

# Ontology-based approach to reuse of business process knowledge

## Donatas Čiukšys

Vilniaus universiteto Matematikos ir informatikos fakulteto lektorius  
Vilnius University, Faculty of Mathematics and Informatics, Lector  
Naugarduko g. 24, LT-03225 Vilnius  
Tel. (+370 5) 2193 075  
El. paštas: donatas.ciuksys@mif.vu.lt

## Albertas Čaplinskas

Matematikos ir informatikos instituto vyriausiasis mokslo darbuotojas, profesorius, habil. dr.  
Institute of Mathematics and Informatics, Principle Researcher, Professor, Dr. Habil.  
A. Goštauto g. 12, LT-01108 Vilnius  
Tel. (+370 5) 2626 107  
El. paštas: alcapl@kti.mii.lt

*The paper proposes an approach to reuse of business process knowledge based on domain engineering, knowledge engineering and ontology-based systems engineering. The main idea of the proposed approach is to separate business process ontology and application domain ontology, and reuse the process ontology in different application domains. A notion of generic business process is introduced and is defined as a family of similar business processes. The two life cycles activity of location of generic business process in application domain is discussed.*

A lot of reuse approaches, methods, technologies and techniques, including recently proposed business process management (BPM) (Smith, Fingar, 2003) and service-oriented architecture (SOA) (Erl, 2005), have been proposed in the fields of information system engineering and software engineering. However reuse of business process knowledge still remains an open problem. The aim of this paper is to discuss a sketch of business process knowledge reuse method that combines different reuse techniques developed in domain engineering (Czarnecki, Eisenecker, 2000), knowledge engineering (Chandrasekaran, 1986), and ontology-based systems engineering (Davies, Studer, Warren, 2006). The main idea of the proposed approach is to separate process ontology and domain ontology, similar as task knowledge and domain knowledge have been separated in knowledge engineering, and reuse the process ontology in different application domains. The paper develops further, refines and improves ideas proposed in (Čiukšys, Čaplinskas, 2005).

## Notion of generic business process

A business process is a partially ordered set of linked activities that create value by transforming an input into a more valuable output. Both input and output can be artefacts and/or information and the transformation can be performed by human actors, machines, or both. A generic business process is an abstraction of a family of similar business processes. All members of this family include a set of common core parts (commonalities) and each particular member includes some additional parts (variabilities), which may differ for different members of the family. A generic business process is described by a kind of feature model (Kang et al., 1990) and by ontology. The feature model can be seen as a view of generic process ontology (Czarnecki, Kim, Kalleberg, 2006). The generic business process does not include any control knowledge about the sequencing of business activities. Control knowledge is added later, reusing process ontology in a particular application domain.

We suggest that generic processes should be expressed in terms of abstract roles (actors, inputs, outputs, resources, capabilities). Generic processes are used to generate particular processes (members of family) that are *located* in chosen application domains. The purpose of generation is to produce the required configuration of the process or, in other words, to decide which variabilities are not relevant to this particular member of family and reject them. After that, roles should be replaced by the entities of application domain in which this member of process family is located. We call this activity *role assignment*.

Thus, the proposed approach provides two main activities: *engineering of process domain* and *process engineering*. The term “*process domain*” is used here to denote a group of particular processes that exhibit similar behaviour and are used to achieve similar goals. Indeed, it is a synonym for the term “generic business process”.

### Engineering of process domain

Engineering of process domain is an activity that is analogous to the domain engineering activity in the two life cycles model (Czarnecki, Eisenecker, 2000). Its purpose is to develop particular process domain. This activity includes three sub-activities referred as analysis, design and implementation of process domain (Fig. 1).

Analysis of process domain provides domain scoping (definition of the boundaries of process family) and discovering commonalities and variabilities among the processes in this domain. The result of analysis is a feature model that describes variabilities and commonalities within business process family. Design of process produces generic business process ontology.

It refines terms defined by feature model and adds to ontology epistemic knowledge. It is important to point out that the resulting ontology is based on the upper business process ontology defining concepts such as *activity*, *input*, *output*, *resource*, *capability*, etc. required to model any generic business process. Upper business process ontology is described in detail in section “Upper process ontology”.

The purpose of implementation of process domain is to pack feature model and process ontology in a package of reusable assets. The process ontology as a reusable asset is represented using Web-Ontology Language (OWL 2004).

### Process engineering

Process engineering is an activity that is analogous to the application engineering activity in the two life cycles model (Czarnecki, Eisenecker, 2000). Its purpose is to generate a particular business process and to locate it in a chosen application domain. Process engineering starts with two parallel activities – analysis of application domain and configuration of generic business process (Fig. 2). The result of analysis is application domain ontology. This ontology is based on the upper application domain ontology that defines concepts required to model application domains, such as *active entity* (e.g. job position, application system, organisational unit, etc.), *provided capabilities* possessed by active entity, *passive entity*, *state*, etc. Process engineering rejects variable parts of the process that are not relevant for chosen application domain and produces final configuration of the located process. A software tool (configurator) is used to support this activity. The main responsibility of

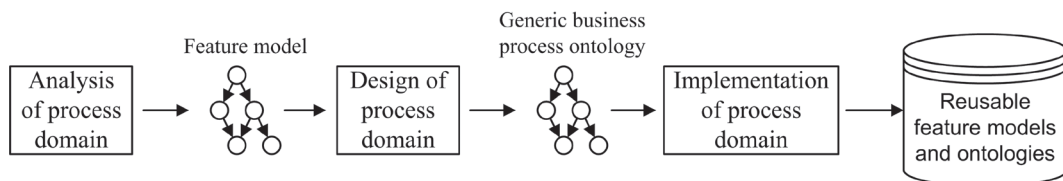


Fig. 1. Engineering of process domain

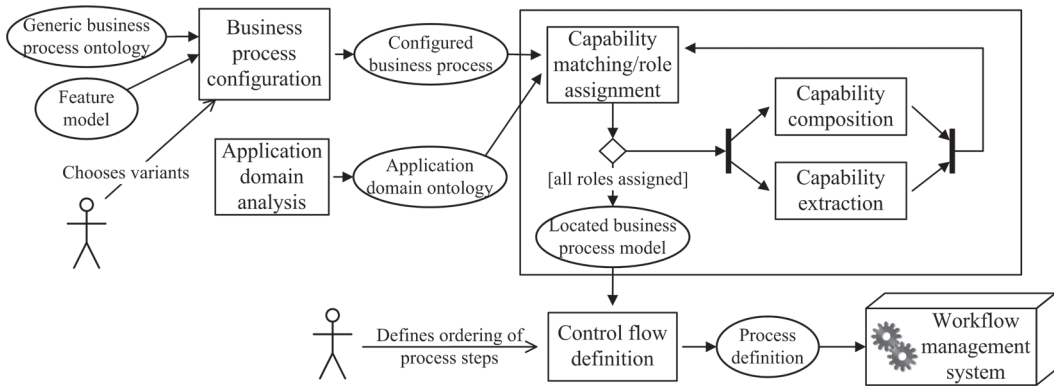


Fig. 2. *Process engineering – business process knowledge reuse*

this tool is to prevent violation of dependencies between variants of feature model.

The configured process and application domain ontology are inputs for the next step, role assignment. Business process is still described in terms of roles (actors executing process’s activities). Requirements for actors that can pretend to play these roles are expressed in form of *required capabilities*. Application domain ontology defines active entities and their *provided capabilities*. So, both roles and entities are characterised in terms of capabilities. Role assignment is done by matching required capabilities to provided capabilities. If an active entity is too coarse-grained for business process role (provided capabilities subsume required capabilities), this entity must be re-engineered and split into several more fine-grained entities (so called *capability extraction*). If an entity is too fine-grained (provided capabilities are subsumed by required capabilities), then usually there will be some number of them and, in this case, these entities have to be composed to one, more coarse-grained, composite entity (so called *capability composition*). If no active entity candidates to play some role, a new entity must be created, for example, by employing a new person or by developing a new application system (not shown in Fig. 2). Iteration “role assignment – capability extraction/composition” is being repeated until all roles are assigned. The result is located business process. However, the control knowledge is still undefined. To add control

knowledge defining execution order of business activities is responsibility of next activity, called control flow definition. As a result executable business process model is produced. It is described in WS-BPEL language (Web Services..., 2007) and can be executed by some workflow management system. This system orchestrates execution of business process activities and at certain times (as defined in business process model) requests services provided by appropriate active entities. Web Services interfaces are used to request services provided by application systems. Human service providers are requested through special interfaces. They are informed about activities that they must perform.

### Upper-level ontology

Application domain ontology captures domain knowledge independently of its use. Both application domain ontology and process ontology should be described by some common system of metaconcepts. It means that some higher-level ontology is required. We call this ontology upper-level ontology (Fig. 3).

This ontology introduces generic concepts that are shared by all lower-level ontologies and reflect underlying theory about the nature of enterprise’s social reality (discourse of interest). Our upper level ontology has been influenced strongly by Uschold’s enterprise ontology (Uschold et al., 1998). The most important dif-

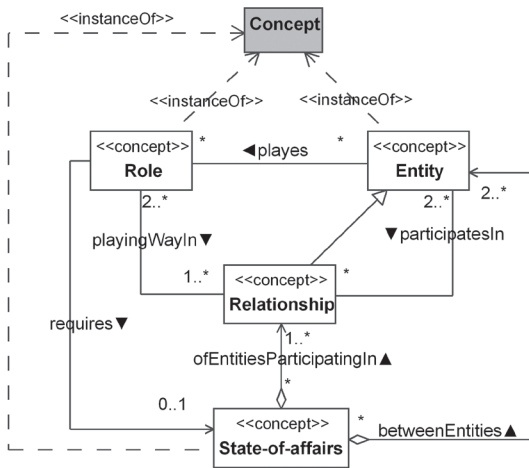


Fig. 3. Upper-level ontology

ference between Uschold’s and our ontologies is that we allow states for roles.

Upper-level ontology (Fig. 3) is two-level ontology. The top level provides the only concept “Concept” that is used to define second level concepts “Entity”, “Relationship”, “Role”, and “State-of-affairs”. These concepts, in turn, are used to define third level concepts in application domain ontology and in process ontology. It means that we follow scheme provided by MOF standard (Meta Object Facility..., 2006).

According to this approach, instances of meta-concepts are concepts themselves.

In this paper upper-level ontology and others, described below, are represented using informal UML-like diagrams. In the process engineering tool these ontologies are represented using formal OWL notation.

### Upper application domain ontology

Let us consider now the upper application domain ontology that serves as a basis to define concepts in particular application domain ontologies (Fig. 4).

All concepts defined by this ontology are instances of concepts defined by upper-level ontology. The ontology refines the notion of entity and classifies all entities into: *active entities* and *passive entities* (Fig. 4). They may overlap. Active entities must provide capabilities required to achieve some business goals or subgoals. Business goal is a state of passive entity that candidates to play output role in some business activity. Such an organisation of concepts is introduced in order to facilitate role assignment. Active entities are further subtyped into *job positions*, *application systems*, and *organisational units* (“OrgUnit” in Fig. 4). All process provided capabilities and may candidate to play roles defined by process

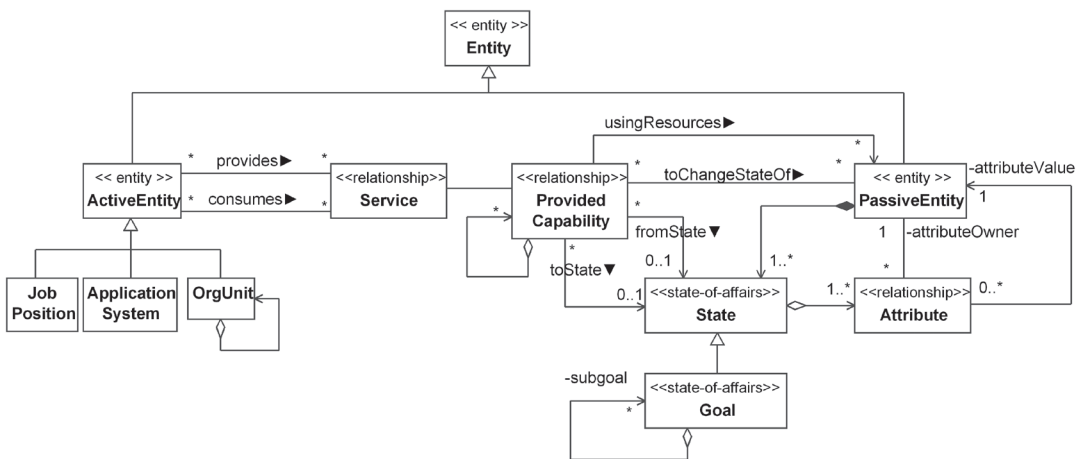


Fig. 4. Upper application domain ontology

ontology. They can change states of passive entities. For example, a job position may provide writing capability and consequently the ability to prepare a document (change its state). Similarly an application system may provide order-processing capability and be able to change the state of order from unprocessed to processed one.

Finally, concept “goal” models business goals. They form a hierarchy. It means that we make assumption that an application domain (as a specific area of business) explicitly states business goals that must be achieved for successful operation of enterprise within this business area.

### Upper process ontology

Up to date exists none standard and commonly accepted business process conceptualisation. A new business process conceptualisation is developed usually when a new business process related project is started or a new tool is developed. In 2003 OMG consortium announced an initiative that aims to standardise the conceptualisation of business processes and to develop so called Business Process Definition Metamodel (BPDM). The draft that candidates to be the final submission is already prepared (Business Process..., 2007). It describes following groups of concepts:

- **course model:** introduces control flow concepts, such as *transition*, *gateway*, *fork*, *join*, etc.;

- **activity model:** introduces structuring concepts, such as *process*, *activity*, *sub-activity*, etc.;
- **interaction protocol model:** introduces interaction and data flow concepts, such as *interaction* and *data* (documents) being exchanged with these interactions;
- **event model:** introduces concepts, describing events that happen during the course of business process, such as *start*, *finish*, *error*, *abort*, etc.

BPDM defines more than 100 concepts. Our upper process ontology is subset of BPDM. It includes only concepts required to describe all kinds of roles provided by business processes (Fig. 5). *Actors*, *inputs*, *outputs* and *resources* are all modelled as roles and domain entities must be assigned to these roles when business process is located in a particular application domain: active entities may candidate to actor roles, passive entities – to input, output and resource roles.

These concepts are not sufficient to represent variabilities provided by process feature model. So, concepts such as *variability*, *variant*, *variation point*, etc. must be included into upper process ontology (Fig. 6).

In the proposed ontology, commonalities are modelled by *activities*, parts that include variabilities – by *generic activities*. A generic activity contains variability and consequently represents a whole family of activities. *Variation point* is a re-

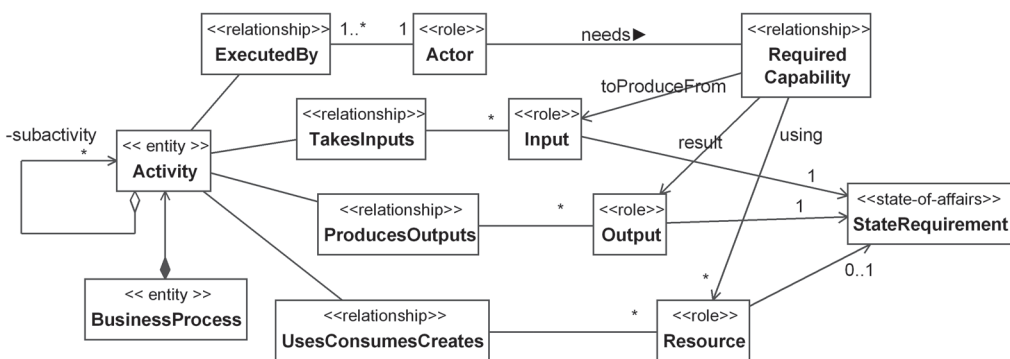


Fig. 5. Upper process ontology (process conceptualisation)

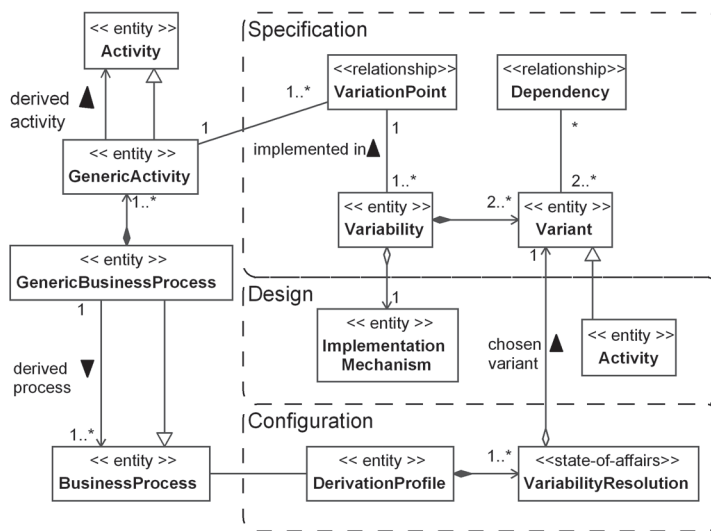


Fig. 6. Upper process ontology (variability conceptualisation)

relationship that associates variability with generic activity. *Variability* is defined as a set of *variants*, one of which must be chosen during business process configuration. *Dependencies* constrain choices of variants, for example, choice of one variant may require choice or removal of other one (e.g. the choice of payment type “Credit card” within e-shop business process may render otherwise optional activity “Connect with bank” as required one). Only one category of variants – activities – is introduced. The reason is that major part of business process variabilities are found namely in activities. Variation points, variabilities, variants and dependencies are identified in the analysis of process domain step and represented by FODA feature models (Kang et al., 1990). In design of process domain step all variabilities are represented by one of so called *Implementation*

## LITERATURE

*Business Process Definition MetaModel (BPDM)* (2007). Object Management Group, Inc. Access through Internet: <http://www.omg.org/cgi-bin/doc?bmi/> [2007-03-01].

CHANDRASEKARAN, B. (1986). Generic tasks in knowledge-based reasoning: High-level building blocks for expert system design. *IEEE Expert* 1(3), p. 23–30.

CZARNECKI, K.; EISENECKER, U. (2000). *Generative Programming: Methods, Tools, and Applications*. Addison-Wesley Professional.

CZARNECKI, K.; KIM, C. H. P.; KALLEBERG, K. T. (2006). Feature Models are Views on Ontologies. In *Proceedings of the 10th International on Software Product Line Conference*, p. 41–51.

ČIUKŠYS, D.; ČAPLINSKAS, A. (2005). Mod-

*Mechanisms*, discussed in (Puhmann, 2005). According to Puhmann, the following variability implementation mechanisms are suitable for modelling of business process variabilities: *Encapsulation of Varying Sub-processes*; *Addition*, *Replacement*, *Omission of Encapsulated Sub-processes*; *Parameterisation*; *Variability in Data Types*; *Inheritance*; *Design Patterns*; *Extensions/Extension Points*. The size of the paper does not allow discussing these mechanisms in more detail. The configurator should be able to process all implementation mechanisms

and to choose only one variant for each of variabilities. As a result, it produces a description of fully configured process and stores this description in *derivation profile*.

## Conclusions

The proposed approach is a part of ontology-based enterprise engineering methodology. It allows reusing of business process knowledge in different application domains. The knowledge to be reused is represented as process ontology. Main advantage of the proposed approach comparing it to ERP approach is that process is adapted to the needs of enterprise and not vice versa. So, better business and information systems alignment should be expected.

elling of reusable business processes: an ontology-based approach. In *Proceedings of ISD'2005 conference*. Karlstad, Sweden.

DAVIES, J.; STUDER, R.; WARREN, P. (2006). *Semantic Web Technologies: Trends and Research in Ontology-based Systems*. Wiley.

ERL, T. (2005). *Service-Oriented Architecture: Concepts, Technology, and Design*. Upper Saddle River: Prentice Hall PTR.

KANG, K.; COHEN, S.; HESS, J.; NOVAK, W.; PETERSON, A. (1990). Feature-Oriented Domain Analysis (FODA) Feasibility Study. *Technical Report CMU/SEI-90-TR-21*, SEI. Carnegie Mellon University, Pittsburgh, Pennsylvania.

*Meta Object Facility (MOF) Core Specification, version 2.0* (2006). Object Management Group, Inc. Access through Internet: [http://www.omg.org/cgi-bin/doc?formal/\[2006-01-01\]](http://www.omg.org/cgi-bin/doc?formal/[2006-01-01]).

*OWL Web Ontology Language Guide* (2004). W3C Recommendation. Access through Internet: <http://www.w3.org/TR/owl-guide/>

PUHLMANN, F.; SCHNIEDERS, A.; WEILAND, J.; WESKE, M. (2005) Variability Mechanisms for Process Models, *PESOA-Report No. 17/2005*. Access through Internet: <http://www.pesoa.org/pages/Publications-en.html>

SMITH, H.; FINGAR, P. (2003). *Business Process Management (BPM): The Third Wave*. Meghan-Kiffer Press.

USCHOLD, M.; KING, M.; MORALEE, S.; ZORGIOS, Y. (1998). The Enterprise Ontology. In *The Knowledge Engineering Review*, vol. 13, Special Issue on Putting Ontologies to Use.

*Web Services Business Process Execution Language v2.0* (2007). OASIS standard. Access through Internet: <http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.pdf>

## ONTOLOGIJOMIS GRINDŽIAMAS VERSLO PROCESŲ ŽINIŲ PAKARTOTINIO NAUDOJIMO METODAS

Donatas Čiukšys, Albertas Čaplinskas

### Santrauka

Straipsnyje pristatomas verslo procesų žinių pakartotinio naudojimo metodas, grindžiamas dalykinių sričių inžinerija, žinių inžinerija ir ontologijomis grindžiama sistemų inžinerija. Pagrindinė siūlomo metodo idėja yra atskirti verslo proceso ontologiją nuo dalykinės srities ontologijos ir pakartotinai panaudoti proceso ontologiją skirtingose dalykinėse srityse. Pasiūloma apibendrinto verslo proceso sąvoka, apibrėžiama kaip

panašių verslo procesų šeima. Straipsnyje aptariama apibendrinto verslo proceso lokacija dalykinėje srityje, susidedanti iš dviejų gyvavimo ciklų. Pirmame cikle yra atliekama apibendrinto proceso inžinerija, antrame – konkretaus proceso inžinerija. Pastaroji susideda iš trijų žingsnių: proceso konfigūravimo, dalykinės srities esybių priskyrimo proceso vaidmenims ir valdymo srautų tarp proceso veiklų apibrėžimo.