in [**Van der Aalst,** 2003a; pp. 38]: *"It is interesting to think about the reason why many workflow products have problems dealing with the state-based patterns"*. This point made by the authors suggests that the chosen modeling mechanisms in BPM have traditionally left out the need to incorporate state-based processing. Unquestionably, the modeling of cases has not been a target of these information technology approaches to business processes.

On the other hand, this 'divide' among types of processes does not exist from a business point of view. Processes that call for frequent change and human intervention have been commonplace in business operations for a long time. Enterprise Resource Planning and Supply Chain Management have been exemplary domains for decades, with their processes implemented by means of packaged solutions from vendors like SAP or Oracle. The fact that operations flexibility, complexity or propensity to heavy human-

[25] Most probably, the year of publication of this diagram in the presentation posted on-line is 2007 (see bibliography).

centric decision-making may have been a second-thought for a specific IT approach to modeling processes it does not make the underlying processes or their requirements for automation new.

Interestingly, these subjects did not seem new either to some well-recognized contributors from the Systems Design and Structural Analysis schools 15 years ago, such as Jackson and Twaddle show in their book entitled *'Business Process Implementation – Building Workflow Systems'* [**Jackson,** 1997]. The authors start Chapter 4 with examples like insurance policy management and customer order management as targets of entity life cycle definition, the latter being the main goal and title of their chapter (page 69). It is noteworthy that these two examples lie also at the realm of those used in explaining the uniqueness of Case Management. Quoting the exact words from Jackson and Twaddle is very insightful : *"… a new life assurance policy may progress through the stages of proposal entry and validation; underwriting, where the risk is assessed and the premiums fixed accordingly; acceptance by the client; and finally, issue of the policy document* […]. *Usually, each stage of a life cycle is the responsibility of one workgroup of users* […]. *These life cycle stages are not invented by the system developers; nor are they the product of any kind of top-down design.* […]. *A life cycle is ultimately composed of tasks to be performed by the users in the office".*

Another prominent group of researchers [**Van der Aalst et al,** 2003a] have undertaken a careful investigation of the topic of *"states"* in workflows, back then in a world dominated by explicit control structures. While almost a decade has gone from the work done by Eindhoven and Queensland among other schools, there has not been much change in the workflow constructs used in BPM standards since then. This persistence shown by standard bodies is noteworthy [26], particularly in the presence of the many clear value-proposition made in support of adding 'state-based expressiveness' such as *milestones*, *deferred selection*, etc. [27] [**Van der Aalst et al,** 2002], [**Van der Aalst et al,** 2003a].

All the contributions reviewed above provide evidence about the viability of the convergence suggested in this subsection through more than a decade of work. Had approaches based on entity life cycle and related state-based modeling concepts been widely known, the two separate worlds of "conventional BPM" and "Case Management" would be much closer to each other than they actually are today. Unfortunately, as clearly said in [**IBM Case Manager,** 2011; pp. 9]: "*case and process modeling are different"*, a statement agreed upon by many other firms and independent authors. Whether any such difference remains under the light shed by entity life cycle-based modeling of processes is truly questionable.

---

[26] BPMN is already complex in terms of the number of constructs offered. From an Ontology point of view, it also exhibits significant redundancy [**Recker et al,** 2009] and its rich functionality still goes underutilized in practice [**Recker,** 2010]. This may explain the emergence of *Case Management Process Model* as a distinct standardization effort led by Open Management Group (OMG), the same organization hosting BPMN, so to avoid adding complexity to BPMN.

[27] The terminology here follows the work in [**Van der Aalst et al,** 2003c] and other related documents. For example, *milestone* was defined as a construct that helps to disable selected activities while maintaining others active; to check whether some part of a process is in a given state; etc. These concepts would have been useful to capture and instrument the flexibility needed in Case Management.

3. Conclusions and Further Work

This paper presented a thorough review of the state-of-the-art on entity-based dynamic modeling for enterprise operations. The entity-based life cycle approach was published for the first time about three decades ago. Specifically, the notion of entities as the main subjects of processes and entity life cycle as a technique for dynamic modeling of operations were introduced by K. Robinson in 1979, C. Rosenquist in 1982 and Jackson in 1983. Since then, Structural Analysis and Information Engineering schools have made significant contributions to modeling enterprise operations and closing the gap between process and data. As stated by different authors and proven by recently published works, these concepts are still foreign to most people in the BPM industry. This review focused entirely on modeling and methodology, although classic entity life cycle techniques also addresses implementation and bridge the IT domain to the business modeling of operations. Hopefully, this paper makes all this rich and highly inter-disciplinary intellectual capital on dynamic modeling of operations more visible to BPM communities.

The potential of using entity life cycle for modeling cases has also been addressed in this paper. Some BPM practitioners have just begun to recognize that for certain types of process analysis, entity life cycle diagrams are a valuable alternative to process maps: "*A state diagram can be an excellent modeling approach when a single entity is the focus of interest or when a natural sequencing of the activities is not obvious*" [**Ramesh,** 2011]. Indeed, this has been the technical foundation and practical belief in other sister disciplines for about 30 years. Tardieu has documented in 1992 [**Tardieu**, 1992] that the more non-deterministic the behavior is the more suitable entity-based dynamic modeling becomes. Cases and BPM appear often as separated disciplines addressing different types of processes. The entity life cycle approach offers the opportunity to achieve a conceptual convergence of modeling techniques across families of processes addressed in conventional BPM and Case Management. This work paves the creation of a research agenda to unify the two categories of processes so far disconnected. Also, the ideas discussed in this paper should hopefully help the BPM industry reunite the corresponding information technology systems into a single stream.

This paper also reviewed methods for selecting *critical* or *significant entities*. The origins of these techniques again go back to the work of different structured analysis schools, with some important contributions from selected BPM practitioners in the nineties. Cross-insemination with the data-base tradition is also important as entity finding is intimately related to the goal of chunking the business. The identification of significant entities in an organization is a consultative intervention that requires brainstorming sessions with different stakeholders. It is hoped that as time goes by, entities will become standardized in different industry segments so that reuse of content can help expedite new modeling needs. The fact that selecting entities is an ad-hoc technique was stressed by some contemporary researchers in [**Kumaran et al,** 2008]. Indeed, the edifice of process architecture needs a solid foundation as was suggested and addressed in

[**Sanz et al.,** 2011a]. Significant entities provide a more stable organizing principle than actions and their decompositions [**Ould,** 1995; 2005].

In modeling complex operations, several significant entities will have to be linked with each other as it is rare the case where one entity in isolation will fill the entire representation of the real-world being modeled. The necessary links among entities are established across their life cycles. There are some systematic patterns in the real-world where two or more entities appear in close interdependency by following some pre-specified forms of interaction. Part 2 of this series will address the concept of *aggregation* as a pattern encountered frequently in practice, i.e., operations whose main subject is described by an entity that aggregates other equally important entities necessary for its life cycle. Aggregation is also a useful approach to discover significant entities for dynamic operations modeling. On the other hand, the concept of entity that has dominated the literature in the last decades is based either on the notion of type, i.e., all real-world things within a selected set are abstracted into a common *king* or things treated as individual instances. These entities are common in workflows and cases, called *factory processes* in this series. While the "factory" is a critical part of an organization, other models are necessary beyond production processes. Part 2 of this series will reexamine these topics by presenting a wider view of entities and different classes of processes in real-world operations whose modeling benefits from these extensions. One category of such entities is a subject that comes under the form of an *ensemble of entity instances* whose importance goes beyond an entity type in the conventional sense. The corresponding processes being modeled by *ensembles* have been identified as *oversight processes* in enterprise operations which embed critical decision-making.

The notion of entity progression used in systems modeling literature has also been revisited. The development of the *central subject* dealt with in processes by information engineering, systems design and computer science viewpoints is immanent, i.e., intrinsic to its nature. This explains the term "life cycle" which abounds in the information science literature. But life cycle is a special form of development and non-factory processes call for a combination with other modes of evolution, such as teleology. Part 2 of this series starts a new journey along developmental models for processes beyond "factory". Interestingly, these operations include but are not limited to the classical scope of strategy processes.

A theory of process consists of statements that explain how and why process unfolds over time. Such a theory is needed not only to ground the conceptual basis of a process modeling and implementation study, but also to guide the design and conduct of empirical research. This work will be continued in Part 2 of this series and in a companion paper [**Sanz et al,** 2011a] where process taxonomy is revisited. There are many forms of progression beyond life cycle that are useful for thinking about and observing processes. For example, Van den Daele [**Van den Daele**, 1969; 1974] and Flavell [**Flavell**, 1972] propose a typology of developmental progressions that goes beyond simple unitary stages and includes multiple, cumulative, conjunctive, and iterative progressions of convergent, parallel, and divergent streams of activities that may unfold as a strategy process develops over time. Cross-fertilization with these seminal investigations from social sciences [**Van de Ven & Poole,** 2005] will shed new light to the BPM field.

Another important aspect needing further foundational work is the deep liaison between entity-based dynamic models and the ontology work of Bunge [**Bunge**, 1977]. The seminal work from Bunge has been extensively studied, applied and extended by Wand, Weber and other information systems pioneers resulting in the so-called *BWW ontology* [**Wand & Weber**, 1990; 1995]. There has also been a revamped interest in BWW propelled by several colleagues from the BPM discipline in recent years [**Green & Rosemann**, 1999], [**Recker et al,** 2005; 2009], [**Rosemann et al,** 2006]. BWW has been used to build a thorough comparison of the representational capabilities of different process design and modeling techniques [**Recker**, 2009]. The inclusion of entity-based life cycle modeling in the comparison of business process modeling techniques will provide a complete spectrum of methods. Quite interestingly, this study reveals a very close alignment between the actual outcomes of entity-based life cycle techniques and the BWW ontology. This alignment is obtained by making the *subject* of a process be associated to the *basic thing* proposed in the BWW ontology. This research is being completed at this time and will be reported elsewhere [**Sanz et al,** 2011b].

Finally, programming models and tools related to entity-based dynamic modeling are not included in this review. These are important subjects and thus, they merit and require a separate paper. These topics have also been addressed in the past decades and for example, the methodology work from Jackson, CCTA and several others include software instrumentation of the models into computer programs for execution. There has been work done relatively recently that goes well-beyond classical contributions from the 80's, 90's and early 2000's. For example, in [**Cohn et al,** 2008], [**Bhattacharya et al,** 2007b], [**Gerede et al,** 2007], [**Hull,** Narendra, Nigam, 2009], and more recently in [**Damaggio,** Deutsch, Hull, Vianu, 2011], [**Liu et al.,** 2011] and [**Hull,** 2011], new horizons are addressed, including semantic verification of entity-based models.

Acknowledgements

The author would like to thank a number of colleagues for many insightful discussions and comments on earlier versions of this paper. In particular, a special recognition is dedicated to R. Hull, P. Nandi, J. Vergo, I. Terrizzano, S. Glissmann, P. Irassar, V. Becker, L. Flores, R. Liu and K. Holley.

Bibliography

Aguilar-Saven, R. *Business Process Modelling: Review and Framework.* Int. J. Production Economics 90, pp. 129–149, 2004.

Becker, M. *Organizational Routines: A Review of the Literature.* Industrial and Corporate Change Vol 13, No. 4, pp. 643-677, 2004.

Berry, P. *Intelligent Workflow - State of the Art in Workflow, 1998.* http://www.ai.sri.com/~swim/resources/SOA-workflow.html

Bhattacharya, K.; Guttman, R.; Lyman, K.; Heat III, F.F.; Kumaran, S.; Nandi, P.; Wu, F.; Athma, P.; Freiberg, C.; Johansen, L.; Staudt, A.: *A Model-Driven Approach to Industrializing Discovery Processes in Pharmaceutical Research*, IBM Systems Journal Vol. 44, No. 1, 2005.

Bhattacharya, K., Caswell, N., Kumaran, S., Nigam, A. and F. Wu. *Artifact-centric Operational Modeling: Lessons Learned from Engagements*, IBM Systems Journal, 46(4), 2007a.

Bhattacharya, K.; Gerede, C. Hull, R.; Liu, R.; Su, J.: *Towards Formal Analysis of Artifact-Centric Business Process Models, Business Process Management*, pp. 288-304, 2007b.

Bhattacharya, K., Hull, R., Su, J. *A Data-Centric Design Methodology for Business Processes.* IBM Technical Report, 2009.

Bunge, M. *Treatise on Basic Philosophy Volume 3: Ontology I – The Furniture of the World.* Kluwer Academic Publishers, 1977.

Central Computer and Telecommunications Agency (CCTA); *Behavior and Process Modeling*, Series on Business Systems Development with SSADM, published by The Stationary Office, London, ISBN 0113308744, 2000.

Chao, T., Cohn, D., Flatgard, A., Hahn, S., Mark Linehan, M., Nandi, P., Anil Nigam, A. Pinel, F., John Vergo, J., Wu, F. *Artifact-based transformation of IBM Global Financing.* IBM Corporation, 2009.

Chen, P. *The Entity-Relationship Model – Toward a Unified View of Data.* ACM Transactions on Database Systems, Vol. 1, No. 1, pp. 9-36, March 1976.

Cohn, D.; Dhoolia, P.; Heath III, F.F. (Terry); Pinel, F.; Vergo, J.: *Siena: From powerpoint to web app in 5 minutes.* In Intl. Conf. on Services Oriented Computing (ICSOC), 2008.

Cohn, D., Hull, R. *Business Artifacts: A Data-centric Approach to Modeling Business Operations and Processes,* IEEE Data Eng. Bull. 32(3), pp. 3-9, 2009.

Cordys. *Modeling Case Management Processes.* On-line presentation. http://www.slideshare.net/jamet123/case-management-processes-203614

Damaggio, E., Deutsch, A., Hull, R., Vianu, V. *Automatic Verification of Data-Centric Processes*, Business Process Management Conference, Clairmont-Ferrand, France, 2011.

Davenport T. *Process Innovation: Reengineering work through information technology*, Harvard Business School Press, Boston 1993

Debevoise, T. *Business Process Management with a Business Rules Approach*, Roanoke, VA: Business Knowledge Architects, 2005.

Della Bordella, M, Ravarini, A., Wu, F., Liu, R. "*Key Business Artifacts: A Methodology for the Strategic Model within the MDBT (Model Driven Business Transformation) Framework.* MCIS Proceedings, 2010.

De Man, H. *Case Management: A Review of Modeling Approaches.* BPTrends, January 2009a.

De Man, H. *Case Management: Cordys Approach.* BPTrends, February 2009b. http://www.bptrends.com/publicationfiles/02-09-ART-BPTrends%20-20Case%20Management-DeMan%20-final.doc.pdf

DeMarco, T. *Structured Analysis and System Specification.* Yourdon Press Computing Series, 1978.

Deutsch, A., Hull, R.; Patrizi, F.; Vianu, V.: *Automatic verification of data-centric business processes,* in Proceedings of the 12th International Conference on Database Theory, ACM: St. Petersburg, Russia. 2009

Ebpml 1: *IBM's BEDL Analysis***.** Available at**:** ttp://www.ebpml.org/blog2/index.php/2010/04/21/mde-bpm-ibm-s-bedl-analysis

Ebpml 2: *Lifecycle Example***.** Available at**:** http://www.ebpml.org/blog2/index.php/2010/09/14/soa-bpm-rest-lifecycle-example

Ensor, D.; Stevenson, I. *Oracle Design,* O'Reilly, 1997.

Felin, T., Foss, N. *Organizational Routines: A Skeptical Look.* Technical Report from Coizueta Business School and Copenhagen Business School, October 23, 2004.

Finkelstein, C. *An Introduction to Information Engineering – From Strategy Planning to Information Systems.* Addison Wesley, 1989.

Fisher, A.; Gilat, D.; Wasserkrug, S. *Device, Method and Computer Program Product for Determining an Importance of Multiple Business Entities*, United States Patents. Patent No. US 2007/ 0100674, May 3, 2007.

Flavell, J.H. *An Analysis of cognitive-developmental sequences.* s.l. : Generic Psychology Monographs. 1972. 86: pp 279-350.

Forrester Report. *The Forrester Wave: Dynamic Case Management, Q1 2011.* by C. Le Clair, D. Miers, January 31, 2011.

Fritz, C.; Hull, R.; Su, J.: *Automatic construction of simple artifact-based business processes,* in Proceedings of the 12th International Conference on Database Theory. ACM: St. Petersburg, Russia, 2009.

Gerede, C.; Su, J.: *Specification and Verification of Artifact Behaviors in Business Process Models*, in Service-Oriented Computing – ICSOC 2007. pp. 181-192. 2009

Gerede, C.; Bhattacharya, K.; Su, J.: *Static analysis of business artifact-centric operational models.* Proceedings - IEEE International Conference on Service-Oriented Computing and Applications, SOCA 2007, pp. 133-140, 2007,

Green, P., Rosemann, M. *An Ontological Analysis of Integrated Process Modelling.* In Jarke, M. and Oberweis, A. (Eds.), CAiSE'99, LNCS 1626, pp. 225 – 240, 1999

Guha, S. Grover, V. Kettinger, W. & Teng, J. *Business process change and organizational performance: exploring an antecedent model.* Journal of Management Information Systems 14(1), pp. 119-154, 1997.

Hammer, M.; Champy, J. Re-engineering the Corporation: A Manifesto for Business Revolution, Harper Business, New York, 1993.

Harrison-Broninski, K. *Human Interactions - The Heart and Soul of Business Process Management,* Meghan-Kiffer Press, 2005.

Hares, J. *Information Engineering for the Advanced Practitioner.* Wiley, 1992.

Hawryszkiewycz, I. T. *Introduction to Systems Analysis and Design*, First Edition. Prentice Hall, 1988.

Hawryszkiewycz, I. T. *Introduction to Systems Analysis and Design*, Third Edition. Prentice Hall, 1994.

Hay, D. *Object Oriented Data Modeling: Entity Life Histories*, 1997. Available at http://www.essentialstrategies.com/

Heath, C., Sitkin, S. *Big-B versus Big-O: What is Organizational about Organizational Behavior?* Journal of Organizational Behavior 22, pp. 53-58, 2001.

Herzum, P., Sims, O. *Business Component Factory.* OMG Press. 2000.

Hull, R. *Artifact-Centric Business Process Models: Brief Survey of Research Results and Challenges.* OTM 2008 Part II, R. Meersman and Z. Tari (Eds.), LNCS 5332, Springer-Verlag, pp. 1152–1163, 2008.

Hull, R.; Narendra, N.C.; Nigam A.: Facilitating workflow interoperation using artifact-centric hubs. In Proc. Intl. Conf. on Service Oriented Computing (ICSOC), 2009.

Hull, R. *Toward Flexible Service Interoperations using Business Artifacts*, IEEE EDOC, 2011.

IBM Case Manager. *Advanced Case Management with IBM Case Manager.* IBM Redbooks, 2011. www.ibm.com/redbooks.

Indulska, M., Recker, J., Rosemann, M., and Green, P. *Business Process Modeling: Current Issues and Future Challenges*, in Advanced Information Systems Engineering - CAiSE 2009, P. van Eck, J. Gordijn and R. Wieringa (eds.), Springer, Amsterdam, pp. 501-514, 2009.

Jackson, M. *Principles of Program Design.* Academic Press, NY, 1975.

Jackson, M. *System Development.* Prentice-Hall International Series in Computer Science, 1983.

Jackson, M., Twaddle, G. *Business Process Implementation – Building Workflow Systems.* ACM Press, Addison-Wesley. 1997.

Johansson H. et al. *Business Process Reengineering: BreakPoint Strategies for Market Dominance*. John Wiley & Sons, 1993.

Kavakli, V., Loucopoulos, P.: *Goal-driven Business Process Analysis Application in Electricity Deregulation.* Information Systems Vol. 24, No. 3, pp. 187-207, 1999.

Khoyi, D. *Data Orientation*, Chapter 6 in "Mastering the Unpredictable", edited by Keith Swenson. Meghan-Kiffer, 2010.

Klein M.; Petti, C.: *A Handbook- Based methodology for Redesigning Business Processes*, MIT Sloan School of Management, Working Paper 4569-06, 2007.

Ko, R.; Lee, S., Lee, E. *Business Process Management (BPM) Standards: A Survey*, Business Process Management Journal Vol. 15 No. 5, pp. 744-791, 2009.

Kumaran, S., Nandi, P., Heath, T., Bhaskaran, K., Das, R. *Adaptive Documents, ADoc-oriented programming*, Applications and the Internet, (SAINT) pp.334-343, 2003.

Kumaran, S., Liu, R., Wu, F. *On the Duality of Information-Centric and Activity-Centric Models of Business Processes.* CAiSE '08 Proceedings of the 20th international conference on Advanced Information Systems Engineering, 2008.

Küster, J., Ryndina, K.; Gall, H. *Generation of Business Process Models for Object Life Cycle Compliance,* in Business Process Management. pp. 165-181. 2007.

Latour, B. *Reassembling the Social - An introduction to Actor-Network Theory.* Oxford, 2005.

Levin, D. *When Do Organizational Routines Work Well***?** *A New Approach to Knowledge Management. Technical Report.* Rutgers University, Newark, 2002.

Liu, R., Bhattacharya, K., Wu, F. *Business Contexture and Behavior Modeling Using Business Artifacts.* Lecture Notes in Computer Science, Volume 4495, Springer Verlag, 2007.

Liu, R., Vaculin, R., Shan, Z., Nigam, A. Wu, F. *Business Artifact-Centric Modeling for Real-Time Performance Monitoring,* BPM Conference, Clairmont-Ferrand, France, 2011.

Lu, R.; Sadiq, S. *A Survey of Comparative Business Process Modeling Approaches*. Business Information Systems Lecture Notes in Computer Science, Volume 4439, pp. 82-94, 2007.

MacDonald, R. A. *The Styles of Art History: Entities or Processes?* The Journal of Speculative Philosophy, New Series, Vol. 7, No. 1, pp. 48-63, 1993.

Martin, J. *Recommended Diagramming Standards for Analysts and Programmers.* Prentice Hall, 1987.

Martin, J. *Information Engineering, Book I - Introduction.* Prentice Hall, 1989.

Martin, J. *Information Engineering, Book II – Planning and Analysis*. Prentice Hall, 1990.

Miers, D. *Process Architecture: The Key to Effective Process Relationships*, Presentation from BPM Focus, 2009.

Nandi, P.; Dieter, K., Moser, S., Hull, R., Klicnik, V., Claussen, S., Kloppmann, M., Vergo, J. *Introducing Business Entities and the Business Entity Definition Language (BEDL): A first class representation of data for BPM applications.* IBM Corporation, April 2010.

Nigam, A.; Caswell, N. *Business Artifacts: An approach to operational specification.* IBM Systems Journal Volume 42, Issue 3, pp. 428- 445, July 2003.

Ould, M. *Business Processes – Modelling and Analysis for Re-engineering and Improvement.* Wiley, 1995.

Ould, M. *Designing a Re-engineering Proof Process Architecture,* Business Process Management Journal, 3 (3), pp. 232–247, 1997.

Ould, M. *Business Processes: Modelling and Analysis for Re-Engneering and Improvement, Wiley, 2005.*

Ould, M. *Business Process Management: A Rigorous Approach*. Presentation from Venice Consulting Ltd, September 2006.

Pentland, B., Feldman, M. *Designing Routines: On the Folly of Designing Artifacts while Hoping for Patterns of Action.* Information and Organization, 18, pp. 235-250, 2008.

Propp, V. *Morphology of the Folktale.* Translation published by University of Texas Press, Austin and London. Original in Russian, 1929.

Ramesh, R. *Using State Diagrams in Business Architecture.* Documented posted on business architecture community website: http://bizarchcommunity.com/ in 2011.

Recker, J., Indulska, M., Rosemann, M., Green, P. *Do Process Modeling Techniques Get Better? A Comparative Ontological Analysis of BPMN.* 16th Australasian Conference on Information Systems, Sydney, Nov. 2005.

Recker, J., Rosemann, M., Indulska, M., and Green, P. *Business Process Modeling: A Comparative Analysis,* Journal of the Association for Information Systems (10:4), pp 333-363, 2009.

Recker, J. *Opportunities and Constraints: The Current Struggle with BPMN.* Business Process Management Journal Vol 16, Number 1, 2010.

Redding, G., Dumas M.; ter Hofstede A.; Iordachescu, A.: *Generating business process models from object behavior models.* Information Systems Management, 25(4): p. 319-331, 2008.

Reijers, H., van Wijk, S., Mutschler, B., Leurs, M. *BPM in Practice: Who is Doing What ?* BPM 2010, LNCS 6336, pp. 45-60, 2010.

Rescher, N., *Process Metaphysics: An Introduction to Process Philosophy",* Albany, N.Y., SUNY Press, 1996.

Robinson, K. A. *Entity / Event Data Modeling,* The Computer Journal 22, pp. 270-281, 1979.

Rooze, E. J., Paapst, M. and Sombekke, J. *eCase Management, An international study in judicial organization*s, 2007, available at:
http://www.rechtspraak.nl/NR/rdonlyres/67AEE8E1-7A06-4E05BB64-C64245D13049/0/84120052_eCaseManagement.pdf

Rosemann, M., Green, P., Indulska, M., Recker, J. *Using Ontology for the Representational Analysis of Process Modeling Techniques.* International Journal Process Integration and Management, 2006.

Rosenquist, C. J. *Entity Life Cycle Models and their Applicability to Information Systems Development Life Cycles: A Framework for Information Systems Design and Implementation.* The Computer Journal, Vol. 25, No. 3, 1982.

Ryndina, K., J. Küster, Gall, H.: *Consistency of Business Process Models and Object Life Cycles,* in Models in Software Engineering. p. 80-90. 2007.

Sanz, J., Leung, Y., Terrizzano, I., Becker, V., Glissmann, S., Kramer, J., Ren, G-J. *Industry Operations Architecture for Business Process Model Collections,* BPM 2011, Clairmont-Ferrand, 2011a.

Sanz, J., et al. *Entity-based Process Modeling and the BWW Ontology Foundation*, to be published, 2011b.

Scheer, A., *ARIS – Business Process Frameworks.* Springer-Verlag, Berlin, 1998.

Sharp, A., McDermott, P. *Workflow Modeling: Tools for Process Improvements and Application Development. First Edition.* Artech House, 2001.

Siegel, J. *In OMG's OCEB Certification Program, What is the Definition of Business Process ?* Certification Whitepaper, OMG, May 2008.

Smith, H., Fingar, P. Business *Process Management: The Third Wave*, Meghan Kiffer. 2006

Strosnider, J.K.; Nandi, P.; Kumarn, S.; Ghosh, S.; Arsanjani, A. *Model-driven synthesis of SOA solutions*. IBM Systems Journal, 47(3):415–432, 2008.
Swenson, K. *Mastering the Unpredictable: How Adaptive Case Management Will Revolutionize the Way That Knowledge Workers Get Things Done,* 2010.MK

Tardieu, H. *Issues for Dynamic Modelling,* in Dynamic Modelling of Information Systems II, H. Sol and R. Crosslin (eds.), Elsevier, 1992.

Tsoukas, H. *What Does it Mean to be Sensitive to 'Process' ?* Keynote address European Group Organization Studies Colloquium 17, Lyon, July 2001.

Tsoukas, H., Chia, R. *On Organizational Becoming: Rethinking Organizational Change.* Organization Science 13/5, pp. 567-582, 2002.

Tsoukas, H., *Complex Knowledge: Studies in Organizational Epistemology*. Oxford University Press, 2005.

Toussaint, P., Bakker, A., Groenewegen, L. *Constructing an Enterprise Viewpoint: Evaluation of Four Business Modelling Techniques.* Computer Methods and Programs in Biomedicine 55, pp. 11–30, 1997.

Trkman, P. *The Critical Success Factors of Business Process Management*, International Journal of Information Management, 30(2), pp. 125-134, April 2010.

Van de Ven, A. *Suggestions for Studying Strategy Process: A Research Note*. Strategy Management Journal, vol. 13, pp. 169-188, 1992.

Van de Ven, A., Poole, M. *Explaining Development and Change in Organizations.* Academy of Management Review, vol. 20, No. 3, pp. 510-540, 1995.

Van de Ven, A., Poole, M. *Alternative Approaches for Studying Organizational Change.* Organizational Studies 26(9), pp. 1377-1400, 2005.

Van den Daele, L.D. *Qualitative models in developmental analysis.* Developmental Psychology 1(4), pp. 303-310, 1969.

Van den Daele, L.D. *Infrastructure and transition in developmental analysis.* Human Development 17, pp. 1-23, 1974.

Van der Aalst, W., *Formalization and Verification of Event-Driven Process Chains.* In Backhouse, R. Baetenm J.C. (Eds.), Computing Science Report 98/01. University of Technology, Eindhoven, 1998.

Van der Aalst, W., *Business Process Management Demystified: A tutorial on Models, systems and standards for workflow management*. Eindhoven: Springer Berlin / Heidelberg, 2004.

Van der Aalst, W., Barros, A.P., Hofstede, A., Kiepuszewski. *Advanced Workflow Patterns.* Eindhoven University of Technology, Technical Report, 2000.

Van der Aalst, W., Berens, P. J. *Beyond Workflow Management: Product-driven Case Handling.* In S. Ellis, T. Rodden and I. Zigurs (Eds.), International ACM SIGGROUP Conference on Supporting Group Work, ACM Press, pp. 42-51, 2001.

Van der Aalst, W., Hofstede, A., Kiepuszewski, Barros, A.P. *Workflow Patterns.* Queensland University of Technology, Brisbane, 2002.

Van der Aalst, W., Hofstede, A., Kiepuszewski, Barros, A.P. *Workflow Patterns.* Journal of Parallel and Distributed Data Bases 14, pp. 5-51, 2003a.

Van der Aalst, W., Hofstede, A., & Weske, M. *Business Process Management: A Survey*. In *Business Process Management*, 2003b.

Van der Aalst, W., Weske, M. and Grünbauer, D., *Case Handling: A New Paradigm for Business Process Suppor*t, Eindhoven Uuniversity of Technology, 2003c.

Veldhuizen, R. August 2010. Blog can be found: http://harx.nl/component/content/article/6-general/14-definitions-of-business-process-managment

Von Brocke, J., Rosemann, M. *Handbook on Business Process Management 1*. Springer, 2010.

Wand, Y., Weber, R. *An Ontological Mode of an Information System.* IEEE Transactions on Software Engineering 16, pp. 1281-1291, 1990.

Wand, Y., Weber, R. *On the Deep Structure of Information Systems.* Information Systems Journal, 5, pp. 203-223, 1995.

Wang, J. and Kumar, A., *A Framework for Document-Driven Workflow Systems*, BPM 2005, Van der Aalst (eds). LNCS 3649, pp. 285-301, 2005.

Weaver, P., Lambrou, N., Walkley, M. *Practical Business Systems Development Using SSADM – A complete tutorial guide.* Third Edition, Prentice Hall, 2002.

Weske, M. *Business Process Management: Concepts, Languages, Architectures*. Springer 2007.

Whetten, D. *Albert and Whetten Revisited: Strengthening the Concept of Organizational Identity*. Journal of Management Inquiry, 15 (3), pp. 219-234, September 2006.

Yongchareon, S., Liu, C.: *Process View Framework for Artifact-Centric Business Processes*, in R. Meersman et al. (Eds.): OTM 2010, Part I. LNCS, vol. 6426, pp. 26–43. Springer-Verlag Berlin Heidelberg, 2010.

Yourdon, E., Constantine, L. *Structured Design*. Prentice Hall, 1979.