Reifying Model Integration Abilities from Natural Language

Alexandra Galatescu, National Institute for R&D in Informatics,
8-10 Averescu Avenue, 71316 Bucharest 1, ROMANIA,
Fax: 40 1 224 05 39, E-mail: agal@std.ici.ro

The paper is a study on natural language (NL) from the viewpoint of the conceptual model integration. The universality, unifying abilities, natural extensibility, logic and reusability of NL, as well as its morphological and syntactic stability, made the modelers think of transposing these features to object and process modeling. A correspondence is established here between the main concepts and relationships in NL and the basic concepts and relationships which the designers use in different models like: object, process, dataflow and workflow models. Two main directions are proposed for the study on the integration abilities of NL: (1) the *morphological and syntactic categories* and (2) the *semantic and intersentential relations* between objects and actions. These categories and relations are supposed to support the seamless integration of the models, as the background either of a *meta-model with linguistic features* or of a *unifying translation algorithm* from NL to several conceptual models.

1. Introduction and motivation

This paper is devoted to the conceptual modelers and has as objectives: (1) to enumerate the requirements for the conceptual model integration, (2) to emphasize the limits of the symbolic notation and the benefits from the application of natural language (NL) patterns to conceptual modeling, (3) to remind the modelers the basic structures and features of NL and to establish their counterparts in conceptual modeling, (4) to sketch a possible solution to the model integration by the abstraction of the linguistic patterns.

Basic Requirements for the Conceptual Model Integration. In order to shorten the engineering period of the complex information systems (ISs), the convergence of the models is needed for uniformly (i.e. using the same conceptual and representation means for): (1) representing the most important aspects in the real world (at least those in Table 2, first column), (2) traversing *all phases* of IS's life cycle, (3) representing *all perspectives* on the business domain (at least the four perspectives (a-d) in Table 1), (4) representing both *business* and *infrastructure* objects, processes, constraints, rules etc, (5) externally and internally representing the system, using the same conceptual means.

TABLE 1
Four perspectives upon the information systems

DATA Perspective (a)	FUNCTIONAL Perspective (b)	CONTROL Perspective (c)	ORGANIZATIONAL Perspective (d)
OBJECT - Structure - Behaviour - Interoperability	DATA/ KNOWLEDGE FLOW - Process functional decomposition - Object/ Concept/ Rule transfer (between activities)	PROCESS - Events - Object / Activity's state transition - Control strategy and flow of the activities	GROUP Organization - WORKflow Supervising -TASK Distribution and Control - TEAM work

The seamless integration of the models (from conceptual, notational, semantic and logical viewpoints) has not yet a satisfactory solution and the modelers still search for the ideal modeling language. The existing tools direct the designers mainly to object-oriented (OO) modeling (possibly, combined with use case and state transition diagrams), along all phases and for all perspectives in IS development. Unfortunately, the functional, organizational and control perspectives are not fully and well integrated in the OO models. Also, the expansion of the agent-based technology entails *a new perspective*, the *epistemological* one, that aims at knowledge and rationale modeling, integration and distribution.

Important methodologies, like UML (OMG (1997), Harmon (1997)) or IDEF family (IDEF conference) (for data, functional and control perspectives), WfRM (Lawrence (1997)) (for organizational, control and functional perspectives), KADS (Schreiber et al. (1993)) (for data, control, functional and epistemological perspectives) etc, mainly *merge models* (one model for each perspective). Their real integration is usually devolved to the developers of the CASE tools or to the human designers and is accomplished in the coding phase.

Benefits from Natural Language (NL) in Model Integration. The existence of the integration abilities in NL, our primary, universal and vital abstraction of life, appears from the observation that, in NL, people describe any kind of information about objects, processes, information flows, about the organization of their life and work, and about their knowledge, beliefs, intentions, rationale etc. So, the universality, as well as the morphological and syntactic stability of a NL-oriented model, would facilitate the *communication* among distributed programs, ISs, users, Internet sites etc.

Toward the model integration using linguistic means, the first step should be the unification of the four perspectives (see Table 1) in a *meta-model*, able to abstract the morphological, syntactic and semantic features of NL and able to synthesize, stylize, adapt and apply the NL patterns to IS modeling. This step implies (1) the examination of the NL patterns (tried in Section 2) and (2) the comparison of the NL patterns with the modeling ones (tried in Section 3 and 4). The semantic representation of NL should lack ambiguities and should express the meaning of the sentences without information loss.

A next step (beyond the scope of the paper) is to analyse, then to synthesize, adapt and apply, the most important and general ideas/ rules/ algorithms etc, already discussed and accepted in the NL processing theories and tools and appropriate to object and process modeling.

The analysis of NL is mainly the work of the specialists in computational linguistics (e.g. Allen (1997, 1995), Sag (1999) etc). However, the objectives of the *linguistics* are different from those of the *conceptual modeling* field, because NL and the conceptual modeling languages are on *different abstraction levels* of the real life. For instance, an objective in the computational linguistics is to solve the NL ambiguities and to correctly identify the morphological and syntactic categories. Instead, the discourse in conceptual modeling languages is a priori considered unambiguous, because usually they are expected to have a sound theoretical background.

Also, the final goal of the NL analysis should be different if it is intended for the model integration. It should give a linguistic theoretical foundation to the modeling interface, as well as to the natural and uniform translation of NL to, as many as possible, existing models, without information loss. It also should not bias to any particular representation of the grammar. For this reason, the study in this paper starts from the basic grammar structure, not from a particular linguistic theory.

Limits of the Symbolic Notation for the Model Integration. These limits will be motivated by briefly examining UML (OMG (1997)) which is proposed as the most important solution today to the integration of the object and process models. But, as a model based on a symbolic notation, it has *disadvantages* regarding:

- *model extensibility*, because any new type of operators/ concepts/ relationships etc introduces a new symbol/ icon/ marker, resulting into an overwhelming notation;
- *model learning*, because too many distinct symbols increase the designers' confusion;
- *system internal structure and evolution,* because there are not universal rules to uniformly translate symbols with heterogeneous semantics and logic into internal structures, irrespective of the implementation models;
- *control of the system's logical and semantic consistency* during the early phases of the system's life cycle, because there are not universal rules to generalize and formalize, hence to control, the logic and semantics of the correlated symbols;
- *integration of the object and process models* because, usually, there is not a sound and generalized solution to the seamless integration of the symbols, from the semantic and logical points of view.

As Table 2 synthesizes, UML notation has a counterpart to the NL main morphological units and functions, but the most UML symbols are indirectly associated with the NL morphological constructions and have a heterogeneous notation. For instance, (1) the events in the interaction diagrams and the activities in the activity diagrams are differently represented, although both are verbs in NL; (2) the objects and the objects' attributes are differently represented, although the same notion (e.g. 'red', instance of COLOR) may have both roles (noun and adjective), even in the same sentence (e.g. 'My shirt is red because I like the red'); (3) an explicit (outside the code) object-action relationship exists only in the interaction diagrams (the object-event relationship), although one may expect it in the activity diagrams as well. These are only few examples of heterogeneity found in UML and, also, in the common attempts to integrate objects and process models using a symbolic notation.

TABLE 2

A parallel between the conceptual characteristics in the real world and their abstraction in NL and in the symbolic notation of UML

Real world	Natural Language	Unified Modeling Language
Complete assertion, attitude, question	Sentence	- Diagram/ Subdiagram
Complex idea, situation, application	Phrase/ Paragraph etc	- Complex diagram (e.g. Use Case or Package diagram)
Object, abstract notion, place, event, quality (simple/complex)	Noun (simple/ compound)	Object (flat/ composite)
Action, state, attitude, event (simple/ complex)	Verb (simple/ compound)	- Message (method/ event) - Activity/ Complex transition - Type of Association
Object feature, quality (simple/ complex)	Adjective (single/ multiple)	Object's attribute (value/ complex structure)
Action characteristic	Adverb	- Operation property - Attribute of the association

Real world	Natural Language	Unified Modeling Language
Anaphora (backward reference)	Pronoun / Anaphoric article	- Reflexive association - Object lifeline
Object – Action Relationship (Role of the Object in Action)	Preposition / Conjunction / Adverb (see Table 8)	- Role in association/ interaction - Relation between a message/ event and its parameters (objects)
Inter-Object Relationship	Conjunction (copulative/ disjunctive etc)	- Association/ Dependency - Aggregate composition - Message
Inter-Operation Relationship	Conjunction / Conjunctive adverbs	Simple/complex transition / Control icons (both in activity diagrams)
Object Plural	Cardinal of the individual noun/ Collective noun	- Object multiplicity in association - Multiobject
Object Quantification and Modality	Noun quantifier/ 'there is' construction	Object multiplicity in association
Action Modality	Modal verb	Decision symbols in activity diagram
Object Type–Instance dichotomy	Noun, Proper noun	object: Class
Specialized meanings of Objects/ Actions	Word Polysemy	Class specialization/ Interface
Encrypting the knowledge on Objects/ Actions	- Human Knowledge/ Metonymy/ Anaphora	Packing class definition /diagram
Disjunctive meanings/ possibilities of Objects/ Actions	Word Homonymy	Separate classes/ Decision among activities / OR split in transition/ Stereotypes
Question	Interrogative sentence	' Query' property of the operation
Embedded Objects/ Ideas / Actions	Embedded sentences	Composite objects/ Nested states

In the second and third columns of Table 2, one may notice that the same aspect/ phenomenon in the real world may have several abstractions in NL and UML. Regarding NL, they reveal the flexibility of NL but, at the same time, sources of ambiguities and heterogeneity in the pure NL expressions. Regarding UML, they reflect the heterogeneous abstraction, hence further difficulties in a uniform implementation.

Related Research. Many theoretical or practical results have been obtained lately with respect to the application of NL paradigm to IS engineering. The most important research directions are: the *linguistic interpretation of the models* (mainly ER model, e.g. Johannesson (1996, 1997), Lewrenz (1999) and OO models, e.g. Moreno&Riet (1997)), the *semantic integration of the conceptual schemas* (Bouzeghoub (1997), Johannesson (1997)), the *modeling of the systems' dynamics* (e.g. Burg et al. (1995, 1997), Riet et al. (1998)), the *human-computer interaction* (e.g. Lewrenz (1999)), the *requirements engineering* (e.g. Fliedl, (1999)), the *organization modeling* (e.g. Medina-Mora (1992)), the *knowledge representation* (e.g. Sowa (1988, 1991), ANSI (1995)), the *formal ontologies* and their application to the search on Web (e.g. Miller (1990), Guarino (1999)), the *business communication* (e.g. Steuten (1999)), the *communication modeling* based on speech act theory (e.g. Johannesson (1997, 2001), Reijswoud&Mulder (1997)) and many other domains.

An important research on the application of NL patterns to conceptual modeling has been starting with Functional Grammar (FG) (Dik (1989, 1990)). In FG, NL is seen from the functional point of view, i.e. the predicates form the central structuring unit of the theory. The predicate frame specifies: the roles entities play within the predicate, the meaning of the predicate arguments, the taxonomic and antonymic inter-predicate relationships. The grammatical, structural and conceptual information in the predicate frames has an elaborate description and classification. A functional point of view of NL, based on the description of the predicate frame, has also been adopted in the abstraction of NL proposed in this paper, where the predicates are predefined or domain specific operations and the predicate frame relies on the roles of the objects in each operation.

FG has been used in the Conceptual Prototyping Language (CPL), which was intended as a specification language and, also, as a general knowledge representation language (Dignum (1989, 1991), Weigand (1989)). Its formal semantics relies on the predicate, modal, deontic and temporal logic and its linguistic structures can be used in an inference process. CPL can be used to specify unambiguously the requirements of the information and communication systems. From the point of view of the model integration, the limits of the linguistic patterns provided by CPL are: (1) they mainly focus on NL simple sentences. Although the simple sentences can be correlated, the intersentential relations in the compound and complex sentences in NL are not explicitly revealed; and (2) they do not approach, in a general way, the semantic relations inside the lexical categories e.g. noun/ verb synonymy, antonymy, homonymy etc).

COLOR-X (with its Static Object Model and Event model) (Burg et al. (1995, 1997), Riet et al. (1998), Vos (1997)) is the most important result today, closer to an expected solution to model integration. It combines the formal and linguistically based conceptual modeling technique CPL with graphical modeling techniques. Its tools facilitate the creation of consistent and complete models of the information and communication systems, linguistically correct, from the textual descriptions in NL of these systems. Eventually, COLOR-X is intended to generate OO programming code from a NL-based modeling technique. The meanings of the words are found in the underlying lexicon WordNet. Regarding the model integration, the authors of COLOR-X have chosen to integrate the modeling aspects approached in OMT (Object Modeling Technique, see Rumbaugh et al. (1991)). The processes are represented by the event model that, like in any OO model, merely trace the events that could appear and be performed in a universe of discourse. Informal sequences of events compose scenarios close to 'use case diagrams', that describe the behaviour of parts of the system (Burg&Riet (1997)). However, in the practical use, only the sequences of events and the state transition diagrams have been proved insufficient for representing complex processes. For this reason, in UML, activity diagrams have been considered as well.

Another application of the Functional Grammar is to the lingustically-based modeling of the communication actions in Steuten et al. (1999). The proposed solution relies on DEMO (Dynamic Essential Modeling of Organizations) based on the theory of the communicative action. The integration of the communication model in DEMO with the static object model and event model in COLOR-X is also foreseen, because of their common underlying linguistic theory: Functional Grammar.

A comparison between NL and OO modeling and a formalization of this relation is provided in Moreno et al. (1997, 2000). The NL-OO model equivalence is proved by the equivalence of the mathematical representation of the linguistic patterns in first order logic and of the conceptual patterns in the set theory. The emphasis here is on the relationships between the nominal syntagma and the complement in a simple sentence in NL. These relationships are either operators in conceptual modeling (e.g. 'type-of', 'is-a', 'class-of') or verbs (predefined like 'is', 'exist', 'may be', 'classify as', 'identifies' or domain-specific verbs). A similar idea is in the solution proposed in this paper, where the inter-object relationships are predefined or domain specific operations.

Another result is KISS (Hoppenbrouwers et al. (1997)) that provides a set of heuristics and procedures to extract the model information from NL sentences. It performs a grammatical analysis to extract the subject, object and action specifications from the textual description of the real world. It also extracts the synonyms (two words for the same concept) and homonyms (one word for two concepts).

The conceptual graph (CG) formalism (Sowa (1988, 1991), ANSI (1995)) is the inspiration source for several linguistic conceptual models (Levreau (1991), Johannesson (1996, 1997), Ambrosio (1997), etc). However, a pure CG-based solution is acceptable only for a data-centric approach, where the system's dynamics (object behaviour and functional operations) is not important as a modeling goal. This restriction is partly removed in Lukose (1996). CGs have also been proved to be useful for representing the content of a lexicon. They are the main inspiration source for the representation proposed in this paper.

In Vos et al. (1997), the meanings of the objects and actions in the real world are explicitly declared by the noun and verb-related features, taken from the linguistic theory. This idea (explicit representation of the object/ action meaning) was also adopted in the solution proposed in this paper where these meanings are extracted into object-action roles and into generic operators that define, describe or correlate objects and actions.

In summary, the limits claimed here of the existing practical and, also, theoretical comparisons between NL and conceptual modeling are: (1) the established correlations usually confine to the inter-object relationships and do not consider the inter-activity relationships; (2) the most relationships identified are those already used and formalized in the conceptual modeling; (3) the meaning of the objects/ actions and of the relationships between them is rarely explicitly extracted and, usually, is incomplete; (4) the intersentential relations and many semantic relations between the words in the same lexical category are usually not considered in the existing comparisons. The proposed analysis and solution in this paper try to go beyond these limits and to explore a possibility to partly alleviate them.

Steps in the Human Understanding of NL. People are not aware about the connections and steps that allow them to understand and integrate the meaning of the NL sentences. A technical and rather simple view (but enough for a starting point in the model integration) suggests the following steps in the human understanding of NL (not necessarily in a chronological order):

1. Associate the words with the *morphological categories* they belong to (nouns, verbs, adjectives, adverbs, etc) and then with their description parts (noun head/ substitute/ determiner/ modifier, verb determiner/ modifier).
2. Identify the *syntactic roles* of the nouns relative to the action of the verbs.
3. Correlate the words by *semantic relations* inside a lexical category (e.g. synonymy, antonymy, etc) or between them.
4. Correlate the words by *intersentential relations* in compound and complex sentences, in paragraphs and sections.

These steps stand here for axioms and directions toward the model integration using linguistic means. They are justified by the NL patterns and relations summarized in this paper.

Structure of the Paper. Section 2 summarizes and reminds to the modelers the main morphological and syntactic patterns in NL. Section 3 parallels the morphological and syntactic categories in NL with their counterparts in the conceptual modeling. In Section 4, the parallel is extended to the semantic and functional relations in NL, which are compared with the corresponding relations among the basic modeling concepts. The NL versus modeling comparison in this paper is supplemented with a brief presentation of a possible translation of the NL patterns to object and process modeling (details are in Galatescu (2001)).

2. Morphological and Syntactic Patterns in Natural Language

Traditionally (Allen (1987, 1995)), NL incorporates several types of knowledge:

- *phonetic and phonological knowledge*: how words are realized as sounds;
- *morphological knowledge*: how words are constructed out of basic meaning units;
- *syntactic knowledge*: how words can be put together to form sentences that look correct;
- *semantic knowledge:* what words mean and how these meanings combine in sentences to form the sentence meanings;
- *discourse knowledge*: how previous sentences affect the interpretation of the subsequent sentences;
- *pragmatic knowledge*: how sentences are used in different contexts and how the contexts affect the interpretation of the sentence;
- *world knowledge:* general knowledge about the structure of the world that the language users must have in order to maintain a conversation.

In this section, the basic morphological and syntactic knowledge in NL is approached, as much as possible *common to several speech languages* and *independent of any theory or parsing tree* in the computational linguistics (a comparative analysis of these theories is in Sag (1999)). The examples in this paper are in English, in or about the reference scenario in Figure 1.

The NL patterns, either morphological (formal) or syntactic (functional), focus on the following structures and relationships: *sentences* (simple, compound, complex); *verb phrases* (around finite verbs); *noun phrases*; *verbal phrases* (around non-finite verbs); *prepositional phrases*; *adjectival phrases*; *clauses* (noun, adverbial, relative, appositive).

The NL patterns described in this section synthesize the structure of English grammar (proposed in Budai (1994)). The functions of the NL basic components (verb, noun, article, adjective, pronoun, adverb, numeral, preposition, conjunction) are supposed to be familiar to the reader.

Reference Scenario. The exemplification in this section of the NL patterns relies on the steps of the reference scenario in Figure 1. In Harmon (1997), this scenario completes the UML use case diagrams for a sales application. It represents a combination of pure NL with an incipient abstraction of the objects, which are referred to by their implementation names: Salesperson, Sales Order, CustInfo, Accounting etc. These names suggest that, in OO modeling, the objects passive at certain points in time (e.g. application names) become *responsible objects* further on during the system's execution (e.g. Sales Order, CustInfo, Accounting, etc).

---

0. *Salesperson* **turns on** laptop, **brings up** *SalesWeb program* and **chooses** *Report Sales Order*.
1. *Salesperson* **enters** the employee name and number.
2. *Sales Order* **checks** to see if name and number are valid.
3. *Salesperson* **enters** *Customer* name and address on *Sales Order* form.
4. *Sales Order* **checks** *Customer* information to find out *Customer* status.
5. *CustInfo* **checks** *Accounting* to determine *Customer* status.
6. *Accounting* **approves** *Customer* information and supplies *Customer* credit limit.
7. *CustInfo* **accepts** *Customer* entry on *Sales Order*.
8. *Salesperson* **enters** first *Item* being *Ordered* on *Sales Order* form.
9. *Salesperson* **enters** second *Item* being *Ordered* etc.
10. When all *Items* **have been entered**, *Items Ordered* **are checked** to determine availability and to check pricing.
11. *Items Ordered* **checks** with *Inventory* to determine availability and to check pricing.
12. *Inventory* **supplies** all availability dates (when it can ship), **approves** prices, **adds** shipping and taxes and **totals** order.
13. Complete *Sales Order* **appears** on *Salesperson*'s screen.
14. *Salesperson* **can print** order, **check** with *Customer* etc.
15. *Salesperson* **submits** the approved *Sales Order*.
16. *Sales Order* **is submitted** to *Accounting* and *Inventory*.

FIGURE 1 Reference scenario for a sales application (Harmon (1997)). The main *objects* are in italic and the main **actions** are in bold.

2.1 MORPHOLOGICAL (FORMAL) PATTERN OF THE SIMPLE SENTENCE

Simple sentence = Noun Phrase + Verb Phrase, with

***Noun** Phrase* =Noun Determiner(s) + Noun Pre-Modifier(s) + Noun Head/ Substitute + Noun Post-Modifier(s)

Noun Determiner ∈ {Article, Pronoun, Numeral, Adjective, Combination of determiners}
Noun Pre-modifier ∈ {Adjective, Noun, Adverb, Verbal Phrase, Clause, Sentence}
Noun Substitute ∈ {Pronoun, Numeral}
Noun Post-Modifier ∈ {Adjective, Adverb, Noun, Verbal/ Prepositional/ Clause Phrase}

***Verb** Phrase* = Verb + Verb Determiner(s) + Adverbial Modifier(s)

The main characteristics of the nouns and verbs are summarized in Table 3. With few exceptions, these characteristics are common to several speech languages. The morphological components of the verb determiners and adverbial modifiers are correlated, in the next section, with the syntactic categories in the simple sentence.

TABLE 3
Verb and noun characteristics

Characteristic of the verb in NL	Characteristics of the noun in NL
1. Finite Full Verb	1. Noun Head
2. Auxiliary Verb - tense former (be, have, shall, should, etc) - mood-former (can, may, must, ought to, etc)	2. Simple Noun 3. Common (concrete/ abstract) Noun 4. Proper Noun
3. Multi-word Verb	5. Compound/ Parasynthetic (with prefix and suffix) Noun
4. Non-finite Verb (Verbal)	6. Derivative Noun
5. Verb Mood - indicative (including interrogation) - imperative - subjunctive (e.g. I wish (that) I were there)	7. Plural Noun 8. Collective Noun 9. Countable Noun 10. Case / role of Noun relative to a Verb
6. Verb Aspect (common, continuous)	11. Noun Quantification and Modality
7. Verb Tense	12. Noun Qualifier (modifying/ determinative adjective)
8. Verb Voice	13. Noun Polysemy (specialized, not necessarily disjunctive, meanings of the noun)
9. Verb Number	
10. Verb Person	14. Noun Substitute: Pronoun (personal, demonstrative/ relative, possessive/ reciproc, indefinite (partly))
11. Verb Polysemy (different, not necessarily disjunctive, meanings of the verb)	
12. Verb Homonymy (identic verbs, with disjunctive meanings)	15. Noun Homonymy (identic nouns with disjunctive meanings)

2.2 SYNTACTIC (FUNCTIONAL) PATTERN OF THE SIMPLE SENTENCE

Simple sentence = Subject + Predicate, with

Subject∈ {Noun Phrase, Noun Substitute, Verbal Phrase, Clause, Indefinite/ Formal Subject}

$$\textbf{\textit{Predicate}} = \text{Finite-Verb} + \underbrace{\left\{\begin{array}{l}\text{Link-verb + Subject Complement (SC) /}\\ \text{Direct Object (DO) /}\\ \text{Direct Object + Direct Object/}\\ \text{Indirect Object (IO) + Direct Object/}\\ \text{Prepositional Object (PO) /}\\ \text{Object (O) + Object Complement (OC)}\end{array}\right\}}_{\textit{Verb Determiners}} + \text{Adverbial Modifier(s) (AdvM)}$$

The *verb determiners and modifiers* are synthesized in Table 4 and exemplified in Table 5. Table 4 also suggests the *correspondence between the syntactic form* (first column) *and the morphological one* (third column) of the NL sentence and Table 5 exemplifies this correspondence.

NOTE
Obviously, some parts of the patterns above may be missing or may be implicit in the common use of NL. This paper only tries to summarize the basic elements of the NL sentence, for which the modelers should establish couterparts in the models or modeling languages they create or use.

TABLE 4

Verb determiners and modifiers

Verb determiner/ modifier	**Function in sentence**	**Corresponding morphological element**
Subject Complement (SC)	Gives information on the subject. It is related to a link-verb (finite verb that expresses being/ passing/ remaining/ seeming or appearing in a certain state)	Noun Phrase/ Noun Substitute/ Adjectival Phrase/ Verbal Phrase/ Prepositional Phrase/ Adverb Phrase/ Clause Phrase
Direct Object (DO)	Person/ thing the verb acts upon (accusative case)	Noun Phrase/ Noun Substitute/ Verbal Phrase/ Clause Phrase
Indirect Object (IO)	Person/ thing receiver of the verb's result (dative case)	Noun Phrase/ Noun Substitute
Object (O)	Functions of DO or IO	All elements for DO or IO
Prepositional Object (PO)	Governs the objects of the verb by means of a *preposition* (Prep. + O)	Prepositional Phrase/ All elements for Object (O)
Object Complement (OC)	Gives information on an object of the verb.	Noun Phrase/ Adjectival Phrase/ Verbal Phrase/ Clause Phrase
Adverbial Modifier (AdvM)	Verb modifier and determiner at the same time	Adverb/ Adverb Phrase/ Adverbial Clause/ Verbal Phrase/ Prepositional Phrase

TABLE 5

Examples of syntactic and morphological elements in the reference scenario (Figure 1)

Syntactic element	**Examples in/ about the reference scenario**	**Morphological counterparts** (some particular examples)
Subject (if predicate in active voice)	- *Salesperson, Sales Order, Items Ordered* - CustInfo, Accounting, Inventory - *Complete* Sales Order (Step 13) - *it* (Step 12)	-> Compound Noun -> Proper Nouns -> Noun + Noun Pre-modifiers -> Noun Substitute
Verb	- chooses, enters, checks, approves etc - *can* print - turns on, brings up	-> Finite simple Verbs -> Verb + Modal -> Finite Prepositional Verbs
Subject Complement (SC)	- Salesperson looks *content* of the sale. - His wish is *selling more*. - He became *an expert in sales*.	-> Adjective -> Participial Verbal Phrase -> Noun Phrase
Direct Object (DO)	- turns on *laptop*, enters *name* - checks Customer *information* - enters first *Item* (Step 8) - checks *to see if* ... (Step 2)	-> Compound/ simple Noun -> Noun + Noun Pre-modifier (noun or numeral) -> Infinitive Verbal Phrase with adverbial subclause of purpose
Indirect Object (IO)	to *Accounting* and ... (Step 16)	(in this example) Proper Nouns
Prepositional Object (PO)	- enters *on* ... form (Step 3) - appears *on* ...screen (Step 13) - Customer is waiting *in front of* the Salesperson *during* the checking of his account.	-> Simple Preposition -> Phrasal/ Group Preposition -> Participial Preposition
Any Object (DO/ IO/ PO) used as Subject of a passive construction	- *John* was told to close the computer - *Sales order* is submitted to Accounting and Inventory (Step 16)	-> Noun/ Proper Noun
Object Complement (OC)	- Item *being Ordered* (Step 8, 9) - dates *when it can ship*	-> Participle Verbal Phrase -> Relative Clause
Adverbial Modifier (AdvM)	- checks *to determine*... (Step 5) - This is his last sale *this month* - *Looking at the screen*, he saw the new message.	-> Infinitive Verbal Phrase -> Adverbial Phrase -> Clause

2.3 VERBAL PHRASE, PREPOSITIONAL PHRASE, ADJECTIVAL PHRASE, ADVERB AND ADVERB PHRASE IN NATURAL LANGUAGE

Verbal Phrases are built around *non-finite verbs* (or *verbals*), which have a general semantics, without number, person and mood. Verbals cannot act by themselves as predicates in sentences. Instead, they can replace other components of the sentence (subject/ subject complement/ object/ object complement/ adverbial modifier/ noun modifier/ adjective modifier etc, see the examples in Table 5). In English, they have one of the forms:

- (long/ short) *infinitive* (e.g. to ask/ ask),
- *present participle* (or *gerund*) (the form in -ing, e.g. asking) or
- *past participle* (asked).

The present tense of the verbals indicates the simultaneous/ future time and their past tense indicates the past time, all relative to the predicates of the main sentences the verbals are involved in.

Prepositional Phrase is any type of phrase that can be preceded by a (simple/ compound/ participial/ group) *preposition.* It has the general form:

Preposition + {Noun/ Verbal/ Clause/ Adverb/ Adjective Phrase etc}

Prepositional phrase can act as prepositional object (see examples in Table 5), adverbial modifier, noun/ adjective post-modifier, subject complement etc.

Adjectival Phrase has the general form: Pre-Modifier + Adjective + Post-Modifier.

It is built around a qualitative/ relative *adjective*, that can have a simple/ compound/ derivative/ parasynthetic (with both prefix and suffix) form. There are three degrees of comparison for adjectives: positive, comparative and superlative. Alternatives for the modifiers of the adjective are:

Premodifiers ∈ {Adverbs/ Noun phrases/ other Adjectives}. E.g. '*very* expensive items', '*even* more expensive than the other item', '*sales* order', '*light* blue screen' etc.

$$\text{Post-modifiers} \in \left\{\begin{array}{l}\text{'enough' Adverb /}\\ \text{Adjectives (like \textit{possible}, \textit{imaginable} etc) /}\\ \text{Prepositional Phrases /}\\ \textit{As}\text{ + Noun Phrase or Clause (after positives) /}\\ \textit{Than}\text{ + Noun Phrase or Clause (after comparatives) /}\\ \text{Clauses/ Infinitive Phrases}\end{array}\right\}$$

E.g. 'it is good *enough* for this price', 'it is as expensive *as last year*', 'it is less expensive *than last year*' etc.

Adverb and Adverb Phrase can be used instead of an adverbial modifier (verb modifier and determiner at the same time). According to their *form*, the adverbs can be: simple/ compound/ phrase (e.g. *at once*, *at least* etc)/ derivative. According to their *meaning*, the adverbs are: of time or frequency, of place, of manner, of degree, of cause or reason etc.

Adverb phrase is any combination of {Noun, Verbal Phrase, Preposition}, that has the role of adverbial modifier.

2.4 COMPOUND SENTENCE, COMPLEX SENTENCE, CLAUSES IN NATURAL LANGUAGE

Compound Sentences join independent simple sentences by *coordinating conjunctions* (copulative, disjunctive, adversative, resultative, explanatory) or *adverbs* or *asyndetically* (without conjunctions) (e.g. Steps 0, 12, 14) .

Complex Sentences are composed of *dependent* (subordinated) *sentences* (subclauses) correlated to a *main sentence* (clause), e.g. Steps 2, 10, 12. Although the *subclause* is a sentence around a finite verb, it cannot function independently, but only depending on a main clause. The most important types of subclauses are briefly described below.

Noun Subclauses are introduced by *subordinators* like conjunctions, pronouns or adverbs. They can function as subject/ subject complement/ object or object clause (a statement or a question)/ object complement/ apposition. Example of object subclause: "Salesperson thinks that *the customer has money*".

Adverbial Clauses are connected to the main clause by *subordinating conjunctions* that suggest: time, place, cause or reason, result, manner, comparison, concession, purpose, condition etc. Figure 1 contains an adverbial subclause of purpose (Step 2) and an adverbial subclause of time (Step 10).

Relative Clauses are introduced by relative pronouns (who, whose, which, etc) or by relative adverbs (where, when, why etc, instead of 'preposition+ relative pronoun'). Step 12 contains a relative subclause.

Appositive Clauses are introduced by 'that' and refer to (or qualify or identify) nouns in the main clause. They have the general form: Noun + that + Clause. E.g. "Salesperson that *just arrived*".

3. Morphological and Syntactic Categories in Conceptual Modeling. Comparison with NL Patterns

Table 6 and 7 synthesize the morphological and syntactic elements (basic categories or parts) in NL and their counterparts in object and process models. Then, a possible solution to represent these elements in the conceptual modeling is given as well.

TABLE 6

Morphological elements in NL and their counterparts in conceptual modeling

Morphological element in NL	Common element in conceptual modeling
Noun	Object (flat/ composite)
Noun Modifier	Object's Attribute/ Property
Noun Determiner	Object's Type/ Quantifier/ Plural/ Cardinality/ Multiplicity in association
Noun Substitute	Object's Identifier
Verb	Operation (Activity)/ Message (Event-like Operation)
Verb Determiner	Parameter in the Operation/ Message signature
Verb Modifier	Operation's Property / State / (Pre/ Post) Conditions

TABLE 7

Syntactic elements in NL and their counterparts in conceptual modeling

Syntactic role in NL	Common element in conceptual modeling
Predicate	Operation (Activity)/ Message (Event)/ Task/ Procedure
Subject with predicate in active voice (agent role)	Object, sender of the message / Initiator of the operation
Subject with predicate in passive voice (receiver or patient role)	Input parameter in the Operation / Message signature
Subject Complement	Attribute of the Subject-like object
Direct Object	Input parameter in the Operation / Message signature
Indirect Object	Object, receiver of the Operation's result
Prepositional Object	Parameter in the Operation signature
Object Complement	Attribute of the Indirect/ Direct Object
Adverbial Modifier	Operation's Property/ State/ (Pre/ Post) Conditions

From Tables 5, 6 and 7, one may notice that:

- in NL,
    - the integration unit is the simple sentence,
    - the morphological and syntactic categories are *positional* and *context-dependent* because people identify these categories by the word order and the word meaning;
    - the syntactic categories may have different morphological counterparts (see Table 5);
    - the morphological categories (see first column in Table 6) may be heterogeneously represented (see the morphological pattern of the simple sentence in Sect. 2.1).

- in the conceptual modeling,
    - the integration unit is the operation signature,
    - the predicate may have different representations (Operation (Activity)/ Message (Event)/ Task/ Procedure, usually represented by different symbols and in different types of diagrams);
    - the roles of the objects in the operation/ message signature are *positional* and *domain-dependent parameters* and usually are identified at the implementation time (coding phase);
    - the objects may be heterogeneously represented. For instance, the objects' attributes (e.g. 'sales' in Sales Order) can be objects as well (e.g. Report of the sales). They can be represented either as independent objects or as attributes of other objects.
    - the operation' properties, states and pre/ post conditions are also heterogeneously represented and can be found in different diagrams, although they all have the same syntactic role in NL (adverbial modifiers).

In conclusion, the heterogeneity, ambiguities and the domain dependence in NL and, also, in many existing modeling structures, make them inappropriate for the seamless integration of the objects, processes, knowledge, workflows etc, as well as for the model reusability. The study above makes us think of a *meta-model* where:

- the morphological and syntactic units should be *stable* and *general* (e.g similar to the linguistic categories in column 1 in Table 6 and 7) and should have a *simple and uniform representation* for all design perspectives (see Table 1);
- the morphological and syntactic units should be *known in the requirement analysis phase,* i.e. they should be explicitly represented outside the code;
- the objects should be domain independently represented in the code and is better to have a *non-positional place* in the operations' signature. If the code depends only on *domain-independent elements,* it could be further reused for objects/ operations/ knowledge etc from any other domain.

A Modeling Simple Sentence. A possible solution to the abstraction of the NL sentence, in its functional (syntactic) form, is its translation into a *modeling simple sentence* composed of the following elements:

(I)

ACTION *(Verb (predicate) in active or passive voice. It specifies a static or dynamic action/ operation)*
AGNT <Object(s), as agent(s) for the action of the verb> (*Subject(s), if predicate is in active voice)*
PTNT < Object(s), as patient(s) the verb acts upon> *Direct Object(s) or Subject(s) if verb in passive voice)*
RCPT <Object(s), as recipient(s) of the results of the action> *(Indirect Object(s) or Subject(s) if verb is in passive voice)*
preposition-like role <Object(s) participant(s) in the action> ...... *(Prepositional Object(s))*
adverbial role <Object/ Property that qualifies/ modifies the action> *(Adverbial Modifier(s))*

where a *domain independent and non-positional role* is interposed between the verb (action) and each noun (type of object). The elements in the modeling simple sentence (I) could be separated by demarcation symbols like in Sowa (1984, 1988, 1991), ANSI (1995) or Galatescu (1999a, b, 2001). This modeling simple sentence can be interpreted as the *description of the* (static or dynamic) *action* the verb specifies, as well as an operation-like relationship between objects.

This idea is similar to the rules in Entity-Relationship (ER) model where the classes of entities are represented by nouns, the relationships betwen classes are represented by verbs and, when necessary, the ambiguity is removed by the representation of the class role in the relationship.

A similar idea is also used in the description of the frame around a verb, in FrameNet project (Baker (1998)), in CPL (Dignum (1989, 1991)), in KISS (Hoppenbrouwers (1997)), in the modeling languages bases on CGs.

Figure 2 exemplifies the modeling simple sentences of some steps in the reference scenario in Figure 1.

Order_SELECT^{Step0}	**CustInfo_ENTER**^{Step3}	**Customer_CHECK**$^{Step4+5}$	**APPEAR**^{Step13}
AGNT Salesperson *PTNT* Sales Order *LOC* Client_SalesWeb	*AGNT* Salesperson *RCPT* Sales Order *PTNT* Customer	*AGNT* Accounting_system *PTNT* Customer *RCPT* Salesperson *COND* Result:{yes/no}	*AGNT* Sales Order *LOC* Screen

FIGURE 2 Examples of modeling sentences for the Steps 0, 3, 4 and 5, 13 in the reference scenario in Figure 1

In the modeling interfaces based on conceptual graphs (Sowa (1984, 1988, 1991), ANSI (1995), Levreau (1991), Johannesson (1996, 1997), Ambrosio (1997), etc), the syntactic categories are suggested by the *conceptual relations* (or *thematic roles*, similar to the cases proposed in the case grammars by Filmore (1968)).

In Table 8 (see also ANSI (1995), Allen (1987)), some of the most important thematic roles (cases) are enumerated, as a set of generic labels for the conceptual relations 'noun-verb', along with their meaning and linguistic interpretation. These are the roles of the (direct/ indirect/ prepositional) objects relative to the action of the verb (operation), as well as the role of the adverbial modifiers in the operation description.

The meaning and the English synonym in the last two columns in Table 8 could replace the generic roles (column 1 in Table 8) in the user interface, in order to make the user easier understand the modeling sentences and to easier acquire/ give information from/ to him.

TABLE 8

Object-Action roles and the roles of the Adverbial Modifiers in the modeling simple sentences

Role and meaning	**Question it answers**	**English synonym**
AGNT (agent of the action: person/ object that produces the action) **(subject if verb is in active voice)**	who/ what acts?	by
PTNT (patient: object the action operates upon) **(direct object** or subject if verb is in passive voice)	(up)on whom does one act?	(up)on
RCPT (recipient: person or object receiving the	for whom does one act?	for, to

Role and meaning	Question it answers	English synonym
result of the action) **(indirect object** or subject if verb is in passive voice)		
INST (instrument/ procedure to achieve the action)	with/ through/ by what does one act?	with, by, through
RSLT (result of the action)	what does the action result into?	into, to
SRC (source of an action or its initial state)	where does one act from?	from
DEST (destination of an action or its final state)	where does one act to?	to, toward
CHRC (characteristic of the action)	what kind of action?	as, of
DSCR (description of the procedure attached to the action)	how does one act?	like, as
PART (part of another concept meaning a whole)	what is it composed of?	of, out of
STAT (state of the action)	in which state is the action?	in, as
EVNT (event that stimulates the action execution)	what stimulates the action?	when
LOC (location of the action)	where does one act?	at, in, on, over
DIST (distance of the action's impact) + **DISTU** (distance unit)	how far does one act?	up to, as far as, far away
DIR (direction of the action)	which way does one follow?	to, toward
TIME/ BEGN (point in time when the action begins)	since when does one act?	since, at
DUR (duration of the action) + DURU (time unit)	how long does one act?	for, as long as
END (point in time when the action ends)	till when does one act?	till, until, by
REP (repetition/ frequency of the action) + **REPU** (repetition unit)	how often does one act?	times
CAUS (cause of the action)	what causes the action?	because, for, as
GOAL (goal of the action)	for what reason does one act?	in order to, to
QTY (measure/ quantity/ degree/ approximation of the action) + QTYU (quantity unit)	how much/ to what extent does one act?	about, as
COND (pre or post-condition of the action)	what conditions the action's start/ end?	if, while
EXPT (exception for the action)	excepting when does one act?	unless
THME (theme of the action)	what about is the action?	about, on
TENS (tense (position in time: past/ present/ future) of the action. Implicitly is present)	when is the action situated in time?	ago/ now/ then
MOD (modality of the action: ability/ possibility, permission/ concession, prohibition, obligation, necessity, wish/ intention/ determination)	which is the agent's attitude with respect to the action?	can, may/ to be permitted , cannot, must/ have to, should/ need, will/ would/ wish

NOTES

1. A noun in NL can have different surface cases (nominative, accusative, dative) relative to a verb, depending on its position in the sentence and, for certain speech languages, on its form. Instead, an object in a modeling sentence can have only one role relative to the action it participates in (agent, patient, instrument etc). For this reason, *in the modeling sentence*, these *roles* and the objects they label *are non-positional.* Also, these roles are domain and language independent.
2. From the third column of Table 8, one may notice that, in NL, the same word in English may connect nouns with different roles (e.g. the words 'to', 'as', 'on', 'for' etc) and different English words may connect nouns with the same role. Also, in NL, the same word may belong to different linguistic categories (e.g. 'for' corresponding to the role RCPT is preposition and corresponding to the role CAUS is a conjunction). These ambiguities in NL will not be transposed in the conceptual modeling using the generic roles in the first column of Table 8.
3. The tense and modality of the actions/ operations are context dependent, because they may depend on some previous operations or on previous results in a process. For this reason, in modeling, it is better to consider them *procedural caracteristics* of the operations and to represent them in the process description, separately from the operations' generic description. This idea complies with the NL rule, where the verb's tense and modality are represented by auxiliary verbs (see Table 3), that explicitly precede the main verbs of the sentences.

Other roles that we can use in the modeling simple sentences stand for *'noun-noun'* or *'noun-adjective'* relations. Some examples are: CHRC (characteristic), POSS (possessor), DSCR (description), PART, ATTR (attribute), SUBT (subtype) etc. The role POSS helps for representing the possessive (genitive) case of the noun. These roles can be used to represent *object qualification* operations, see Figure 4.

The uniform formalization and the logical interpretation of the general modeling simple sentence in (a) below is expressed in the first order logic formula in (b), where the operation is defined/ described by a λ-expression with $(x_{i1},\ldots,x_{ik})$ (input/ output parameters) as bound variables and $(x_{i1},\ldots,x_{ik}) \subset (x_1,\ldots,x_p)$:

(a) OPERATION role $_1$ $\forall$Object_Type$_1$ role $_2$ $\exists$ Object_Type$_2$ … role $_p$ $\exists$? Object_Type$_p$	**(b)** OPERATION = $\lambda(x_{i1},\ldots,x_{ik})\ (x_{j1},\ldots,x_{j(p-k)})(\forall x_1)(\text{Object\_Type}_1(x_1) \wedge \text{role}_1(x_1)) \supset (\exists x_2) \ldots (\exists x_p)\ (\text{Object\_Type}_2(x_2) \wedge \text{role}_2(x_2) \wedge \ldots \wedge (\text{Object\_Type}_p(x_p) \vee \text{NULL}) \wedge (\text{role}_p(x_p) \vee \text{NULL})) \wedge \text{OPERATION}(x_1,\ldots,x_p)$

$\forall i \in 1{:}p$, role$_i$ is one of the roles in Table 8. The universal quantifier ∀ replaces the indefinite pronouns 'any', ‘all’, ‘every’, 'each' in NL. The two existential quantifiers: ∃, meaning *compulsory existence* ('must exist') and ∃?, meaning *optional existence* ('may exist'), replace the definite or indefinite articles in NL. NULL helps for representing the quantifier ∃?. In addition, each role in Table 8, may have its own logical interpretation.

The *double semantics* of the objects by type and roles has the advantage that the operation can be internally described and processed only depending on the domain independent roles. Their correlation with the domain specific types can be external and dynamically changed.

Besides generic predefined roles (see above), *domain specific roles* might also be used (see Figure 5(a)), with a domain specific interpretation.

4. Semantic and Intersentential Relations in Conceptual Modeling. Comparison with NL Patterns

This section insists on the semantic and intersentential relations abstracted in the object and process models, compared with the corresponding relations in NL. Two kinds of semantic relations are distinguished: inside the lexical categories and between them. Then, a solution to represent these relations by modeling simple sentences of form (I) (in Section 3) is given as well.

4.1 SEMANTIC RELATIONS INSIDE LEXICAL CATEGORIES

The main correlations of the words in the same lexical category in NL are summarized in Table 9, along with their counterparts in modeling. Only the noun and verb categories are taken into consideration. The semantic relations in Table 9 are the most important ones in the WordNet lexicon (Miller (1990)).

TABLE 9

Semantic relations inside categories

Semantic relation in NL	Semantic relation in conceptual modeling
Noun meronymy/ holonymy	Object aggregation or composition/ fragmentation or decomposition
Noun hyponymy/ hypernymy	Object specialization/ generalization
Noun synonymy	Class synonymy (identical structures for classes with different names)
Noun antonymy	Class antonymy (classes with opposite meanings)
Noun homonymy (identical form and different meanings)	Class homonymy (classes with the same name, but different structures)
Verb meronymy/ holonymy	Functional composition/ decomposition
Verb hyponymy/ hypernymy	Functional specialization/ generalization
Verb synonymy	Operation synonymy (identical functions for operations with different types / names)
Verb antonymy	Functional antonymy (operations with opposite functions. Case: two actions that undo each other)
Verb homonymy	Functional homonymy (operations with the same name, but different functions. Case: polimorphism)
Verb troponymy (manner relation)	Operation specialization (by the manner of the action)
Verb entailment	Functional, semantic, temporal entailment
Cause-effect relations between verbs	Relation between an event (cause operation) and the operation whose execution it stimulates

The main conclusions from Table 9 are:

- the semantic relations inside the lexical categories are heterogeneously reflected into the existing conceptual models, usually by different symbols for different categories of relations. Also, usually, only object/ operation meronymy/ holonymy and hyponymy/ hypernymy are represented. The other relations are rarely explicitly represented.

- the meaning of the semantic relations usually results from the human interpretation of the symbols. It cannot be explicitly extracted and uniformly formalized in the respective models.

Generic Operators for the Defininition and Description of the Objects, Actions and of the Relationships between them. A unifying solution for explicitly representing the relations in Table 10 can be their uniform representation by means of generic operators, one operator for each category of relations. In the conceptual modeling, already exist standardized operators such as: part-of, type-of, subtype-of, tuple-of, set-of etc. Unfortunately, they are usually represented by a heterogeneous notation of symbols.

The *description of a generic operator* can be a modeling simple sentence of form (I) (see Section 3). Some examples of generic operators close to the reference scenario in Figure 1 are given in Figure 3 (where the colon (:) separates the name/ type of the class from the individual object and the conditions (restrictions) are not formalized).

(a) Object specialization/ object hyponymy **Object_SUBTYPING** *RCPT* **Smith's Screen** - *subtype* *PTNT* Screen - *supertype* *INST* RESTRICTION: (POSS of Screen is Salesperson: Smith)	**(b) Action specialization/ hyponymy** **Action_SUBTYPING** *RCPT* ACCEPT - *subtype* *PTNT* CHECK - *supertype* *INST* RESTRICTION: (COND of CHECK is ‘yes’)
(c) Definition of a holonic object/ object holonymy **Object_COMPOSITION** *RCPT* **On_Action_Constraint** - *whole (abstract) object* *PART1* PRECondition - *part objects* *PART2* POSTCondition *PART3* TransitionCondition	**(d) Temporal entailment of an action** **Action_ENTAILMENT** *PTNT* ACCEPT *(ACCEPT entails the previous* *RCPT* CHECK *CHECK operation)*

FIGURE 3 Examples of generic operators for object and action correlation

The operators may be considered:

- *generic*, if they have a generic function and their instances have the same number and type of roles. This is the case of the operators: Object_SUBTYPING, Action_SUBTYPING, Action_ENTAILMENT in Figure 3.
- *partly generic,* when the number or type of the roles may differ in different instances of the operator and only its function remains generic. This is the case of the operator Object_COMPOSITION in Figure 3, that can have a different number of parts in different instances; and also the Object/ Action_QUALIFICATION operator in Figure 4, that can involve characteristics of different types in different instances.

Whatever is the case, the code for processing these operators will be domain independent, because it will depend only on the type of the operator, on the generic roles of the objects involved in that operator and on their general and uniform logical interpretation.

For example, each object ‘x’ defined by the generic definition operator, denoted by Object_DEFINITION, can be expressed in predicate calculus by a λ-expression, where ‘x’ is a bound variable:

$$\begin{aligned}\text{Defined\_Object\_TYPE} = (\lambda x)\,(x_1,\ldots,x_n)\,(\forall x)(\ &\text{Defined\_Object\_TYPE}\,(x) \wedge \text{RCPT}(x)\ ) \supset \\ &(\exists\, x_1) \ldots (\exists x_n)\,(\text{Object\_Type}_1(x_1) \wedge \text{role}_1(x_1) \wedge / \vee \ldots \wedge / \vee \\ &(\text{Object\_Type}_n\,(x_n) \vee \text{NULL}) \wedge (\text{role}_n\,(x_n) \vee \text{NULL})) \wedge \text{Object\_DEFINITION}(x, x_1,\ldots,x_n)\end{aligned}$$

4.2 RELATIONS BETWEEN LEXICAL CATEGORIES

The relations between the lexical categories are summarized in Table 10. The first column enumerates the main inter-category dependences in NL. The relations involving the complementary lexical categories as *pronouns* (noun substitutes), *verbals* (non-finite verbs), *articles* (noun determiners), *clauses* (subordinated sentences) are considered as well.

The third column in Table 10 shows how the the inter-category relations in NL are abstracted in the conceptual modeling. The main conclusion is that, in the conceptual modeling, the knowledge about these relations is only partly represented explicitly in diagrams, the most of it is encoded and can hardly be extracted from the programming code (even with modern software reengineering tools)

The main idea that arises from the second column in Table 10 is that, the inter-category relations in NL, with few exceptions (the noun-verb relation and the anaphoric reference), can be seen as generic operators that could be uniformly transposed into the conceptual modeling. This column reveals the main types of generic operators that describe the semantic relations inside and across sentences. These operators can be represented in the form (I) (see Section 3 and 4.1).

TABLE 10

Relations between the basic lexical categories

Dependent categories in NL	Type of relation in NL	Relation in conceptual modeling
1. Noun -- Verb	Case (Role) of the noun relative to verb’s action (see Table 8)	Object -- Operation (Position and formal type of the object in the operation signature)
2. Noun -- Adjective	Noun Qualification	Object -- Object Property
3. Common Noun -- Proper noun	Instantiation of the common noun	Class -- Object
4. Subunits of the compound Noun (noun + noun / adjective + noun / adjective + verb / verb + noun / adverb + verb / gerund + noun etc)	- Definition / Identification of the compound noun - Noun Subtyping (specialization)	- Compound name of the Class and Class structure (intension) - Name of the Subclass and Class specialization relation
5. Noun -- Indefinite Pronoun (all/every/each/no/any/neither, none..)	Universal / negation Quantification of noun	Object -- Object Quantifier/ Cardinality/ Multiplicity
6. Noun -- Definite/ indefinite Article	- Existential Quantification of the noun - Anaphoric reference	Object -- Object Quantifier/ Cardinality/ Multiplicity - Object lifeline
7. Verb -- adverb	Verb Description / Qualification	Operation -- Operation Property
8. Subunits of the multiword Verb (verb + adverbial particle / verb + preposition / verb + adverbial particle + preposition)	- Verb Definition/ Identification - Verb Subtyping (specialization)	- Compound name of the Operation and Operation description (param.) - Name of the operation Subtype and Operation specialization relation
9. Noun -- Pronoun	- Noun Substitution - Anaphoric Reference	Object -- Object ID
10. Noun -- Verbal	Noun Qualification	Object -- Object Property
11. Adjective -- Adverb	-- Derivation..	No relation. Because it has different qualification targets, it is only a lexical relation
12. Noun -- Noun	- Noun Qualification *or* - Noun Synonymy/ Antonymy/ Hyponymy/ Hypernymy/ Meronymy/ Holonymy/ Homonymy	Object -- Object. This relation can be: (1) a message, (2) object intension (correlates an object to its attributes), (3) object synonymy/ antonymy etc (see Table 9)
13. Adjective -- Adjective	Adjective Synonymy/ Antonymy	Attribute -- Attribute (reference)
14. Verb -- Verb	-Verb Correlation/ Verb Control/ Verb Modality/ Verb Tense (verb-auxiliary verb) -Verb Hyponymy/ Hypernymy/ Synonymy etc (Table 9) - Verb semantic Decomposition	- Functional relations between operations in activity diagrams (control and decision statements) - Operation Hyponymy /Hypernymy etc (Table 9) - Process semantic decomposition
15. Main Clause -- -- subordinated Clause	It means the dependence 'Main verb – Subordinated verb' (see 14)	- Control and decision statements - Conditions in a transaction

Using generic operators, these relations can be uniformly represented, either inside or between modeling simple sentences. Some of these relations have been represented in Figure 2 and 3 by generic operators of form (I) (see Section 3). Figure 4 gives two examples of generic operators for object and action qualification.

(a) Noun qualification	**(b) Verb qualification**
Object_QUALIFICATION	**Action_QUALIFICATION**
PTNT Screen *Screen qualification with its possessor*	*PTNT* APPEAR
POSS Salesperson	*CHRC* completely:{yes/ no}

FIGURE 4 Examples of generic operators for object and action qualification

Table 10 also reveals (1) other elements that should complete the modeling simple sentence of form (I), such as: quantifiers, (anaphoric) references, cardinality, and (2) other types of operators, like: *definition / identification operators* of the objects, actions or verbals (non-executable actions) (see examples in Figure 5).

The (partly) generic operator :

Object_QUALIFICATION
PTNT active_object_type
ROLE attributive_object_type

can have the logical interpretation:

$$(\exists a)\ (\text{attributive\_object\_type}\ (a)) \supset (\exists o)(\text{active\_object\_type}\ (o))\ (\text{PTNT}(o) \wedge \text{ROLE}\ (a)) \wedge \text{Object\_QUALIFICATION}\ (o, a)$$

where any *attributive object* 'a' necessarily qualifies an *active object* 'o' (that directly participates in an operation). However, the attributive object can become active object in other circumstances.

(a) Object definition/ identification	**(b) Action definition/ identification** (generic form)
Object_DEFINITION *-- definition of Person* *RCPT* Person *NAME* Person Name *SSN* Social Security Number *SEX* Sex:{M/F} *LOC* Department: {Accounting/ Sales/ Marketing/ Admin/ Production/ Personnel ...}	**Action_DEFINITION** *RCPT* ActionID *- generic definition of an action* *NAME* ActionName *TYPE* ActionType:{atomic/ subprocess/ process} *LOC* ProcessID
(c) Step 8 in Figure 2-1- Verbal usage	**(d) Verbal definition**
Item_ENTER *AGNT* Salesperson *PTNT* Item *RCPT* Sales Order *reference to the verbal identifier* *COND* Item_ORDER *LOC* Sales Order_Form	**VERBAL_DEFINITION** *RCPT* **Item_ORDER** *--Item_ORDER is not executable. It only associates (qualifies) the item with the customer that has ordered it and with the order date* *PTNT* Item *AGNT* Customer: Smith *TIME* Order Date

FIGURE 5 Examples of generic operators for object and action (executable or not) definition

4.3 INTERSENTENTIAL RELATIONS

The most important intersentential relations in NL are inside compound or complex sentences or inside paragraphs or sections. Practically, they are *relations between the actions of the verbs* in simple (dependent or independent) sentences. Table 11 summarizes the types of intersentential relations in NL and the corresponding ones in the conceptual modeling.

TABLE 11
Intersentential relations in NL and in the conceptual modeling

Intersentential relations in NL	**Intersentential relations in the conceptual modeling**
Compound sentence with (copulative, disjunctive, adversative, resultative, explanatory) conjunctions between sentences	Diagram with: - sequential operations - AND/ OR join/ split of operations
Complex sentence with a main clause and subclauses (noun/ adverbial/ relative/ appositive subclause)	Transaction module with: - control statements - decision conditions - actions modality (obligation, permission, necessity)
Paragraph with anaphoric/ generic reference between sentences	Sub-process with data/ object/ object identifier transfer between transactions
Section with inter-paragraph reference	Process with subprocess control

Some remarks on Table 11 are:

- the compound sentences in NL correlate independent operations, implicitly in a chronological sequential order. The parallel operations must be explicitly stated, e.g. by the expression, '*A* should happen at the same time with *B*'. In modeling, distinct symbols separate the sequential execution from the parallel one, where join or split are represented.
- in modeling, the intersentential relations are superficially represented in the analysis/ design diagrams. They are detailed just in the transaction code.
- the transfer of information is usually known at the implementation time, when the actual parameters are explicitly associated with the operation's formal signature (see, for example, the correlation between the class diagram and the activity diagram in UML). The main idea in this section is that, the designers should be forced to think of the information flows between objects/ operations/ processes, at the

conceptual level, in order to correctly define and describe them and refer to them, before the system implementation.

In NL, the correlation between two verbs in two different *simple sentences* are usually established by anaphoric or generic references introduced by the definite article or by a pronoun in the second sentence. In the conceptual graph formalism, they are represented by coreferences that correlate the coreferent concepts in two or several graphs. The most important intersentential correlations in the NL *compound and complex sentences* will be considered below and transposed to the conceptual modeling.

Explicit Representation of the Intersentential Relations. Compound and Complex Sentences in Conceptual Modeling. In ANSI (1995), the starter set of conceptual relations for the conceptual graphs includes intersentential relations as complex connections expressed by adverbs and conjunctions that relate clauses, sentences, paragraphs or contexts (sets of related propositions). Such relations are: BFOR (before), AFTR (after), CAUS (cause), PURP (purpose), CNSQ (consequence), METH (method), AND, OR. This idea has been further developed in Galatescu (2001), where the operations (and implicitly, the modeling simple sentences that define/ describe them) are explicitly correlated by intersentential relations, as navigational links, and controlled by conditions. The correlated operations (actions) compose a *subprocess-like sentence* (hierarchical diagram).

The most important of the proposed intersentential relations are summarized in Table 12, together with their logical interpretation. Each relation may be associated with a *constraining inference rule*, whose *premise* is the last activated operation (denoted by PO) and, implicitly, all the previously activated operations and whose *conclusion* is the conjunction (or disjunction) of the operations proposed for the process continuation and denoted by O, $O_{i1}, \dots, O_{in}$.

In order to represent the mapping rules of the intersentential relations into predicate calculus, the following notations will be used: 1) *x, y* are individuals of output object types of PO operation, transferred to its child operations, 2) $\supset$ means the compulsory activation of the implied operation, 3) $\wedge$ means the conjunction of the operations or rules, 4) $\vee$ means the disjunction (exclusive or not) of operations or rules, 5) ==> means the transformation into predicate calculus of an intersentential relation, 6) NULL is a null (ineffective) operation.

The transfer (or sharing) of at least an output object of the operation in the premise to the operations in the conclusion of the inference rule is implicitly supposed. It justifies the logical correlation between operations by, at least, a common subject/ object.

TABLE 12

Types and logic of the intersentential relations (relations between operations/ actions in a (sub)process)

Intersentential relation	Logical interpretation in predicate calculus
Operation Modality	
PO **MUST** $Oi_1, \dots, Oi_n$	== > $(\forall x)((PO(x) \supset Oi_1(x)) \wedge \dots \wedge (PO(x) \supset Oi_n(x)))$
PO DO O	== > $(\forall x)(PO(x) \supset O(x))$ -- Imperative execution
PO **MAY** $Oi_1, \dots, Oi_n$	== > $(\forall x)((PO(x) \supset (Oi_1(x) \vee NULL)) \wedge \dots \wedge (PO(x) \supset (Oi_n(x) \vee NULL)))$
PO (MAY $\wedge$ condition) $Oi_1, \dots, Oi_n$	==> $(\forall x)(((PO(x) \wedge condition) \supset (Oi_1(x) \vee NULL)) \wedge \dots \wedge$ $((PO(x) \wedge condition) \supset (Oi_n(x) \vee NULL)))$
Operation Specialization	
PO **SPEC** $Oi_1, \dots, Oi_n$	== > $(\neg PO) \wedge (\exists x)(Oi_1(x) \text{ XOR } \dots \text{XOR } Oi_n(x))$
Sequential Execution	
PO **THEN** / **BFOR** O	== > $(\forall x)((PO(x) \supset O(x)) \wedge \neg(O(x) \supset PO(x)))$
PO **AFTR** O	== > $(\forall x)(\neg(PO(x) \supset O(x)) \wedge (O(x) \supset PO(x)))$
Alternative Execution	
I F condition **THEN** O_1/ **ELSE** O_2	==>$(\exists x)((condition(x) \supset O_1(x)) \wedge (\neg condition(x) \supset O_2(x)))$
PO **CASE** $Oi_1, \dots, Oi_n$	== > $(\forall x)(PO(x) \supset Oi_1(x) \vee Oi_2(x) \vee \dots \vee Oi_n(x))$
PO (CASE $\wedge$ condition_value) $Oi_1 \dots, Oi_n$	==>$(\forall x)((PO(x) \wedge condition\_value_1) \supset Oi_1(x) \vee$ $(PO(x) \wedge condition\_value_2) \supset Oi_2(x) \vee$ $\dots \vee (PO(x) \wedge condition\_value_n) \supset Oi_n(x))$
Iterative Execution	
WHILE condition DO O	==>$(\exists x)((condition(x) \supset O(x)) \wedge (\neg condition(x) \supset NULL))$
PO **MUST REPEAT**	== > $(\forall x)(\exists y)(PO(x) \supset PO(y))$. - To avoid the infinite loops, NULL will be instead of 'y' when the repetition ends. This is a declarative end.
PO **MAY REPEAT**	== > $(\forall x)(\exists y)(PO(x) \supset (PO(y) \vee NULL))$ - This is a procedural end of the repetition that stops before PO expansion/ start.

Intersentential relation	Logical interpretation in predicate calculus
Logical Relations between Operations	
PO **AND** O	$==> (\forall x) ((PO (x) \supset O (x)) \wedge (O (x) \supset PO (x)))$ (sequential AND)
PO **OR** O	$==> (\forall x) ((PO (x) \supset (O (x) \vee NULL)) \wedge (O (x) \supset (PO (x) \vee NULL)))$
PO **XOR** O	$==> (\forall x) ((PO (x) \supset \neg O (x)) \wedge (O (x) \supset \neg PO (x)))$
NOT O	$==> (\forall x) (\neg O (x))$ - Execution negation. ‘x’ an input concept of O
Grouped Execution PO **GROUP** $Oi_1, \ldots, Oi_n$, (with compulsory $Oi_1, \ldots, Oi_k$ and optional $Oi_{k+1}, \ldots, Oi_n$)	$==> (\forall x) (PO(x) \supset (Oi_1 (x) \wedge \ldots \wedge Oi_k (x) \wedge (Oi_{k+1}(x) \vee NULL) \wedge \ldots \wedge (Oi_n (x) \vee NULL)))$
Operation Description PO **DSCR** DO	$==> (\forall x) (PO (x) \supset (\exists d) DSCR(d))$, where $DSCR(d) = (\lambda d)$ λ-definition(DO)[d], ‘d’ is an instance of PO’s description, defined by another operation DO.
Operation Motivation PO **GOAL** O	$==> (\forall x) ((PO (x) \supset O (x)) \wedge (\neg O (x) \supset \neg PO (x)))$
Operation Stimulation/ Cause EO **EVNT** / **CAUS** O	$==> (\forall x) ((\exists e) EVNT (e) \supset O(x))$, where $EVNT(e) = (\lambda e)$ λ-definition(EO)[e] ‘e’ is a particular event/ cause and ‘λ-definition (EO)’ is the λ-expansion of the event/ cause operation EO.
Operation Starting time_condition **TIME** O	$==> (\forall x) ((\exists t) (time\_condition (t) \supset O(x)))$ ‘t’ is a certain point in time when the time_condition becomes true.

If the correlating relations are AND, GROUP, REPEAT, OR, CAUS/ FOR, BUT, NOT etc, the subprocess-like sentences correspond to the *compound sentences* in NL (see Figure 6 (a) for the sentence "Sales person turns on the laptop, brings up the program and chooses Report Sales Order").

When the modeling sentences are linked by subordinating relations like IF-THEN-ELSE, DSCR, TIME, GOAL, EVNT, DO, WHILE, subordinating CAUS, MUST, THEN, CASE, SPEC etc, the subprocess-like sentences correspond to the *complex sentences* in NL. These relations are preceded in Figure 6(b) by the symbol ⊃. The subordinated operations necessarily belong to the same complex sentence as the subordinating ones.

In Figure 6(a), two ways are given for representing the compound sentence in Step 0 of the reference scenario: (1) using GROUP relation and (2) using consecutive (sequential) AND relations between the component actions TURN_ON, BRING_UP and CHOOSE (and implicitly, between the modeling sentences that describe them).

Figure 6(b) represents a possible correlation (in a complex sentence) of the sentences that describe the Steps 0, 1, 2, 3, 4, 5 in the reference scenario (Figure 1), using some of the subordination relations mentioned above.

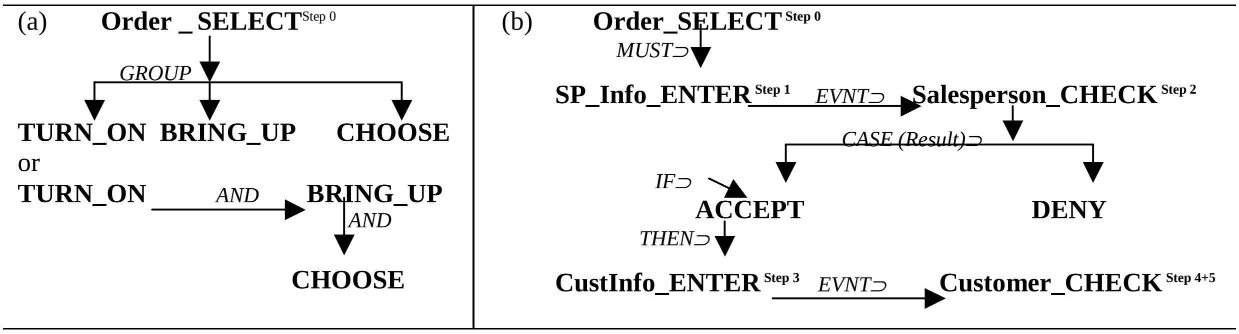

FIGURE 6 Examples of *intersentential relations* (in italic) in compound (a) and complex (b) modeling sentences. Each operation is described by a modeling simple sentence (see examples in Figure 2)

The compound and complex sentences in Figure 6 have been represented graphically. They could also be represented in a linear notation, similarly to a pseudocode.

Advantages of the Simple, Compound and Complex Sentences in Conceptual Modeling. The most important advantages from the sentences proposed above are:

- The activities and processes can be detailed and explicitly declared, integrated and stored outside the code. Hence, they can be reused and customized very easy and, relying on their logic, they can be processed using a general and domain-independent algorithm.
- Using explicit relations between the operations of a process, in earlier phases of the system development. It has the advantage that the designer can think of and easier change the mandatory operations (e.g. preceded by MUST), the operations that will be implicitly executed (e.g. if preceded by

EVNT), the alternative operations (if precede by CASE), the conditional operations (e.g. if preceded by IF) etc, that usually are devolved to the programmer.

- The evaluation of the object and process models, from the logical and semantic viewpoints, in early phases of the system development. As mentioned above, the logic of the simple/ compound/ complex modeling sentences (expressed in first or higher-order logic) derives from (1) the logic of each (predefined or user-defined) operator that governs the sentence, (2) the logic of the intersentential relations, (3) the domain independent semantics and logic of the roles the concepts have in the operation execution. The logical support might further be implemented into a process coordination mechanism.
- The uniform representation of the most important aspects in NL and real world, with benefits in at least two directions: the system automation and the user interface.
- The natural implementation of these sentences in relational databases.

A deeper and exemplified comparison between the unifying abilities of these sentences and the symbolic notation in UML diagrams is in Galatescu (1999a, b). Also, a detailed presentation of the unifying transformation of the NL patterns to the conceptual models by means of these sentences is in Galatescu (2001).

These sentences have been previously used for multidatabase modeling and have been implemented in ORACLE*Forms. At present, they are going to be used in a multi-agent system for assistance in Business Process Reengineering (BPR) sessions. They will help for the uniform representation and integration of the domain-specific and BPR-specific objects and processes with the agents' communication acts. In this application, the compound and complex sentences will be implemented as dynamic scenarios for guiding the users through the BPR methodology (see Galatescu & Greceanu (2002a, b)).

5. Conclusions

The paper is a study on NL with respect to its modeling and model integration abilities. This study is motivated by the need for the integration of the object, process, data flow, workflow, knowledge models in the early phases of the system development. The disadvantages of the symbolic notation for the model integration have been motivated by a brief analysis of UML with respect to the main concepts and relationships in the real world.

The first goal of the paper was to summarize and remind to the modelers the main features of NL and to establish their counterparts in the conceptual modeling. The comparison NL versus conceptual modeling has relied on four steps in human understanding of NL. These steps have directed the study toward a possible integration of the models by morphological and syntactic categories and patterns and by semantic and intersentential relations.

The second goal of the paper was to sketch a possible solution to the integration of the conceptual models, inspired by the conceptual graph formalism and relying on modeling simple, compound and complex sentences. The claimed motivation for this solution is that the translation of the NL patterns to OO modeling should be extended to the other perspectives in information system engineering (see Table 1 (b-c)).

References

ALLEN J. (1987, 1995). Natural Language Understanding, Benjamin/Cummings Publ. Comp., USA

AMBROSIO A.P., METAIS E., MEUNIER J.-N. (1997). The linguistic level: Contribution for conceptual design, view integration, reuse and documentation, Data&Knowledge Engineering, Vol. 21(2)

ANSI (1995). IRDS Conceptual Schema. Part 1: Conceptual Schema for IRDS. Part 2: Modeling Language Analysis, ANSI Report X3H4/93-196

BAKER C., FILMORE C. (1998). The Berkeley FrameNet Project. Intl. Computer Science Institute, USA

BOUZEGHOUB M. (Ed)(1997). Natural Language for Databases, Data&Knowledge Eng., Vol. 21(2)

BUDAI L. (1994). English Grammar, Budapest, 1994

BURG J., RIET R. P. (1995). COLOR-X Object modeling profits from linguistic. In Towards Very Large Knowledge Bases: Knowledge Building & Knowledge Sharing, IOS Press

BURG J., RIET R.P (1997). The impact of linguistics on conceptual models: consistency and understandability. Data&Knowledge Engineering, Vol. 21(2)

DIGNUM F. (1989). A Language for Modelling Knowledge Bases. Based on Linguistics. Founded in Logic. PhD thesis, Vrije University, Amsterdam

DIGNUM F., RIET R.P. (1991). Knowledge Base Modelling Based on Linguistics and Founded in Logic, Data&Knowledge Engineering, Vol. 7

DIK S.C. (1989). The Theory of Functional Grammar. Part I: The structure of the Clause. Foris, Dordrecht

DIK S.C. (1990). Some Developments in Functional Grammar: Predicate Formation. In Contemporary Dutch Linguistics

FILMORE C.-J. (1968). The case for case. In Bach. E, Harms R. (Eds): Universals in Linguistic Theory: Holt Rinehart and Winston, New York

FLIEDL G., WINKLER C. (1999). Natural Language based requirements engineering - The NTMS approach. Conf. on Software and System Engineering and their Applications (ICSSEN'99), Paris

GALATESCU A. (1999a) On a Unifying Notation of UML Diagrams Based on Natural Language Paradigm. In vol. ‘Managing Information Technology Resources in Organizations in the Next Millenium’, Idea Group Publishing, USA

GALATESCU A. (1999b). Graphical NL Versus Symbolic Notation in Object and Process Modeling. Conf. on Applications of NL to Information Systems (NLDB'99), Klagenfurt, Austria

GALATESCU A. (2001). A Unifyng Translation of Natural Language Patterns To Object and Process Modeling, In vol. 'Information Modeling in the New Millenium', Idea Group Publishing, USA

GALATESCU A., GRECEANU T. (2002a). Ontologies Supporting Business Process reengineering. Accepted for publication in Proc. of 4th Conf. on Enterprise Information Systems, April, Ciudad Real, Spain

GALATESCU A., GRECEANU T. (2002b). Using Ontologies in Virtual Branstorming for Business Process reengineering. Accepted for publication in Proc. of 3rd IFIP Working Conf. on Infrastructures for Vortual Enterprises, May, Sesimbra, Portugal

GUARINO, N., MASOLO, C. AND VETERE, G. (1999). OntoSeek: Content-Based Access to the Web. In IEEE Intelligent Systems, 14(3)

HARMON P., WATSON M. (1997). Understanding UML: The Developer’s Guide. With a Web-based Application in Java: Morgan Kaufmann Publishers

HOPPENBROUWERS J., VOS B, HOPPENBROUWERS S. (1997). NL Structures and Conceptual Modelling: Grammalizing for KISS, Special Issue of Data&Knowledege Engineering, Amsterdam

IDEF Users Group Annual Conference

JOHANNESSON P., BOMAN M., BUBENKO J. et al. (1996). Conceptual Modeling: Prentice-Hall

JOHANNESSON P. (1997). Supporting schema integration by linguistic instruments. Data&Knowledge Engineering, Vol. 21(2)

JOHANNESSON P. (2001). A Language/action based Approacg to Information Modeling. In vol. Information Modeling in the New Millenium, Idea Group Publishing Press, USA

LAWRENCE P. (Ed) (1997) Workflows Handbook, John Wiley&Sons

LEVREAU G., MEUNIER J., BOUZEGHOUB M., METAIS E. (1991). Definition d’une Interface Language Naturel pour la Conception de Bases de Donnees. Rap. tech. MASI91.45, Inst. Pascal, Paris

LEWERENZ J. (1999). On the Use of Natural Language Concepts for the Conceptual Modeling of Interaction in Information Systems. Conf. on Applications of NL to ISs (NLDB'99), Klagenfurt, Austria

LUKOSE D. (1996) MODEL-ECS: Executable Conceptual Modelling Language. Knowledge Acquisition Workshop (KAW96), Banff, Canada, 1996

MEDINA-MORA R., WINOGRAD R., FLORES R. AND FLORES F. (1992). The ActionWorkflow Approach to Workflow Management Technology. Conf. on Computer Supported Cooperative Work, New York

MILLER G., BECKWITH R., FELLBAUM C., GROSS D., MILLER K. (1990) Introduction to WordNet: An On-line Lexical Database. Intl. Journal of Lexicography 3(4)

MORENO A.M., RIET R.P. (1997). Justification of the Equivalence between Linguistic and Conceptual Patterns for the Object Model. Conf. on Applications of NL to Information Systems, Vancouver, Canada

MORENO A.M., JURISTO N., RIET R.P. (2000). Formal Justification in Object-Oriented Modeling: A Linguistic Approach. Data&Knowledge Engineering, Vol. 33(1)

OMG (1997). UML (Unified Modeling Language) Notation Guide v1.1

REIJSWOUD V., MULDER H. (1997). Speech Act based Communication and Information Modeling with Demo. Proc of Ws. on Evaluation of Modeling Methods in System Analysis and Design, Barcelona

RIET R., Burg H., Dehne F. (1998). Linguistic Instruments in Information System Design. Conf. on Formal Ontology in Information Systems, Trento, Italy (IOS Press)

RUMBAUGH J., BLAHA M. et al. (1991). Object-Oriented Modeling and Design. Prentice-Hall

SAG I., WASOW T. (1999) Syntactic Theory: A Formal Introduction. Stanford Univ., Center for the Study of Language and Information (http://www-csli.stanford.edu/csli/projects/cogsci9495-hpsg.html)

SCHREIBER G., WIELINGA B., BREUKER J. (1993) A Principled Approach to Knowledge-based System Development, Academic Press London

SOWA J.F. (1984). Conceptual Structures. Information Processing in Mind and Machine, Addison Wesley

SOWA J.F. (1988). Lexical Structures and Conceptual Structures. Semantics in the Lexicon, Kluwer Academic Publishers

SOWA, J.-F. (1991). Toward the expressive power of the natural language. Principles of Semantic Networks. Explorations in the Knowledge Representation: Morgan Kaufmann Publ.

STEUTEN A., RIET R., DIETZ J. (1999). Linguistically Based Conceptual Modeling of Business Communication, Conf. on Applications of NL to ISs (NLDB'99), Klagenfurt, Austria (and Data&Knowledge Engineering, 35-2)

THALHEIM B., DUSTERHOFT A. (1999). Metaphor Development for Internet Sites. Conf. on Applications of Natural Language to Information Systems (NLDB'99), Klagenfurt, Austria

VOS B., GULLA A., RIET R. (1997). Verification of conceptual models based on linguistic knowledge. Data&Knowledge Engineering, Vol. 21(2)

WEIGAND, H.(1989). Linguistically Motivated Principles of Knowledge Base Systems. PhD thesis, Vrije University, Amsterdam