T

Translation of Natural Language Patterns to Object and Process Modeling

Alexandra Galatescu
National Institute for R&D in Informatics, Romania

INTRODUCTION

In order to shorten the engineering period of the complex information systems (ISs), the *integration of the models* is needed for uniformly: (1) traversing all phases of ISs' lifecycle (analysis, design, coding, testing), and (2) representing object, functional, process and organizational models on the business domain.

The seamless integration of the four models has not yet been a satisfactory solution from the conceptual, notational, semantic and logical viewpoints. The existing tools direct the designers to object-oriented modeling, possibly combined with "use case" and "state transition" diagrams. But, the functional, process and organizational models are not well and completely integrated with the object models. Moreover, using the agent-based technology for IS implementation, the knowledge modeling and integration will be necessary.

The integration solutions proposed in important methodologies like UML (Unified Modeling Language) (OMG, 2003) and IDEF (KBSI, 2000) (for object, functional and process models) or Workflow Reference Model (WfMC, 2003) (for organizational, process and functional models) mainly merge the models. Their conceptual integration is devolved to the developers of CASE tools or to the human designers and is accomplished during the coding phase. The existing methodologies, usually relying on the symbolic notation, do not provide the seamless and explicit (outside the code) integration of the object and activity-like symbols, from the semantic and logical viewpoints.

The integration abilities of natural language (NL) appear from the observation that people describe in NL any kind of information about objects, processes, information flows, the organization of their life and work, their knowledge, belief, intention, rationale and so forth. The universality and syntactic stability of a linguistic model is supposed to facilitate the communication among distributed ISs and users.

In computational linguistics, the main objective is to solve NL ambiguities (when more than one meaning is possible in a sentence) and to correctly identify the syntactic categories. Some NL analyzers build models for knowledge representation (logical models, semantic networks, frames, conceptual dependencies, conceptual graphs, etc.).

Instead, the discourse in the conceptual models (CMs) is a priori considered unambiguous and is expected to have a sound theoretical background. The objectives of NL analysis for model integration should be: (1) a linguistic theoretical foundation to the modeling interface, and (2) a uniform translation of NL to all types of CMs, without information loss. This article will focus on the second objective.

BACKGROUND

This section gives technical reasons for the research on natural language (NL) translation to conceptual models (CMs) (basically for model integration purpose) and a review of the state of the art and of the basic concepts and problems in this domain.

The complexity of information system (IS) representation comes from the complexity and diversity of the concepts it should integrate, defined in object, functional, process and organizational models. The integration of these models should be accomplished during the IS analysis or design phase.

These models result from the abstraction of the IS requirements, expressed by the analysts in NL. A comparison between the NL model (Allen, 1995; Sag, 1999), and the existing CMs reveals the conceptual relationships between NL and conceptual modeling. The most important relationships are:

- Between *categories* in CMs and NL. For example, between objects and nouns, activities and verbs, object attributes and adjectives, activity attributes and adverbs. Also, the noun and verb determiners/modifiers/substitutes in NL have counterparts in CMs.
- Between *semantic relationships* in CMs and NL. For example, between object aggregation/fragmentation and noun meronymy/holonymy; between object specialization/generalization and noun hyponymy/hypernymy; between functional composition/decomposition of processes and verb meronymy/ holonymy, and so forth.
- Between *syntactic roles* in NL (subject, predicate, direct/indirect/prepositional object, complement,

adverbial modifier) and primitives in CMs. For example, the predicate is represented by an activity/ event; the subject in active voice is the object-like sender of the message or initiator of an activity; the direct, indirect, prepositional objects are object-like parameters in activity execution; and so forth.

- Between *structures* in NL and CMs. For example, the simple sentence in NL is represented by the activity signature (list of object-like parameters that participate in the activity execution); the complex sentences are represented by (sub)process diagrams, and so forth.

Lately, the *co-references* between concepts in two or several sentences and the *ellipses* in NL are also represented in CMs.

This similarity made the researchers think that conceptual modeling could be as powerful as NL for representing the reality. But, the translation of NL patterns into a CM that integrates the four models is still an open problem.

NL-CM translation has been tried in several research domains: linguistic interpretation of the models (mainly, entity-relationship and object-oriented models), semantic integration of the conceptual schemas, modeling the systems' dynamics, human-computer interaction, requirements engineering, organization modeling, knowledge representation, formal ontologies and their application to the search on Web, business communication modeling based on speech act theory, and so forth.

The most important results have been obtained for the translation of NL to object and event models. Among the *NL-oriented representations* of these models, the most important are the functional grammar (FG) and the semantic networks (especially, conceptual dependencies and conceptual graphs). They propose the representation of the object models by syntactic categories and rules.

Functional grammar represents the functional aspects in NL, by the description and classification of predicate frames. FG has been further used for defining CPL (conceptual prototyping language), which focuses on NL simple sentences (the intersentential relations are not explicitly revealed). Also, CPL does not approach general semantic relationships inside the lexical categories (e.g., noun/verb synonymy, antonymy, homonymy, etc). COLOR-X (Riet, 1998) can be considered the most important application of CPL and practical result for CMs' integration. It integrates static object and event models of information and communication systems, abstracted from their textual descriptions. It relies on OMT (object modeling techniques). Like in any object-oriented model, the processes merely trace events that compose scenarios, similar to use case diagrams. Lately, the textual requirements are transformed into UML-schemata, for example (Fliedl, 2000).

The *conceptual dependencies* and, lately, the *conceptual graphs* (CG) (Sowa, 2000) are other linguistic representations for CMs. The syntactic categories are suggested by their roles to each other (meaning relationships between nouns and the verb that governs them in a simple sentence, e.g., agent, patient, instrument, recipient, location, time, source, destination, etc). A similar representation is the frame description in FrameNet (Filmore, 2002). But, all these representations are data-centric (ISs' dynamic behaviour is not important as a modeling goal).

The translation of NL to organizational models (workflows) has been obtained mainly with respect to the modeling of the business communication (e.g., Steuten, 2000). The theory of communicative actions and, lately, the *speech act theory* (Johannesson, 2001) are the main linguistic representations of the communication aspects in ISs.

The globalization of the organizations has a great impact on IS representation, especially with respect to the common vocabularies and the interoperability between distributed and heterogeneous applications. In this context, the conceptual modeling must step into a new era and intersect a new field: formal *ontologies*. A first benefit from ontologies for IS representation is that they describe, categorize and constrain concepts and relationships at the development time (Guarino, 2000). Using CMs, the constraints are basically imposed at run time. Another benefit is that the ontology specification is outside the code, while many object-oriented modeling specifications (especially constraints) are implemented inside the code.

Unfortunately, for the conceptual integration and for the representation of the ISs' dynamics, the existing ontologies have the same limits as the CMs. Most of them are object-oriented, relying on OKBC (open knowledge base connectivity) specification. For building business process ontologies, PSL (process specification language) (initiated by NIST) is recommended (e.g., Gruninger, 2003), mainly because it can be logically integrated with KIF (knowledge interchange format), appropriate for object and knowledge description and exchange.

For *ontology integration,* two alternatives can be considered: (1) by an upper-level ontology, able to represent all aspects in the real world, or (2) by a translation and correlation algorithm between the concepts and rules in different ontologies. Such an algorithm is mostly encoded and the ontology integration is recommended to be accomplished at the development time. So, the first alternative appears as a better solution. For the conceptual integration of all aspects in the real world (and, implicitly,

Related Content