CMMN Implementation in Executable Model of Business Process at Order-Based Manufacturing Enterprise

Vadim Kuzin and Galina Kuzina

Mosflowline, Russian Federation, 125599, Moscow, Izhorskaya str., 6
zilbernstein@gmail.com, kuzina@mosflowline.ru

Abstract. Agility - capability of an enterprise to function in the highly competitive and dynamic business environment. To survive and successfully develop companies should have flexible, adaptive business processes and management system that enforces the strategy and ensures achievement of target (commercial) goals. Case management is a paradigm for supporting flexible and knowledge intensive business processes. It is strongly based on data as the typical product of these processes. This paper presents implementation of this paradigm at the manufacturing enterprise based upon principles of CMMN emerging standard, declarative approach to business process modeling and the systems theory. The implementation uses first order logic (Prolog) and elements of lambda calculus. It has been in operation for more than 3 years.

1 Introduction

Modern business environment is highly competitive and dynamic. To survive and successfully develop companies should be not only agility – focused and have flexible, adaptive business processes but also the business management system that enforces implementation of the business strategy and ensures achievement of target (commercial) goals. The operation of business management system includes the online monitoring current state of the business, comparing it to the planned one, analyzing causes of deviation and generating the corrective responses by updating the plan, launching, pausing/stopping the relevant business processes. Therefore for businessmen to have the tool ensuring the achievement of business goals by managing business processes these processes should be closely linked to the state of the business and its management system.

This paper presents the experience obtained in development and operation of flexible and adaptive business process management system at ZAO "Mosflowline" (www.mosflowline.ru) - the manufacturing enterprise with order-based engineering and production processes. It produces polyurethane-insulated components for oil and district heating pipelines. The enterprise operates in highly competitive environment imposing strict requirements on prices, delivery times, payment adjourning periods. The business processes in the enterprise of this type have the following distinctive features: (a) a customer can make changes to the list of the ordered products, their

Y.T. Demey and H. Panetto (Eds.): OTM 2013 Workshops, LNCS 8186, pp. 112–123, 2013.

configurations, scope and time of delivery "on the run" while the order is being executed; (b) a contract manager can pick different ways to execute the contract: production, resale, outsourcing of certain operations depending on the book of orders, availability of production capacity and other factors determining the state of the company; (c) large number of customers resulting in large number of simultaneously executed orders; (d) the size of nomenclature is more than several thousand items due to various combinations of product parameters affecting manufacturing processes. Features (a) and (b) indicate the high level of agility of these processes: they can be implemented in a number of ways depending on specific case and personal preferences of the client and the contract manager while each of the ways is limited to a set of template implementations. The order size can vary from several to several hundred different items. The number of orders in work varies from several dozens to several hundreds. Product data is also continuously changed (both in terms of structure and content) due to requests from the customers and R&D efforts aimed at improving quality and characteristics of the product line.

The main goal of the project was development and introduction of the adaptive and flexible business process management system that should ensure achievement of the enterprise commercial goals. The project required resolving two large problems. The first one – development of the agile enterprise model that in addition to business process definitions should include the definitions of:

- goals represented as integrated metrics identifying the planned state of the business at different levels of management;
- management system ensuring the monitoring of business process execution and achievement of the identified goals;
- resources (material, human, information) used in business processes;
- organization structure identifying subordination of business process participant and responsibility for goal achievement.

The enterprise model should allow for computer-aided verification at the stage of initial development as well as at the stage of modification due to process improvement or adaptation.

The second problem – is development of the system supporting the execution of flexible and adaptive business processes. The system should:

- ensure integration of definitions of goals, resources, organization structure, planned and actual business states into the single information model of the business;
- provide selection of the specific business process implementation depending on the current state of the business and user preferences;
- enable modification of business processes in the course of their execution;
- enable direct execution of declarative definitions of the business processes without the need to translate them into software code.

Solution to the first problem rests upon the mathematical systems theory that deals with the dynamical systems/processes in the physical world [1]. The advanced approach to development of the agile enterprise model and non-workflow theory of business processes based upon the systems theory is described in [2, 3, 4]. According to this approach an enterprise is represented as three-layered model consisting of assets, sensors and business process instances (BPI). Enterprise is defined as multidimensional state space. A BPI is then defined as a trajectory in the space-time, and business process model - as a set of formal rules describing all "valid" trajectories. Business processes support system should assists BPI participants to follow one of the valid trajectories and provide them with a common interactive "map" where they can together try to find their path towards the BPI's goal despite the "roadblocks" appearing where they are not expected.

Development of the system supporting the execution of flexible and adaptive business processes where the information context has the dominating influence (the second problem) is based upon data-centric approach to BPM. It draws on several influences [5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 20].

The work has brought the following results:

- the model of a case has been developed based upon SADT [25] declarative modeling of the semantics of business processes as hierarchical network of business-activities transforming input resources into output resources using mechanisms and guided by control signals;
- the process information model has been developed as a semantic network complemented with the first order logic constructs (Prolog) and the elements of lambda-calculus (Lisp) to define calculated (derived) properties of classes (e. g. aggregated indicators of the business's condition);
- the model of a case has been complemented with process information model (its information context) containing the knowledge of the application domain represented using the basic elements of semantic networks (triplets, relationships, classes), the combined model enables automatic verification of model's consistency (since a model is represented as composition of logical statements);
- the case model interpreter has been developed for direct execution of cases implementing the basic concepts of CMMN standard: (a) rich planning during execution of case instance; (b) declarative/rules – based behavior specifications; (c) tasks, (hierarchical) stages and milestones;
- the information model interpreter has been developed in the form of self-adapting user interface to input and display the data needed to perform specific business activity.

The section 2 contains a real life example of modeling the part of the business-system that implements the process of preparing the price proposals for engineering, production and shipment of oil pipeline components with evaluation of economic efficiency considering the current financial and operational condition of the company. The section 3 contains the fragment of the information model of the process and short description of the business-system information metamodel. The section 4 briefly describes the basic elements of business-system behavior model. The section 6 describes related work. The section 7 contains brief conclusions.

2 Example of Developing Business Process Model

The paper considers an enterprise as a complex adaptive business system that achieves the target goals through execution of business processes. It can be described as the composition:

$$BS = <S, \{G\}, \{F\}, SF, \{R\}>$$

where: S – state space of the system, {G} – set of goals organized in a tree, {A} – set of the activities implemented by the business processes to achieve the goals (e. g. order-based manufacturing of products, making changes to the order, etc.), SF – structure of the system that ensures implementation of the activities and achievement of the goals, {R} – set of resources utilized and consumed in the business processes (materials, equipment, staff, information). The state of a business system is defined through the states of its resources, departments and finished, ongoing and scheduled business processes. The papers considers a business process as transition between states of the business system resulting from coordinated interaction of business system departments that produce and exchange material and information resources.

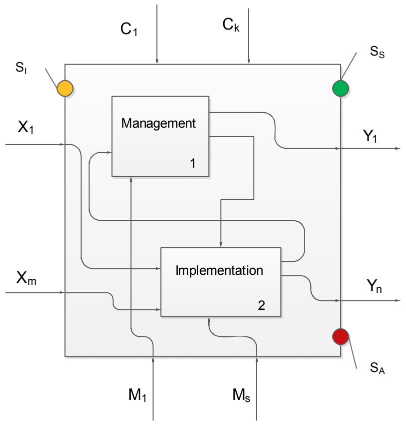

Fig. 1. Business Activity metamodel

The processes causing transitions of the business system and its components between states are defined using the concept of business- activity. The definition of a business - activity (BA) includes (fig. 1): input resources {X1 … Xm} (materials, parts, documents), output resources {Y1 … Yn} (products, documents), control signals {C1 … Ck} (planned indexes), supporting resources {M1 … Ms} (staff, equipment, instrument), a start state (SI) (a guard counterpart), a successful completion state (SS) (a milestone counterpart), an abort state (SA). The BA execution is represented as two interacting processes: an implementation process and a control process. The BA's implementation process ensures transformation of the BA's input resources into output resources using the supporting resources. The BA's control process ensures that parameters of implementation process tend to the control signals and is a combination of standard functions: planning, accounting, analysis, monitoring/adaptation implemented for the specific BA.

Business system operation can be represented as hierarchical framework of BAs interacting with each other. The top level BAs implement the company's strategy, the terminal level BAs correspond to activities performed by ultimate performers. Consider the business function of preparation of orders to deliver oil pipeline components from some intermediate level of BAs hierarchy (fig. 2). The BA ensures the transition of the business system from its current state to the state where it has achieved the total value of open orders specified in the sales plan. The BA is implemented by a cross-functional business process involving several departments: commercial (CD), product engineering (ED), procurement (PD) and process engineering (PED). The implementation business process (iBP) starts with search for order resulting in a request for price proposal from customer. The request contains technical description of the products needed by the customer. The ED employee (design engineer) uses the request to create the order specification containing a list of items identifying type and parameters of the products to be delivered. If while creating the order specification the ED employee fails to select the appropriate product from the company's catalog of products then he starts the BA of product development based upon the customer's requirements specified in the received request. The business process implementing the product development BA starts with creating a request for new product development. The request is used in development of product drawings and process maps. The oil pipeline components have complicated configuration. Consequently the description of an oil pipeline component includes parametric geometrical model, the required amount and list of materials required to fabricate the product. In developing the process map the process engineer selects materials and parts from the company's catalogue of materials. If the engineer fails to find some material in the catalogue he queries the procurement department if it is possible to deliver the material with specified parameters within required time period (specified by customer in the request for proposal). The PD employee searches for vendors that could deliver the material/part requesting the price and delivery times. If the PD employee succeeds then the material is input into the company's catalogue of materials. Otherwise the process engineer should change the process map and the requirements to materials. When the process map is finished the notification is generated and sent to the design engineer. When the order specification is finished (and notifications of completing the process maps for all non-standard products are received) the ED employee sends the order specification to CD for commercial proposal preparation. Upon receiving the order specification the CD employee is to choose between various scenarios of contract execution. For example if the business system lacks free production facilities in the period when certain specification item is to be delivered then the CD employee should indicate that the item is to be resold rather than produced. Also if the business system has got products available on stock with parameters close to the parameters of certain specification items then the CD employee starts the BA of agreeing new parameters of the items with the customer. Other parameters of order execution include: payment adjourning period, usage of customer's materials, outsourcing of certain production operations. Therefore the CD employee is to choose the scenario of the order execution considering the current financial condition of the company, forecast of revenue, liabilities, assets, production facilities available in the period when the order is to be executed. After the scenario has been chosen the CD employee specifies the profitability of each specification item, calculates its sell price and the overall order budget. Then the budget is sent to the head of CD department for approval. The head of CD department can return the price proposal to the CD employee for modification if its parameters do not satisfy the control signals of the proposal preparation BA. While the price proposal is being prepared the customer usually makes a number of changes to the contents of order specification as well as technical parameters of its items. Any change to the order specification results in the need to agree the change with the departments involved in the order execution: PD, warehouse and the shop in order to ensure that it is possible to fabricate and deliver the products according to the order specification within the required time period.

The formalized model of the business system has been developed on the basis of IDEF0 notation [24] that has been substantially extended to capture the semantics of the notation constructs. The notation has been selected because: (a) it ensures

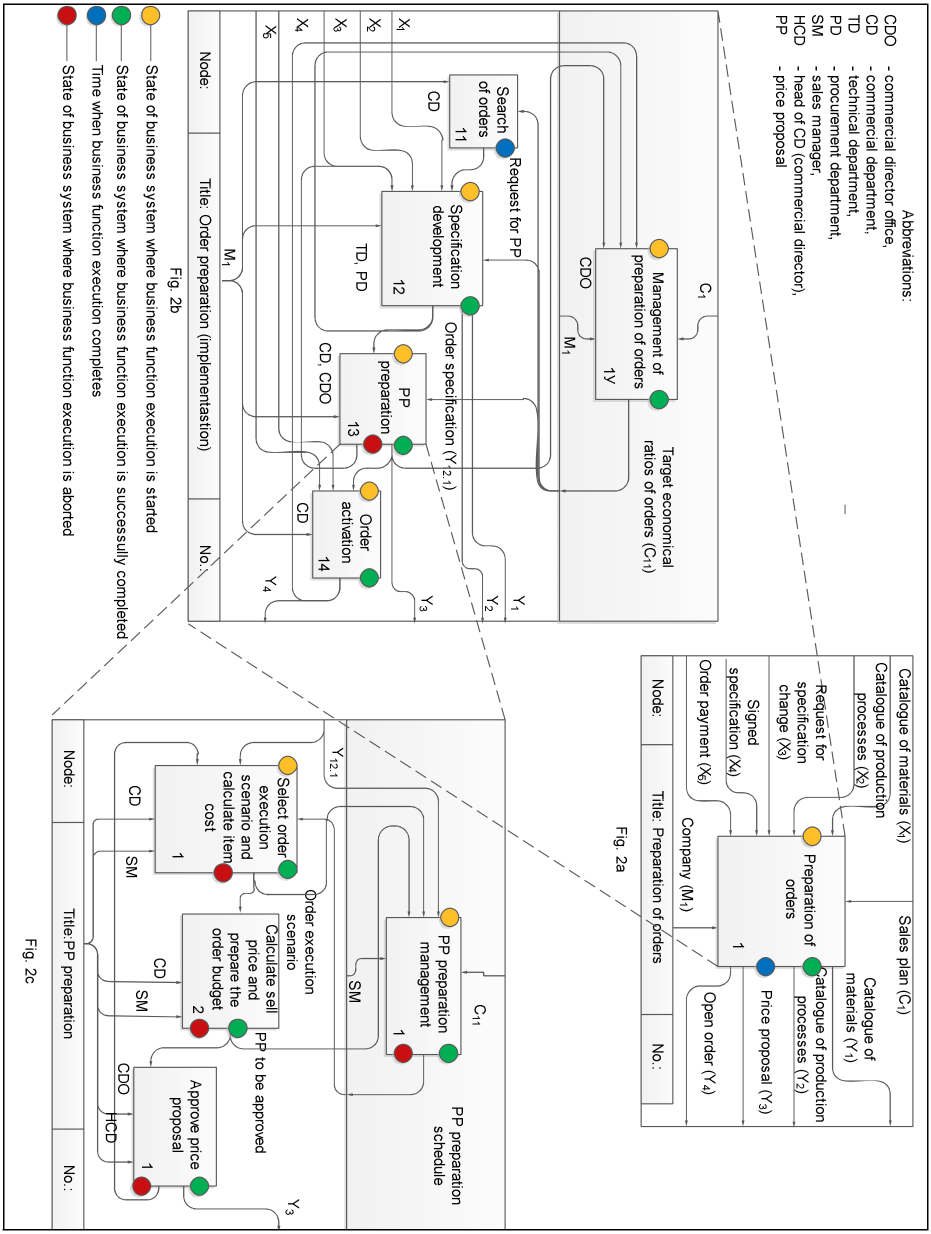

Fig. 2. The fragment of "Preparation of orders to deliver oil pipeline components" case model

top-down consistent design enabling "completeness" of the model; (b) provides most of the constructs to define business system behavior and organization; (c) can be represented using semantic networks; (d) SADT methodology [25] treats processes as transitions between states. It is necessary to note that IDEF0 arrows stand for not sequence of execution but for cause-and-effect relationships between business activities (boxes). Using extended IDEF0 notation the above example can be formalized in the following way (Fig.2). "Preparation of orders" BA (Fig. 2a) ensures transition of the business system from S0 state where it has received the sales plan C1 to the one of the states: S1 where C1 has been fulfilled (the total of open orders in the period T exceeds or equal C1), S2 where C1 has not been fulfilled but the period has finished.

The Fig. 2b presents hierarchical decomposition of "Preparation or order" BA. It includes the following BAs: "Search of orders", "Specification development", "Price proposal (PP) preparation" and "Order activation". Each "child" BA is provided with necessary input, supporting resources and control signals. Output of each BA's serves either as output of the parent BA or the input of one of its "siblings". Also pre- and post-conditions of execution for each BA are identified. "Management of preparation of orders" BA generates the control signals correcting the target economical ratios of the orders such as: profitability, adjourning period, shipment period, etc. The signals are generated upon receiving the feedback from "implementation" BAs that includes: the number of the prepared specifications, the total revenue of prepared price proposals, the total revenue of activated orders. The "Price proposal preparation" BA is decomposed into: "Select order execution scenario and calculate item cost" BA, "Calculate sell price and prepare the order budget" BA, "Approve price proposal" BA, "Management of price proposal preparation" BA. The most interesting BA in the decomposition is "Select order execution scenario…". Depending on the condition of the business system the execution of the BA can involve taking various decisions on whether produce or resell certain items of the order specification, use customer's materials or own materials, outsource certain production operations or perform it at own production facilities, etc. So the list of decisions to be taken and their sequence is not known in advance and depend on the current situation in the business system (book of orders, available production facilities, etc.). When the BA starts and the current condition is evaluated then it becomes clear what decisions should be made. When all necessary decisions have been made the final cost calculation is performed.

3 Business System Information Metamodel

To develop the business system information model first order logic (Prolog) has been used. The following features of the business system have determined the choice of Prolog: (a) need to use the single method to describe complex systems (business systems); (b) need to make automatic formal verification of consistency of the resulting models; (c) easy modification of the model at runtime; (d) easy integration with relational databases of corporate information systems. To use Prolog the entire contextual information about the business process should be represented as a semantic network describing the knowledge about the application domain with use of triplets (object-attribute-value) as well as various relationships and classes. The information metamodel graph can be viewed at: http://en.acm-systems.ru/business-system-information-metamodel. The basic concepts of the information metamodel are: term (concept), entity, relationship, class, lambda-expression, predicate. The term is the abstract supertype of entity and class. Class – is the counterpart of a mathematical set. Class consists of the elements of certain type that is specified by ItemType relationship. Class can be defined by enumeration, operation over sets (union, intersection, subtraction) or with use of predicate. Each class definition method is encapsulated in a certain subclass (Enumeration, Union, Intersection, Subtraction, DefinedByPredicate). Class members are identified by SetInstances relationship. Properties of a class are defined by relationships. Relationship is a subset of Cartesian product of classes (RElement0 x RElement1 x … x RElementn, где 0, 1, .. n – domainindex attribute values). The metamodel includes directed relationships only. Depending on the "direction" of the underlying relationship class properties can be "direct" (where class occupies first position in the relevant Cartesian product (domainIndex attribute = 0)) and "reverse" (in all other cases when domainIndex attribute > 0). The above example demonstrated that business system state can be represented using integrated indexes like the total workload of the equipment, expected profit margin of the book of orders, total revenue of order, etc.

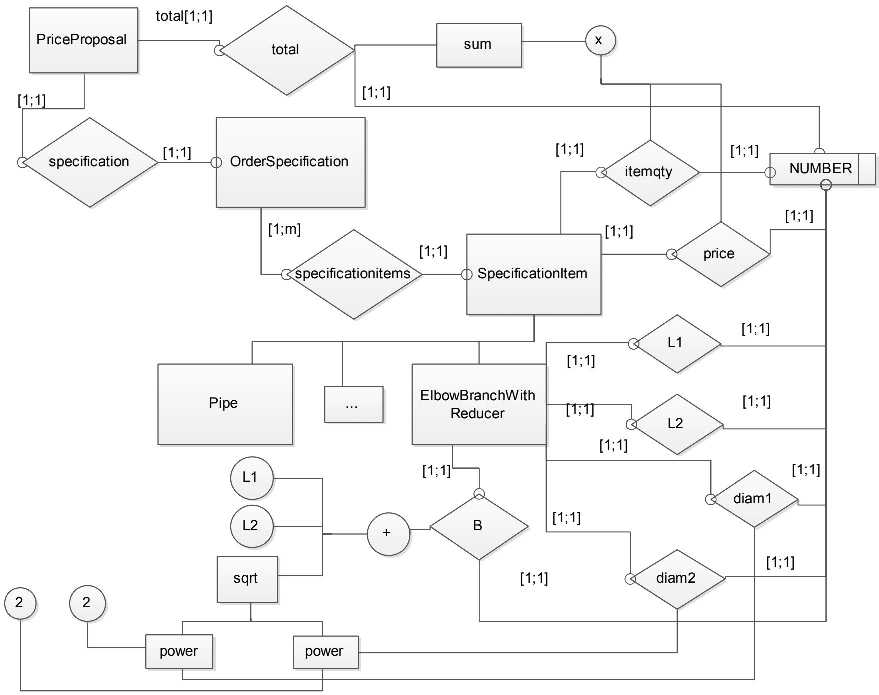

Fig. 3. Fragment of "Preparation of order" information model

To enable definition of such indexes the business system metamodel includes the construct of "derived" property using lambda expressions. Lambda expressions use standard functions like sum, sqrt, power, etc and properties of classes represented as function calls. Consider the fragment of the information model of "Preparation of orders" business process presented in Fig. 3. The model includes the classes: PriceProposal, OrderSpecification, SpecificationItem, Pipe, ElbowBranchWithReducer (subclasses of SpecificationItem with the relationships: specification, specificationitems, total, itemqty, price, L1, L2, diam1, diam2, B. The PriceProposal class has the derived property "total" that can be represented as lambda-expression: (sum (* price(specificationitems(specification(P))) itemqty(specificationitems(specification(P)))), where P is the instance of PriceProposal. In the expression specification, specificationitems, itemqty, price properties are represented as "functions": specification(P) – gives instance of OrderSpecification for the instance P of PriceProposal, specificationitems(specification(P)) gives instances of SpecificationItem. Another example of derived property is geometrical parameter B of ElbowBranchWithReducer class. The lambda-expression of the parameter: (+ (L1(I) L2(I) sqrt (+ power(diam1(I) 2) power(diam2(I) 2))), where I – the instance of SpecificationItem.

4 Executable Model of Business System Represented in Prolog

To execute business processes implementing business activities case model interpreter (CMI) and information model interpreter (IMI) have been built. CMI performs the following main tasks: (a) initial planning of business system track in its state space; (b) creating, starting, pausing, resuming, completing and aborting execution of business activities based upon dynamic evaluation of business system state as well as definitions of the business activities; (c) registration of actual track of the business system in its state space. The IMI performs the following main tasks: (a) auto-generation of data input forms; (b) auto-generation of reports; (c) information model-based creation of instances. CMI operation is based upon the business system operation metamodel developed on the basis of the business system information metamodel. The graphical representation of the business system operation metamodel can be viewed at: http://en.acm-systems.ru/business-system-operation-metamodel. The description of CMI and IMI operation is available at: http://en.acm-systems.ru/ description-of-acm-and-im-interpreters/.

5 The Related Work

In the introduction we mentioned the approach proposed in [2, 3, 4]. It is based upon system-related view of business process and consider business process instance (BPI) as a point moving in multi-dimensional state – space. A BPI is then defined as trajectory in space – time, business process model – as a set of formal rules describing all valid trajectories, goal - as a point or surface in the state space, enterprise - as a complex multilevel adaptive system. We have used the same concepts and our approach to agile enterprise modeling (first problem) is very similar to this one. Differences of our model are driven by the practical goal: to develop and introduce the working system at the real-life enterprise which operation is aimed at achieving commercial goals. The differences include: a) introduction of organization structure as the backbone of goal-based management system; b) classification of resources (materials, equipment, staff, information) as predicate-defined sets of objects used in state space definitions; c) application of lambda-calculus and first order logic in calculation of complex information properties needed to evaluate the current business state. Interestingly that the need in flexible "navigator" that can suggest a path from any given position to the goal indicated in [2] has evidently arisen in the course of the operation of our system. Currently we have already developed such a navigator and are testing it.

In our development of the agile and flexible business process support system we have used the basic concepts of CMMN draft standard [19, 20]: guard - pre-condition, milestone - post-condition, stage – business activity with decomposition. As in GSM approach [15, 17] our agile and flexible business process support system uses precise mathematical definition for operation semantics. This aspect enables verification of the executable models. In contrast to GSM approach our model is Prolog-based and is executed directly by the interpreter we have developed. Another difference of our implementation from GSM approach is using the semantic network to represent the enterprise information model. The semantic network allows a) describe complex information context b) provide easy modification of the information model c) definition of calculated properties using first order logic and lambda calculus.

ConDec [21] is fully declarative business process language in which the possible sequencings of activities are governed entirely by constraints expressed in temporal logic. The main problem of using this approach at the real-life manufacturing enterprise is integration of business process definitions and constraints in its information space so that the definitions are described using the enterprise information objects.

Existing commercial products [16, 22, 23] provide support for flexible business processes but their data representation is document-oriented and not suitable for the manufacturing enterprise with parametric product data models, complex integrated indicators of the business state.

6 Conclusions

Designing the flexible business process management systems needs the theory of business processes to be developed using the system-based approach to the modeling of the business processes. The business processes at a manufacturing company are executed in complicated information context. Semantic networks have been chosen to build the formal representation of the context that in business process execution models got naturally mapped on first order logic (Prolog constructs). The declarative definition of the business process models using first order logic has enabled automatic verification of the consistency of the definition. To execute business processes we have built the interpreters of business process models and information models. In introducing the system we have discovered that business process participants are interested in: a) simulation of different cases/situations; b) flexible "navigator" that can suggest the most appropriate way to handle particular case by choosing specific implementation of the corresponding business process; c) more advanced tools to design the case information models; d) further development of reporting tools.

The implementation of basic concepts of CMMN standard in practice has demonstrated that they cover all of the major aspects of designing the adaptive business process management systems: declarative approach to definition of business process semantics; logic-based evaluation of the activities to be run depending on the state of the business; definitions of business process instances can be modified during execution.

We have started to introduce the system more than three years ago by gradually replacing the old ERP system. Currently the system has about 100 online users (each of them can have several dozens of processes running). Scalability of the system is limited to the hardware capacity only (now it runs on 4 x Intel Xeons with 16 GB of RAM). Users have accepted it well in owing to its functional capabilities: a) timely notification about encountered problems (arrears of payments, arrears of receiving originals of the documents, shortage of materials on stock, availability of finished goods to be shipped, etc) and advising on how to handle them by launching specific business processes; b) fast adaptation of the system to the new processes, new conditions of process execution, new information objects and new properties of existing information objects.

The system has been integrated with the accounting system, PDM system, production scheduling system and other applications using ESB (ApacheMQ). The system generates and consumes information objects in XML format that are transferred through the bus. The structure of the objects is native to the applications they go to or they come from. The system generates objects to be exchanged as "reports" obeying the format of the consumer system. When the system receives an object from the external application it parses the object into a fragment of the semantic network in accordance with the enterprise information model.

Acknowledgments. The authors would like to thank the anonymous reviewers for their comments on initial draft of this work that helped to make this paper more readable.

References

1. Kalman, R.E., Falb, P.L., Arbib, M.A.: Topics in Mathematical System Theory. McGraw-Hill (1969)
2. Bider, I.: Towards a Non-workflow Theory of Business Processes. In: Proc. ACM 2012 Workshop (2012)
3. Bider, I., Bellinger, G., Perjons, E.: Modeling an Agile Enterprise: Reconciling Systems and Process Thinking. In: Johannesson, P., Krogstie, J., Opdahl, A.L. (eds.) PoEM 2011. LNBIP, vol. 92, pp. 238–252. Springer, Heidelberg (2011)
4. Bider, I., Perjons, E., Elias, M.: Untangling the Dynamic Structure of an Enterprise by Applying a Fractal Approach to Business Processes. In: Sandkuhl, K., Seigerroth, U., Stirna, J. (eds.) PoEM 2012. LNBIP, vol. 134, pp. 61–75. Springer, Heidelberg (2012)
5. Nigam, A., Caswell, N.S.: Business artifacts: An approach to operational specification. IBM Systems Journal 42(3), 428–445 (2003)
6. Bhattacharya, K., Caswell, N.S., Kumaran, S., Nigam, A., Wu, F.Y.: Artifact-centered operational modeling: Lessons from customer engagements. IBM Systems Journal 46(4), 703–721 (2007)
7. Bhattacharya, K., Gerede, C.E., Hull, R., Liu, R., Su, J.: Towards formal analysis of artifact-centric business process models. In: Alonso, G., Dadam, P., Rosemann, M. (eds.) BPM 2007. LNCS, vol. 4714, pp. 288–304. Springer, Heidelberg (2007)

8. Gerede, C.E., Bhattacharya, K., Su, J.: Static analysis of business artifact-centric operational models. In: IEEE International Conference on Service-Oriented Computing and Applications (2007)
9. Gerede, C.E., Su, J.: Specification and verification of artifact behaviors in business process models. In: Krämer, B.J., Lin, K.-J., Narasimhan, P. (eds.) ICSOC 2007. LNCS, vol. 4749, pp. 181–192. Springer, Heidelberg (2007)
10. Liu, R., Bhattacharya, K., Wu, F.Y.: Modeling business contexture and behavior using business artifacts. In: Krogstie, J., Opdahl, A.L., Sindre, G. (eds.) CAiSE 2007. LNCS, vol. 4495, pp. 324–339. Springer, Heidelberg (2007)
11. Kumaran, S., Liu, R., Wu, F.Y.: On the duality of information-centric and activity-centric models of business processes. In: Bellahsène, Z., Léonard, M. (eds.) CAiSE 2008. LNCS, vol. 5074, pp. 32–47. Springer, Heidelberg (2008)
12. Küster, J.M., Ryndina, K., Gall, H.C.: Generation of BPM for object lifecycle compliance. In: Alonso, G., Dadam, P., Rosemann, M. (eds.) BPM 2007. LNCS, vol. 4714, pp. 165–181. Springer, Heidelberg (2007)
13. Damaggio, E., Deutsch, A., Vianu, V.: Artifact systems with data dependencies and arithmetic. In: ICDT 2011, pp. 66–77 (2011)
14. Marin, M., Hull, R., Vaculín, R.: Data Centric BPM and the Emerging Case Management Standard: A Short Survey. In: La Rosa, M., Soffer, P. (eds.) BPM Workshops 2012. LNBIP, vol. 132, pp. 24–30. Springer, Heidelberg (2013)
15. Damaggio, E., Hull, R., Vaculín, R.: On the equivalence of incremental and fixpoint semantics for business artifacts with guard-stage-milestone lifecycles. In: Rinderle-Ma, S., Toumani, F., Wolf, K. (eds.) BPM 2011. LNCS, vol. 6896, pp. 396–412. Springer, Heidelberg (2011)
16. van der Aalst, W.M.P., Weske, M.: Case Handling: a new paradigm for business process support. Data & Knowledge Engineering 53(2), 129–162 (2005)
17. Hull, R., Damaggio, E., De Masellis, R., Fournier, F., Gupta, M., Heath, T., Hobson, S., Linehan, M., Maradugu, S., Nigam, A., Sukaviriya, P., Vaculín, R.: Business Artifacts with Guard-Stage-Milestone Lifecycles: Managing Artifact Interactions with Conditions and Events. In: Proc. ACM Intl. Conf. Distributed Event-Based Systems (DEBS) (2011)
18. de Man, H.: Case Management: A Review of Modeling Approaches. BPTrends (January 2009)
19. Object Management Group. Case Management Process Modeling (CMPM) Request for Proposal. Document bmi/2009-09-23 (October 2009)
20. BizAgi, Cordys, IBM, Oracle, SAP AG, Singularity (OMG Submitters), Agile Enterprise Design, Stiftelsen SINTEF, TIBCO, Trisotech (Co-Authors). Proposal for: Case Management Modeling and Notation (CMMN) Specification 1.0. (OMG Document bmi/2012-07-10). In response to: Case Management Process Modeling (CMPM) RFP (OMG Document bmi/2009-09-04). Object Management Group (July 2012)
21. van der Aalst, W.M.P., Pesic, M.: Decserflow: Towards a truly declarative service flow language. In: Bravetti, M., Núñez, M., Zavattaro, G. (eds.) WS-FM 2006. LNCS, vol. 4184, pp. 1–23. Springer, Heidelberg (2006)
22. Zhu, W., Becker, B., Boudreaux, J., Braman, S., Do, T., Gomez, D., Marin, M., Vaughan, A.: Advanced Case Management with IBM Case Manager. IBM Redbooks, New York (2011)
23. de Man, H.: Case Management: Cordys Approach. BPTrends (February 2009)
24. ICAM Architecture Part II-Volume IV - Function Modeling Manual (IDEF0)
25. AFWAL-TR-81-4023, Materials Laboratory, Air Force Wright Aeronautical Laboratories, Air Force Systems Command, Wright-Patterson Air Force Base, Ohio 45433 (June 1981)