Usability Issues for Systems Supporting Requirements Extraction from Legal Documents

Nicola Zeni
Dept. of Information Engineering and Computer Science
University of Trento
Trento, Italy 30332–0250
nicola.zeni@unitn.it

Luisa Mich
Dept. of Industrial Engineering
University of Trento
Trento, Italy 30332–0250
luisa.mich@unitn.it

***Abstract*—Usability as ease of use and learnability, is critical for systems supporting requirements elicitation for regulatory compliance. The main problem is that these systems have to analyze documents in a specialized natural language, a task that is far from being completely automated. Usability issues are also related to a variety of other characteristics of such systems. Reasons why an early adoption of usability practices is desirable and beneficial in their development are described. Main lessons learned in developing and applying a complex framework for requirements elicitation from regulatory documents are presented to illustrate some of the most relevant usability concerns.**

***Index Terms*—support system, natural language analysis, usability design, users, roles.**

I. Introduction

There are many projects aimed to develop tools and systems to support requirements analysts in eliciting legal requirements from regulatory documents (see the proceedings of past RELAW workshops, http://gaius.isri.cmu.edu/relaw and of the Jurix conferences http://jurix.nl/proceedings). Systems range from editors for tagging laws, contracts, rules and other kinds of legal texts to frameworks integrating natural language processing (NLP) functions. NLP systems are mandatory for a completely automatic requirements extraction, as the process of requirement extraction would require a deep semantic and pragmatic analysis of textual documents. The latter is the ultimate goal of researches in natural language and is far from being reached (http://journals.cambridge.org/action/displayJournal?jid=NLE). This implies that existing tools have different degree of maturity and only some of them – among which, modules for the manual tagging of the documents, search tools, structural tagging, conversion from and to different format – could be considered in the pre-competitive development phase and integrated in commercial systems.

Whatever the approach and tools to be developed, there are some characteristics common to systems supporting requirements extraction (or SSRE, for short) from textual regulatory documents. In the next section, such characteristics are described in relation to usability concerns. These concerns have to be addressed according to the general principles of usability engineering, that is as an up-front and on-going activity [1], [2]. Section III reports lessons learned in developing and applying GaiusT [3], [4], a complex framework for requirements elicitation from regulatory documents which thanks to its comprehensive architecture allowed exposing a variety of usability aspects.

II. Usability: Why

Usability is defined by ISO as "the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use." [5]. In software systems development, usability is one of the quality dimensions included in the ISO model [6] and quality has to be obtained including quality goals since the very beginning of the development process. To underline this aspect, literature distinguishes a number of types of quality, among which, designed and delivered quality [1], [7], [8] so that quality goals have to be taken into account at the different steps of the projects: quality and usability in particular cannot be obtained adding features at the end of the project [9]. Usability is even more critical for systems developed in research projects. In this projects, solutions to new problems have to be invented and implemented, testing results at different stages of the project, often collaborating with many research teams, sharing gathered knowledge in scientific reports and papers, involving experts in a variety of domains. This is the case for SSRE, in which beside technical skills – in requirements analysis, software engineering, programming, web engineering, human computer interfaces, linguistics, statistic – other skills are necessary, as jurisprudence and specialized knowledge of laws for specific domains, e.g., health, patents.

The core task a SSRE from legal documents has to accomplish is the identification of concepts included in a given model of regulations. Automatic recognition of concepts can be supported by semantic annotation modules. The definition of a conceptual model – usually using UML (Unified Modeling Language, www.uml.org) or other modeling languages – is another activity that a SSRE can support, which implies that graphic functions are also needed. Besides, modules for manual tagging or to revising tagged documents are included. Some SSRE allow creating different layers of analysis and tagging, from structural to semantic tagging according to the conceptual model. Both in-line and off-line tagging or

978-1-4799-6324-9/14/$31.00 © 2014 IEEE

markup requires the creation and management of databases. Multi-language tagging is also in place in some projects. As regards the implementation of a SSRE, many programming languages and technical frameworks could be used. It is quite common that some of the linguistic activities (e.g., parsing, work frequency analysis) or editing features (both on texts and on graphical models) are developed customizing open source programs or looking in existing libraries. These modules have then to be integrated into the framework, solving transportability and inter-operability problems.

The degree of automation of all these activities varies greatly between the modules of an SSRE. Experience gathered in a large scale project which goal is to design and develop a general framework named GaiusT [4], has highlighted how the success of sub-projects focused on specific modules or features are also strongly related to usability issues. Common characteristics of a SSRE are:

- A SSRE includes many functions addressing the variety of activities of the requirements extraction process [4], [10].
- There are a number of different roles involved in this kind of projects: researchers, developers, requirements analysts, business analysts, lawyers, experts of standards, statisticians, domain experts (e.g., health experts for laws like HIPAA, www.hhs.gov/ocr/privacy/hipaa/understanding),etc.
- Research projects often adopt an incremental development process in which prototypical modules are designed to test research questions and, if successful, added to a comprehensive framework.

All these three characteristics demand attention to usability for a SSRE: In general terms, the three aspects of usability ? effectiveness, efficiency and ease of use ? are all present in an enhanced way. In particular:

- The modular structure may imply technical issues related to different implementation languages and other technical and architectural requirements as for example all those needed to implement the SSRE as a Web-based system.
- The presence of many types of users ? besides final users of a SSRE – highly increases usability concerns and can be critical to the success of the project. Cooperative work has also to be supported because researchers are working in different places (e.g., European projects involve research groups in different countries). Ease of use and learnability are hampered by the trade-off between the need to implement new solutions in an effective and efficient process, often with throwaway prototypes, and the need to develop interfaces allowing researchers to test their analysis and markup approaches without the help of a developers.
- The evolution of the systems requires feedback from many people and such feedback can be effectively gathered only if the prototypes are usable. On the other side, in usability engineering, prototypes are suggested to test and improve usability of the system to be realized. However, features to be checked for usability are a lot, so that it demands a focused view that prototypes help to enforce but it does not happen automatically.

III. Usability: Lessons Learned

Usability engineering suggests addressing usability concerns according to a multi-stakeholder approach. Experiences gained with the GaiusT project [11] are illustrated referring to three relevant roles and to their needs in terms of usability requirements: researchers, developers and final users.

A. Researchers Needs

SSRE are often developed by researchers working in large scale international research projects. This is also the case with GaiusT. The system evolved from a research project which goal was to design and implement semantic annotation tools, and the final framework, named Cerno, was a multilingual semantic annotation system [12]. The Cerno project started about ten years ago [13], and since the beginning researchers working on it were distributed on different organizations and countries. Application of Cerno to legal documents required a re-design and a deep revision of the modules of Cerno and the development of new ones [3], and to support distributed collaboration recently, the framework has been extended by adding multiusers and groups management so that different users can work on the same project of annotation. Another requirement is that the SSRE had to deal with different conceptual models,. describing laws in a different abstraction level, as for example, the model used in GaiusT (Fig. 1) and the model used by Nomos (Fig. 2) [14] (for another model see, e.g., [15]).

These evolution exposed some usability needs that can be addressed with: agile documentation (to allow researchers to use the SSRE in an almost intuitive way); a glossary (to establish a common language and avoiding ambiguities and limiting misunderstandings); interfaces to the core modules to support the definition of tagging rules for the new conceptual model; modules for the analysis and comparison of output produced in all the steps of the requirements extraction process. In GaiusT a set of modules have been developed to support the preprocess of legal documents, through statistical and semantic analysis of texts, using lexical resources or ontologies such as WordNet (http://wordnet.princeton.edu/) or Google N-gram (https://books.google.com/ngrams/), the construction of the conceptual model, the structure elements definition and the analysis of tagged documents (an example is given in Fig. 3).

B. Developers Needs

Developers of a SSRE have not only to implement solutions for a variety of tasks, but they also have to choose among a large number of platforms, technologies, libraries, programming languages, standards, etc. These choices have a strong impact on the efficiency component of usability, that is on development time and costs. The most critical decisions made for the GaiusT project, are related to the need of integrating modules, libraries and new software and, whenever possible,

of using open source resources. Their mesh-up resulted more challenging than expected. For example, available off-the-shelf modules often turn out to be incompatible, or with low performances, or did not implement all the needed features.

For GaiusT, C# and not Java was chosen to develop on the Windows platform. Both languages hide the complexity of the best-performing programming language C++, but C# is more integrated with the operative system, facilitating I/O activities, accesses to HW resources and the native creation and integration of databases. Java is also multi-platform, but performances are lower [16]. Parsing modules in Cerno were in TXL (www.txl.ca), but it was less scalable than expected to process long documents and regular expressions were then used. The extraction of texts from I/O files in different formats resulted almost straightforward and is based on existing libraries.

More critical are the functions and tools for the linguistic analysis. As open source resource are most of the time available in Java source, if they need to be integrated in a platform that does not support such language, they need to be developed from scratch. This was the case, for example, of the module to lemmatize words and to deal with declension of verbs and noun for English and Italian language. Technical needs, communication and collaboration with and among the researchers, required the migration of GaiusT to a Web-based system. These migrations are risky, as guidelines for web development and for the choice of a web framework are not always available. In the first development phase of GaiusT, NodeJs was adopted (http://nodejs.org). Later, for maintenance and robustness reasons, it was migrated to another framework, ASP:NET MVC (http://www.asp.net/mvc). To his end it was necessary to re-install and re-customize everything, increasing time and costs (this migration required 2 out of a 12 man-months for the implementation of the web-based version of the system).

This experience confirmed that a web framework can speed-up development. However, such frameworks have a number of limitations and the main lessons learned are:

- the learning curve is flat, and costs to customize a framework are high;
- efforts to finalize the web application can be higher than the benefits gained through the framework;
- installing the framework (including a variety of programs) requires a lot of time;
- if you have adopted a given framework, exit barriers to change it are very high.

From our experience, general criteria to make a decision (which framework to adopt) are: rate of diffusion, existing versions, number of developers using it, blog and FAQ, documentation, portability, maintenance, client/server; do the framework run on the server you have to refer to.

C. Users Needs

A web-based version of a SSRE system also increases its usability in terms of ease of use. This aspect turned out critical for users needs in two main ways: (a) to interact with lawyers;

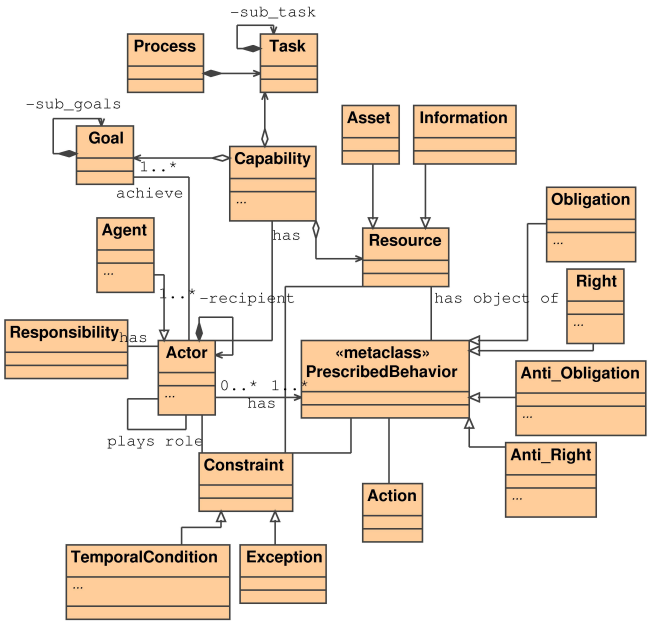

Fig. 1. Conceptual model for regulations used in the GaiusT framework.

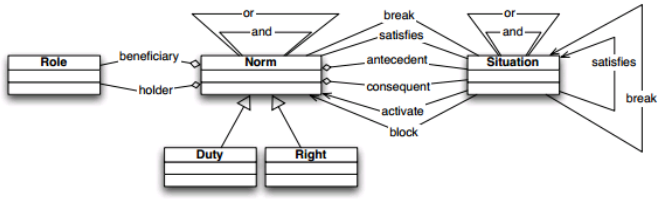

Fig. 2. Conceptual model for regulations used in the Nomos project.

(b) to set-up and realise experiments. In both cases, the goal is to gather feedback and data to improve the system or to compare different solutions to specific problems. Besides issues related to the usability of the different features of GaiusT, lawyers underlined the need to identify and use standard mark-up languages for legal documents, as LegalRuleML [17] of the OASIS organization (https://www.oasis-open.org/committees/membership.php?wg_abbrev=legalruleml). The language has been adopted in some SSRE projects [18], but there are risks related to its adoption, being it proposed by a consortium other than the W3C (www.w3c.org) consortium for the standards of the Web (http://en.wikipedia.org/wiki/OASIS_(organization).

A particular class of users of SSRE are the participants of experiments, often students whose role and needs are similar to those of (junior) requirements analysts. For these users, needs are also related to the support of the analysis and tagging of legal texts. Manual inputs have also to be supported in many steps of the requirements extraction process and a (graphical) visualization of intermediate and final output.

IV. Conclusion

Usability for SSRE can be enforced with an early focus on the needs of three main roles, researchers, developers and

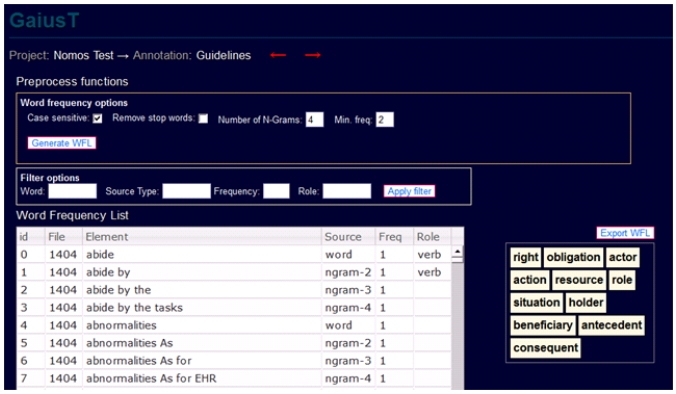

Fig. 3. The interface of the word frequency analysis module.

users. Concerns demanding attention for each of these categories of users have been highlighted and described referring to the experience gained with the design and development of a large scale framework for SSRE.

The main benefits of usability - increased productivity, decreased training and support costs, reduced development time and costs [19], [20] – are relevant (maybe even more attractive) for SSRE research projects.

Future work should include the evaluation of the usability with the application of usability methods and techniques [21], [22].

Acknowledgment

This research was partially supported by the ERC advanced grant 267856 'Lucretius: Foundations for Software Evolution' and by VISCOSO is financed by the Autonomous Province of Trento through the "Team 2011" funding programme.

References

[1] A. Parasuraman, V. A. Zeithaml, and L. L. Berry, "A conceptual model of service quality and its implications for future research," *The Journal of Marketing*, vol. 49, no. 4, pp. 41–50, 1985. [Online]. Available: http://dx.doi.org/10.2307/1251430

[2] M. Imai, *Kaizen: The Key to Japan's Competitive Success*. New York, NY: McGraw-Hill, 1986.

[3] N. Zeni, L. Mich, J. Mylopoulos, and J. R. Cordy, "Applying gaiust for extracting requirements from legal documents," *2013 6th International Workshop on Requirements Engineering and Law (RELAW)*, vol. 0, pp. 65–68, 2013.

[4] N. Zeni, N. Kiyavitskaya, L. Mich, J. Cordy, and J. Mylopoulos, "Gaiust: supporting the extraction of rights and obligations for regulatory compliance," *Requirements Engineering*, pp. 1–22, 2013. [Online]. Available: http://dx.doi.org/10.1007/s00766-013-0181-8

[5] I. O. for Standardization, *ISO 9241-11: Guidance on Usability*. International Organization for Standardization, 1998. [Online]. Available: http://books.google.it/books?id=TzXYZwEACAAJ

[6] I. O. for Standardization, *ISO/IEC 9126-1, Software Engineering – Software product quality – Part 1: Quality Model*. International Organization for Standardization, 2001. [Online]. Available: http://books.google.it/books?id=TzXYZwEACAAJ

[7] V. Zeithaml, A. Parasuraman, and L. Berry, *Delivering Quality Service: Balancing Customer Perceptions and Expectations*. Free Press, 1990. [Online]. Available: http://books.google.it/books?id=RWPMYP7-sN8C

[8] J. Juran and F. Gryna, *Quality planning and analysis: from product development through use*. McGraw-Hill, 1980. [Online]. Available: http://books.google.it/books?id=Gt5TAAAAMAAJ

[9] D. Gause and G. Weinberg, *Exploring requirements: quality before design*. Dorset House Pub., 1989. [Online]. Available: http://books.google.it/books?id=QedTAAAAMAAJ

[10] N. Kiyavitskaya, N. Zeni, T. D. Breaux, A. I. A. n, J. R. Cordy, L. Mich, and J. Mylopoulos, "Automating the extraction of rights and obligations for regulatory compliance," in *ER*, 2008, pp. 154–168.

[11] N. Zeni, N. Kiyavitskaya, L. Mich, J. Cordy, and J. Mylopoulos, "Gaiust: supporting the extraction of rights and obligations for regulatory compliance," *Requirements Engineering*, pp. 1–22, 2013. [Online]. Available: http://dx.doi.org/10.1007/s00766-013-0181-8

[12] N. Kiyavitskaya, N. Zeni, J. R. Cordy, L. Mich, and J. Mylopoulos, "Cerno: Light-weight tool support for semantic annotation of textual documents," *Data & Knowledge Engineering*, vol. 68, no. 12, pp. 1470–1492, 2009.

[13] N. Kiyavitskaya, N. Zeni, J. R. Cordy, L. Mich, and J. Mylopoulos, "Applying software analysis technology to lightweight semantic markup of document text," in *ICAPR (1)*, 2005, pp. 590–600.

[14] A. Siena, I. Jureta, S. Ingolfo, A. Susi, A. Perini, and J. Mylopoulos, "Capturing variability of law with nó mos 2," in *ER*, 2012, pp. 383–396.

[15] R. Winkels and R. Hoekstra, "Automatic extraction of legal concepts and definitions," in *JURIX*, 2012, pp. 157–166.

[16] A. H. Yousef, T. A. El-lateef, and M. F. Ismail, "Benchmarking and performance enhancement framework for multi-staging object-oriented languages," *Ain Shams Engineering Journal*, vol. 4, no. 2, pp. 241 – 257, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S209044791200072X

[17] T. Athan, H. Boley, G. Governatori, M. Palmirani, A. Paschke, and A. Wyner, "Oasis legalruleml," in *ICAIL*, 2013, pp. 3–12.

[18] M. Palmirani and F. Vitali, *Akoma-Ntoso for Legal Documents, in: Legislative XML for the Semantic Web: Principles, Models, Standards for Document Management*, ser. Law, Governance and Technology Series. Springer, 2011. [Online]. Available: http://books.google.it/books?id=2yBVDmy-MCkC

[19] J. Nielsen. (2008) Usability roi declining, but still strong. [Online]. Available: http://www.useit.com/alertbox/roi.html

[20] R. G. Bias and D. J. Mayhew, *Cost-Justifying Usability: An Update for the Internet Age*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2005.

[21] L. Olsina, F. Papa, and H. Molina, "How to measure and evaluate web applications in a consistent way," in *Web Engineering: Modelling and Implementing Web Applications*, ser. Human-Computer Interaction Series, G. Rossi, O. Pastor, D. Schwabe, and L. Olsina, Eds. Springer London, 2008, pp. 385–420. [Online]. Available: http://dx.doi.org/10.1007/978-1-84628-923-1_13

[22] L. Mich, M. Franch, and N. Kiyavitskaya, "A quality gaps roadmap for a website usability process," in *Computational Intelligence for Modelling Control Automation, 2008 International Conference on*, Dec 2008, pp. 169–174.