Automatic Business Process Model Assembly on the Basis of Subject-Oriented Semantic Process Mark-up

Alexander Gromoff[1], Nikolay Kazantsev[1], Pavel Shapkin[2] and Leonid Shumsky[2]
[1]*National Research University Higher School of Economics, BPM Department, Moscow, Russia*
[2]*National Research Nuclear University MEPhI (Moscow Engineering Physics Institute), Cybernetics and Information Security Department, Moscow, Russia*

Keywords: Enterprise Architecture, Semantic Mark-up, Real-Time Business Architecture, Formal Process Specification.

Abstract: Business processes of large companies often lack the flexibility of their smaller competitors. This fact results in slower reaction to the changes in business environment which usually require to launch complicated IT projects. Nevertheless many tasks faced in these situations by IT departments have standard solutions which could be expressed in form of semi-formalized best practices and recommendations. Authors propose a business process model enriched by the semantic mark-up that enables to represent these recommendations in form of fully formalized high-level process descriptions. Using the proposed technical approaches concrete business process models can be generated from these descriptions. Authors discuss formal methods of searching the relevant high-level descriptions as well as rating and evaluation of the generated concrete models based on different criteria.

1 INTRODUCTION

Nowadays there is recommended process configuration that can execute almost every goal of a company target. For this process in turn there are several possible realisations by means of different programs. The key enterprise architecture metricise its ability to connect new processes and IT-means for their support. The moving towards this direction is done from several aspects. Firstly, the approaches to improvement of the process goal setting and strategic planning are developed. The goal, which is formulated correctly should not only satisfy the future owner by its form but also be a process-setting model of actions. For ensuring such level of goal decomposition different techniques of semantic modelling and semantic analysis of received constructions are used. Secondly, there is a tendency of putting standard process models and IT means for their realization into cloud and this tendency is currently empowered. In general this is the development of process modelling and process realisation *automation*. The usage of cloud technologies allows composing difficult processes from standard components and reference or etalon solutions for developing processes realisation.

Both described approaches seriously increase the agility of enterprise architecture and approximate it to the real-time architecture, when any change of strategic goal immediately is reflected in change of settings or/and content of software used. The aim of this work is to make next step toward conjunction of both described approaches.

The case is that, when semantic modelling of goal formalisation is widely used then in the sphere of process modelling the situation is different. Suggested elementary canonical processes are modelled sufficiently informal and heterogeneously and the semantic modelling is used at best only for description of inputs, outputs of the processes and for notion of its pre and post conditions. This brings us to the fact that from one point of view the task of searching the solution becomes difficult and composite process construction considerably lost in visualisation and in understanding, as a result – in acceptance. From another point of view, the task of connection of the constructed process model with technological wares also becomes difficult and uneven. The outcome of these disadvantages is that there is a big lag between the model and strategic goal changes and this changes implementation. This lag appears as a result of the need to update the pro-

Gromoff A., Kazantsev N., Shapkin P. and Shumsky L..
Automatic Business Process Model Assembly on the Basis of Subject-Oriented Semantic Process Mark-up.
DOI: 10.5220/0005120001580164
In *Proceedings of the 11th International Conference on e-Business* (ICE-B-2014), pages 158-164
ISBN: 978-989-758-043-7

cess model (or develop new model) and then again «link» the technological wares to this new model.

Currently situation in relatively big enterprises is close to dramatic misunderstanding what to do and where to go. Actually in 10 observed entities with a staff from 5K to 40K of personnel the lag of implementation of the required changes in IT or workflows is about 6-9 months. That means when actually it's a turn of change one can guaranty no planed change is required anymore since situation has changed again and new changes are necessary.

Considered approach is in application of process description formal model, which should allow integration of strategic goal's semantic model and process technological implementation model. For description of process formal model it is suggested to use the variant of π-numeration. This model allows describing the technological aspects of realization thus more time will be spend to the algorithm description of connection the model with strategic goal semantic model.

2 THE DESCRIPTION OF REAL-TIME BUSINESS ARCHITECTURE AND COMPREHENSIVE CASE STUDY

This part is devoted to the description of target state of projected system and to the description of the case study where described approach can be implemented.

Breath of an architecture new wave smells as an interdisciplinary combination of the approaches and getting out of concept layer: influenced by IT information flows are growing rapidly eroding the barriers between the departments and even companies. Possibility of generation ad-hoc real-time business processes on the global cloud-based self-generated business service basis, however, is not yet provided.

The feedback loop makes business processes as visible for corrections as locally efficient. The architecture itself takes the responsibility for the global efficiency and strategy goals achievement. Main obstacle for such business development or reengineering on a platform of basic or referential patterns is a 'human factor', which is a key issue in resource usage for such processes.

SOA, it seems, had an ability to overcome threshold of human factor nihilism. Nowadays when "SOA" term is mentioned, professionals interpret it as IT architecture with a fixed set of integrated services. Since "fixed" is a key issue here due to standardization and cost saving strategies, the question of responding to business agility trend is one of the most vital for SOA today. The concept of real-time business architecture is in charge of rapid changes of business requirements (Gromoff et al., 2012). The list of authors suggested an alternative view on business modelling using free services in order to meet variable business demands by means of creating the prototype of app store - is-store in network where services of various vendors are collected and simultaneous feedback from the users that attached their personal experience (rating) is received.

The state-of-the-art for today's technological facilities of the Big Data centres, which are grown faster than mushrooms in rainbow forest, has started unique process of dictating new mental paradigm for traditional business mentalities. NOW we have to understand that not only referential models of business processes, or even executable blocks of business units can be simply bought from Clouds traders, but much more, something what always considered as a main assets of any business — expertise and intellectual capital. This futuristic reality of automated business engineering could be considered as an approach to the newer business vision, which is dictated by modern technological abilities and tendencies.

From this respect it's easy to see that in such business architecture only strategic targeting and monitoring is a responsibility of the true human executive level while the rest is compiled automatically from patterns and basics, best practices, and concepts 'as to be best'. In a final phase of business orchestration intellectual resources are selected from Clouds and after legal formalities are switched into action of the ready-to-go processes.

Moreover, the scheme of functioning of suggested service is the following:

1. The user formulates the goal in the most general form in the natural language.
2. The formal goal statement is generated on the base of domain ontology.
3. As a result of reflection on formalized goal, the set of formalized solutions with weights by different criteria is build.

In the fact, each solution is a business process model. At the same time the weights allow to make the choice of concrete solution on the base of one or another preference: the speed of goal reaching, cost of solution, the quality of result and so on. The search of solutions as well as the weights calculation is done on the base of semantic mark-up, which is included in the process description. In its turn the search of process description is done on the base of reference models, which are created by the experts or imported from external sources.

3 GOAL FORMALISATION

The semantic mark-up is needed in order to "query" the set of executable canonical process models using the formalised goal statement.

The formalised goal conjuncts the structure of entities of setting task and the set of strategies – the definition of process on the level of domain ontology. It is based on the domain conceptual model (Baader et. al. 2007), (Roslovtsev et. al. 2013) and describes the entities executable by the process: the sequence of the steps that should be done under these entities; the description of input data, expected and intermediate results. One of the main semantic characteristics of goal is the description of the required resources and correlation of these resources with the resources of organization. Such, the strategy prescribes that for setting goal execution the employees are needed and suggests the description of human resources of concrete organization.

In the formal strategy description there should be highlighted the following elements:

- Entities and their types;
- Actions;
- Parameters of actions and their types;
- Preconditions;
- Post conditions;
- Effects.

Entities are objects, which interact with one another in the stated task. An object can be e.g. the organisation itself – in case of goal formulation – or different data objects etc. The types of the entities are the categories from the domain ontology.

Actions are different methods of interaction between entities. Each action is characterised by source (subject) and target (object). Also, the action can be connected with some set of parameters, preconditions, post-conditions and the effects.

The set of action parameter values is the information connected with the action besides the source and target. When the source and target of action can be only some entities, the action parameters values can be also some numeric values, text and so on. In many cases the action parameters values cannot be determined prior the process execution, for example, during the description of actions of data transferring the data transferred should be entered into the system by the user. In such cases the types of parameters should be indicated, setting the limitations on the allowed set of values without specifying the values themselves.

For addition the action description can include pre-conditions, post-conditions and effects.

4 EXTENSION OF Π-CALCULUS FOR INCLUDING THE SEMANTIC MARK-UP

This part is devoted to the extension of the grammar and rules of abstract machine of π-calculus. The needed extension should solve two basic tasks. Firstly, the process model with linked semantic mark-up should allow differentiating the processes by their characteristics for comparison and ranging the models. Secondly, the model process execution with semantic mark-up should allow picking out the redox that answers the set semantic characteristics.

At first we will shortly describe the formal model being used. The alphabet of π-calculus consists of the following components:

- The set of names of calculus $\mathcal{N}$, designated by lowercase Latin letters;
- The set of processes, designated by uppercase Latin letters;
- The typed calculus includes atomic types designated by lowercase Greek letters.

π-calculus terms are inductively builds by adding special prefixes to existing processes or by combining existing processes. These operations are described by the following grammar:

$$P ::= \mathbf{0} \mid \bar{x}y.P \mid x(y).P \mid (x)P \mid (P|Q) \mid !P$$

For better expensiveness of model ability we will use poliadic variant of calculus (Milner, 1992), (Turner, 1996), which describe the transfer and receiving the list of names, but not the pressed name.

Let us examine the data designation more detailed. The **0** symbol is used to designate the empty process, or the process that does not execute any activity, The $\bar{x}[y_1, \dots, y_n]$ prefix designates the transfer of names $y_1, \dots, y_n$ by the link *x*, the $x(y_1, \dots, y_n)$ prefix designates the data receiving. The prefix of data receiving bounds the received names during the process-continue (P); the set of free names of the process designates as $FN(P)$, the set of bound names is $BN(P)$, data receiving is in the substitution of all inputs of received name into the received value: if $x(y).P$ – Is the process that receives data and the process $\bar{x}z$ transferes data then the result of their communication will be the process $P[z/y]$. The communication (receive-transfer) is done when the one process transfers and the second process receives data by one the same name. Except for the prefix of data receiving there is an opportunity for bounding the name in the process with the help of creating the local link $(x)P$. This entry bounds the name x in the process P – this name becomes the locally determined in the process: no external process can interact using this name. Thus instruction is used for determination of internal, protected or temporary channel of data exchange.

In the π-calculus there determined one manner of process interaction writing – their parallel execution. The parallel-executed processes can interact by certain rules. The replication of processes means $!P$ the process for which there always can be recieved the new copy. The example of such process can be the data transmition from the system when by the request the system can provide the data set which is connected with the names y_1, y_n by the channel x: - this can be written by the process $!\left(\bar{x}(y_1, \dots, y_n)\right)$. The notion of this process without the replication will mean the one-time transmition, which cannot be repeated in the frames of the process – this is the difference between message-request transmition and the transmition of stored master data.

4.1 Semantic Process Mark-up

Semantic mark-up of the process is the set of parameters of the same ontology where the formalised goal is described. The mark-up shows the characteristics of business processes at each step of its execution. The mark-up helps to define which actions and with which costs are in concrete process variant.

The request to the set of canonical processes is in that for each strategy from formalised goal the process model (or its part) is chosen. This model is closer to strategy and for each chosen model the difference between existing and needed resources of organisation is calculated in accordance with this model.

Several general directions will do the semantic mark-up of process. Firstly, we will consider how in that model the pre and post conditions can be represented. Under precondition we will understand some predicate which validity shows if the process can be executed. Under post conditions we will understand the predicate failure to execute which will be a failure and the process should be terminated. These predicates will be modelled using the terms of λ-calculus (Boudol, 1989) coded (Milner, 1992), (Sangiorgi, 2003) in the process. For coding the terms the operator $[|\_|]$ will be used, which assosiate the term λ calculus with correct process of π-calculus. This operator extends the grammar of π-calculus processes by the following manner, where the Λ is the random term of λ-calculus:

$$P ::= \cdots \mid [|\Lambda|].P$$

There are rules, which code the terms of λ – calculus.

$[\|x\|]a$	$= \bar{a}x$	1
$[\|\lambda x.M\|]a$	$= (f)(\bar{a}f \mid !f(x,b).[\|M\|]b)$	2
$[\|MN\|]a$	$= (b)(c)\left([\|M\|]b \mid b(f).\left([\|N\|]c \mid c(x).\bar{f}(x,a)\right)\right)$	3

The first rule associates the variable with process of transition of its value using the stated channel. The abstraction is coded as a process which creates the local link and immediately transfers it through the stated channel and in parallel triggers the process replication, which after receiving the input data and channel by the link returns the value of the term *M*. Application is coded in a form of sequence – in the beginning the first term is coded, after that the second term id coded and the application of calculated function to the argument is executed returning the result for *a,* point of this term installing.

The preconditions are written in the following manner. If the process P can start its execution only if the predicate Λ_P, dependant from the parameters $x_1, \dots, x_n$, received by the link *l*, is known, it is written as

$$(a,b)(l(x_1, \dots, x_n).[|\Lambda_P|]a \mid a(b).\mathit{if}\ b\ \mathit{then}\ P\ \mathit{else}\ 0)$$

This process starts it execution only if the precondition has been executed. By analogy the post conditions are written. The coded predicate receives data from executed process but calculate them according to the described logic, i.e. can operate be the entities of conceptual model.

For domain entities description that are included in the process model, the following grammar extension will be used. Let us pick out the set of available names of π-calculus $\mathcal{N}$ subset $\mathcal{L} \subseteq \mathcal{N}$ of different by two variables. We will call the set $\mathcal{L} \cup \{\tau\}$ as the set of labels. The label can be either the name of π-calculus or the special flag τ, represents the absence of known label. The difference of labels from the ordinary names of π-calculus is in the relation of labels to connection of variables. The characteristic of the labels is that the label can not be connected and can not be α transformed. Adding the labels the grammar of term construction is changed as the following:

$$P ::= \bar{x}[y_1, \dots, y_n].P, \{x, y_1\ \dots, y_n\} \subseteq \mathcal{N}$$

$$P ::= x(y_1, \dots, y_n).P, x \in \mathcal{N}, \{y_1, \dots, y_n\} \subseteq \mathcal{N} \setminus \mathcal{L}$$

$$P ::= (x)P, x \in \mathcal{N} \setminus \mathcal{L} \qquad P ::= P|Q \qquad P ::= !P$$

We will designate this grammar as Γ_l. If process P is represented in this grammar let us write this as $\Gamma_l \vdash P$. As it is seen from the grammar definition, labels can designate either transmitted data or names-channels using which the data transmition is done. The connection of entities of domain to concrete transmited data brings the sense of calculation of some fixed objects in the domain. This objects can be corresponding structure divisions, regulative documents and other objects of domain that do not depend on concrete execution. The usage of label as the channel of data transition/receiving will be used for process execution assessment. For doing this with each label a lot of entities of domain that represent the using resources, defined in the strategy, will be connected. The execution of reduction step, included such label will be estimated as the usage of this resource, all such costs are saved during the process execution. This makes it possible to estimate the process execution in the context of all resources, defined in the frames of strategy. All calculated costs could be estimated in accordance with certain rules and criteria, defined in the frames of strategy.

For further concrete process description let us define the term of context of execution Δ for term $P \mid \Gamma_l \vdash P$. Under this context we will understand the vocabulary the key of which is label $l \in \mathcal{L}$, and the value – object of conceptual model, connected with this label. The sense of this context is in the connection of abstract names of π-calculus with using objects without the violation of the structure that is defined by calculus and semantics. The term of π-calculus that is having the empty context or does not include the values of each label, we will call the structure process model M_s. The term the context of which contain the values for all labels we will call the conceptual process model M_C.

$$P,\Delta \in M_C \Rightarrow \Gamma_l \vdash P \;\&\; \forall n \in FN(P) \cap \mathcal{L} \rightarrow \; n \in \Delta$$

$$P,\Delta \in M_S \Rightarrow \Gamma_l \vdash P \;\&\; \exists n \in FN(P) \cap \mathcal{L} \rightarrow \; \neg\,(n \in \Delta)$$

The difference between these two types is in the non-redex terms of the structure model, which can be reduced in the conceptual model by the functions execution that are connected with labels from the context of process execution that are coded in the same way as predicates in pre and post conditions.

In addition to writing in the context of execution for label will be defined the term if the name type. In the classical typed π-calculus for each name the type is assigned that is an atomic type or channel type (Pierce and Sangiorgi, 1993), (Barendregt et. al, 1977). We will extend this term for supporting the semantic mark-up – will call the full type of the name the pair $[\alpha, C]$, where the first element is the ordinary type of calculus and the second element is the type, defined by the semantic object, connected with this label. For the names that do not have the connection with the semantic object, the second type equals «null» type of using means of semantic modelling. These changes will paired with using extended type system (Barendregt, 1991) for functions defined by λ-calculus terms.

By its nature the process modelling using the π-calculus does not operate with the definitions of actions (steps of the process) however they can be used in the form of requirements in the strategy and using the request to lots of model can be formulated. In the strategy, the action will be represented as an individual object, which is characterized, by the type of the entity received as the input, the type of the result and, optionally, the resources that are used for this action execution.

For supporting the requests that operate the term of action we will use the following agreement. Action in the term of π-calculus is the term that is conceptual process model reducing to the process, consisting from prefixes of data transition, parallel composition and empty process. The execution of actions starts from receiving the data of label or from executing some function where at least one of the arguments is the label. The example of action will be the process $l_1(x), (a)(\bar{t}[x, a]), [\langle l_1, c\rangle]$, where the following interpretation in frames of the considered domain can be presented: the information system l_1 for new employee *x* creates the unique code *a* and transmits the employee data with its code for further legislation $\bar{t}[x, a]$. This action will be characterised by the type of recieving object (new employee) and the type of returning object (empty type as the process terminates by the local data transmition). For additional data identification there can be identified the action according to the context of process execution after the action reduction. The set of actions that the process contains from is defined by the process reduction and the selection of all appropriate for action definition sub processes.

Summarising the receiving extensions we will get:

1) The semantic mark-up of the process allows to use in the processes pre and post conditions, regulating the process execution depending on the executing objects;
2) The context definition in the conceptual process model allows to estimate the «price/value» of process execution;
3) For the model there can be received the set of executing actions in the terms of canonical model, the received set can be compared with set in the strategy by the coincidence of general characteristics;

4) The extended typification of process allows to define the applicability of the process for the required data.

Complete complex of actions, that are included into semantic mark-up allows to organise the marked up process model in such a manner that to the set of these model requests, compiled on the base of goal, can be directed and to receive the optimal model..

4.2 Case Study Formalization

Let us return to our motivating example of staff recruitment task. On the formal level the specified goal there is some concept "staff recruitment" action with input parameter "profile" and the result of it is the legalisation of new employee on the certain position.

Suppose that the reference process models base contains the descriptions for the following processes of staff recruitment:

1) Internal recruitment (in filial of the company etc.)
2) Staff recruitment through the staff agencies.

In the simplest form these processes can be formalised in the following manner. We will assume that the internal requests for the employees transferred to the hr channel and the requests for agencies are transferred through the hh channel. Moreover, the employee should be legalised on the new position for that the ERP system e is used. If we assign the profile description as p and the data of found employee is x, we receive the following models:

1) $\overline{hr}[p] \mid hr(p).(x)(\bar{e}[x,p])$
2) $\overline{hh}[p] \mid hh(p).(x)(\bar{e}[x,p])$

It should be mentioned that hr, hh and e are the labels. The type of the systems hr and hh correspond to the action that receives as the inputs the descriptions of profiles and returns the description of found employee. The type of the e system corresponds to the action of legalization of employee on the certain position. In such interpretation the types of processes mentioned match conforms to the stated goal formalization and thus will be included into resulted set of solutions.

The conceptual model of described processes contains characteristics of used systems hr, hh and e. The most interesting are those characteristics that define the weight of the solution. For example, let us examine such parameters as time consumption and the level of hired employee. The time costs on the personnel selection inside the company can be lower than in the case of agency involvement, however, the search of employee through the agency potentially allows to find a highest-level employee. Of course, the examined two borderline cases can have different variations different from each other by different weights. These variations can also be included in the reference base.

On the base of data from conceptual model for found processes the weight is calculated by the described parameters which gives the opportunity to user to make argumented decision towards one or another solution.

5 CONCLUSIONS

We have shown how the cloud real-time business architecture can be implemented by means of formal business process specification in π-calculus enriched by semantic mark-up. The semantic mark-up links business process models to the domain conceptual model; it helps to find the processes, which match the specified goal and also to weight different solutions on different semantic criteria.

The considered example describes most simple cases. The incorporation of pre- and post-conditions into the process description allows to automate the construction of complex solutions – solutions, which combine several processes. These solutions correspond to complex processes that are compositions of other (sub-) processes. In these cases pre- and post-conditions could be used to determine the sequence of sub-process execution. Proposed approach can simplify implementation of process in semantic web SOA environment (Martin et. al. 2007).

REFERENCES

Milner, R., 1992. Functions as processes // Mathematical structures in computer science. T. 2. № 02. P. 119–141.

Milner, R., 1993. The polyadic π-calculus: a tutorial. : Springer.

Pierce, B., Sangiorgi, D., 1993. Typing and subtyping for mobile processes // Logic in Computer Science. LICS'93., Proceedings of Eighth Annual IEEE Symposium on., P. 376–385.

Sangiorgi, D., Walker, D., 2003. The pi-calculus: a Theory of Mobile Processes. : Cambridge university press.

Turner, D., 1996. The polymorphic pi-calculus: Theory and implementation.

Barendregt, H.P., Dekkers, W., Statman, R., 1977. Typed lambda calculus. P. 618.

Barendregt, H., 1991. Introduction to generalized type systems. J. Funct. Program. 1, 125–154.

Boudol, G.,1989 Towards a lambda-calculus for concurrent and communicating systems // TAPSOFT'89. P. 149–161.

Baader, F., et. al., 2007. The Description Logic Handbook: Theory, Implementation, and Applications. : Cambridge University Press. 622.

Vladimir, R., Leonid, S., Nikolay, K., et al., 2013. A synthetic approach to building a canonical model of subject areas in the integration bus // In the proceedings of ISKO-Maghreb, 2013 3rd International Symposium: IEEE Xplore : Marrakech, Morocco - 2013 - C. 1 - 7; DOI 10.1109/ISKO-Maghreb.2013.6728118

Gromoff, A., Kazantsev, N., Kozhevnikov, D., Ponfilenok, M., Stavenko, Y., 2012. Newer Approach to Create Flexible Business Architecture of Modern En-terprise./ Global Journal of Flexible Systems Management, Vol. 13, No. 4, p. 207-215

Martin, D., Burstein, M., McDermott, D., et. al, 2007. Bringing Semantics to Web Services with OWL-S. World Wide Web 10, 243–277. doi:10.1007/s11280-007-0033-x