Transferring Research Into the Real World: How to Improve RE with AI in the Automotive Industry

Sven J. Körner, Mathias Landhäußer and Walter F. Tichy
Karlsruhe Institute of Technology
Karlsruhe, Germany
Email: {sven.koerner, landhaeusser, tichy}@kit.edu

***Abstract*—For specifications, people use natural language. We show that processing natural language and combining this with intelligent deduction and reasoning with ontologies can possibly replace some manual processes associated with requirements engineering (RE). Our prior research shows that the software tools we developed can indeed solve problems in the RE process. This paper shows this does not only work in the software engineering domain, but also for embedded software in the automotive industry.**

We use artificial intelligence in the sense of combining semantic knowledge from ontologies and natural language processing. This enables computer systems to "understand" requirement texts and process these with "common sense". Our specification improver RESI detects flaws in texts such as ambiguous words, incomplete process words, and erroneous quantifiers and determiners.

I. Introduction

Brooks' statement from 1987's *silver bullet* paper still holds true: "the hardest part of the software task is arriving at a complete and consistent specification, and much of the essence of building a program is in fact the debugging of the specification" [1]. Requirements are necessary for product management. They are the starting point for every implementation, whether they are of software, mechanical, or logical nature [2]. Sommerville understands (software) requirements as the description of or services offered by a system and its operational constraints [3]. And except for a few exceptions [4], requirements are mostly in natural language [5]–[7]. The information stored in natural language descriptions is gathered in the analysis elicitation phase by experts and laymen [8] and translated and documented in actual requirements.

Our previous research [9]–[13] showed that requirements can be improved and processed automatically using approaches from artificial intelligence (AI) like natural language processing (NLP), ontology reasoning, and deduction. That is, finding flaws in requirements texts, detecting the semantic meaning of words, and automatic UML diagram generation directly from natural language requirements. Preliminary results hinted that our approach is industry and sector independent. Requirements engineering researchers deem the transfer from research to practice an important challenge (e.g. GRRIP 2013 [1]). So we took our approach to a test by cooperating with industry partners and scrutinizing their requirements with our RECAA tool set. This paper shows the results of our cooperation with Daimler AG, which is continuously working on improving their requirements engineering processes [14], [15]. For instance, automatically finding mistakes in requirements is an important part of their agenda. Not all contents from the provided documents can be published in this paper, but are available on inquiry from Daimler. Still, this paper covers details of automatically detected requirement flaws and explains the technical processing and the integration of AI in detail.

II. Related Work

Research in requirements engineering (RE) focuses on many aspects. Our focus in the requirements engineering domain is on the possible shortcomings of faulty requirements and their effect on the actual implementation. How to discover those flaws (manually) has been discussed and examined in the last decades [16]–[24]. Our approach focuses on the tool support of this part of the RE process.

Many researchers have tackled this challenge in the past and were limited due to the lack of AI and NLP tools. Therefore they focused on rather constricted approaches [25]–[30] instead of supporting the entire RE process. The mandatory deduction, reasoning, and semantics gathering could not be done or assumed elaborate user workload to add the necessary *intelligence*. Our approach aims to limit the user workload to only interact with the tools' interfaces if feedback from the stakeholders is necessary. Extensive case studies in Körner's dissertation [13] have shown that software requirements of many types can be processed with the RECAA tool suite and revealed a significant speed-up and improvement of the RE documents.

The tools currently applied by the automotive industry are the usual suspects when it comes to requirements tracking, tracing, and management: IBM Rational DOORS, Rational Requirements Composer, CaliberRM, or RequisitePro [2].

III. Concept

Our approach RECAA comprises four tools: Requirements Engineering Specification Improver (RESI), AutoAnnotator (AA), Sale Model eXtractor (SaleMX), and Requirements Engineering Feedback System (REFS) as depicted in Figure 1. Each of these tools uses different NLP tools and semantic

[1] Workshop on Gaps between Requirements Research and Industrial Practices.

[2] c.f. http://www-142.ibm.com/software/products/ and https://sites.google.com/site/grripworkshop/

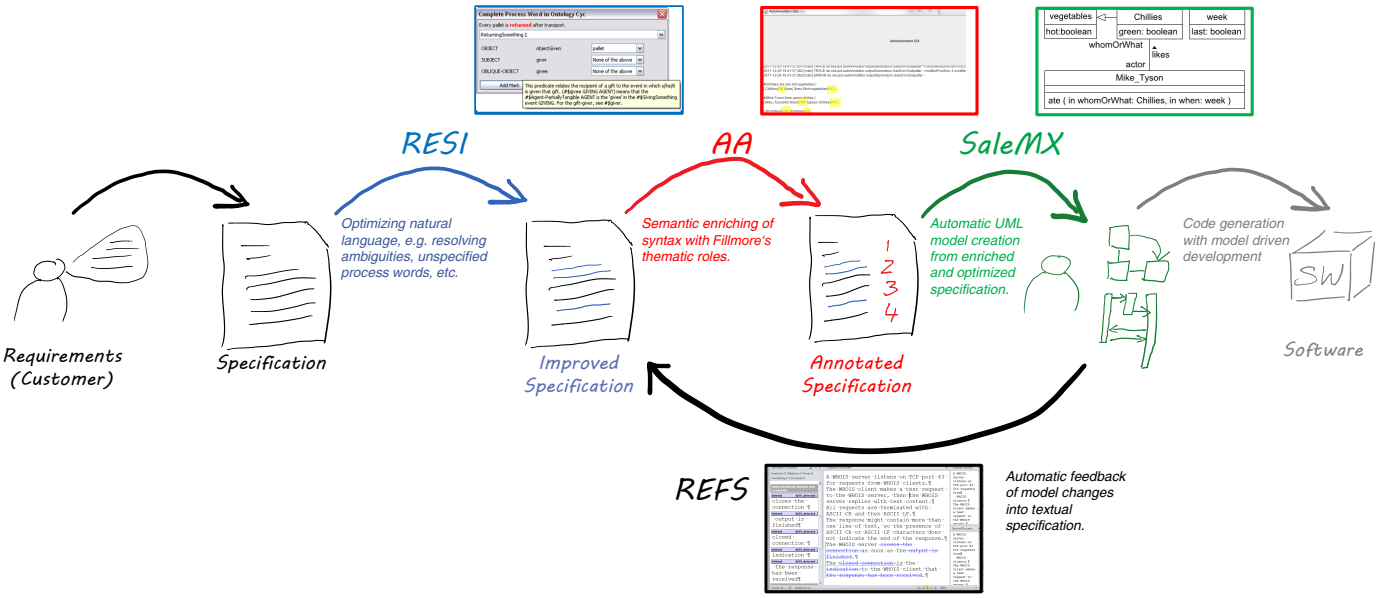

Fig. 1. The RECAA process considers requirements elicitation, quality assurance, model generation and change management. This study concentrates on improving specifications with RESI.

enriching through ontologies: Stanford Parser, NER, POS-Tagger, JavaRAP, WordNet, ResearchCyc to name a few.

Figure 2 shows RESI's processing cycle: At first, it transfers natural language into an model for the Eclipse Modeling Framework (EMF) and then applies (linguistic) rules to find flaws such as ambiguities, wrong quantifiers, etc. [13]. Those rules are based on the flaw detection approaches of Berry and Rupp et al. (see related work). For every flaw that RESI can detect (c.f. Table I), there is one rule. Rules cover ambiguity, quantifiers, synonyms, incomplete process words (see Section V), nominalizations, and a few more not relevant to the contents of this paper. Rules (R1-R3 in Figure 2) can use NLP to extract information from the text and ontologies to reason about it. This alters the EMF model by updating, adding, and deleting parts. Then, the altered EMF model is fed back to the textual specification.

Sometimes the application of a linguistic rule requires feedback from the user. RESI makes suggestions, but lets the users decide. If the users, typically requirements analysts, cannot decide on the spot, they need to clarify with the stakeholders. Then they can rewrite the requirements according to the stakeholders' expectations. Evaluation showed that our rules find most flaws in the requirements texts automatically and sometimes perform better than highly-skilled requirements engineers [13].

The following section discusses RESI's results on the specification of a lighting system. UML is not used in the specification of the lighting system, therefore this paper does not cover the automatic creation of UML models from natural language text with AA, SaleMX, and REFS. Daimler focuses on natural language descriptions of requirements and we focus on finding and remedying flaws in these requirements texts with RESI. Section V shows in detail how RESI applies a rule.

IV. Results

The specification we used is a anonymized and modified sample specification prepared by Daimler AG describing the requirements for the adaptive exterior lighting control (high beams) and its interaction with the user display. The specification is of high quality and already has been used in the production process to develop the corresponding features for the car. The specification came under scrutiny because we wanted to check if our tools were able to find flaws in existing specifications. We tested the specification for flaws with RESI and found a number of possible issues. The results are listed in Table I and an excerpt of the erroneous sentences can be found in Listing 1. The specification comprised 712 sentences with 3391 words. The large number of sentences is due to the fact that about half of the specification is in table format and each table cell counts as a new sentence. The cells contain either a sentence, half a sentence, or just a single word.

Our tool found 28 ambiguous words in the specification, eight of which could be specified more accurately with suggested substitutions from RESI. RESI determines suggestions by using the ontologies WordNet [31] and Cyc [32]. We decided to choose *standard* ontologies rather than creating our own specific ontologies, to show the validity of the approach without running the risk of creating a self fulfilling prophecy. As ontology research shows, domain specific and specially targeted ontologies lead to better results. Our idea was to show that processing requirements works with standard means and could be improved even further by adopting and maintaining specific ontologies.

Ontologies deliver more precise descriptions of processes and nouns. The results are more concise formulations in the requirements. For instance, in sentence (1) `restore(d)` was substituted for the clearer `reinstatement`. RESI tells

Listing 1. Excerpt from the Adaptive Exterior Lighting System Specification

```
1 If no advancing vehicle is recognized
    any more, the high beam illumination
    is restored within 2 seconds. If
    necessary, the high beam illumination
    is restored within 1 second.
2 If a defective illuminant is detected,
    the information about the defective
    illuminant is quickly transmitted to
    the instrument cluster.
3 This process cannot be started if the
    result cannot be displayed.
4 With overload, the illumination area
    requirements do not need to be
    respected.
5 If the threshold is underrun, the low
    beam is activated.
6 If the threshold is overrun, the low
    beam is deactivated.
7 Correction factor for distance
    calculation time to the vehicle in
    front with audio warning.
8 Correction factor for distance
    calculation time to the vehicle in
    front with optical warning.
9 If the threshold is overrun, the brake
    assist system gets active.
```

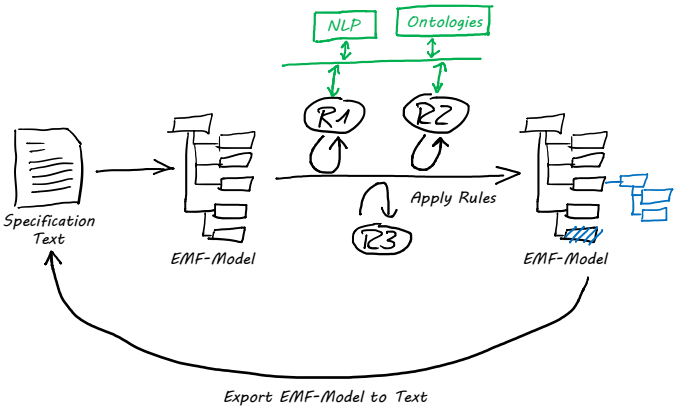

Fig. 2. RESI converts text into EMF models, applies linguistic rules, improves the specification and feeds this information back.

the user that *restore* could be mistaken for *refinishing* or *restoring an object* which is not the desired sense in this sentence. Rather, the explanation denotes that the high beams will be turned back on and therefore their status reinstated, especially compared to the previous state. RESI suggests to use `reinstatement` and explains the meaning of it as follows: *Each instance (of reinstatement) is the condition of being brought back into original existence, use, function, or position.* Further, the word `second` was substituted with `SecondsDuration` to assure not to overlook the time measuring aspect of the formulated sentence. Cyc explains the meaning of `SecondsDuration` as follows: *A UnitOfTime function that takes one or two real numbers as arguments and returns, as its value, a comparable Time-Quantity measured in seconds. More precisely, an expression of the form (SecondsDuration NUM) denotes the (point-value) Time-Quantity of being exactly NUM seconds in duration, and an expression of the form (SecondsDuration MIN MAX) denotes the (properly interval-like) Time-Quantity of being at least MIN and at most MAX seconds in duration.* The fact that `SecondsDuration` allows a minimal and maximum value leads to further feedback from RESI when checking determiners and quantifiers. Sentence (1) includes 3 quantifiers which RESI clarifies by suggesting to change `no advancing vehicle` to `none (w/o exception) advancing vehicles` and `2 seconds` to `exactly 2 seconds` and `within 1 second` to `exactly 1 second`.

TABLE I
QUANTITATIVE EVALUATION OF AUTOMATICALLY FOUND FLAWS IN THE ADAPTIVE EXTERIOR LIGHTING SYSTEM SPECIFICATION

Flaw Category	#Flaws Found
#Ambiguous Words	28
#Suggested Meanings	66
#Suggested and More Accurate Meanings	8/66
#Nominalizations	2
#Incomplete Process Words	1
#Synonyms	0
#Erroneous Quantifiers and Determiners	3

In sentence (4) we specified the `illumination area` more closely as `area of the illumination object`. `Calculation time` was substituted with `calculation TimeInterval` in sentences (7) and (8), `threshold` with `ThresholdValue` in sentences (6) and (9), and `need` with `Need-SystemCondition` in sentence (4).

To clarify the meaning of the remaining 20 ambiguous words, RESI suggested a total of 58 additional meanings. Here, none of the suggestions did apply. In this case, the analyst can decide to set a *marker/reminder* in the requirements document and to gather feedback from the stakeholders on how the word is to be understood. Additionally, the tool discovered two nominalizations: `process` in sentence (3) and `warning` in sentence (7) which should be replaced by verbal phrases instead, as demanded by Rupp, Berry, etc.

RESI found the incompletely specified process word `displayed` in sentence (3). Here, one argument for the verb is missing: it is not clear who the sender of the information to be displayed is. Also, one could speculate that the object displaying the information is most likely the instrument cluster as suggested by RESI and depicted in Figure 3. Pleasantly enough, RESI does not find any synonyms in the specification which may be due to Daimler's efforts of getting rid of weak words, synonyms, and other easy to find ambiguities. Another positive aspect of this specification we found is that indefinite articles are not used in an ambiguous way where "a" is used in the sense of "one, and only one".

To sum it up, the specification includes 34 possible flaws, three of which Daimler deemed problematic after an initial

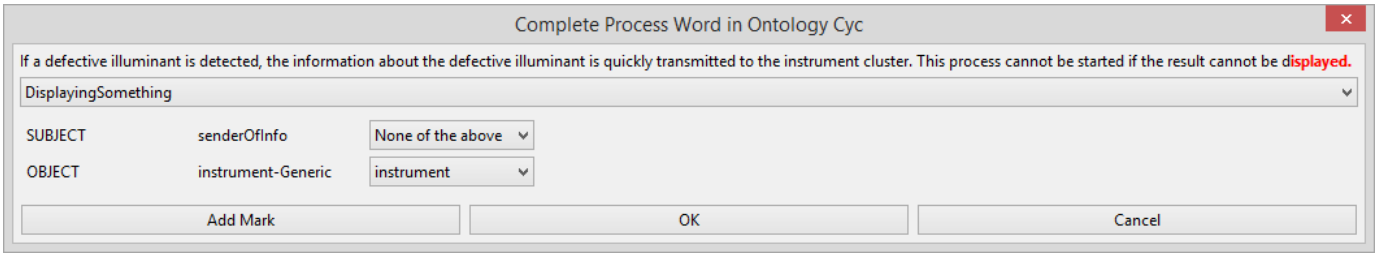

Fig. 3. RESI finds `display` as incompletely specified process word and suggests/suspects that the information is displayed on the instrument cluster, but points out that it is unclear who the sender of the information to be displayed is.

evaluation of our results. These were `quickly`, `underrun` and the incompletely specified process word `display`. According to Daimler, the other errors in the requirements document cannot be considered flaws since Daimler's glossary precisely defines the meaning of the terms involved. According to Daimler, all teams working with the specification have the same understanding of the vocabulary. In the document we assessed, no glossary was included.

V. How It Works

First, the requirements text is tokenized and parsed into an EMF model as described in Figure 2: Sentences, words, punctuation, and named entities are pushed into an EMF tree and traversed by RESI. The pre-processing includes retrieving the infinitive form of verbs to improve ontology query results using WordNet.

To explain how RESI works, we use the example of the incompletely specified process word. To find this flaw, RESI makes a query to an ontology to recover the argument lists of the process word. The example uses sentence (3) and shows the queries for ResearchCyc only. The queries to ResearchCyc are written in CycL [33], a ResearchCyc specific language. Predicates and fixed terms are prefixed with `#$` and variables with `?`. ResearchCyc is structured into so called micro theories. The relevant queries for RESI are being made to the micro theory *GeneralEnglishMt* which covers the general English language. We expect even better results with domain specific ontologies and micro theories for the automotive domain.

ResearchCyc provides semantic information about words in argument lists. One retrieves the list for a given word with a single query. Before that, the process word (in this case `displaying`) has to be tranformed into a word constant. This is done be capitalizing the first letter of the word and adding `-TheWord` to the corresponding word string.

```
(#$verbSemTrans #$Displaying-TheWord
 ?SENSECOUNTER ?FRAMETYPE
   ?FRAME)
```

The variable `?FRAME` then holds the argument list in the following format:

```
(#$and
   (#$isa :ACTION #$DisplayingSomething)
   (#$senderOfInfo :ACTION :SUBJECT)
   (#$instrument-Generic :ACTION
    :OBJECT)))
```

The line with the predicate `#$isa` denotes the meaning of the word in its context. The argument list is returned only if the meaning of the process word matches the argument list. The other lines show the arguments in their semantic (e.g. `#$senderOfInfo`) and their syntactic (e.g. `:ACTION`) roles. RESI determines the descriptions of the corresponding objects and combinations with the following query. Later, the descriptions are used in the user interface for feedback:

```
(#$comment #$senderOfInfo ?COMMENT)
```

To fill these argument lists with recommendations (in this case the `instrument-Generic`) from the corresponding sentence, RESI checks every argument with which value/word it could be filled:

```
(#$arg2Isa #$instrument-Generic
 ?WHATFITS)
```

This leads to the result that an `#$instrument-Generic` can be every `#$instrumentalRole` or `#$actors` object.

Finally, every word of the sentence is checked whether it suffices as one of the needed arguments. RESI checks if the word which is supposed to be used as argument (`#$instrumentalRole`) is a generalization of the meaning of the word (`#$instrument-Generic` for `instrument`) from the sentence. RESI inserts the word into the corresponding list of argument candidates, if this query returns true. After checking all arguments and all words as described, the candidate lists are being returned and RESI suggests process word arguments as shown in Figure 3.

We use similar Cyc queries in the implementation of the other linguistic rules (such as nominalizations, ambiguous words, etc.). The RECAA tool suite also supports other ontologies, such as WordNet, ConceptNet [34], Yago [35], and OWL ontologies.

VI. Conclusion

This paper reports our findings when applying RESI to a real world specification from the automotive domain. Daimler provided us with a clean specification. Still, our tool RESI produced 34 warnings, three of which were deemed problematic and had been overlooked. As a lesson learned, we have seen that the automatic requirements processing techniques we developed for RECAA not only work in the software engineering domain, but can also be applied to other requirements. The results show that our tool suite is too

"picky" for usage in real world domains, because even the slightest flaws are detected and listed. To make the approach usable in industry, we have to integrate it with the CASE tools and to improve the warning strategies. Currently, RESI finds and reports all identified flaws without assessing the possible weight and impact of certain flaws compared to others. Further tests need to show if thresholds for flaws will help the user. We expect that it might be reasonable to ignore some flaws to reduce the users' workload and to make sure one does not oversee potentially expensive flaws. Transferring research results into the real world seems partly about tailoring the tools for fast and easy use with the right set-up and mix between precision and recall. We already setup future studies to improve usability.

We used our RECAA setup with standard ontologies without using specific micro theories. They might improve results further if company-wide wordings and terms are used and introduced and therefore detected. The next step is to transfer these promising results into the current infrastructure with IBM Rational Doors, etc. and see if the amount of additionally detected flaws makes up for a business case. After that, creating a Daimler specific ontology (e.g. on the basis of already existing requirements) could be the next step to improve the precision of the suggestions.

A threat to internal validity is the small number of requirements we were able to test so far. We are hoping to receive more specifications that we can process to get a better understanding of the connection and background of different domains and requirements types in different industries.

Acknowledgment

We'd like to thank Daimler AG for sharing the requirements specification and cooperating with our research.

References

[1] F. P. Brooks, Jr., "No silver bullet essence and accidents of software engineering," *Computer*, vol. 20, pp. 10–19, April 1987. [Online]. Available: http://dx.doi.org/10.1109/MC.1987.1663532

[2] A. Maglyas, U. Nikula, and K. Smolander, "What do practitioners mean when they talk about product management?" in *RE*, M. P. E. Heimdahl and P. Sawyer, Eds. IEEE, 2012, pp. 261–266.

[3] I. Sommerville, *Software engineering*, 7th ed., ser. International computer science series. Boston: Pearson, Addison-Wesley, 2004. [Online]. Available: http://www.gbv.de/dms/ilmenau/toc/383981964.PDF

[4] P. Shaker, J. M. Atlee, and S. Wang, "A feature-oriented requirements modelling language." in *RE*, M. P. E. Heimdahl and P. Sawyer, Eds. IEEE, 2012, pp. 151–160.

[5] L. Mich, M. Franch, and P. Inverardi, "Market research for requirements analysis using linguistic tools," *Requir. Eng.*, vol. 9, pp. 40–56, 2004.

[6] B. H. C. Cheng and J. M. Atlee, "Research directions in requirements engineering," in *Proc. Future of Software Engineering FOSE '07*, 23–25 May 2007, pp. 285–303.

[7] C. Rolland and C. Proix, "A Natural Language Approach for Requirements Engineering," in *Proceedings of the Fourth International Conference CAiSE'92 on Advanced Information Systems Engineering*, P. Loucopoulos, Ed., vol. 593. Manchester, United Kingdom: Springer-Verlag, 1992, pp. 257–277, 10.1007/BFb0035136. [Online]. Available: http://dx.doi.org/10.1007/BFb0035136

[8] A. Niknafs and D. M. Berry, "An industrial case study of the impact of domain ignorance on the effectiveness of requirements idea generation during requirements elicitation." in *RE*. IEEE, 2013, pp. 279–283.

[9] T. Gelhausen, "Modellextraktion aus natürlichen Sprachen: Eine Methode zur systematischen Erstellung von Domänenmodellen," Ph.D. dissertation, Karlsruhe Institute of Technology, Jul. 2010.

[10] S. J. Körner and T. Brumm, "Natural language specification improvement with ontologies," *International Journal of Semantic Computing (IJSC)*, vol. 03, no. 04, pp. 445–470, 2010.

[11] S. J. Köerner and M. Landhäußer, "Semantic enriching of natural language texts with automatic thematic role annotation," in *Natural Language Processing and Information Systems*, ser. Lecture Notes in Computer Science, C. Hopfe, Y. Rezgui, E. Mtais, A. Preece, and H. Li, Eds. Springer Berlin Heidelberg, 2010, vol. 6177, pp. 92–99. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-13881-2_9

[12] W. F. Tichy and S. J. Koerner, "Text to software: Developing tools to close the gaps in software engineering," in *Proceedings of the FSE/SDP Workshop on Future of Software Engineering Research*, ser. FoSER '10. New York, NY, USA: ACM, 2010, pp. 379–384. [Online]. Available: http://doi.acm.org/10.1145/1882362.1882439

[13] S. J. Körner, "Recaa - werkzeuguntersttzung in der anforderungserhebung," Ph.D. dissertation, Karlsruhe Institute of Technology, February 2014. [Online]. Available: http://www.ksp.kit.edu/9783731501916

[14] D. Ott, "Defects in natural language requirement specifications at mercedes-benz: An investigation using a combination of legacy data and expert opinion." in *RE*, M. P. E. Heimdahl and P. Sawyer, Eds. IEEE, 2012, pp. 291–296.

[15] D. Ott and A. Raschke, "Review improvement by requirements classification at mercedes-benz: Limits of empirical studies in educational environments," in *Empirical Requirements Engineering (EmpiRE), 2012 IEEE Second International Workshop on*, Sept 2012, pp. 1–8.

[16] L. Kof, "Treatment of Passive Voice and Conjunctions in Use Case Documents," in *Application of Natural Language to Information Systems*, ser. LNCS, Z. Kedad, N. Lammari, E. Mthais, F. Meziane, and Y. Rezgui, Eds., vol. 4592. Paris, France: Springer–Verlag, June 27–29 2007, pp. 181–192, copyright Springer-Verlag, available at http://www.springerlink.com/content/a602k461043n4322/.

[17] E. Kamsties, D. M. Berry, and B. Paech, "Detecting Ambiguities in Requirements Documents Using Inspections," in *Proceedings of the First Workshop on Inspection in Software Engineering (WISE'01)*, 2001, pp. 68–80.

[18] D. M. Berry, A. Bucchiarone, S. Gnesi, and G. Trentanni, "A New Quality Model for Natural Language Requirements Specifications," 2008. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.96.5268

[19] D. M. Berry, E. Kamsties, and M. M. Krieger, *From Contract Drafting to Software Specification: Linguistic Sources of Ambiguity - A Handbook*, November 2003. [Online]. Available: http://se.uwaterloo.ca/~dberry/handbook/ambiguityHandbook.pdf

[20] M. Ceccato, N. Kiyavitskaya, N. Zeni, L. Mich, and D. M. Berry, "Ambiguity identification and measurement in natural language texts."

[21] C. Denger, D. M. Berry, and E. Kamsties, "Higher quality requirements specifications through natural language patterns," vol. 0. Los Alamitos, CA, USA: IEEE Computer Society, 2003, p. 80.

[22] N. Kiyavitskaya, N. Zeni, L. Mich, and D. M. Berry, "Requirements for tools for ambiguity identification and measurement in natural language requirements specifications," *Requir. Eng.*, vol. 13, no. 3, pp. 207–239, 2008.

[23] A. Dwarakanath, R. R. Ramnani, and S. Sengupta, "Automatic extraction of glossary terms from natural language requirements." in *RE*. IEEE, 2013, pp. 314–319.

[24] C. Rupp, *Requirements-Engineering und -Management*, 4th ed. Carl Hanser Verlag, 2007, ch. Das SOPHIST-REgelwerk – Psychotherapie fr Anforderungen, pp. 139–176.

[25] D. K. Deeptimahanti and R. Sanyal, "An innovative approach for generating static UML models from natural language requirements," in *Advances in Software Engineering*, ser. Communications in Computer and Information Science, vol. 30. Springer, 2009, pp. 147–163.

[26] L. Mich, "Nl-oops: from natural language to object oriented requirements using the natural language processing system lolita," *Nat. Lang. Eng.*, vol. 2, pp. 161–187, June 1996. [Online]. Available: http://portal.acm.org/citation.cfm?id=974662.974666

[27] P. Kroha, R. Janetzko, and J. E. Labra, "Ontologies in checking for inconsistency of requirements specification," *Advances in Semantic Processing, International Conference on*, vol. 0, pp. 32–37, 2009.

[28] S. P. Overmyer, B. Lavoie, and O. Rambow, "Conceptual modeling

through linguistic analysis using LIDA," in *Proc. of the ICSE '01*. Washington, DC, USA: IEEE Computer Society, 2001, pp. 401–410.

[29] F. Fabbrini, M. Fusani, S. Gnesi, and G. Lami, "The linguistic approach to the natural language requirements quality: Benefit of the use of an automatic tool," in *SEW '01: Proceedings of the 26th Annual NASA Goddard Software Engineering Workshop*. Washington, DC, USA: IEEE Computer Society, 2001, p. 97. [Online]. Available: http://ieeexplore.ieee.org/search/wrapper.jsp?arnumber=992662

[30] K. Verma and A. Kass, "Requirements analysis tool: A tool for automatically analyzing software requirements documents," in *7th International Semantic Web Conference (ISWC2008)*, October 2008.

[31] G. A. Miller, "WordNet: A lexical database for English," *Communications of the ACM*, vol. 38, no. 1, pp. 39–41, 1995.

[32] Cycorp Inc., "ResearchCyc," [last visited on 07/03/2014]. [Online]. Available: http://research.cyc.com/

[33] *Cyc 101 Tutorial*, Cyc.com.

[34] C. Havasi, R. Speer, and J. Alonso, "ConceptNet 3: a Flexible, Multilingual Semantic Network for Common Sense Knowledge," in *Recent Advances in Natural Language Processing*, Borovets, Bulgaria, September 2007. [Online]. Available: http://web.media.mit.edu/~jalonso/cnet3.pdf

[35] J. Hoffart, F. M. Suchanek, K. Berberich, E. Lewis-Kelham, G. de Melo, and G. Weikum, "Yago2: Exploring and querying world knowledge in time, space, context, and many languages," in *Proceedings of the 20th international conference companion on World Wide Web (WWW 2011)*, S. Srinivasan, K. Ramamritham, A. Kumar, M. P. Ravindra, E. Bertino, and R. Kumar, Eds., Association for Computing Machinery (ACM). Hyderabad, India: ACM, 2011, pp. 229–232.

[36] *21st IEEE International Requirements Engineering Conference, RE 2013, Rio de Janeiro-RJ, Brazil, July 15-19, 2013*. IEEE, 2013.

[37] M. P. E. Heimdahl and P. Sawyer, Eds., *2012 20th IEEE International Requirements Engineering Conference (RE), Chicago, IL, USA, September 24-28, 2012*. IEEE, 2012.