AN ONTOLOGY BASED APPROACH TO BUSINESS MODELLING

Gang Zhao, Yanbin Gao, Robert Meersman

Semantic Technology and Applications Research Laboratory, Department of Computer Science
Vrije University of Brussels
Brussels, Belgium

gang.zhao, yanbin.gao, robert.meersman@vub.ac.be

Abstract – Business modelling for IT systems needs to capture complex business semantics. Ontology-based approach for business modelling serves the purpose of capturing, representing and communicating the semantics of the underlying business logic. This article introduces ontology-based knowledge engineering for business modelling. It illustrates the DOGMA framework of ontology representation and its derived AKEM methodology to knowledge engineering development. The approach is illustrated with a real life case of business processes.

Keywords: ontology, knowledge engineering, business process, business modelling.

I. INTRODUCTION

As a result of such IT system requirements as collaboration, interoperability, adaptability, knowledge management, business modelling for IT systems needs to capture the rationale of the underlying business logic rather than merely operational procedures in order to enable the required intelligent processing based on domain and application semantics. This study explores the application of ontology engineering both as representation and modelling methodology in business modelling. The ontology-based approach addresses the basic challenges of business modelling: understanding, change management, communicating and consensus building. It suggests a methodology to manage the development process. It encourages capturing relatively stable conceptual model and thereupon manages the model scalability and reuse, dynamic changes and application-dependent variations of business logic. It seeks to identify basic semantics or meta knowledge underpinning the data and process models in IT systems. It uses the DOGMA (Developing Ontology Guided Mediation for Agents) paradigm of ontology modelling [18, 19] and the AKEM methodology for describing common business assumptions.

Among many other requirements, modelling business logic for system interoperation and collaboration is a recent one. Application needs for access to multiple data resources organized in divergent ways and collaboration of business processes in a business network will benefit from a description of underlying semantics and a corresponding mediation mechanism. Operations on multiple databases can be based on the ontology-based mediation. The diversities of database schemas can be encapsulated by ontology interpreters that enable database access in semantic terms. The dynamically set-up peer-to-peer interoperation over business processes in a collaborative business network can be also based on the ontology of semantic protocols for providing interpretation service of data or messages. Similar models of communication are seen in machine translation systems or software agents [8] or collaborative process manager [3].

II. DOGMA APPROACH TO ONTOLOGY

Ontology is an approximate semiotic representation of agreed conceptualization about a subject domain. It has two basic layers: lexons and commitments. A lexon is the conceptual construct depicting the generic semantic relationship underlying business data and processes. A commitment constrains, instantiates, composes and connects lexons with reference to specific business applications.

A. Lexons

A lexon is a quintuple \( < \gamma, t_1, r_1, t_2, r_2 > \), where \( \gamma \in \Gamma \) is a context identifier, \( t_1 \in T \) and \( t_2 \in T \) are terms referring to the entities in a semantic relationship. \( r_1 \in R \) and \( r_2 \in R \) are roles in the semantic relationship. \( \Gamma \), \( T \) and \( R \) are strings over an alphabet, \( A^+ \). The context identifier indicates an ideational context in which \( t \in T \) and \( r \in R \) in a lexon becomes meaningful. The ideational context is externalized by a set of resources, such as documents, graphs, databases. Through this resource, the semantic extension of a lexon is established, communicated, documented and agreed upon among ontology developers.

The lexon reads as follows. In the ideational context, \( \gamma \), the entity, \( t_1 \), plays the role, \( r_1 \) with the entity \( t_2 \). Alternatively, the entity \( t_2 \) plays the role, \( r_2 \) with the entity \( t_1 \). Here entity refers to a semantic type rather than its tokens. The lexon, therefore, embodies relationship or fact types instead of instances of an application domain.

B. Lexon Commitments

Architecturally, ontology consists of two layers: lexons and lexon commitments in the DOGMA representation framework [14, 23]. On the second layer, application-specific instantiation, integrity constraints and logical connections are applied to lexons. The commitment layer based on the lexon base introduces a modular approach to ontology development of an otherwise monolithic representation of generic and specific, declarative and operational semantics of application domains. It is based on the principles of encapsulating changes, application requirements and multiple perspectives of semantic models. In addition to the representa-
tion. It is also significant in the organization of development activities with dual emphasis on semantic generics and specifics at different tasks.

C. Lexon Base

The lexon base is a set of lexons. Formally it is a set of expressions composed from the ordered pair, \( <A, \Omega> \) where \( A \) is the alphabet and \( \Omega = \Gamma \times T \times R \times T \times R \). It constitutes conceptual system underpinning business knowledge. Similar to natural languages, the lexon base is a semiotic system or semiotic resources for communication. Multiplicity in perspectives and dimensions of representation is a feature of the lexon base as semiotic resources. It constitutes the representational potential in the lexon base and allows the actual perspective to be adopted as commitments to lexons. The lexon base allows for potential semantic redundancy, ambiguous, inconsistent, or even contradictory lexons. Their application-specific deployment as semiotic expression serves to ground lexons in a particular application context, removing ambiguity, inconsistency and contradiction.

III. AKEM For Ontology Development

AKEM stands for Application Knowledge Engineering Methodology. It is based on the DOGMA approach to ontology. It targets the intelligent application development with complex semantic models. Similar to classical knowledge or expert system engineering paradigms, it starts from application-specific requirements and ends with knowledge bases. AKEM distinguishes itself by its marked emphasis on the extraction, abstraction and organization of meta knowledge in the form of ontology, with the intention to support

- Capture and reuse of the application-independent, fundamental principles of business semantics
- Change management in view of the dynamics and change of business models
- Scaling up business modeling in layers to manage modeling complexity

It emphasises the informal specification of ontology in the actual process of knowledge engineering, its scalability, versitility and traceability as compared with other methodologies [6, 7, 12, 16, 24, 25]. It addresses the issue of ‘minimal encoding bias’, ‘minimal ontological commitment’[10] and intuitive representation [22] in both methodological process and deliverables. Its life cycle model is adapted from RUP [17, 19] and implements the controlled iteration of activities in four phases with a set of deliverables.

This article illustrates the ontology development for business modeling through the activities of scoping, analysis, development and deployment.

Figure 1: AKEM Life Cycle Model

A. Scoping

Scoping is to identify a part of the universe of discourse for modeling and development. The instrument for describing a particular semantic space of business is stories. The story not only indicates the part of the semantics proper to focus on, but also conveys the semantic context. It serves a guideline to identifying the relevant document resources from which business semantics is to be extracted and modeled. The main elements of the story are settings, characters, episodes. Stories are used to convey business case and scenarios and their semantic scope to the client, business analyst, knowledge and ontology engineer and developer of intelligent business applications. They are important instrument for requirements management.

B. Analysis

Given the stories and relevant document resources, the activity of analysis is to create the knowledge constituent model to describe how the semantics is decomposed and how each part is elaborated. The knowledge decomposition breaks down the high level business or strategies into a hierarchical structure of subtasks or business logic. Having structured the amorphous semantic space into a constituent structure, the next task is to zoom onto each constituent and elaborate on the relevant concepts and derivational rules of business logic. The elaboration is specified in a restricted natural language. It indicates what and how business entities, relationships, events, activities and organizations are involved.

C. Development

The main tasks are extraction, abstraction and organization of lexons. The methodology stresses and assumes the extensive use of texts as deliverables for informal knowledge specification. This is not only because of traceability of decision making in semantic modelling, but also the natural language text is the most effective means of communicating and building consensus on domain semantics.

The lexon extraction, is text-based and a linguistic activity. Its task is to spot key words and phrases of semantic importance in the text. The abstraction is a process of recognizing recurrent semantic types and
their inter-relationship. It is confined to the key words and phrases of the text under examination. It ‘translates’ the meaning verbalized by key words into a binary relationship. While the extraction and abstraction focuses exclusively on the semantic contents expressed explicitly in the text, the lexon organization goes beyond the semantics of text with the introduction of conceptions assumed or implied. The purpose is either rationalizing the interrelationship among lexons produced so far or integrating with existing top-level business ontologies. The resultant lexon base can be thought of as a knowledge dictionary.

D. Deployment

One task of the deployment relevant to the current topic is to specify commitments to lexons in the light of business tasks. The lexon commitment specification defines tokens of business logic and semantics in terms of lexons. It describes how lexons are grounded in applications and connected to form the network of application logic.

While lexon modelling seeks for the reusable semantic concepts in order to create a flexible ontology model of business, specifying lexon commitments refocuses the attention to a particular task in a context of particular business transaction. The activity is based on the results of lexon modelling in order to explicitly capture both the generic business assumptions and specific business constraints and logic.

The commitment specification used in this article features two constructs: commitment statement and commitment discourse, referred to as statement and discourse in the follow sections. A statement has a tripartite structure: theme, transition and rheme, similar to RDF’s structure of subject, predicate and object except that rhemes are always “classes” [20]. The names come from functional schools of linguistics in Europe, namely, Prague school of linguistics [9]. Using them here is to stress the network structure, multi-variate relationships and functional contextual aspects of lexon commitments, also found in natural language discourses.

1) Commitment statement

The statement is a lexon grounded in perspectives, constraints and instantiation. A lexon, \(<γ, t_1, r_1, t_2, r_2>\), can be grounded in the form of \([t_1, r_1, t_2]\) or \([t_2, r_2, t_1]\), depending on the choice of the perspective on the role. The perspective is expressed in the transition of the statement. In addition, the transition can be qualified with constraints such as modality, time, aspect and negation. The theme and rheme are filled by the terms of the lexon. They can be either instantiated with particular values, referring to application- and task-specific tokens, or constrained as sets, such as relation cardinality. The transition is the selected role, indicating the perspective of the grounded lexon. The constraints associated with the theme, transition and rheme are attribute value pairs. They can be processed by feature unification algorithms.

2) Commitment discourse

The discourse is a set of connected statements or other discourse. The connection operator ranges from logical, sequential to IO relationship. Similar to natural language discourse, the commitment discourse exhibits multi-dimensional relationship among the statements, between transitions, between themes and rhemes, as input and output to another discourse, structurally embedded or not.

In Figure 2, the tripartite box symbolizes statement. The upper box is the theme, the middle the transition and the lower the rheme. Each theme and rheme is constrained in term of cardinality and shared instantiation with equal signs.

![Figure 2- A Commitment Discourse](image)

The anchored connection joins the six statements by shared instantiation. \([\text{Quotation}, \text{Concern}, \text{Product}], [\text{Product, CharacterisedBy}, \text{ProductNumber}]\) and \([\text{Product, CharacterisedBy}, \text{ProductSpec}]\) are linked up in an endocentric configuration whereas \([\text{Customer, Receive, Quotation}], [\text{Quotation, Concern, Product}]\) and \([\text{Product, IncludedIn, Catalogue}]\) are connected in an exocentric configuration that relays instantiation between different commitment statements.

A second basic discourse connection uses logical connectors, such as if...then, and, or, prior to. The logical relationship cements the grounded lexons into business rules, either to capture derivation inferences or a sequence of business tasks and processes. Below are some basic constraints and connectors.

- Qualification
  - Cardinality
  - Quantification
  - Value constraint
  - Negation
  - Modality
  - Time
  - Aspect
  - Identification scheme

- Set constraints
  - Intersection
  - Complementation
E. Commitment specification

The specification format explored in this study is by the use of the program design language (PDL) [4]. PDL is intended for human use, though tools can be built to generate more formal specification or codes. It is a combination of some features in structured programming languages with restricted natural language. It includes a list of reserved words for structuring the specification contents, such as if ... then. Restricted natural language means some constraints on styles of expression, such only imperative, simple sentence. The requirement is to specify business logic in unambiguous natural language texts with reserved words that give clear clues on how to map the contents into an expected formal language or program structure. Our experience shows that it is an effective way to specify rich semantics for IT applications. Besides the operators listed, the reserved words are THEME, RHEME, TRANSITION, WITH, STATEMENT, DISCOURSE. Examples can be reserved words are THEME, RHEME, TRANSITION, WITH, STATEMENT, DISCOURSE.

IV. A CASE STUDY: BUSINESS PROCESS MODELING

This section uses an example to illustrate the ontology-based knowledge engineering approach to business modeling.

A. Scoping with stories

Below is an illustration of a story about order processing work flow of a telecommunication company. Its particular format is an instrument of AKEM for the deliverable of requirements. It is typically created by business analysis and knowledge analyst on the basis of a vision statement, problem definition and knowledge resources collected from clients. The story plays a similar role as use cases in UML and motivating scenarios [11].

<table>
<thead>
<tr>
<th>Setting</th>
<th>S1</th>
<th>CNC is a network and telecommunication services provider. It runs a big telecommunication and IP networking for providing products and services to the customers around the whole country.</th>
</tr>
</thead>
<tbody>
<tr>
<td>S2</td>
<td></td>
<td>In order to adapt to the speedy change in supplies and demands, meet the wide variety of customer requirements, improve time-to-market, low price and delivery, reduce management cost they decide to automate and collaborative its service order processing procedure.</td>
</tr>
<tr>
<td>S3</td>
<td></td>
<td>They envision that their service order processing system is capable of process-driven and customer-focused in service delivery accomplishment.</td>
</tr>
<tr>
<td>S4</td>
<td></td>
<td>They foresee their cooperation will support both short-term and medium-to-long term business management and cooperation.</td>
</tr>
<tr>
<td>Characters</td>
<td>C1</td>
<td>CNC is in charge of providing services to customers.</td>
</tr>
<tr>
<td></td>
<td>C2</td>
<td>Customers who apply for a service.</td>
</tr>
<tr>
<td></td>
<td>C3</td>
<td>Customer department is in charge of accepting the service application and managing the customers</td>
</tr>
<tr>
<td></td>
<td>C4</td>
<td>Orders department receives the service order application, form and manage the service order information.</td>
</tr>
<tr>
<td></td>
<td>C5</td>
<td>Sales department signs the contract for service request.</td>
</tr>
<tr>
<td></td>
<td>C6</td>
<td>Operations Department implements the order of the service.</td>
</tr>
<tr>
<td>Episode</td>
<td>E1.0</td>
<td>Customers apply for and order services.</td>
</tr>
<tr>
<td></td>
<td>E1.1</td>
<td>Customer department manages customer’s information.</td>
</tr>
<tr>
<td></td>
<td>E1.2</td>
<td>Order processing department manages orders and check on the requested service resources.</td>
</tr>
<tr>
<td></td>
<td>E1.3</td>
<td>If resources of ordered services is satisfied, reserve the resources for customer.</td>
</tr>
<tr>
<td></td>
<td>E1.4</td>
<td>When customer order contracted, operate engineering department implement the services.</td>
</tr>
<tr>
<td>Note</td>
<td>P1</td>
<td>The management foresees the dynamic changes of services and their organization. To encapsulate changes and maximise flexibility and adaptability, BS Analysts, Inc decides to take the approach of layered knowledge modelling.</td>
</tr>
</tbody>
</table>

B. Analysing into knowledge constituent model

In the scope indicated by the story, the domain expert and knowledge analyst seeks to work out a knowledge breakdown model. He takes the perspective of business processes, since the purpose is the work flow management. The knowledge decomposition phase as applied to service order processing is devoted to break down the service order processing into a hierarchical structure of subtasks. Starting from the episodes of the story, the knowledge units are refined into subunits, as the following shows.
Contextualized in the knowledge constituent and its elaboration for team communication and change management.

1) Extracting and abstracting lexons

Given the documents of stories and knowledge analysis, the ontology engineer performs the task of lexon extraction and abstraction. Table I shows a list of lexons extracted and abstracted from knowledge analysis of service order processing.

<table>
<thead>
<tr>
<th>Context</th>
<th>Term</th>
<th>Role</th>
<th>Term</th>
<th>Role</th>
</tr>
</thead>
<tbody>
<tr>
<td>order</td>
<td>Customer</td>
<td>ApplyFor</td>
<td>Service</td>
<td>BeAppliedFor</td>
</tr>
<tr>
<td>order</td>
<td>Customer</td>
<td>Order</td>
<td>Service</td>
<td>BeOrdered</td>
</tr>
<tr>
<td>order</td>
<td>SalesPerson</td>
<td>Manage</td>
<td>Customer</td>
<td>BeManaged</td>
</tr>
<tr>
<td>order</td>
<td>OrderManager</td>
<td>Manage</td>
<td>Order</td>
<td>BeManaged</td>
</tr>
<tr>
<td>order</td>
<td>Service</td>
<td>Use</td>
<td>Resource</td>
<td>BeUsed</td>
</tr>
<tr>
<td>order</td>
<td>SalesPerson</td>
<td>Inquire</td>
<td>Resource</td>
<td>BeInquired</td>
</tr>
<tr>
<td>order</td>
<td>Operator</td>
<td>Reserve</td>
<td>Resource</td>
<td>BeReserved</td>
</tr>
<tr>
<td>order</td>
<td>Operator</td>
<td>Release</td>
<td>Resource</td>
<td>BeReleased</td>
</tr>
<tr>
<td>order</td>
<td>Customer</td>
<td>Sign</td>
<td>Contract</td>
<td>BeSigned</td>
</tr>
<tr>
<td>order</td>
<td>SalesPerson</td>
<td>Sign</td>
<td>Contract</td>
<td>BeSigned</td>
</tr>
<tr>
<td>order</td>
<td>Contract</td>
<td>concern</td>
<td>Resource</td>
<td>BeReserved_</td>
</tr>
<tr>
<td>order</td>
<td>Operator</td>
<td>Deliver</td>
<td>Service</td>
<td>BeDelivered</td>
</tr>
<tr>
<td>order</td>
<td>SalesPerson</td>
<td>Inform</td>
<td>Customer</td>
<td>BeInformed</td>
</tr>
<tr>
<td>inquiry</td>
<td>SalesPerson</td>
<td>Make</td>
<td>Inquiry</td>
<td>BeMade</td>
</tr>
<tr>
<td>inquiry</td>
<td>OrderManager</td>
<td>Receive</td>
<td>Inquiry</td>
<td>BeReceived</td>
</tr>
<tr>
<td>inquiry</td>
<td>Inquiry</td>
<td>concern</td>
<td>Resource</td>
<td></td>
</tr>
<tr>
<td>inquiry</td>
<td>Inquiry</td>
<td>concern</td>
<td>Resource</td>
<td></td>
</tr>
<tr>
<td>inquiry</td>
<td>BackBoneResource</td>
<td>IsA</td>
<td>Resource</td>
<td>Include</td>
</tr>
<tr>
<td>inquiry</td>
<td>OrderManager</td>
<td>Inquire</td>
<td>Resource</td>
<td>BeInquired</td>
</tr>
<tr>
<td>inquiry</td>
<td>Operator</td>
<td>Answer</td>
<td>Inquiry</td>
<td>BeAnswered</td>
</tr>
<tr>
<td>inquiry</td>
<td>OrderManager</td>
<td>Confirm</td>
<td>Resource</td>
<td>BeConfirmed</td>
</tr>
<tr>
<td>inquiry</td>
<td>SalesPerson</td>
<td>Receive</td>
<td>Resource</td>
<td>BeReceived</td>
</tr>
<tr>
<td>reserve</td>
<td>SalesPerson</td>
<td>ApplyFor</td>
<td>Resource</td>
<td>BeAppliedFor</td>
</tr>
<tr>
<td>reserve</td>
<td>BackboneOperator</td>
<td>SubtypeOf</td>
<td>Operator</td>
<td>SupertypeOf</td>
</tr>
<tr>
<td>reserve</td>
<td>LocalOperator</td>
<td>SuperTypeOf</td>
<td>Operator</td>
<td></td>
</tr>
<tr>
<td>reserve</td>
<td>BackboneOperator</td>
<td>Answer</td>
<td>Inquiry</td>
<td>BeAnswered</td>
</tr>
<tr>
<td>reserve</td>
<td>OrderManager</td>
<td>Accept</td>
<td>Resource</td>
<td>BeAccepted</td>
</tr>
<tr>
<td>reserve</td>
<td>Operator</td>
<td>Validate</td>
<td>Resource</td>
<td>BeValidated</td>
</tr>
<tr>
<td>reserve</td>
<td>LocalResource</td>
<td>Satisfy</td>
<td>Resource</td>
<td>BeSatisfied</td>
</tr>
<tr>
<td>reserve</td>
<td>Operator</td>
<td>Reserve</td>
<td>Resource</td>
<td>BeReserved</td>
</tr>
<tr>
<td>reserve</td>
<td>OrderManager</td>
<td>Confirm</td>
<td>Resource</td>
<td>BeConfirmed</td>
</tr>
</tbody>
</table>
Different perspectives, such as organizational, data, process and function views can be presented separately or jointly in the form of commitment networks. And all this is done in the same representational framework: in terms of lexons, constraints and connectors.

3) Lexons of business data

As the process consumes and manipulates data, a common semantic model of data will be useful for interoperation of information systems. This meta conceptual data model abstracts away from data models implemented in the distributed and independent database systems and allows for shared process control on an abstract level. The lexons in Table III represent the data semantics.

4) Deriving statements

Relevant lexons are selected from the lexon base and grounded into a statement with a particular perspective.

STATEMENT

\[
\text{<client, Customer, ApplyFor, Service, BeAppliedFor>} \\
\text{THEME Customer} \\
\text{TRANSITION ApplyFor} \\
\text{RHEME Service}
\]

The theme, transition and rheme are constrained and/or instantiated in feature bundles.

STATEMENT <order, Customer, Order, Service, BeOrdered> \\
THEME Customer WITH ref: ctm \\
TRANSITION Order \\
RHEME Service WITH ref: svc \\

STATEMENT <client, Customer, BeCharacterised, CustomerName, Characterise> \\
THEME Customer WITH min:1, max:1 \\
TRANSITION BeCharacterised \\
RHEME CustomerName WITH min:1, max:1 \\

5) Composing discourse about processes

The following example shows a commitment discourse that consists of statements and discourses as well.

DISCOURSE order \\
VAR query, resource, customer, ordered_service, order, salesman, order_manager, contract, service_to_reserve \\

EVENT \\
STATEMENT <order, Customer, Order, Service, BeOrdered> \\
THEME Customer WITH ref: customer
The following example shows one of discourses embedded in the above example.

DISCOURSE inquiry (IN customer_inquiry OUT query_answer)
VAR salesman, order_manager, resource
EVENT
STATEMENT <inquiry, SalesPerson, Make, Inquiry, Be-Made>
RHEME SalesPerson WITH ref:salesman
TRANSITION Make
RHEME Inquiry WITH ref: customer_inquiry
ACTION
STATEMENT <inquiry, OrderManager, Process, Inquiry, BeProcessed>
RHEME OrderManager WITH ref:order_manager
TRANSITION Process
RHEME Inquiry WITH ref: customer_inquiry
IF
STATEMENT <inquiry, Inquiry, Concern, Resource, >
RHEME Inquiry WITH ref: customer_inquiry
TRANSITION Concern
THEN
STATEMENT <inquiry, LocalOperator, Answer, Inquiry, BeAnswered>
RHEME LocalOperator
TRANSITION Answer WITH ref:query_answer
RHEME Inquiry WITH ref: customer_inquiry
IF
STATEMENT <inquiry, Resource, PartOf, BackboneResource, Include>
RHEME Resource WITH ref:resource
TRANSITION Include
RHEME BackboneResource
THEN
STATEMENT <inquiry, BackboneOperator, Answer, Inquiry, BeAnswered>
RHEME BackboneOperator
TRANSITION Answer WITH ref:query_answer
RHEME Inquiry WITH ref:customer_inquiry
STATEMENT <inquiry, OrderManager, Confirm Inquiry, BeConfirmed>
RHEME OrderManager WITH ref:order_manager
TRANSITION Confirm
RHEME BeAnswered_Inquiry WITH ref:query_answer
STATEMENT <inquiry, Receive, Inquiry-Answer, BeReceived>
RHEME SalesPerson WITH ref:salesman
TRANSITION Receive
RHEME BeAnswered_Inquiry WITH ref:query_answer
END

6) Composing discourse about data

The discourse defining data structure can be written as follows.

DISCOURSE Customer
This study is partially funded by the EU 5th framework program, IST 2001-38248, the project of FF POIROT (www.ffpoirot.org)

REFERENCES:


