Conceptual Modeling and Natural Language Analysis

Colette Rolland

Abstract The CAiSE'92 paper presented a tool called OICSI that used Natural Language Processing (NLP) techniques to support both the generation of an Information System (IS) conceptual schema from textual requirements and in the reverse way, schema paraphrasing to ease schema understanding and evaluation by stakeholders. Both topics have been of interest during the next 20 years among other new usages of NLP techniques in the context IS development. For sake of space, this paper concentrates on an overview of NLP techniques used as elicitation techniques.

1 The Initial Paper and Related Works

The initial paper was written at a stage of IS engineering maturity at which it was clear that an IS represents some excerpt of the World and that IS engineering shall focus on modeling the concepts of the world on which IS users need information. As a consequence of this assumption, a number of conceptual modeling languages were developed such as [1–4] to name a few. It was also becoming clear that these languages were not understandable by people other than modelers. This evidence raised the issue of how to master the creation of a conceptual model (called schema at that time) as long as the process implies exchanges between domain experts & stakeholders (who know their wishes, needs and requirements) on one hand, and modelers (who master conceptual languages) on the other hand. For the same reason, validation was also an issue as stakeholders can hardly validate whether the conceptual model really reflects their needs. As long as natural language is used during this conceptualization process, the end-users and all stakeholders participating in the IS project have a chance to be involved. Then, the idea to support

C. Rolland (✉)
Université Paris1 Panthéon Sorbonne, 90, rue de Tolbiac, 75013 Paris, France
e-mail: Rolland@univ-paris1.fr

J. Bubenko et al. (eds.), *Seminal Contributions to Information Systems Engineering*,
DOI 10.1007/978-3-642-36926-1_4,

NL communication during both phases of model creation and model validation came in mind.

OICSI was developed as a CASE tool that supports conceptual model generation from requirements statements expressed in NL and helps in the validation of this model by paraphrasing it, i.e. reformulating its semantic content as natural language sentences. The generation phase is based on the adaptation of the Fillmore's 'Case to Case' theory [5] whereas the paraphrasing phase uses Chomsky's approach [6].

Whereas OICSI partly automates the generation of a conceptual model, some early attempts to support this task were based on manual guidelines [4, 7–9]. The Functional Grammar [10] was preferred to the Case Grammar used in OICSI in [11] and [12]. Some other few approaches of the same period presented tools, which used parsers to extract model elements from NL sentences [13, 14].

From these beginnings, the ways NLP techniques have been used in IS development and particularly during requirements engineering and conceptual modeling are manifold. The analysis of literature suggests to organize them according to four strategies, which (a) support the generation of models from NL input texts, (b) support model paraphrasing, (c) help in the general understanding of NL input texts and (d) Improve NL texts quality. For space restriction this paper only comments on point (a).

2 Generation of Models from NL Input Texts

Typically, these approaches take as input a document expressed in full or structured Natural Language (most of the time in English) and generate model elements of a given conceptual model formalism. They can be further classified into techniques focusing on (a) structural (static) aspects of an IS conceptual model (e.g. entity-relationship diagrams) or (b) behavioral aspects (e.g. uses cases & scenarios). Some approaches (c) deal with the generation of other types of models such business rule models, ontologies or traceability models or with different activities such as compliance with regulation documents.

Like in OICSI, many researchers have used NLP techniques to generate structured or formal models from requirements documents expressed in NL. NL-OOPS [15] is a Case tool that supports requirements analysis and generates object-oriented models from NL requirements documents. It uses the LOLITA NLP toolkit that linguistically analyze texts. CICO [16] transforms tagged requirements statements into various forms of structured models based on rules. Moreno [17] has developed a method that transforms NL requirement statements into an object model. Her method is based on a grammatical analysis of requirement statements, and initially transforms them into a restricted form of NL. Subsequently, the restricted statements are transformed into object structures; this is based on patterns that transform linguistic structures into conceptual structures. The approach was further developed in [18, 19]. The authors of [20] describe an approach that uses part-of-speech tagging and morphological analysis for the generation of candidate elements of a

class diagram. Additionally, an ontology is used to refine the candidates according to the specificity of the real world domain. Finally, classes that do not appear in relationships and relationships which do not involved at least two classes are deleted. In [21] controlled NL is used to express requirements that are automatically transformed into formal specifications.

Approaches concentrating on the extraction of modeling concepts for dynamic models can be found [22–26]. COLOR-X [22] supports non-automatic construction of formal events languages from lists of events described in natural language). In [23] we used a linguistic pattern approach inspired from OICSI to transform a textual scenario of a use case into a so-called conceptualized scenario. The process includes a parsing of the full NL scenario text to identify linguistic structures that allow the identification of linguistic semantic patterns, which in turn, are mapped to scenario conceptual elements. The conceptualized scenario serves as a basis to reason about missing requirements and to suggesting additions in the requirements specification [27]. Extraction of use cases and scenarios are also described in [25, 26]. Vice-versa the approach presented in [24] uses use-cases as inputs to generate behavior specifications. In the LIDA approach [28] candidate objects, attributes and methods as well are extracted from textual requirement statements.

NLP techniques have been also used to generate models other than the typical IS conceptual models. For example, in the area of *ontology engineering*, the Text2Onto approach [29] uses machine-learning techniques together with linguistic processing in order to derive an ontology from a text. A lightweight NLP is used in [30] to automatically generate and maintain *traceability relations* between different types of software requirements artifacts. In the BROCOM approach [31] the targeted output are *business rules*. In the SMART approach [32] NLP techniques are used to automate the generation of a *business process model* from textual requirements. *Requirements specifications* can also be the target of a transformation process, which uses textual requirements as inputs and generates a precise list of requirements expressed in some controlled NL [33]. In the area of *compliance with regulations* [34] presents a technique to check the compliance of requirements with regulations while eliciting requirements. The approach checks compliance by trying to match a newly discovered requirement to regulations represented by combinations of case frames resulting from the Case Grammar technique. In [35] we used the Case Grammar to define a Goal template and to develop a tool supporting *a controlled formulation of a goal*. A similar linguistic approach to goal formulation was used in [36] to reason about *variability in requirements*. The approach considers the goal linguistic frame elements as variability concerns.

To conclude it seems that NLP techniques remain useful in conceptual modeling but only occasionally. The reason might be that requirements/concepts elicitation is part of a decision process that cannot be automated from analysis of NL texts.

Acknowledgements The original paper was co-authored by Christophe Proix, who was my doctoral student at that time. Christophe deserves an equal share of the credit for the work that we have accomplished. Unfortunately, I was unable to contact Christophe for the purpose of this article.

References

1. Chen, P. P. S. (1976) The entity relationship model: toward a unified view. In: ACM Trans. on Data Base Systems (TODS), 1(1), pp. 9–38. ACM
2. Dubois, E., Hagelstein, J., and Rifaut, A. (1989) Formal requirements engineering with ERAE. In: Philips Journal of Research, 43(3). Philips
3. Rolland, C., and Richard, C. (1982) The Remora methodology for information systems design and management. In: Proc. Int'l Conf. on Comparative Review of Information Systems Methodologies. CRIS. IFIP WG8.1. North Holland, 1982
4. Nijssen, G.M, Halpin, T.A. (1989) Conceptual schema and relational database design: a fact oriented approach. Prentice-Hall, Englewood Cliffs, New Jersey
5. Fillmore, C.J. (1968) The Case for Case. In Holt, Rinehart and Winston, Inc., E. Bach/R.T. Harms (eds.) Universals in linguistics theory
6. Chomsky, N. (1965) Aspects of the theory of syntax. MIT, Cambridge Massachusetts
7. Chen, P. (1983) English Sentence Structure and Entity Relationship Diagrams. In: Int'l Journal of Information Sciences. Vol. 29, pp. 127–149
8. Abbott, R.J. (1983) Program Design by Informal English Descriptions. In: Communications of the ACM. 26 (11), pp. 882–894. ACM
9. Saeki, M., Horai, H., Enomoto, H. (1989) Software Development from Natural Language Specification. In: Proc. Int'l Conference on Software Engineering (ICSE), pp. 64–73, IEEE
10. Dik, S. (1980) Studies in Functional Grammar. Academic Press
11. Dignum, F., van de Riet, R.P. (1991) Knowledge base modelling based on linguistic and founded in logic. In: Data & Knowledge Engineering. Vol. 7, pp. 1–34. Elsevier
12. Burg J.F.M. (1997) Linguistic Instruments in Requirements Engineering. IOS Press, Amsterdam
13. Buchholz, E., Cyriaks, H., Düsterhöft, A., Mehlan, H., and Thalheim, B. (1995) Applying a Natural Language Dialogue Tool for Designing Databases. In: Int'l Workshop on Applications of Natural Language to Databases. NLDB'95, pp. 119–133, 1995
14. Tjoa, A.M., Berger, L. (1993) Transformation of Requirement Specification Expressed in Natural Language into an EER Model. In: Proc. 12th Int'l Conf. on Entity Relationship Approach. ER1993, pp. 127–149, Springer, Heidelberg, 1993
15. Mich.: NL-OOPS (1996) From natural language to object oriented requirements using the natural language processing system LOLITA. In: Natural Language Engineering. Cambridge Universal Press
16. Ambriolla V., Gervazi V. (1997) Processing Natural Language Requirements. In: Proc. of Int'l Conf. on Automated Software Engineering. ASE '97, pp. 36–45. IEEE, 1997
17. Moreno A. (1997) Object-Oriented Analysis from Textual Specifications. In: Proc. 9th Int'l Conf. on Software Engineering and Knowledge Engineering. SEKE 97, 1997
18. Juristo, N., Morant, J.L, and Moreno, A., M. (1999) A formal approach for generating OO specifications from natural language. In: Journal of Systems and Software. Elsevier
19. Juristo, N., Moreno, A.M., López, M. (2000) How to use linguistic instruments for object-oriented analysis. Software. IEEE
20. Harmain, H.M., Gaizauskas, R.: CM-Builder (2000) An Automated NL-based Case Tool. In: 15th Int'l Conf. on Automated Software Engineering. ASE'00, pp. 45–54. IEEE, 2000
21. Cabral, C., Sampaio, A. (2008) Formal specification generation from requirement documents. In: Electronic Notes in Theoretical Computer Science. Elsevier
22. Burg J., van de Riet R. (1995) COLOR-X: Linguistically-based Event Modelling: A General Approach to Dynamic Modelling. In: Proc. 17th Int. Conf. on Advanced Information System Engineering. CAiSE1995. LNCS, pp. 26–39, Springer, Heidelberg, 1995
23. Rolland, C., Ben Achour, C. (1998) Guiding the construction of textual use case specifications. In: Data & Knowledge Engineering, Vol.25, pp. 125–160. Elsevier
24. Menck, V. (2004) Deriving behavior specifications from textual use cases. In: Proc. Workshop on Intelligent Technologies. Reference.kfupm.edu.sa, 2004

25. Kof, L. (2007) Scenarios: Identifying Missing Objects and Actions by Means of Computational Linguistics. In: Proc.15th Int'l Requirements Engineering Conference (RE 2007), pp. 211–130. IEEE, 2007
26. Santos, J., Moreira, A., Araujo, J., Amaral, V., Alferez, M., Kulesza, U. (2008) Generating Requirements Analysis Models from Textual Requirements. In: First Int'l Workshop on Managing Requirements Knowledge. MARK2008, pp. 32–41. IEEE, 2008
27. Rolland, C., Souveyet, C., and Ben Achour, C. (1998) Guiding Goal Modelling Using Scenarios. In.: Transactions on Software Engineering (TSE). 24(12), pp. 1055–1071, IEEE
28. Overmyer, S.P. Lavoie, B. Rambow, O. (2001) Conceptual modeling through linguistic analysis using LIDA. In: Proc. 23rd Int'l Conference on Software Engineering. ICSE2001, pp. 401–410. IEEE, 2001
29. Cimiano, P., Völker, J. (2005) Text2Onto: A Framework for Ontology Learning and Data-driven Change Discovery. In: Proc. 10th Int'l Conf. on Applications of Natural Language to Information Systems (NLDB). LNCS, vol. 3513, pp. 227–238. Springer, Heidelberg, 2005
30. Zisman, A., Spanoudakis, G., and Pérez-Miñana, E. (2003) Tracing software requirements artifacts. In: Proc. Int'l Conf. on Software Engineering Research and Practice. SERP, 2003
31. Herbst, H. (1997) Business rule-oriented conceptual modeling. (Physica Verlag). Springer, Heidelberg
32. Rayson, P., Emmet, L., Garside, R., Sawyer, P. (2001) The REVERE Project: Experiments with the application of probabilistic NLP to Systems Engineering. In: M. Bouzeghoub et al (eds.). Natural Language Processing and Information Systems. LNCS, pp. 288–300, Springer, Heidelberg, 2001
33. Gervasi, V., Zowghi, D. (2005) Reasoning about inconsistencies in natural language requirements. In: Transactions on Software Engineering and Methodology. TOSEM 14(3), pp. 277–330. ACM
34. Saeki, M., Kaiya, H. (2008) Supporting the elicitation of requirements compliant with regulations. In: Z. Bellasene, M. Leonard (eds.). CAiSE 2008. LNCS, vol. 5074, pp. 228–242, Springer, Heidelberg, 2008
35. Prat, N. (1997) Goal formalisation and classification for requirements engineering. In: Proc. 3rd Int'l Workshop on Requirements Engineering: Foundations of Software Quality. REFSQ1997. LNCS, pp. 145–156. Springer, Heidelberg, 1997
36. Liaskos, S., Lapouchnian, A., Yu, Y., Mylopoulos, J. (2006) On goal variability acquisition and analysis. In: Proc. Int'l Conf. on 14th Requirements Engineering Conference. RE2006, pp. 79–88. EEE, 2006