View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by AIS Electronic Library (AISeL)

Association for Information Systems
AIS Electronic Library (AISeL)

BLED 2016 Proceedings

BLED Proceedings

2016

A Framework for Traceability of Legal Requirements in the Dutch Governmental Context

Koen Smit
HU University of Applied Sciences Utrecht, Nijenoord 1, 3552 AS Utrecht, Netherlands, koen.smit@hu.nl

Martijn Zoet
Zuyd University of Applied Sciences, martijn.zoet@zuyd.nl

Matthijs Berkhout
HU University of Applied Sciences Utrecht, matthijs.berkhout@hu.nl

Follow this and additional works at: http://aisel.aisnet.org/bled2016

Recommended Citation

Smit, Koen; Zoet, Martijn; and Berkhout, Matthijs, "A Framework for Traceability of Legal Requirements in the Dutch Governmental Context" (2016). *BLED 2016 Proceedings*. 38.
http://aisel.aisnet.org/bled2016/38

This material is brought to you by the BLED Proceedings at AIS Electronic Library (AISeL). It has been accepted for inclusion in BLED 2016 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

29th Bled eConference

Digital Economy

June 19 - 22, 2016; Bled, Slovenia

A Framework for Traceability of Legal Requirements in the Dutch Governmental Context

Koen Smit

HU University of Applied Sciences Utrecht, Nijenoord 1, 3552 AS Utrecht, the Netherlands, koen.smit@hu.nl

Martijn Zoet

Zuyd University of Applied Sciences, Havikstraat 5, 6135 ED Sittard, the Netherlands, martijn.zoet@zuyd.nl

Matthijs Berkhout

HU University of Applied Sciences Utrecht, Nijenoord 1, 3552 AS Utrecht, the Netherlands, matthijs.berkhout@hu.nl

Abstract

In the past decades, research and practice focused a lot of attention towards traceability in the context of software requirements, food supply chains, manufacturing, and aviation industry. As legislation and regulations in software systems become increasingly relevant, traceability of legal requirements is of great importance. In this study, we aimed to create a framework in which the basis for traceability of legal requirements is addressed. To be able to do so we conducted five case studies at five Dutch governmental institutions, which was followed by a three-round focus group. The resulting framework comprises 22 (layered) traceability elements in relation to three domains that offers a reference model to determine how traceability can be applied in software system design, in the context of the Dutch government.

Keywords: *Traceability, Legal Requirements, Legal Sources*

1 Introduction

An industry that is influenced by changes in laws and regulations comprises the governmental institutions that deliver public administration services. As more and more public administration services are offered digitally, the need to trace the delivered services to their legal sources, laws and regulations, becomes more complex. This type of traceability is absent in most of the current public administration services (Van Engers & Nijssen, 2014).

To be able to create new or change existing public administration services that adhere to laws and regulations, these legal sources need to be interpreted and transformed from natural language into specifications for computer-executable business rules (van Engers & van Doesburg, 2015). These activities are often defined in a specific process to guide and structure the transformation of legal requirements into software systems. An example of this is the '*agile execution of law',* developed and employed by the Dutch Tax and Customs Administration (Boer & Van Engers, 2013). In these processes, traceability is a core capability.

However, currently, traceability is often of secondary importance when a public administration service is designed. This influences the transparency governmental institutions can provide when a service is delivered to a customer. The importance of traceability usually changes when errors are made providing the services (Van Engers & Nijssen, 2014). The main reason for this delay is that the implementation of adequate design and management of traceability often costs a significant amount of organizational resources, which should be justified by a proper business case (Cleland-Huang, Gotel, Huffman Hayes, Mäder, & Zisman, 2014). In this paper, we propose a traceability framework which enables governmental institutions to select what form of traceability and to what extend traceability should be implemented. Moreover, our traceability framework enables governmental institutions to choose what elements to utilize in software systems design, based on what elements are usual when addressing traceability of legal requirements. To be able to do so, we addressed the following research question: "*Which elements are useful to trace with regards to legal requirements in the context of the Dutch government*"

The remainder of this paper is organized as follows: First, we provide insights into how traceability is intertwined with software systems design and how it affects decision making as part of public administration (e-)services. This is followed by the research method used to construct the traceability framework. Furthermore, the collection and analysis of our research data are described. Subsequently, our results which led to our traceability framework are presented. Finally, we discuss which conclusions can be drawn from our results, followed by a critical view of the research methods utilized and results of our study and propose possible directions for future research.

2 Background and related work

In the previous decades, much research focused on traceability. For example, traceability in food supply chains (Opara, 2003) and manufacturing chains in the aviation industry (Ngai et al., 2007). Furthermore, traceability is utilized in the context of requirements (Gotel & Finkelstein, 1994), software artifacts (Gao, Zhu, Shim, & Chang, 2000), model-engineering (Jouault, 2005), jurisprudence (ECLI) (van Harten & Jansen, 2013), and the protection of copyrighted material (Staddon, Stinson, & Wei, 2001). As laws and regulations need to be transformed and processed into software systems of governmental institutions, we focus on traceability of legal requirements in the context of software artifact traceability.

Software artifact traceability is defined as: "*Software artifact traceability is the ability to describe and follow the life of an artifact (requirements, code, tests, models, reports, plans, etc.) developed during the software lifecycle in both forward and backward directions*" (Gotel & Finkelstein, 1994). A proper implementation of software artifact traceability can provide insights into system development and evolution, assisting in both top-down and bottom-up program comprehension, impact analysis, and reuse of existing software artifacts, and is therefore defined as a critical success factor in software development (Domges & Pohl, 1998). In this context, traceability knows two dimensions. The first dimension comprises vertical and horizontal relations. Horizontal relations refer to traceability relations that associate elements of the same type of artifact (i.e. relationships between facts) while vertical relations refer to associations from an artifact towards different type of artifacts (i.e. a relationship between a decision and its underlying business rule) (Lindvall, Tvedt, & Costa, 2003). The second dimension comprises pre and post-traceability, which is also referred to as forward and backward traceability (Gotel & Finkelstein, 1994). Pre-traceability refers to the relations between requirement specifications and the sources that have given rise to these specifications, i.e. the stakeholders that have expressed the views and needs which are reflected in them while post-traceability refers to the relations between requirement specifications and artifacts that are created in subsequent stages of the software development life cycle.

A lot of research is performed on software artifact traceability (Lucia, Marcus, Oliveto, & Poshyvanyk, 2012; Lucia, Fasano, Oliveto, & Tortora, 2007; Sundaram, Hayes, Dekhtyar, & Holbrook, 2010). However, a recent study by Cleland-Huang et al. (2014), who analyzed the knowledge base regarding software artifact traceability, still uncovered research directions that are not adequately covered by current research effort. For example, traceability strategizing, creation of intuitive forms of query mechanisms, and visualization of trace data. Their study resulted in a collection of research directions that are defined as useful for both complementations of the body of knowledge and applicability in practice. One of those research directions is that of the development of traceability reference models to guide the design of traceability solutions. Cleland-Huang et al. (2014) state that, to date, most research on traceability reference models focused on the creation of a reference model for standard (generalized) projects. According to (Ramesh & Jarke, 2001), a traceability reference model can be defined as: "*A traceability reference model specifies the permissible artifact types and permissible link types that can form a trace on a project, and is derived from an analysis of the queries that the resulting traceability is intended to answer.*" The problem with most of the currently proposed traceability reference models is that none of them are universally accepted of widely used in industry, due to the fact that most of them are too general of nature (Cleland-Huang et al., 2014). An example of a traceability reference model which is tailored for application in a specific domain is the work of Katta (2012), which proposed a traceability reference model for use in the highly-regulated nuclear domain. One of the key factors of its acceptance by the industry was that the creation and tailoring of the traceability reference model were driven by the industry itself.

This particular study was initiated and driven by five executive governmental institutions. These institutions are responsible for delivering public administration services. Due to this, traceability between software systems and legal sources is an important component in their software development lifecycle. An example of a Dutch public administration service which is offered as an e-service would be, on a yearly basis, the declaration of taxes. For this e-service, it is essential that the decision-making is transparent and thus, all components that are part of the e-service are linked to legal sources. This ensures a legally valid execution of decision-making that is supported by software systems and/or executed in a fully automated manner. In this study, we define a legal source as a source of law or regulation, stated by supranational, national, regional or local stakeholders within the legal rights to do so (Tarantino, 2008). Examples of legal sources are 1) international treaties on human rights, 2) the European Community Law, 3) national laws and regulations, 4) civil rights, and 5) internal policies. Moreover, we also utilize the concept of a legal requirement, which we define as a requirement that is extracted from a legal source which influences software system design. Legal requirements are different from conventional software requirements in three distinct ways (Breaux, 2009): 1) legal requirements govern multiple industries, goods, and services, whereas traditional practice focuses on software requirements target specific systems, 2) Legal requirements are not elicited by engineers from stakeholders, they are codified in legal language and interpreted therefrom, and 3) Ambiguity cannot be removed from legal requirements by software engineers, it can only be classified and interpreted in the context of organizational practices, goods, and services. An example of a method that is tailored to the definition of legal requirements based on legal sources is the Frame-Based Requirements Analysis Method (FBRAM), see (Breaux & Antón, 2007) and (Breaux, 2009).

3 Research method

The goal of this research is to propose a validated traceability framework which can guide the design of the traceability capability at governmental institutions. In addition to the goal of the research, also, the maturity of the research field is a factor in determining the appropriate research method and technique(s). In this study, traceability is considered in combination with the research field of legal requirements. The maturity of the traceability research field, in general, is very mature. Still, research on traceability reference models is less mature (Cleland-Huang et al., 2014). The research areas of legal requirements and business rules management, in general, is less mature to nascent (Kovacic, 2004; Nelson, Peterson, Rariden, & Sen, 2010; Anonymous, 2014). Focus of research in nascent research fields should lie on identifying new constructs and establishing relationships between identified constructs (Edmondson & Mcmanus, 2007). Summarized, to accomplish our research goal, a research approach is needed in which elements that should be traced and the actual traces are explored and combined into one traceability framework. To achieve our goal, we analyze traceability demands regarding legal requirements in five case studies at five governmental institutions. Based on this round of data collection a traceability framework is constructed and proposed. Then, to increase the generalizability of the traceability framework, three rounds of validation are conducted in the form of a focus group where experts of all five case study organizations participated.

Case study research is selected so that the researchers were able to gather data on how traceability is implemented. Therefore, the case studies are exploratory of nature. The organizations are selected from a pool of Dutch governmental institutions that provide public administration services based on laws and regulations that are provided by the Dutch legislative governmental branches. Our study comprised a holistic case study approach, featuring one context, traceability of legal requirements, and five cases within this context. The unit of analysis are the traceability demands of the individual case organizations. As the case study approach is exploratory of nature, the data collection and analysis consisted of secondary data and semi-structured interviews, which is a combination of first and third-degree data collection. This approach has several advantages and is thoroughly discussed in (Runeson & Höst, 2009).

Adequate research methods to explore a broad range of possible ideas and/or solutions to a complex issue and combine them into one view when a lack of empirical evidence exists consist of group-based research techniques (Delbecq & Van de Ven, 1971; Okoli & Pawlowski, 2004; Ono & Wedemeyer, 1994). Examples of group based techniques are Focus Groups, Delphi Studies, Brainstorming and the Nominal Group Technique. The main characteristic that differentiates these types of group-based research techniques from each other is the use of face-to-face versus non-face-to-face approaches. Both approaches have advantages and disadvantages, for example, in face-to-face meetings, provision of immediate feedback is possible. However, face-to-face meetings have restrictions with regard to the number of participants and the possible existence of group or peer pressure. To eliminate the disadvantages, we combined the face-to-face and non-face-to-face technique by means of applying case studies and three focus group meetings.

4 Data collection and analysis

Data for this study is collected over a period of six months, between August 2014 to February 2015, through five case studies and three rounds of focus groups. Between each round of the focus group, researchers consolidated the results. Both methods of data collection are further discussed in the remainder of this section.

4.1 Case Studies

Over a period of three months, between August 2014 and November 2014, five case studies were conducted by a group of seven researchers. The case studies were performed in two phases. The first phase comprised the collection and analysis of secondary data. The second phase comprised the semi-structured interviews. The selection of the participants should be based on the group of individuals, organizations, information technology, or community that best represents the phenomenon studied (Strauss & Corbin, 1990). For this study, the phenomenon studied is represented by organizations and individuals that deal with traceability of legal requirements. Such organizations are often financial and government institutions. The organizations that agreed to cooperate with the focus group meetings were the: 1) Dutch Tax and Customs Administration, 2) Dutch Immigration and Naturalization Service, 3) Dutch Employee Insurance Agency, 4) Dutch Education Executive Agency, Ministry of Education, Culture and Science, and 5) Dutch Social Security Office.

First, the experts of the case study organizations were prompted to gather and send all relevant and available documentation to the research team to analyze in advance of the semi-structured interviews. As this yielded a large amount of secondary data, the researchers needed a month to structure the data so that it was understood by the researchers and that it could serve as a basis for the semi-structured interviews, in terms of topics to be discussed.

Second, we conducted two semi-structured interviews with subject-matter experts at each case organization. The subject-matter experts were in all cases responsible for the traceability capability at the case organization and had more than five years of experience. Based on our findings from the first phase, an interview protocol was followed, comprising the following questions: 1) “Are all elements and traces described correctly?”, “2) Do I want to remove an element or a trace?” 3) “Do we need additional elements or traces?“, and 4) “Does the element or trace contribute to the traceability of legal requirements throughout software systems design?” The interviews were all audio-taped and were protocolled and consolidated on the same day.

4.2 Focus Group

Subsequently to the case studies, the focus groups were prepared and conducted between November 2014 to February 2015. Before a focus group is conducted, first, a number of key issues need to be considered: 1) the goal of the focus group, 2) the selection of participants, 3) the number of participants, 4) the selection of the facilitator, 5) the information recording facilities, and 6) the protocol of the focus group.

The goal of the focus group was to assemble and validate a traceability framework. We utilized the same selection of Dutch governmental institutions which collaborated in the case study phase, also to increase generalizability. Based on the written description of the goal and consultation with employees of each government institution, participants were selected to take part in the three focus group meetings. In total, thirteen participants took part who fulfilled the following positions: four business rule architects, three business rule analysts, two project managers, one IT architect, one enterprise architect, one software engineer, and one tax advisor. Each of the participants had, at least, five years of experience with traceability and traceability issues in practice. Each focus group was chaired by one experienced facilitator. Besides the facilitator, five additional researchers were present during the focus group meetings. One researcher participated as 'back-up' facilitator, who monitored if each participant provided equal input, and if necessary, involved specific participants by asking for more in-depth elaboration on the subject. The remaining four researchers acted as a minute's secretary taking field notes. They did not intervene in the process. All focus groups were video and audio recorded. A focus group meeting took on average one and a half hours. Each focus group meeting followed the same overall protocol, each starting with an introduction and explanation of the purpose and procedures of the meeting, after which ideas were generated, shared, discussed and/or refined.

Prior to the first round, participants were informed about the purpose of the focus group meeting and were invited to study the traceability model of their corresponding organization, derived from the case study results. In addition, the first version of the traceability framework that was constructed from the results of the case studies was also included. All participants were asked to bring any comments, which came up while studying the results, with them to the first focus group meeting. The first round started with the presentations of the individual traceability models derived from the case study results. After the individual presentations, participants discussed the usefulness of each traceability element. Also, additional traceability elements were proposed. For each proposed traceability element, the name, description, rationale, domain, and organization-specific examples were discussed and noted. After the first focus group, the researchers consolidated the results. Consolidation comprised the construction of the first version of the traceability framework and detection of double traceability elements (conceptually equal). The results of the consolidation were sent to the participants of the focus group two weeks in advance for the second focus group meeting. During these two weeks, the participants assessed the consolidated results in relationship to four questions: 1) "Are all elements and traces described correctly?", "2) Do I want to remove an element or a trace?" 3) "Do we need additional elements or traces?", and 4) "Does the element or trace contribute to the traceability of legal requirements throughout software systems design"?" This process of conducting focus group meetings, consolidation by the researchers and assessment by the participants of the focus group was repeated two more times (round 2 and round 3). After the third focus group meeting (round 3), saturation within the group occurred, leading to a consolidated traceability framework.

5 Results

In this section, the results of the case studies and the focus group are presented. First, we report on the results of the case studies conducted. This is followed by the results from the comparative analysis in which the case study results are compared. Lastly, we report on the results of the focus group meeting, which had the goal to validate our findings and come to a traceability framework a basis for traceability of legal requirements in software systems.

5.1 Case Study Results

As mentioned in section three, five case studies were conducted. Based on the analysis of both the secondary data and interview results a traceability map is created that visualizes the traceability elements deemed important per case study, see for example figure 1. To improve the readability of this section, we label the Dutch Tax and Customs Administration as case A, the Dutch Immigration and Naturalization Service as case B, the Dutch Employee Insurance Agency as case C, the Dutch Education Executive Agency as case D, and the Dutch Social Security Office as case E. In our results we refer to elements and traces as a singular form, while, in practice, it is possible that elements are referred to in the plural form.

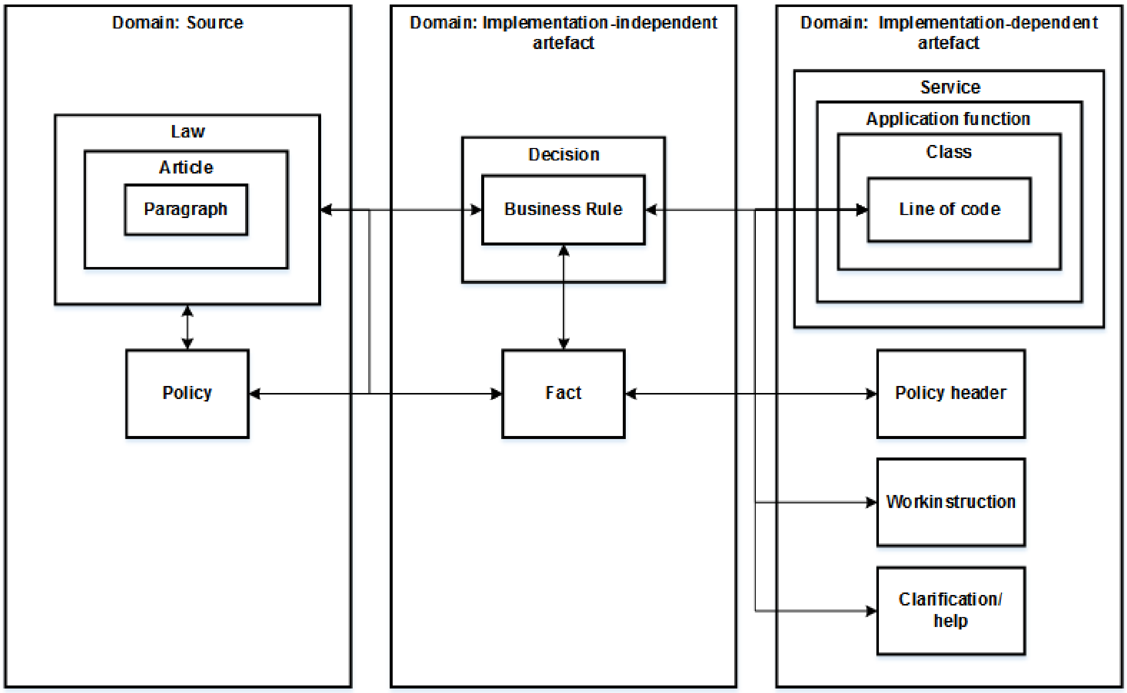

Figure 1: Example traceability model of the Dutch Education Executive Agency.

A similarity that we identified was that all five cases utilize three domains in which elements are managed and traces are implemented. Additionally, all five case organizations utilize those domains to trace between as well. The first domain is the source domain. This domain comprises the laws and regulations as defined by the legal writers of the house of representatives of the Netherlands. The second domain is defined as the implementation-independent artifact domain. This domain comprises artifacts that are established without incorporating language or properties that are affiliated to the use of specific technology (i.e. from specific vendors). The third domain comprises the implementation-dependent artifacts domain. This domain utilizes, for example, vendor specific instantiations of artifacts. An example of this would be the use of knowledge models specifically created and used in the application BeInformed.

In total, the results of the case studies identified multiple similarities and differences between the involved case organizations. However, due to space limitations, we do not cover each individual difference but summarize the differences. In summary, there were seven elements that were included by all case organizations, four elements that were included by all but one case organization, two elements that were included by all but two case organizations, and four elements were included by two of five case organizations.

Further summarized, our findings show some noteworthy design decisions by the case organizations regarding traceability demands. First, we identified a difference in the traceability towards laws and regulations in the source domain. Case A and B reported to trace to the lowest level possible; individual words, whilst case C, D and E report to trace on the level of paragraphs. Case A and B indicate to require these extra levels of traceability due to the fact that both organizations need process less structured laws and regulations compared to case C, D, and E (i.e. often lacking structuring in articles or paragraphs). Moreover, case A also required lower levels of traceability to be able to compare words as concepts in laws and regulations.

Case A and C trace business rules, while case B, D and E utilize decisions as parent elements for business rules which are also traced. Case A indicates to do so because it allows them to execute a more precise form of traceability. Case C motivation for this design decision is that they are still designing their solution and experimenting with the required precision of traceability. Case B, D, and E utilize decisions as parent level of business rules because it enables them to build business rule architectures with the purpose to structure a large amount of business rules as part of a decision.

Moreover, case C and E include a data-model in addition to the common vocabulary-model (i.e. an Entity Relationship Diagram). Case E needs to trace this element due to the fact that their software systems require a data-model in processing legal requirements and providing their public administration services.

Similarly, both case C and E include implementation-dependent data models, whilst case A, B, and D did not want to trace data related models in the implementation-dependent domain. Case E reported utilizing implementation-dependent data models for the execution of their public administration services, bound to a specific software system supplier.

Lastly, case B reported to not trace to either software systems, services, components, classes or a line of code, while case A, C, D, and E did express the necessity to trace to these elements. This is due to the fact that the chosen software system of case B is built upon design principles that do not adhere to layers as, for example, software systems, services, components. Also, case E was the only organization which reported to also trace towards process activities as part of their Business Process Management System due to their integration with a specific software system supplier.

5.2 Focus Group Results

Based on the case study results the researchers prepared the first focus group session. The goal of these focus group sessions was to, based on the participant’s input and feedback, assemble the traceability framework. Also, as described in section 4, the participants focused on further refinement of the elements to trace in terms of label and description and vertical traceability demands regarding the traceability framework.

The participants agreed in the first focus group round on the consolidated source domain. For this domain two traceability elements were split into different levels of elements to trace; delegated legislation and jurisprudence. Delegated legislation is added due to the fact that the executive organizations of the government are also able to extend or further define constraints for the implementation of laws and regulations. As this kind of regulation can influence how software systems are designed the executive organizations should be able to trace it. Jurisprudence is in this case defined as judgments or decisions by judges from various legislative levels. As these judgments or decisions can influence how the executive organizations should execute laws and regulations (i.e. by constraining them to judge negatively in specific situations which were previously allowed by law), jurisprudence should be traced as well.

Furthermore, little variety was identified regarding the elements in the implementation-independent domain. The participants agreed to split a traceability element into two elements; object model and use-case. An object model is utilized as an Entity Relationship Diagram, serving as a frame of reference how data is used in decision services by the executive organizations. The way the data is structured and used in decision making affects software systems design, and thus should be traced. Furthermore, use cases are important to trace due to the fact that these contain specific end-user scenario's coupled with certain laws and regulations.

Moreover, the participants had the most discussion regarding the implementation-dependent domain. This was due to the fact that the software systems are very diverse (i.e. most suppliers impose self-developed languages or solutions). However, although most consensus amongst the participants was required for the traceability elements in this domain, no additional element was included op top of the elements deducted from the case studies.

5.3 Traceability Framework

To select the elements to be included in the final traceability framework, multiple methods of agreement can be applied, for example, nominal comparison, ordinal comparison or narrative appraisal. In our research, we applied the method of agreement to compare the different cases and to be traced elements. However, a traceable element was added to the framework even it occurred only once. The reason for this is that the framework provides organizations all possibilities to choose from. Therefore also situations that occur only once in the selected organizations can be applicable to other organizations. The final traceability framework derived from the focus groups is built out of three domains, which are elaborated upon in section 5.1. Summarized, each of the domains comprises three or more high-level traceability elements which we will elaborate in this subsection if not already addressed in the previous subsections. Regarding the source domain, a policy refers to internal procedures or protocols inherent to the specific organization.

Regarding the implementation-independent domain, the high-level traceability elements are a scope, fact type model, object model, and use-case. A scope is defined as any unit of analysis, stated by the organization. Examples of this are a selection of business rules part of a specific decision service or one decision with all its underlying business rules. The number of contexts in a scope can vary but consists of a minimum of one context. A decision is built from one or more business rules. The fact type model serves as a domain model containing all possible terms that are utilized in decision making, which are labeled as facts.

Regarding the implementation-dependent domain, the high-level traceability elements are a software system, work instruction, and specification. The software system is built from one, but usually multiple (shared) services. Services are built from (shared) components. A component can be further dissected into classes, and on the lowest possible level, a Line of Code (LoC) that can be traced. The relationships between the different elements are all identical: many to many relationships. Summarized, the consolidated traceability framework is presented in Figure 2.

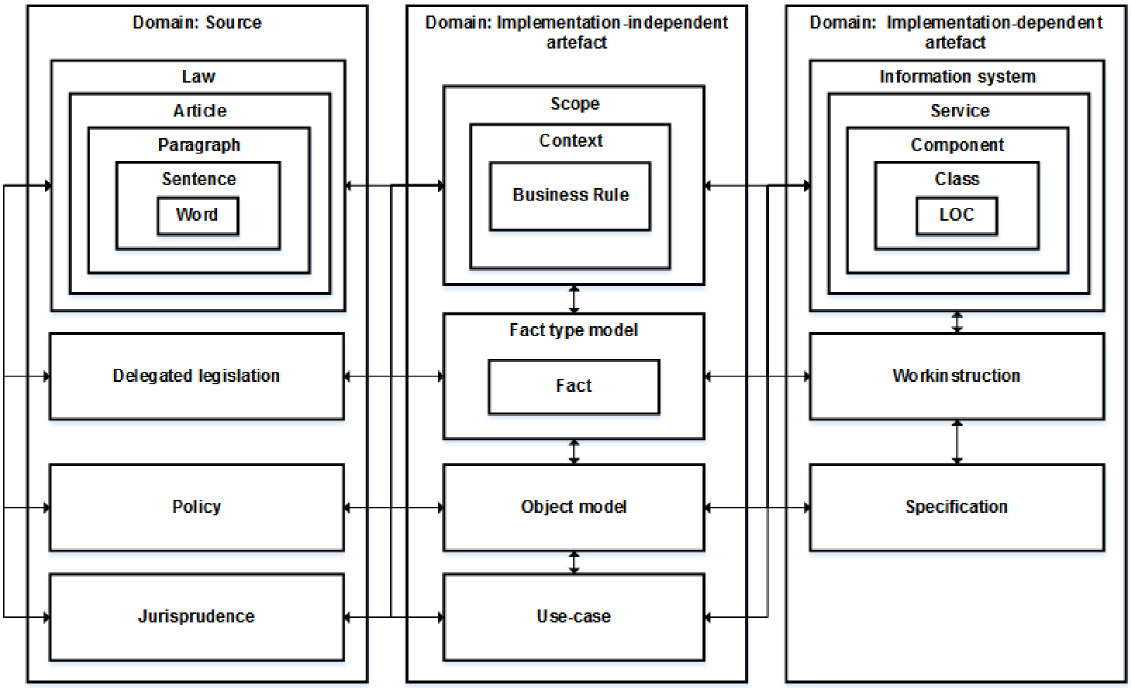

Figure 2: Final traceability framework.

6 Discussion & Conclusion

In this paper, we aimed to find an answer to the following question: "*Which elements are useful to trace with regards to legal requirements in the context of the Dutch government*". To accomplish this goal, we conducted a study conducting five case studies and a three round focus group. Both were applied to retrieve traceability elements from participants, 41 in total, employed by five executive governmental agencies. Our rounds of data collection and analysis resulted in a traceability framework which can be utilized when designing or improving the traceability capability of governmental organizations that execute laws and regulations. From a research perspective, our study provides a fundament for traceability principles and traceability elements focused on the implementation of laws and regulations in software systems design. From a practical perspective, executive governmental organizations could utilize the results of this study to guide the (re)design of traceability of legal requirements in software systems. With this in place organizations can ensure the adequate level of transparency towards legislative branches of the government, judges and judicial systems, and no less significant, towards citizens and businesses. Furthermore, another practical implication of our results could be that the governmental organizations now have a common frame of reference to communicate when addressing traceability. Therefore, our proposed traceability framework can be useful when executive governmental branches need to collaborate in a single chain of services.

Several limitations may affect our results. The first limitation concerns the sampling and sample size. The sample group of case organizations and participants is solely drawn from government institutions in the Netherlands. While we believe that government institutions are representative for organizations implementing traceability of legal requirements to implementation systems design, further generalization towards non-governmental organizations, amongst others, is recommended. Taken the sample size of five case studies and 41 participants into account, this number needs to be increased in future research. This research focused on identifying new constructs and establishing relationships given the current maturity of the traceability research field. Although the research approach chosen for this research type is appropriate, research focusing on further generalization should apply different research methods, such as quantitative research methods, which also allow us to incorporate larger sample sizes to validate our findings.

References

Boer, A., & Van Engers, T. (2013). Legal knowledge and agility in public administration. *Intelligent Systems in Accounting, Finance and Management*, *20*(2), 67–88.

Breaux, T. (2009). *Legal requirements acquisition for the specification of legally compliant information systems*. ProQuest.

Breaux, T. D., & Antón, A. I. (2007). A Systematic Method for Acquiring Regulatory Requirements: A Frame-Based Approach. In *Sixth International Workshop on Requirements for High Assurance Systems (RHAS-6)*.

Cleland-Huang, J., Gotel, O. C., Huffman Hayes, J., Mäder, P., & Zisman, A. (2014). Software traceability: trends and future directions. In *Proceedings of the on Future of Software Engineering* (pp. 55–69). ACM.

De Lucia, A., Marcus, A., Oliveto, R., & Poshyvanyk, D. (2012). Information retrieval methods for automated traceability recovery. In *Software and systems traceability* (pp. 71–98). Springer London.

Delbecq, A. L., & Van de Ven, A. H. (1971). A group process model for problem identification and program planning. *The Journal of Applied Behavioral Science*, *7*(4), 466–492.

Domges, R., & Pohl, K. (1998). Adapting Traceability Environments to Project Specific Needs. *CACM*, *41*(12), 54–62.

Edmondson, A. C., & Mcmanus, S. E. (2007). Methodological Fit in Management Field Research. *Proceedings of the Academy of Management*, *32*(4), 1155–1179.

Gao, J., Zhu, E. Y., Shim, S., & Chang, L. (2000). Monitoring software components and component-based software. In *The 24th Annual International Computer Software and Applications Conference* (pp. 403–412). IEEE.

Gotel, O. C., & Finkelstein, A. C. (1994). An analysis of the requirements traceability problem. In *Proceedings of the First International Conference on Requirements Engineering* (pp. 94–101). IEEE.

Jouault, F. (2005). Loosely coupled traceability for ATL. In *Proceedings of the European Conference on Model Driven Architecture (ECMDA) workshop on traceability* (pp. 29–37).

Katta, V., & Stålhane, T. (2012). *Traceability of safety systems: approach, meta-model and tool support*.

Kovacic, A. (2004). Business renovation: business rules (still) the missing link. *Business Process Management Journal*, *10*(2), 158 – 170.

Lindvall, M., Tvedt, R. T., & Costa, P. (2003). An empirically-based process for software architecture evaluation. *Empirical Software Engineering*, *8*(1), 83 – 108.

Lucia, A. D., Fasano, F., Oliveto, R., & Tortora, G. (2007). Recovering traceability links in software artifact management systems using information retrieval methods. *ACM Transactions on*

Software Engineering and Methodology (TOSEM), *16*(4), 13–23.

Nelson, M. L., Peterson, J., Rariden, R. L., & Sen, R. (2010). Transitioning to a business rule management service model: Case studies from the property and casualty insurance industry. *Information & Management*, *47*(1), 30–41. http://doi.org/10.1016/j.im.2009.09.007

Ngai, E. W. T., Cheng, T. C. E., Lai, K. H., Chai, P. Y. F., Choi, Y. S., & Sin, R. K. Y. (2007). Development of an RFID-based traceability system: Experiences and lessons learned from an aircraft engineering company. *Production and Operations Management*, *16*(5), 554–568.

Okoli, C., & Pawlowski, S. D. (2004). The Delphi method as a research tool: an example, design considerations and applications. *Information & Management*, *42*(1), 15–29.

Ono, R., & Wedemeyer, D. J. (1994). Assessing the validity of the Delphi technique. *Futures*, *26*(3), 289–304.

Opara, L. U. (2003). Traceability in agriculture and food supply chain: a review of basic concepts, technological implications, and future prospects. *Journal of Food Agriculture and Environment*, *1*(1), 101–106.

Ramesh, B., & Jarke., M. (2001). Toward reference models of requirements traceability. *IEEE Transactions on Software Engineering*, *27*(1), 58–93.

Runeson, P., & Höst, M. (2009). Guidelines for conducting and reporting case study research in software engineering. *Empirical Software Engineering*, *14*(2), 131–164.

Staddon, J. N., Stinson, D. R., & Wei, R. (2001). Combinatorial properties of frameproof and traceability codes. *IEEE Transactions on Information Theory*, *47*(3), 1042–1049.

Strauss, A., & Corbin, J. M. (1990). *Basics of qualitative research: Grounded theory procedures and techniques*. Sage Publications, Inc.

Sundaram, S. K., Hayes, J. H., Dekhtyar, A., & Holbrook, E. A. (2010). Assessing traceability of software engineering artifacts. *Requirements Engineering*, *15*(3), 313–335.

Tarantino, A. (2008). *Governance, Risk, and Compliance Handbook: Technology, Finance, Environmental, and International Guidance and Best Practices*. John Wiley & Sons.

van Engers, T. M., & van Doesburg, R. (2015). At your service, on the definition of services from sources of law. In *Proceedings of the 15th International Conference on Artificial Intelligence and Law* (pp. 221–225).

van Engers, T., & Nijssen, S. (2014). Connecting People: Semantic-Conceptual Modeling for Laws and Regulations. *Electronic Government*, 133–146.

van Harten, H. J., & Jansen, R. H. (2013). How do Judges Learn EU Law? A Dutch Narrative. *Review of European Administrative Law*, *6*(2), 57–83.