Case Management: A Review of Modeling Approaches

Henk de Man

Background and objective

In 2008 the Object Management Group (OMG) released an RFI on “dynamic business activity modeling”. The need to model case management processes was the primary motivation towards this RFI. Several vendors of business process management systems (BPMS) have expressed interest in standardization of case management modeling. Cordys, a vendor of a BPMS platform, based on service oriented architecture (SOA) technology, is one of them. Cordys is increasingly focusing on and providing support for management and improvement of business operations. This year Cordys will release the next release of its business operations platform, called BOP-4. BOP-4 includes a model-driven case management solution, smoothly integrated with BPMN. Within the OMG, the Government Domain Task Force works on standardization of records management and case files. It is expected that case management standardization will be a joint effort of the Business Modeling and Integration and Government Domain task forces. It is too early to conclude whether, eventually, a new specification will be adopted, next to BPMN, or whether case management modeling capabilities will be incorporated into the next release of BPMN. In a series of two articles the discussion around “case management modeling versus BPMN” will be taken to the broader business process management community.

Certain executable processes, especially processes that deal with application integration and transaction processing, can adequately be expressed in BPMN. However, BPMN is not providing adequate support for controlling human-driven processes, also known as case management processes. A proper definition of case management will be provided later in this document. BPMN assumes that processes are executed based on a predetermined activity sequence. This assumption is not valid for case management processes. Human workers, especially knowledge workers, do require flexibility that cannot be expressed in BPMN. A standardized control language for case management does not yet exist.

As most state-of-the-art BPMS systems adopt BPMN for process modeling, case management processes cannot yet benefit from the power of these systems yet.

For this reason, in many administrative environments, case management often remains a paper-based process. E-mail is probably the most used tool for many case management processes. However, management of cases that are just administrated on paper, or that are just “automated” by e-mail, is often insufficiently productive, and does not sufficiently comply to today’s regulations, that e.g. aim to protect personal information of clients, citizens, patients, students, etc. Such systems also lack the possibility to adequately standardize, balance, schedule, authorize, audit, monitor and account for case work.

In several market segments, such as call center environments, advocacy, etc. niche applications that focus on management of cases in these specific environments have become available,. However, such systems can become barriers for programs that aim to standardize and optimize business process management, in a model-driven way, throughout the company, i.e. based on a BPMS.

There is a substantial market for modeling, execution and monitoring of case management processes, to be distinguished from, but closely integrated with sequential workflows or BPMN-style processes. This explains why a vendor like Cordys has expanded its footprint in BPMS, going beyond what can be expressed in BPMN, and why several vendors, including Cordys, are aiming for standardization of case management modeling.

In this paper we will further define and characterize case management, and review and evaluate the various approaches in business process management, which aim to support case management.

In a next paper we will present the Cordys approach for case management, based on a use case.

Defining case management

Case management, as a practice, is adopted in many areas. Among them are licensing & permitting in Government, application and claim processing in Insurance, patient care and medical diagnosis in Healthcare, and mortgage processing in banking. More industrial examples are call handling in call centers, sales & operations planning, invoice discrepancy handling, handling business requirements in R&D, engineering to order, etc.

Although case management is a well-known and well-recognized concept (and even profession) in business, there is, as yet, no standard definition. The following definition originated in the Healthcare industry and has become a de facto definition (without Healthcare-specific terminology): "*Case management is a collaborative process of assessment, planning, facilitation and advocacy for options and services to meet an individual's health needs through communication and available resources to promote quality cost-effective outcomes*" (Case Management Society of America (2008)).

This definition suggests the following:

- Central to the process is the "case" itself. In Healthcare: the "individual's health", with all related needs, and the additional information that accrues throughout its life cycle.
- Case management is collaborative and communicative. Case workers will have to respond to internal and external events.
- When events occru, it is often not predetermined which activities have to be executed and in what sequence, etc. The case worker has to assess needs first, after which further action has to be planned accordingly,
- The goal to be reached is clearer than the path to follow. There might be a variety of options to reach the goal, and the case worker's decisions will gradually determine the path. Processes will evolve rather than being predefined.
- The case worker is a knowledgeable person (knowledge worker) whose skills allow him or her the freedom to make decisions, for which (s)he is responsible.

Cases are often managed through a series of milestones. Event management and milestone control are core features of case management. A casual web-search will immediately sugggest that there is a high correlation among the terms "case management", "event" and "milestone."

Although in this article we will most often use the term "case management", one encounters synonomous terms in business parlance as well, such as "case handling", "case file handling", "case file management" and "case folder management". Some sources seem to suggest a distinction between "case management" and "case handling" in instances where "case handling" is concerned with the the operational handling of individual cases, and "case management" is concerned with managing the "flow of cases". A similar distinction is suggested by Ould (2005). He talks about "case processes" and reserves the word "case management" more for managing the flow of cases through the company. According to Ould a case process is "*the process that deals with one case or instance of a "unit of work" (UoW). A case process should be named Handle a… or Prepare a…, depending on whether the UoW is an input/trigger or an output/outcome*". Similarly: "*A case process is one that deals with one something, such as a customer order, or a clinical trial*" (Ould (2004)).

In Government it is very common to talk about a "case file". A case file or case folder is the central artifact in case management. The OMG Government Domain Task Force (2008) has dedicated a standard specification to the structuring and modeling of case files. A potential standard for case management should preferably leverage that case file standard.

According to Rooze et al. (2007), different types of cases can be distinguished. A workflow model for some cases, such as in "district court" (in Judiciary), is more of a "factory model", which is guided by the system and has very littlel room for deviation. At the "high court", a "service model" is followed, which is guided by the user and where the system basically just makes suggestions. Rooze distinguishes the following types of case management:

- Mass cases. These cases can almost be completely automated and be managed via "workflow management".
- Regular cases. The majority of cases fall into this category and can be managed via "planning of milestones".
- Special cases. These cases are the least routine and allow the user considerable discretion in determining how to handle them. Rooze suggests that there is no clear direction in how to manage these cases.

I should mention here that Rooze et al distinguish between "milestones" and "states". In dealing with any type of case, managers need to have "insight in matter status". "Matter status" means case status. A "state" becomes a "milestone" when planned dates, deadlines, etc. are set for it. "States" are becoming a (logistical) planning instrument then, as is suggested by Rooze in discussing "regular cases".

Review of approaches

In the previous section we came across some predominant aspects of case management that make it less feasible to model case management processes as sequential workflows (BPMN-style processes)

- The case data or case file serves as the central and shared context across all activities (and case workers) in the case.
- Case management is highly collaborative and event-driven (and state-controlled) by nature. In event-driven environments, events can be received in any sequence.
- The human factor is predominant. Case worker decisions may highly influence the process, as it evolves. Case workers are knowledge workers, who should not be restricted by what has to be done, but should rather by guided by what can be done.

Flow-based modeling (sequential workflow) seems sufficient for "mass cases" (as discussed above), but not for "regular cases" and "special cases". But what other process modeling paradigms are available to offer better support to "regular" and "special" cases ?

In the literature, three distinct principle types of process-control paradigms are available (see Berry (1998), Manolescu (2002) and Manolescu (2001)):

- Activity-based. Paradigm: Flow of control from activity to activity according to a predetermined sequence. Tasks are ordered based on the dependencies among them. BPMN, BPEL and XPDL are clearly examples of activity-based process modeling standards.
- Artifact-based. Paradigm: Activities are defined in the context of a business artifact, and these activities become available for execution based on data events that occur on the business artifact, possibly also as constrained by rules that evaluate the business artifact. According to Bhattacharya et al. (2007) and Bhattacharya et al. (2005), a business artifact is defined as a business entity that (1) is a record used to store information pertinent to a given business context, (2) has a distinct life-cycle from creation to completion, (3) has a unique identifier that allows identification of an artifact across the enterprise, and (4) is a record meaningful to the business user. Business artifacts are an abstraction to focus businesses on the core information entities that are most significant from an accountability perspective. The artifact is an information record that allows for measuring whether or not the business is on track to achieve their business goals.

Example: Order. The Order stores different aspects such as the date created, the result of the credit check, planned execution date, etc. The Order exists in different states or stages such as Pending Order, Planning, Live Order and Completed Order. Obviously business artifacts can typically play the role of case data context.

- Communication- (or conversation-) based. Paradigm: The process goal is reached as a result of a series of interactions (or communications, conversations) between process participants. The tasks that the individual participants perform are not modeled explicitly.

A similar distinction is recognized by Wang and Kumar (2005), suggesting "Process based", "Information based" and "Organization based" as the three process modeling perspectives.

Hybrid combinations of these approaches are possible as well.

Which modeling paradigm is most adequate will of course depend on which part is most dominant and most stable:

- Is it the sequence flow of activities, then activity-based modeling is the obvious paradigm.
- Is it the business artifact and its related events, life cycle states and rules, then artifact-based modeling should be considered.
- Is it communication or interaction, then communication-based modeling will be most appropriate.

Because of the inadequacy of modeling case management processes (at least the "regular" and "special" ones) based on activity-based (or flow-based) processes, we will explore ideas proposed in relation to the other two process modeling paradigms. We will evaluate the approaches afterwards.

Artifact-based process control

Bhattacharya et al. (2007) provides a concept whereby activity control is based on state transitions of and business rules that execute the business artifact. Activity sequencing is a result of what happens with the artifact and is therefore very flexible. A graphical modeling notation is not proposed. The article refers to case handling, as presented in Aalst et al. (2003), as a related topic.

The case handling concept as outlined by Aalst et al. (2003) and Vanderfeesten et al. (2006) is a hybrid, combining activity-based and artifact- (case-) based control principles. There is still a logical flow, but case workers can be authorized to skip and re-do activities. The case (and its data) is the main driver of the process. Activities can be constrained by the availability of case data elements. And certain case data elements can only be entered by certain case activities. Case data-based restrictions are modeled as explicit relationships between case data elements and activities. See Figure 1 for a graphical notation (C, A, D, R and F represent case, case activities, case data elements, roles and case activity forms respectively).

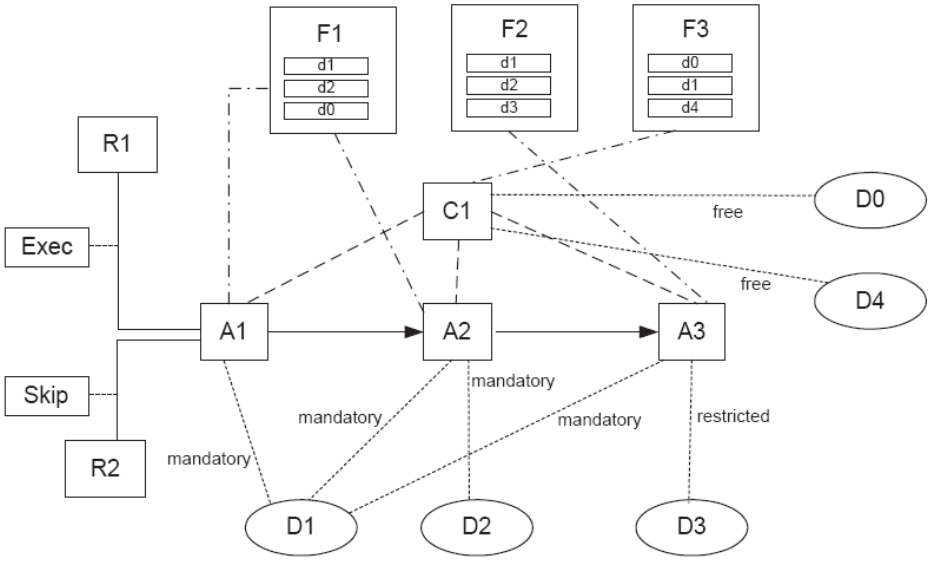

Figure 1: Case handling model of Van Der Aalst (Source: Aalst et al. (2003))

Vanderfeesten et al. (2006) talk about "product-based workflow design". The case is the product that is "manufactured" in the process. The case data structure is like a Bill-Of-Material in manufacturing. Data elements of the case are like "parts" in manufacturing, which are "mounted" into the case via "tasks", which are very much like assembly operations. This analogy with the manufacturing assembly world is striking by the way: manufacturing execution systems (MES) are well-known because of their support for human workers on the shop-floor. MES systems have been introduced, among others, because Enterprise Resource Planning (ERP) systems did not provide sufficient detail and flexibility for shop-floor workers. Knowledge workers in human-driven business processes need similar support. Knowledge workers require an even higher degree of flexibility than shop-floor workers however. We may expect therefore that case management processes will have to be even less activity-based and more artifact-based than proposed by Aalst et al. (2003) and Vanderfeesten et al. (2006). Later on in this article we will revisit the analogy between case worker support and shop-floor worker support. The concept as outlined by Aalst et al. (2003) and Vanderfeesten et al. (2006) has been implemented in FLOWer, by the company Pallas Athena.

Kaan (2005) suggests activity control based on rules that evaluate the case data, whereby activity-dependency on rules is depicted via Venn-diagrams, as indicated in Figure 2. However such diagrams can quickly become unreadable, when the number of rules continues to grow.

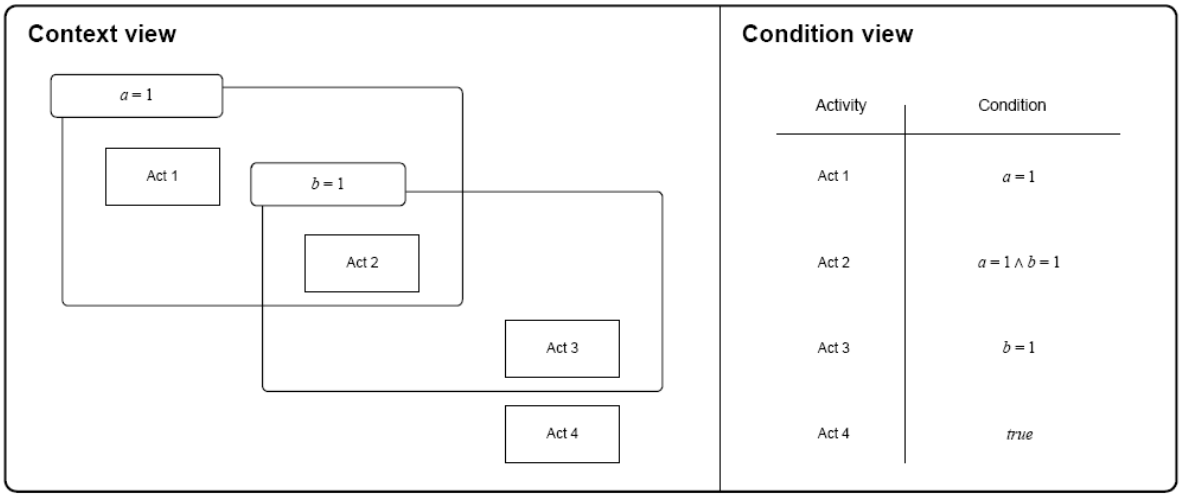

Figure 2: Venn diagram for rule-based case activity control (Source: Kaan (2005))

Ferguson and Stockton (2005) consider a state machine as an adequate approach for business artifact-based control: "*A business state machine is a programming metaphor that a business analyst can create with graphical tools. A state machine can represent a business artifact, that transitions through several well-defined states in response to specific life cycle events*." As proposed by Nigam and Caswell (2003) and Bhattacharya, Caswell et al. (2007), originating from the same school, a business artifact-based process modeling technique and corresponding notation is used, which makes use of states and state transitions of artifacts. The same technique is applied on an R&D process in Bhattacharya et al. (2005). Bhattacharya et al. (2005) also explain how the model is transformed to a native state machine representation for execution purposes. The business analysis notation itself moves a bit away from state machine notation itself, although the notion of states and transitions is still recognizable (see Figure 3).

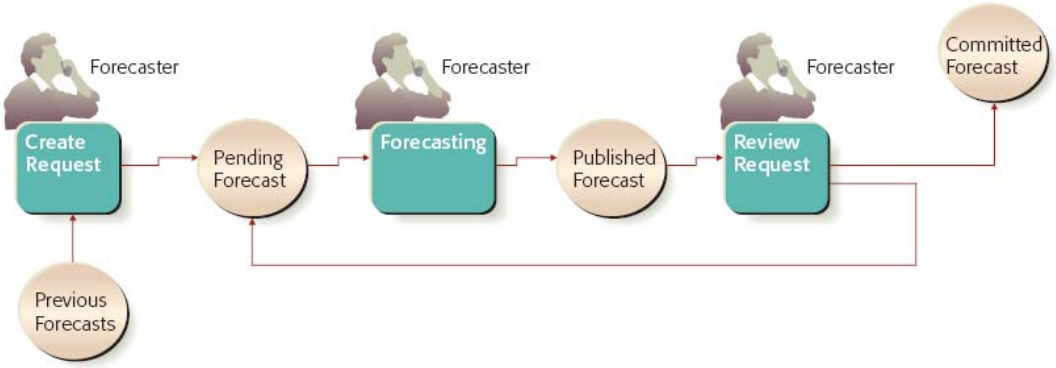

Figure 3: Business artifact-based process (Source: Bhattacharya, Caswell et al. (2007))

Medeiros et al. (1995) propose a concept of flexible workflow whereby activities can be chosen based on the availability of data, via constraints. This concept is applied to workflow in an R&D environment to conduct scientific experiments ("scientific workflow"). They also suggest that workflow instances, as they have evolved, are recorded for later analysis and potential re-use. Recurring workflow patterns can probably be abstracted for later re-use. Next to this, Medeiros et al. (1995) explicitly state that process workers in scientific workflows can decide themselves

which activities to include, and they consider this option a major difference between scientific workflows and traditional workflow. The system is making suggestions, and the users decide. Medeiros et al. (1995) do not show how this works out in terms of a process model however.

A similar concept is supported by Vortex, as presented by Manolescu (2001) and Hull and Su (1999). Vortex is meant to support workflows that don't have a pre-determined number of steps. In Vortex one specifies what tasks might potentially be performed for an incoming event (e.g. an insurance claim or a customer order), and what the artifact-based conditions are under which these tasks should be performed. More Vortex details are provided by Dong et al. (1999) and Fu et al. (2000). Vortex supports a kind of Event-Condition-Action (ECA) flows. A suggested graphical notation looks rather cryptic (see Figure 4).

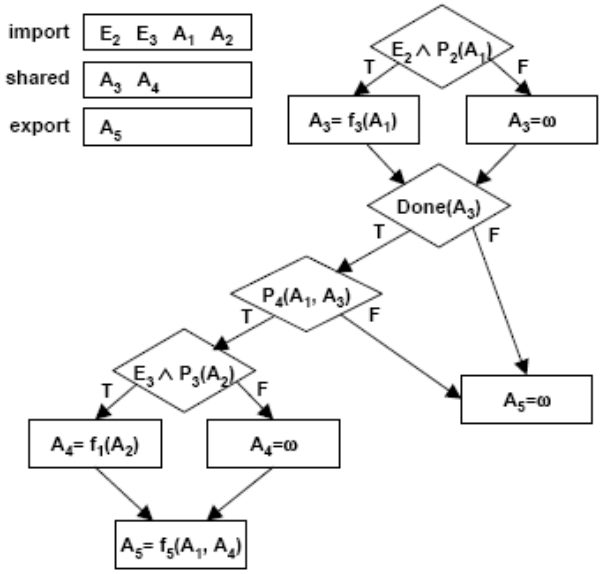

Figure 4: Vortex flow (Source: Dong et al. (1999))

Wang and Kumar (2005) present a similar concept, called "document-driven workflow". There is no explicit control flow. Execution of the process is driven by input documents. Basically control-flow ("hard constraints" or "business policies") and data-flow ("soft constraints") are mixed in a single diagram. For example, a customer cannot start to eat unless the food is produced ("soft constraint"), but whether the customer should pay before or after eating may vary from one restaurant to another ("hard constraint"). Data-based constraints (input-wise as well as output-wise) are depicted via connectors also. This has a drawback: bigger business problems, with many data-dependencies will result in spaghetti models. Data-icons do not only represent data objects or documents themselves, but are also used to represent data objects in their various states (life cycle stages), e.g. "Approved payment", "Rejected payment", "Order fulfilled", "Order cancelled", etc. (see Figure 5). When certain information is available (or is in a certain state), certain tasks can be done, which will create (or update) other information, etc. Wang and Kumar (2005) claim that a process which is based on data-flow dependencies ("soft constraints") is much more flexible than a control-flow based process--at least in situations where the enterprise information structure is more stable than activity sequences in processes, which is often the case. A change of the process will be as easy as adding or changing an activity locally by just specifying or changing what information it requires and/or what information it produces. Process paths need not be reconsidered.

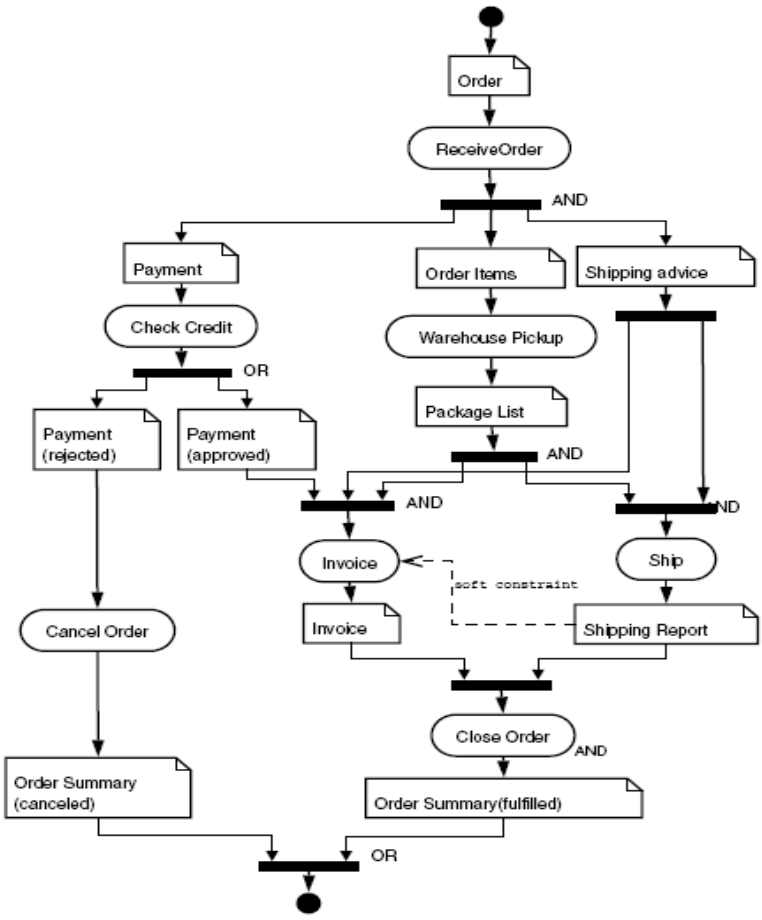

Figure 5: Document-driven workflow (Source: Wang and Kumar (2005))

Strahonja (2005) presents the use of UML state machines to model case management procedures in Legislation, such as bankruptcy, enforcement, litigation and criminal procedures. Strahonja (2005) considers state machine as the ideal language to analyze and understand case management. State machines are used as well to develop case management systems. State machine is recommended particularly because of its intuitive and visual modeling notation.

So-far we reviewed some artifact-based process modeling approaches. We will now look into some communication-based process modeling approaches, before we will evaluate the various approaches.

Communication-based process control

Possibly the most well-known communication-based process modeling technique is the Role Activity Diagram (RAD) technique. As human collaboration is an important aspect of case management processes, RAD as a communication-based process modeling technique was developed for that purpose. Or at least, one of the main proponents of RAD (Ould) wrote his book completely in the context of applying RAD to case management. Bushell (2005) reviews Ould's book on RAD (Ould (2005)), as well as a book from Harrison-Broninski, another RAD advocate (Harrison-Broninski (2005)). Bushell (2005) also presents a graphical representation of a RAD model (see Figure 6).

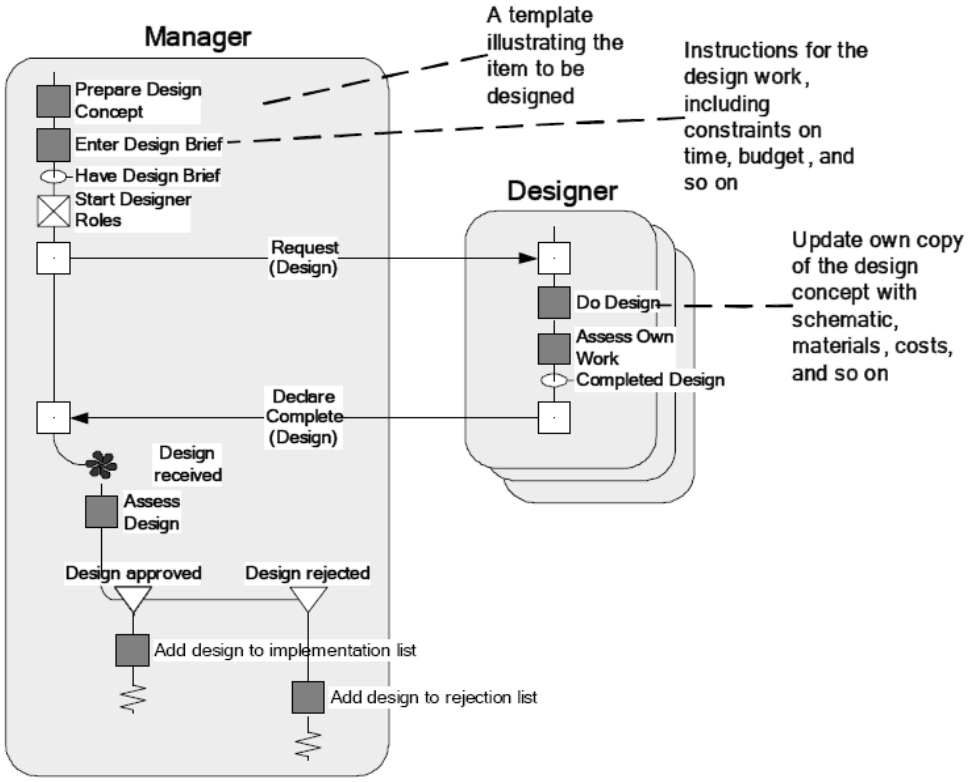

Figure 6: A RAD model (Source: Bushell (2005))

According to Beeson et al. (1999), RAD makes the sequencing of activities (including interactions) too rigid: "*Sometimes we would like to express the fact that a number of activities may take place after some point, but that their order is unimportant. However, there seems to be no notation in RAD modeling that can capture such property*".

According to List and Korherr (2005), the RAD notation is not specific enough to support process execution. It is more suitable as an analysis tool. It does not provide any linkage to an information model. E.g. data inputs and data events are not specified.

We also studied Ould's book (Ould (2005)), to learn to what extent case management modeling needs were addressed by RAD. We came to the following conclusions:

- RAD assumes a pre-defined sequence of activities within a role. This is in conflict with case management requirements. In this respect, RAD is as rigid as BPMN is ("strict ordering").
- The information (case data) cannot be defined in RAD. It is not possible to define control based on case data (there is no notion of data, or its related states, events and rules). RAD focuses on the state of conversation, but not on the state of the case or the business artifact itself.
- An explicit distinction is made between activities (that roles perform individually) and interactions. But in the context of case management, this distinction is often arbitrary and unclear. E.g. "meeting activity", "multi-participant voting activity", "pair programming activity" or a "co-development activity": are these activities or interactions ? In fact, they are multi-role activities. If BPMN were refined a bit, by allowing activities to be projected on more than one swim-lane, the same modeling construct could probably be achieved. In a diagram without swim-lanes, such activities could be modeled right-away.

- RAD can probably provide a complementary view on business collaboration between business partners, in situations where such interaction is pre-determined and contract-driven. But in dynamic case management environments, a predefined sequencing of interactions is too restrictive.
- BPMN is not sufficient to model case management processes. But it raises the question of to what extent RAD offers something that BPMN does not. And if not, RAD would probably not be of help to achieve our objective either. Advocates claim that RAD can model constructs that BPMN typically cannot.. This is questionable however. The central concept of RAD is the "role", and the way a "role" is depicted. But there is no essential difference between the notation of a RAD role and a swim-lane in BPMN, as can also be concluded from the discussion in Harrison-Broninski (2006). Parallel threads of execution (interaction), another assumed key uniqueness of RAD, is neither a real differentiator, since the same can be achieved in BPMN via gateways. A RAD process can begin at multiple points, whereas a BPMN process has a single start. But note that it has already been suggested by others, that BPMN should support that situation as well (see e.g. Silver (2007)). RAD focuses on interactions, but the next release of BPMN is expected to enhance interaction modeling significantly.

Based on an analysis of Harrison-Broninski (2005) for the same purpose, I concluded the following:

- RAD cannot be used to control the detailed work of case workers. "*Division of activity into tasks is not shown in RAD itself, but must be added either via a process modeling system that allows activity detail to be added, or as plain text (e.g. in activity description)*". This suggests that RAD is really too weak for our purpose.
- What Ould states as a core strength of RAD, parallel treads of conversation and all related associated detail, is qualified by Broninski as of no use for human-driven processes. Because human workers would like to work simultaneously on activities from multiple branches, or even better--regardless of any branches. Machines need to be programmed for multiple threading. Human workers need not be.
- Although RAD has no notion of a case data context, Harrison-Broninski suggests that roles are assumed to own their own data sources, which they deploy in the process. But there are two issues here: the model does not demonstrate this , and, in case management we assume a central and shared case data context, rather than a set of personal information sources in the roles.
- Human workers may want to work on activities in any sequence, which is not possible in RAD. In such instances,Harrison- Broninski suggests leaving out a lot of detail from a RAD. The vertical conversation read-line is removed as well, because it implies a fixed activity sequence. Rather each activity might have a pre-condition and a post-condition. Only when the post-condition of one activity matches the pre-condition of another one, are the two activities connected by the vertical line. This causes a radically different interpretation of the vertical line in RAD ! The end result of Harrison-Broninski's relaxations is that the expressiveness of a process model for a dynamic process is very low. The model is hardly telling anything anymore (see Figure 7 for an example of a simplified RAD diagram).

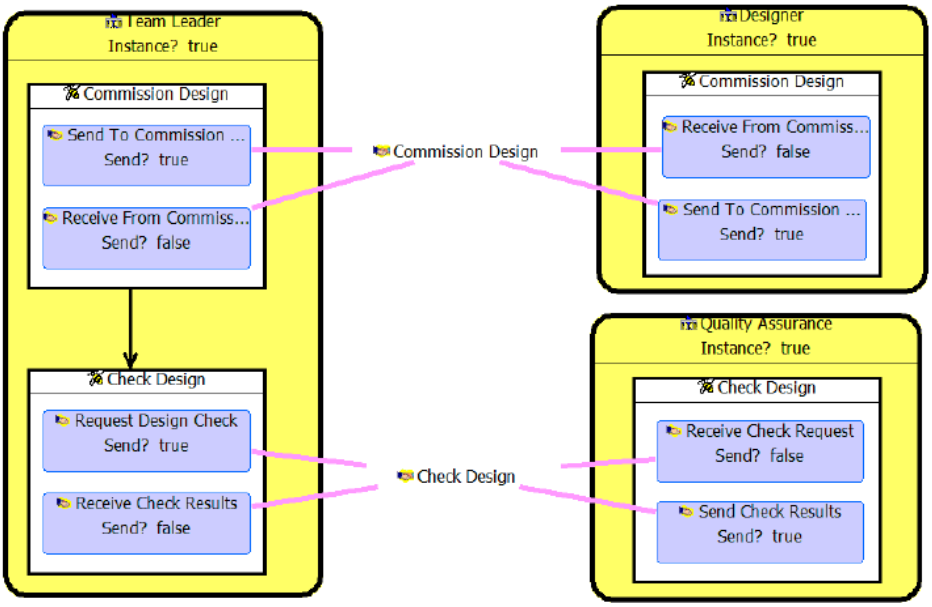

Figure 7: Simplified RAD model of Harrison-Broninski (Source: Harrison-Broninski (2006))

A strong aspect of Harrison-Broninski's contribution is that he suggests that human-driven processes, like case management processes, are different from other processes, in that work preparation and work distribution design are incorporated in the process itself. Process workers define processes, and the process of defining a process is part of the process. Harrison-Broninski explains this based on observations of R&D processes, following the "REACT pattern": Research, Evaluate, Analyze, Constrain, Task. People start to explore a new subject ("Research"), "Evaluate" how it can applied in their own business, decompose the work into tasks for other workers to involve in the project ("Analyze"), determine controls and authorizations for these tasks ("Constrain") and finally get these tasks executed ("Task"). This is a good idea which is very applicable to case management processes, although Harrison-Broninski in his book never made clear how this would lead to something better than RAD.

A different school of communication-based processes is oriented to the Language-Action Perspective (LAP). LAP is discussed by Kethers and Schoop (2000) and Goldkuhl (2003). According to LAP, people reach goals through communication. The basic unit of communication is a "speech act". A speech act is a minimal functional unit in human communication (see Jaworowska (2001)), like "request", "promise", "demonstrate performance", and "accept". ActionWorkflow and DEMO are two prominent applications of the LAP theory to business process management. Both consider processes as composed from interactions (in DEMO also called "transactions"), which are based on speech acts. Tasks that are performed by the individual participants themselves ("the actual work") are not modeled. More DEMO details are provided by Dietz (2003), including graphical representations of DEMO models. Figures 8 and 9 provide a high level abstraction and a lower level abstraction respectively (the rectangles represent roles, and the circles represent transactions).

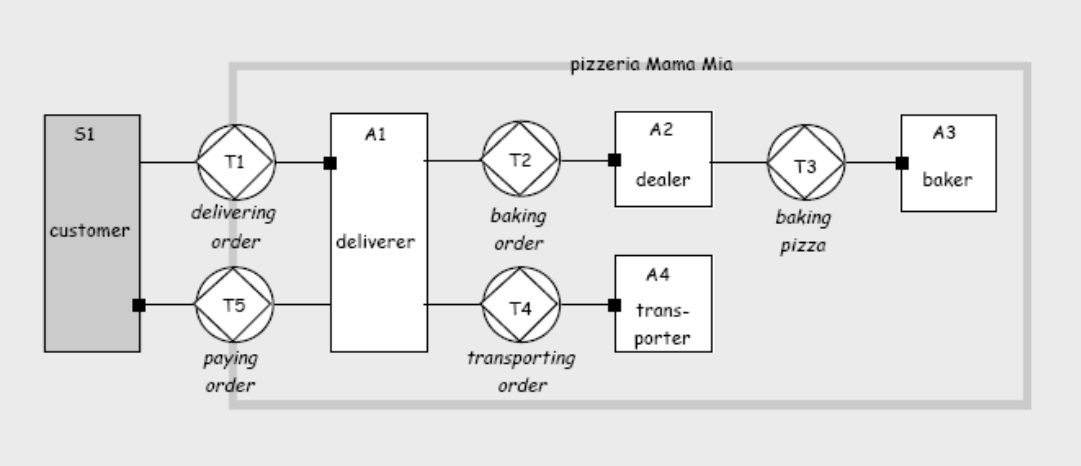

Figure 8: High-level view of DEMO model (Source: Dietz (2003))

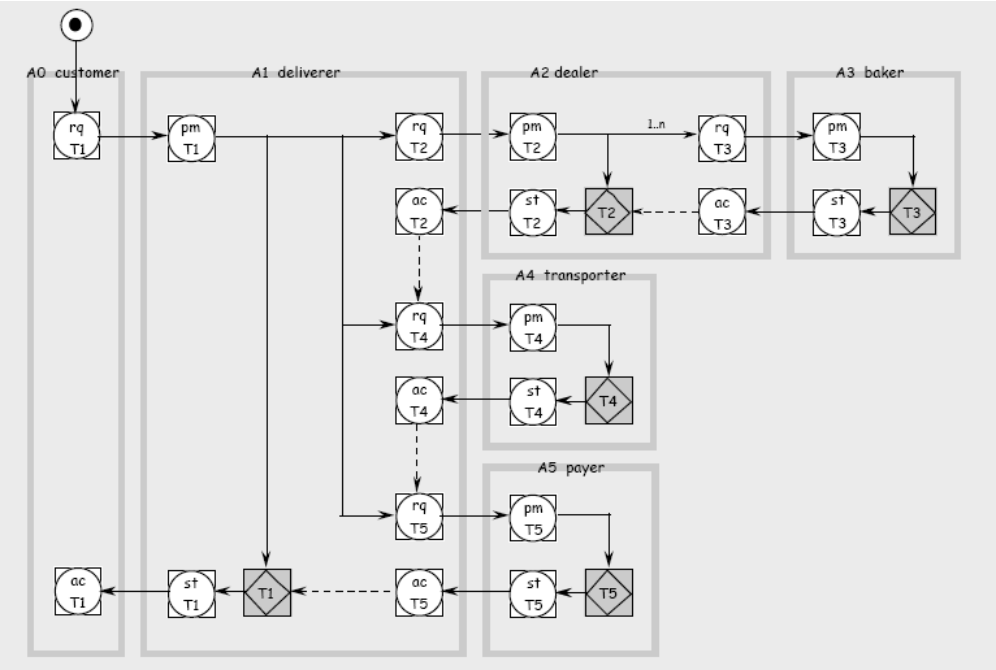

Figure 9: Lower level view of DEMO model (Source: Dietz (2003))

Also Manolescu (2001) mentions ActionWorkflow as an example of communication-based process-control.

LAP has also been much criticized. The following objections are stated by Weigand (2005):

- It is questionable whether interactions should be modeled as the primary means of process control, because interactions and commitments are often left implicit in day-to-day life. Communication should rather be considered a complementary view.
- Communication-based models aren't precise enough. The trigger to do something, the actual thing to be done (the "what"), the authorization to do something, etc. are all implied by a single interaction. This will result in abstract and overly implicit modeling.
- The LAP community remained small over the years. Apparently LAP did not add significant value.

Lyytinen (2004) is even sharper in criticizing LAP:

- LAP is academic. There is no relevancy for industry.
- There is no triggering reason why practitioners would apply LAP to problems that can be approached in other ways.

Kethers and Schoop (2000) are rather critical in evaluating ActionWorkflow (one of the well-known LAP approaches):

- There is no definition of the object of the workflow (basically, there is no case data, or no business artifact as context …). Consequently data-based conditions to determine whether certain communication goals have been reached, or whether certain phases of communication can be started, cannot be defined, which causes the approach to be "soft". Also, alternatives cannot be modeled.
- Interactions are possible between only two stakeholders.
- Only "communication activities" can be represented, and no "material activities" (the activities that the stakeholders perform themselves, by which they create their deliverables). Kethers and Schoop (2000) demonstrate that claim based on case studies in New Product Development: the actual new product development activities, such as prototype testing, or production, cannot be represented. This makes the model fairly incomplete.
- The four phases of an ActionWorkflow pattern are strictly sequential. E.g. once the "performance" phase has been entered, there is no way of backtracking to the "commitment" phase.
- The model does not support any perspective of timeliness (e.g. time events, etc.).

As stated by Weigand (2005), DEMO was inspired by the ActionWorkflow approach. Part of the criticism on ActionWorkflow also applies to DEMO therefore. From Oren and Dietz (2003) it is clear that DEMO also has difficulty in modeling "material activities" (see above), because activities within a role are left implicit. If one wanted to model such activities in DEMO, the activities would have to be modeled as roles. Note that RAD is more detailed than DEMO in this respect: RAD can define some "material activities" within a role, although still at a high-level.

As suggested by Dumay et al. (2005) and Dumay (2005), DEMO is not sufficient and specific enough as a basis for process automation. It is more suitable for capturing high-level business interaction. The more detailed human-to-human interaction is too subtle to formalize easily. Also Lyytinen (2004) states that DEMO is good for business process analysis, not for business process execution. Reijswoud and Rijst (1994) show how DEMO is used for high-level business analysis.

Rittgen (2004) investigates to what extent DEMO can support dynamic processes. Rittgen suggests not modelling dynamic transactions in a DEMO diagram, but to leave them out,and define rules-based behavior instead. Such behavior cannot be defined in DEMO models however. DEMO itself does not provide the means to model dynamic process models (and this is one of the prerequisites for modeling case management processes !).

According to Hommes and Reijswoud (2000), expressiveness in DEMO models is not so high. In additjon, DEMO is not adequate to model larger scale business problems, as the number of arrows and icons will multiply and become unreadable.

In the literature there has also been discussion about combining RAD and LAP. Beeson and Green (2003) investigate how RAD can be detailed further based on LAP. Harrison-Broninski (2005) suggests the same: Incorporating the concept of "speech acts", to refine the interactions in RAD, and to get more support for semantic meaning of conversations (although it wasn't clear in his book how this really materialized into a RAD adaptation). These attempts clearly indicate that there is a general belief that RAD is not providing sufficient detail. But given the criticism of

LAP, ActionWorkflow and DEMO, as discussed above, we cannot expect that a LAP extension of RAD will make RAD suitable for our purpose.

The OMG is currently working on the second release of BPMN. It seems that BPMN 2 will include support for communication-based approaches as well. The OMG seems to be proposing "conversation diagrams", and diagrams for collaboration in more detail, and that does suggest striking similarities with some of the DEMO diagrams as discussed above. As the BPMN 2 specification is still in progress, we will not speculate further on its details in this article.

Evaluation and conclusions

From the analysis above, we can derive the following conclusions:

- Communication-based approaches support business analysis purposes better than process-execution purposes. Whereas our objective is to find a modeling approach for the purpose of process execution (control language).
- Communication-based approaches lack sufficient detail, and their acceptance and adoption is relatively low.
- Communication-based approaches such as RAD (and its adaptations), ActionWorkflow and DEMO cannot cope with the dynamics of case management processes.
- In case management processes, case data or business artifacts and their related life cycle states and events are the stable parts. The notion of which activities *can be* (not necessarily *should be*) executed, as triggered by which events, under which data conditions or in which case life cycle stages, is also stable, especially for "regular cases". Activity sequences (control flow) are dynamic, and evolve in run-time. Collaborations will mostly evolve in run-time as well. This suggests that "artifact-based" modeling is the appropriate paradigm for case management processes.
- Of all the artifact-based approaches as discussed above, possibly the most attractive one is "document-driven workflow" (Wang and Kumar (2005)), although the other approaches have some useful elements that can be applied in the context of case management. It seems to be very useful to represent and integrate case data in the model by case-related life cycle states, events and conditions (rules). As discussed above, Wang and Kumar (2005) are to focused on an input-output representation: data (or data life cycle states) are an input of certain activities and an output of other activities. This results in too many connectors in the diagram. Preferably a more relaxed way of modeling is adopted.

In addition, it is important to stay as close as possible to BPMN notational style--"look and feel", notation of event, activity, decomposition, etc. This will also ease migration later, in the event the OMG should decide to integrate case management process support into BPMN itself. Note that a close integration between control flow-based processes and artifact-based processes is required anyway: it should be possible to start a BPMN process from a case management process, and vice versa. The design-time modeling experience around this integration will have to be smooth.

After having reviewed the various approaches above, we also observe that one important aspect of case management remained under-emphasized: Modeling of the human factor, the degree of freedom for case workers to decide which activities to include in the process. Medeiros et al. (1995) elaborate a little bit on this subject, although not in much detail. Harrison-Broninski (2005) talks a lot about it, but does not show how to model it.

In terms of the distinction as used by Rooze et al. (2007), "artifact-based" modeling approaches seem to focus most on "regular cases", but aren't explicit enough with respect to "special cases". We must understand, however, that the more "special" a type of case is, the less likely it will be that control can be solely based on predefined case-state models and/or case date-based rules. There should also be an explicit way to define "process definition" as part of case work for case workers. Note that, especially in the beginning, when the company starts formalizing case management practice, it will often be difficult to specify business rules to control applicability of case activities. Sometimes it is impossible to specify such rules ("special cases"), and sometimes

the company will have to “learn” first, even in case of some “regular cases”. Maybe many cases will start as “special” and later turn into “regular”. Case instances may initially evolve based on case worker decisions. Based on exploring (or “mining”) case instance history, certain patterns might appear to be recurring. Sometimes, in situations involving “mass cases”, sequential workflows (sub-processes) might be abstracted from these patterns, and sometimes, for “regular cases”, state- and rule based constructs might be abstracted. Case models might evolve over time in this way.

The Cordys case management modeling approach explicitly supports modeling of how case workers can define the process themselves, as part of their process work. We will present this approach in the next article. We will also show how case data-related life cycle states, events and conditions (rules) can further guide the case workers in defining the process as it moves along.. In this way, the Cordys approach is business artifact- or “case file”-based.

References

[1] Aalst, Wil. M.P. van der, Weske, M. and Grünbauer, D., Case Handling: A New Paradigm for Business Process Support, 2003 http://is.tm.tue.nl/staff/wvdaalst/publications/p252.pdf

[2] Beeson, I. and Green, S., Using a Language Action Framework to Extend Organizational Process Modeling, 2003 http://www.cems.uwe.ac.uk/~sjgreen/UKAIS2003.pdf

[3] Beeson, I., Green, S., Sa, J. and Sully, A., Linking Business Processes and Information Systems Provision in a Dynamic Environment, 1999
http://www.cems.uwe.ac.uk/~sjgreen/ISJournal.pdf

[4] Bhattacharya, K., Caswell, N. S., Kumaran, S., Nigam, A. and Wu, F. Y., Artifact-centered operational modeling: Lessons from customer engagements, 2007
http://www.research.ibm.com/journal/sj/464/bhattacharya.pdf

[5] Bhattacharya, K., Gerede, C., Hull, R., Liu, R. and Su, J., Towards Formal Analysis of Artifact-Centric Business Process Models, 2007
http://alumni.cs.ucsb.edu/~gerede/research/papers/bghls-bpm07-Artifact.pdf

[6] Bhattacharya, K., Guttman, R., Lyman, K., Heath III, F. F., Kumaran, S., Nandi, P., Wu, F., Athma, P., Freiberg, C., Johannsen, L. and Staudt, A., A model-driven approach to industrializing discovery processes in pharmaceutical research, 2005
http://www.research.ibm.com/journal/sj/441/bhattacharya.pdf

[7] Berry, Pauline M., Intelligent Workflow - State of the Art in Workflow, 1998
http://www.ai.sri.com/~swim/resources/SOA-workflow.html

[8] Bushell, S., A Review of Two Books Describing a New Foundation for Business Process Management, 2005
http://www.businessprocesstrends.com/publicationfiles/10%2D05%20BR%20BPM%20and%20Human%20Interactions%20%2D%20%20Sue%20Bushell%2Epdf

[9] Case Management Society of America (CMSA), What is Case Management?, 2008
http://www.cmsa.org.au/definition.html

[10] Dietz, J. L.G., Generic Recurrent Patterns in Business Processes, 2003
http://is.tm.tue.nl/bpm2003/download/presentation%20Dietz.pdf

[11] Dietz, J. L.G., Designing Technical Systems as Social Systems, 2003
http://infolab.uvt.nl/research/lap2003/dietz.pdf

[12] Dong, G., Hull, R., Kumar, B., Su, J. and Zhou, G., A Framework for Optimizing Distributed Workflow Executions, 1999
http://citeseer.ist.psu.edu/cache/papers/cs/13639/http:zSzzSzwww.dcs.gla.ac.ukzSzdbpl99zSzfinalzSzDongHullEtAl.pdf/dong99framework.pdf

[13] Dumay, M., Demo or Practice: Critical Analysis of the Language/Action Perspective, 2005
http://arxiv.org/ftp/cs/papers/0410/0410006.pdf

[14] Dumay, M., Dietz, J. and Mulder, H., Evaluation of DEMO and the Language/Action Perspective after 10 years of experience, 2005 http://www.vits.org/konferenser/lap2005/Paper%207-LAP.pdf

[15] Ferguson, D. and Stockton, M., SOA programming model for implementing Web services, Part 1: Introduction to the IBM SOA programming model, 2005 http://www.ibm.com/developerworks/webservices/library/ws-soa-progmodel/

[16] Fu, X., Bultan, T., Hull, R. and Su, J.,Verification of Vortex Workflows, 2000 http://www.cs.ucsb.edu/~su/papers/2001/TACAS01.pdf

[17] Goldkuhl, G., Conversational Analysis as a Theoretical Foundation for Language Action Approaches?, 2003 http://infolab.uvt.nl/research/lap2003/goldkuhl.pdf

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

[19] Harrison-Broninski, K., BPM, Anyone? A Response to Paul Harmon's response to my response to the BPTrends Email Advisor of November 15, 2005, 2006 http://www.businessprocesstrends.com/publicationfiles/02%2D06%2DDIS%2DRe%2DReStandardizingBPNotation%2DHarrison%2DBroninski%2Epdf

[20] Harrison-Broninski, K., Human Interaction Management and the future of BPM, 2006 http://humanedjucation.com/white_papers/Keith%20Harrison-Broninski%20-%20Process%202006.pdf

[21] Hommes, B. J. and Reijswoud, V. van, Assessing the Quality of Business Process Modelling Techniques, 2000 http://csdl2.computer.org/comp/proceedings/hicss/2000/0493/01/04931007.pdf

[22] Hull, R. and Su, J., The Vortex Approach to Integration and Coordination of Workflows, 1999 http://www.cs.ucsb.edu/~su/papers/1999/cross99.pdf

[23] Jaworowska, J., Speech Act Theory, 2001 http://instructional1.calstatela.edu/lkamhis/tesl565_sp04/troy/spchact.htm#The%20Meaning%20of%20Speech%20Acts

[24] Kaan, K., Modeling support for the Case Management paradigm and Expectations Management in process innovation, 2005 http://serv2.ist.psu.edu:8080/viewdoc/summary;jsessionid=84A4B3AD86EBC5D019341279FF376750?doi=10.1.1.102.5338

[25] Kethers, S. and Schoop, M., Reassessment of the ActionWorkflow Approach: Empirical Results, 2000 http://dbis.rwth-aachen.de/conf/lap2000/paper151-169.pdf

[26] List, B. and Korherr, B., An Evaluation of Conceptual Business Process Modelling Languages, 2005 http://publik.tuwien.ac.at/files/pub-inf_3845.pdf

[27] Lyytinen, K., The Struggle with the Language in the IT – Why is LAP not in the Mainstream?, 2004 http://www.scils.rutgers.edu/~aakhus/lap/Lyytinen.pdf

[28] Manolescu, Dragos A., Micro-Workflow: A Workflow Archicture Supporting Compositional Object-Oriented Software Development, thesis, 2001 http://micro-workflow.com/PDF/phdthesis.pdf

[29] Manolescu, Dragos A., Workflow Enactment with Continuation and Future Objects, 2002 http://micro-workflow.com/PDF/wecfo.pdf

[30] Medeiros, C. B., Vossen, G. and Weske, M., WASA: A Workflow-Based Architecture to Support Scientific Database Applications, 1995 http://citeseer.ist.psu.edu/cache/papers/cs/7399/http:zSzzSzwwwmath.uni-muenster.dezSzmathzSzinstzSzinfozSzuzSzdbiszSzResearchzSz..zSzWeskezSzPaperszSzfb02-95.pdf/bauzermedeiros95wasa.pdf

[31] Nigam, A. and Caswell, N. S., Business artifacts: An approach to Operational specification, 2003, http://www.research.ibm.com/journal/sj/423/nigam.pdf

[32] OMG Government Domain Task Force (GovDTF), MDA Models for Electronic Records Management Services, OMG, 2008

[33] Oren, E. and Dietz, J.L.G., Development of a DEMO based workflow management system, 2003 http://www.eyaloren.org/pubs/DEMObasedWFMS.pdf

[34] Ould, Martyn A., Getting your Head Round Spaghetti, 2004 http://www.bptrends.com/publicationfiles/10%2D04%20ART%20Spaghetti%20%2D%20Ould%2Epdf

[35] Ould, Martyn A., Business Process Management, A Rigorous Approach, Martyn A. Ould, BCS, 2005.

[36] Reijswoud, V. E. van and Rijst, N. B.J. van der, The Analysis of Planning Activities using the DEMO Transaction Process Model, 1994 http://citeseer.ist.psu.edu/cache/papers/cs/5362/http:zSzzSzwww.demo.tudelft.nlzSzfullpaperszSzreijswrijst94.pdf/vanreijswoud94analysis.pdf

[37] Rittgen, P., Supporting Planned and Ad-Hoc Changes of Business Processes, 2004 http://ftp.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-236/paper4.pdf

[38] Rooze, E. J., Paapst, M. and Sombekke, J., eCase Management, An international study in judicial organizations, 2007 http://www.rechtspraak.nl/NR/rdonlyres/67AEE8E1-7A06-4E05-BB64-C64245D13049/0/84120052_eCaseManagement.pdf

[39] Silver, B., My BPMN Wish List, 2007 http://www.brsilver.com/wordpress/2007/11/28/my-bpmn-wish-list/

[40] Strahonja, V., Modeling Legislation By Using UML State Machine Diagrams, 2005 http://crosbi.znanstvenici.hr/datoteka/249560.1113.pdf

[41] Vanderfeesten, I., Reijers, Hajo A. and Aalst Wil M.P. van der, Product Based Workflow Design with Case Handling Systems, 2006 http://fp.tm.tue.nl/beta/publications/working%20papers/Beta_wp189.pdf

[42] Wang, J. and Kumar, A., A Framework for Document-Driven Workflow Systems, 2005 http://www.personal.psu.edu/axk41/BPM05-jerry-reprint.pdf and http://bpm2005.loria.fr/files/PDFs/Session3/Wang.pdf

[43] Weigand, H., LAP: 10 years in retrospect, 2005 http://infolab.uvt.nl/pub/weigandh-2005-105.pdf

---

Author

Henk de Man is Research Director of Cordys.

hdman@cordys.com , www.cordys.com

