# Capturing and Sharing Domain Knowledge with Business Rules

## Lessons Learned from a Global Software Vendor

Walid Maalej
University of Hamburg
Hamburg, Germany
maalej@informatik.uni-hamuburg.de

Smita Ghaisas
Tata Consultancy Services
Pune, India
smita.ghaisas@tcs.com

***Abstract*—Business rules represent constraints in a domain, which need to be taken into account either during the development or the usage of a system. Motivated by the knowledge reuse potentials when developing systems within the same domain, we studied business rules in a large software company. We interviewed 11 experienced practitioners on how they understand, capture, and use business rules. We also studied the role of business rules in requirements engineering in the host organization. We found that practitioners have a very broad perception for this term, ranging from flows of business processes to directives for calling external system interfaces. We identified 27 types of rules, which are typically captured as a free text in requirements documents and other project documentation. Practitioners stated the need to capture this tacit form of domain knowledge and to trace it to other artifacts as it impacts all activities in a software engineering project. We distill our results in 17 findings and discuss the implications for researchers and practitioners.**

***Index Terms*—Requirements Knowledge, Business Rules, Empirical Studies, Software Documentation, Domain Knowledge**

## I. Introduction

Business rules are statements that define or constrain particular aspects of the business [9]. They are often used to specify the behavior of a system under development or how it should be used. Business rules are also used to capture constraints and conditional instructions, such as an allowed input range or an important action that must be performed when an event occurs. From the mid-1990s, the information systems and business- process management communities have discovered the potential of systematically capturing and managing business rules. Several initiatives have focused on formalizing business rules, resulting in either rule definition languages such as RuleML [3], Semantic Web Rule Language (SWRL) [11], and extensions of the Unified Modeling Language (UML) [20], or in tools for defining and maintaining business rules such as SAP Netweaver Business Rules component [17] and Be Informed [19].

In the software and requirements engineering (RE) communities, business rules have been mainly studied with the purpose of developing rule engines and rule-based systems that aim to facilitate the "business driving IT" vision [13]. However, there has been less emphasis on capturing the knowledge contained in business rules and using it to facilitate documentation or communication in software projects. In this paper, we study how software, in particular requirements engineering practitioners, understand, capture, maintain, and use the knowledge contained in business rules.

Our work was motivated by two observations, which we made in the company that hosted this study. First, we observed in a large insurance project that a minor change in one business rule (a modification in the allowed growth rate of insurance funds as determined by a country law) resulted in a significant ripple effect and maintenance effort, which lasted for 6 months. This rule was described in numerous documents and its implementation was scattered across different parts of the system. Second, we observed that many rules are domain, region, or company specific rather than application specific. Automatically identifying and extracting the rules and tracing them to related project artifacts would increase reuse when developing a new application for the same domain, region, or company [7]. This is the long-term goal of this research.

To achieve automation, we must first study current practices and problems and carefully derive solution strategies. This paper reports on the findings from detailed interviews with eleven practitioners from a global software vendor based in Asia. We asked practitioners from different backgrounds, domains, and roles about their understanding of business rules, the way they capture and maintain rules, and whether they see any problems and potential improvement in current practices. We also studied participants RE practices and problems to put the answers in the overall project contexts.

The remainder of the paper is structured as follows. Section II summarizes previous related work. Section III introduces the study design, including the research questions, research method, and participants. Section IV and Section V summarize our findings: first on business rules, their definition, maintenance, and usage and then on RE practices and problems, which impact business rules. Finally, Section VI discusses the implications of the findings for practitioners and researchers, while Section VII concludes the paper.

## II. Related Work

Early works from the information system and business-process management communities focused on the definition, classification, extraction, and management of business rules.

**Definition.** According to the Business Rules Group definition [9], a business rule either asserts business structure, controls, or influences its behavior. Gottesdiener [8] defines

business rules as declarative, atomic, distinct, business oriented and business owned rules that are expressed in natural language. Ross [16] defines several dozens of atomic rule types and equates them to elements in the periodic table. Ceri and Fraternale [5] claim that business rules model the reaction to events that occur in the real world. Business rules are also considered as a requirement on condition or manipulation of data [18] and as a computational requirement that determines or affects how business is run [15].

**Classification:** The efforts to classify business rules took various viewpoints into account. Wieden et al. [21] proposed 15 different "semantically-oriented" rule types grouped into structural, behavioral, and managerial categories. Zoet et al. [22] proposed a business rule categorization that is aligned to the business process management lifecycle. Herbst et al. [10] argued that common data-oriented methods are insufficient and inconvenient for a complete modeling of business rules.

**Extraction.** Researcher previously suggested approaches to extract business rules from structured and unstructured text, motivated by the reuse potential of rules. Ali et al. [1] suggested an approach that takes rule repositories either in relational databases or text format as input and convert it into xml syntax by applying transformation method on SQL queries or a parsing and transformation method using xquery. Mahgoub et al. [12] integrated XML technology with Information Retrieval techniques to automatically select the most discriminative keywords for association rules generation and used Data Mining techniques for association rules discovery. Breaux and Antón [4] proposed a method to mine rule semantics for understanding legislative text.

**Management.** Spreeuwenberg et al. [19] suggested using controlled natural language and pattern sentences to enable the involvement of domain experts in system modeling and allowing to define and understand business rules. The authors discussed advantages of mapping pattern sentences with the underlying meta-model. They also highlighted challenges in implementing the approach for a larger audience and in dealing with variations of rule sentences. Döhring et al. [6] discussed the use of business rules in combination with an "eventing semantics" to introduce flexibility in workflows. Becker et al. [2] used business rules to model compliance requirements in the financial sector and presented an extension of the semantic process modeling language for managing the rules.

Despite the long-standing recognition for their business impact, business rules remain rather unpopular. Resch [14] concluded the need to know much more about business rules in order to unleash the full potential of rule management systems. The author suggested that research should start with the rationale view and this reveals many questions, which must be answered by sound empirical and experimental research.

The empirical studies that we came across, e.g., by Weiden et al. [21], Zoet et al. [22] or Herbst et al. [10] start with a hypothetical rule classification. None has attempted to probe what the practitioners' understand when they refer to rules. To our knowledge, there is no published study on how software and requirements practitioners perceive and maintain business rules. Our work is a step towards understanding this perception.

## III. Study Design

We summarize the research questions of this study, introduce the method followed, and describe the participants.

### *A. Research Questions*

The main goal of this study was to qualitatively explore how business rules are being used in software and requirements engineering projects and what are recurrent problems and potential improvements. This implies answering the following specific research questions:

- **RQ1:** How are business rules being perceived in practice and which types of business rules exist?
- **RQ2:** How do stakeholders capture and maintain business rules?
- **RQ3:** How are business rules used in software projects?
- **RQ4:** Which impact does RE practices and tools have on the usage and management of business rules?
- **RQ5:** Which problems related to RE affect the management and usage of business rules?

The goal of RQ1 is to eliminate ambiguities about business rules as different stakeholders might understand them differently. Answering RQ2 and RQ3 includes exploring practices, tools, and templates used by practitioners to define, maintain, and use to business rules. The last two research questions focus on exploring the relationship between RE practices and business rules.

### *B. Research Method*

To answer the research questions, we conducted semi-structured interviews with open questions together with a detailed literature survey (see Section II). We structured the interview questions along three sections:

1. About You: In this section, which lasted for about 10 minutes, we asked the participants to introduce themselves, their backgrounds, and experience. Moreover, we asked about the domain they work in, typical projects, customers, and team characteristics.
2. Requirements Engineering: In this section, we asked the participants about common requirements engineering practices, which tools are used, and whether there are problems related to these practices. This section lasted for about 20 minutes.
3. Business Rules: The last and longest section (60-70 minutes) was dedicated to business rules. We focused on how participants define a business rule and which variants they know, asking for example and experiences. We also asked about maintaining and using the rules and wheteher there are problems related to current practices and tools. In this section, we relflected on participants answers from Section 2 to put them into the RE context.

TABLE I. OVERVIEW ABOUT INTERVIEW PARTICIPANTS

| P# | Roles | Exper. (yrs) | Projects characteristics | Team | Domain |
|---|---|---|---|---|---|
| P1 | Domain consultant, business process manager, requirements reviewer | 13 | Large programs (6 years) on property and casualty insurance, personal insurance for a large company | 1500 people at peak time. 80 in RE | Insurance |
| P2 | Business process manager, domain consultant, program manager | 20 | Large nationwide insurance deployment for a European government, over 22 months | 350 people in Asia and Europe | Insurance |
| P3 | Developer, project manager, requirement analyst | 4 | Online services using specified technology stacks | 30 people, 2-3 for requirements documentation and analysis | Software Industry, Investment Banking |
| P4 | Developer, maintenance lead, project manager | 20 | Development of an electronic shares trading application, Business intelligence, reverse engineering, maintenance for an insurance company | Ranging from 100-300 in varied locations | Telecom, Insurance, Power |
| P5 | Developer, software architect, project manager | 12 | Documents and claims management for public institutions, Customer Relationship management for telecom companies | Ranging from 50-100 people | Finance, Telecom |
| P6 | Software architect, program manager, delivery manager | 10 | Infrastructure software for large telecom companies across the world, a large automotive company | 40-120 people | Automotive and Telecom |
| P7 | Program manager, product manager | 16 | Development and customization of a market infrastructure with 45 components, 5 offerings for numerous large banks | 30-50 people | Financial Services |
| P8 | Project manager, program manager | 21 | Product customization and delivery for seven banks | Ranges from 7/8 to 200/300 depends on extent of customization to be done | Financial Services |
| P9 | Product manager, presales lead | 17 | Customization of a financial services infrastructure | 1 pm to 100 person months | Banking and Financial Services |
| P10 | Tester, lead underwriter, program manager | 14 | Testing product suite for large banks in US | 30-35 people | Banking and Financial Services |
| P11 | Delivery center head | 18 | Large program management | 30-90 people | Financial Services |

The full list of the interview questions is available online at http://www.teamweaver.org/wiki/index.php?title=Business_rules_interviews Each interview lasted for about 90 minutes and was carried out either via phone or face-to-face. While one main interviewer (one of the authors) moderated the interview sessions, one or two other interviewers were present to take notes and ask for clarifications. This reduced the interviewer's bias. The main interviewer summarized the minutes along the questions within 48 hours. Then the other interviewer(s) and the participants were able to refine and extend the minutes. For the summary of the results, a statement that was observed twice was included and extended iteratively with quotes from other minutes.

To summarize the results, we first aggregated the statements along the research questions. Then, the authors independently tagged the answers of the participants and grouped quotes that seemed similar. In the subsequent iteration, we discussed the groups of the quotes and merged them. Finally, we revisited the interview minutes, identified similar statements, and merged them into the findings while taking care of preserving their meanings.

### C. Participants

Between June and September 2012, we interviewed 11 participants. The selection of the participants was based on two criteria. First, we aimed at getting participants from different – at least two – business units (i.e. industrial sectors) in the host organization. Second, we aimed at capturing different project roles and people with different background, and varied experience to increase the validity of our results. Table I summarizes the participants' details. Several participants, e.g., P2, P6, P7, and P9, had also worked in other organizations and reported on their experience in general.

## IV. FINDINGS ON BUSINESS RULES

We summarize the results of RQ1, RQ2, and RQ3.

### A. Definition of Business Rules and their Types

**Finding 1: Stakeholders have different perceptions of business rules depending on their roles and experience.**

While all participants agreed that business rules usually represent constraints and restrictions, there was a disagreement about the nature of restrictions and the characteristics of the business rules. For instance, P2 stated that business rules are constraints for a system that are *non-negotiable* and are driven by corporate policy or regulations. For example, investments in a single fund cannot exceed N% of total investment. P11 said that "repetitiveness" is a way to recognize rules. He cited the example of auctions at central banks. Any auction has a start-time, end-time, cut-off time, and weekly frequency. These parameters "*repeat* for each central bank with different values."

We noticed that a group of participants (P3, P4, and P8) focused more on the *system and product perspective* when describing business rules. They considered business rules as

TABLE II. TYPES OF BUSINESS RULES MENTIONED IN THE INTERVIEWS

| ID | Type | P1 | P1 | P2 | P3 | P4 | P5 | P6 | P7 | P8 | P9 | P10 | P11 | Stated # |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Validations rules & value ranges | | | x | x | x | x | | | x | x | | x | 7 |
| 2 | System/ application specific rules | x | x | x | | | x | | | | | x | | 4 |
| 3 | Calculation rules | | | x | x | x | | | | | | | x | 4 |
| 4 | Access control rules | | | x | x | | | | | | | x | x | 4 |
| 5 | External system interfaces | | | x | | x | x | | | | | | | 3 |
| 6 | Laws & regulations | | | | x | | x | | | | | x | | 3 |
| 7 | Sequencing/ control flow | | | | x | | | | | x | | x | | 3 |
| 8 | Geography specific rules | | | | | | | x | | | x | x | | 3 |
| 9 | Business process rules | x | x | x | | | | | | | | | | 3 |
| 10 | Procedural/ Operational rules | x | x | | | | | | | | | x | | 3 |
| 11 | User interface rules | | | x | x | | | | | | | | | 2 |
| 12 | Company policies | | | | x | | | | | | | x | | 2 |
| 13 | Time restrictions | | | | | | x | | | x | | | | 2 |
| 14 | Dynamic rules | | | | | | | | x | | x | | | 2 |
| 15 | Definitions | x | x | | | | | | | | | | | 2 |
| 16 | Data rules | | | x | | | | | | | | | | 1 |
| 17 | Non-functional rules | | | x | | | | | | | | | | 1 |
| 18 | Boundary scenarios | | | x | | | | | | | | | | 1 |
| 19 | Variation rules | | | x | | | | | | | | | | 1 |
| 20 | Ranking of items | | | | x | | | | | | | | | 1 |
| 21 | Mandatory constraints | | | | x | | | | | | | | | 1 |
| 22 | Industry standards | | | | | x | | | | | | | | 1 |
| 23 | Exceptions to known exceptions | | | | | x | | | | | | | | 1 |
| 24 | Usability rules | | | | | | x | | | | | | | 1 |
| 25 | Configuration rules | | | | | | | | x | | | | | 1 |
| 26 | Domain specific rules | | | | | | | | | | x | | | 1 |
| 27 | Dependency rules | | | | | | | | | | x | | | 1 |

conditions or constraints that are mandatory and affect certain product features. P4 compared a business rule to a software component, which has an input, an output, and some processing. The processing part typically must follow some rules. The input has certain conditions and restrictions. These correspond to the business rules. He gave the example of standards in coloring financial charts, depending on ranges of values, system interfaces, or how a system should behave. P8 focused more on the product perspective. He stated, "A business rule is anything that characterizes a product and *how* it works, such as screen behavior or batch processes."

Other participants had a more general understanding of rules: either from a *requirements* (P5, P9) or a *business process* perspective (P1, P7). P5 explained, "If the customer should, e.g., be able to modify her account, the corresponding business rule is to allow customer to modify some fields." P9 even claimed that every *requirement* is a high level rule "e.g., this process works this way in this geography." P1, P2, and P7 mainly reduced business rules to restrictions on *process* models. P7 stated, "A rule is an encapsulation of some aspect of how a business objective is to be achieved, e.g., trade acceptance should happen between 9am to 5 pm."

**Finding 2: Types of information that practitioners consider as business rules are listed in Table II.**

When asking about the types, participants mentioned in total 27 types of rules. Table II lists all types of business rules that were mentioned in the interviews and by how many participants. These types are not mutually exclusive and might overlap. The most frequently stated type was validation rules. An example stated by P3 was "A particular functionality should not be available on Sunday." The second most frequent type was system/application specific rules (e.g. restricting the deployment of the application and how it should be used), followed by calculation rules (i.e. how to calculate a particular value), access control rules, rules to use external system interfaces, and laws & regulations.

We were surprised that definitions, non-functional rules, and flows were stated as business rule types, as we expected them to be separate requirements concepts. P1 explained that definition rules are like concept specification, e.g., detailing

what is a vehicle for the insurance domain and which different types exist. Like operational rules and definitions rules are never implemented but are important for the comprehension and implementation of other rules" (P2). Non-functional rules include availability restrictions, performance rules, and the number of concurrent users at time. As example of a usability rule, P5 stated the US law that public systems must support people with disabilities. Sequencing/ control flow restricts the functionality of the system, its behavior or the process, e.g. "Unless users fill in the primary details, don't show them the screen to enter other secondary details."

*B. Capturing and Maintaining Business Rules*

**Finding 3: Business rules are embedded as free text anywhere in the project and domain documentation.**

All practitioners reported that rules are not necessarily documented in separate documents or separate sections in a document. They can be related to and embedded in a business process, a process step, functionality, a feature, a screen, a mockup, or a system component. For instance, P5 stated that there is "no separate section for business rules", and that these are "captured as part of functionality." P3 stated, "We can find rules as descriptions of features or a set of validation constraints." P1 and P2 stated that rules can be found in requirements specifications, source documents, process description, marketing brochures, regulations (such as telecommunication standards), or laws.

Participants mentioned a few exceptions. P6 said, "Rules are mainly documented in free text. Only in exceptional, safety critical cases they are modeled or formalized." P4 said, **"**At most in embedded systems project, they try to formally document rules to conduct formal verifications. This is done only if people's life is concerned." P2 stated, "A separate rule sheet is maintained only if the customer insists on this. Otherwise, process steps and rules are not segregated."

One participant (P1) mentioned a tool, which was introduced for maintaining rules. He said, "At the beginning we did not capture rules systematically. Later a new task force was introduced to systematize maintenance of rules. We decided to document and maintain the rules in a database." He argued that this helped to deal with the complexity of the domain. Moreover, the team originally envisioned an automatic maintenance of the system by maintaining the rules. However, the participant admitted that this never succeeded due to the new complexity introduced by formalizing the rules.

**Finding 4: Stakeholders rely on assumptions instead of externalized business rules.**

Six participants stated that, in particular in the early project phases, many business rules remain as assumptions and are neither captured nor discussed. P8 said, "We relied on a lot of assumptions. No business rules were in place, Actions and validations were not documented. A customer wanted to know the validations, but they were neither documented nor implemented. So it was a mess."

P5 explained, "Gaps between what is implemented and what the customer actually wants are rooted in missing rules. These are not captured until we get change requests and the actual development starts." P4 stated, "Constraints on how systems interact with each other are very critical in the telecom domain and often unknown or assumed. Even the customer cannot say much about this." P9 agreed that this applies to the banking domain too and explained, "To find out these rules we need to understand the system, try out its interfaces, what are the conditions, which authorization, and access control are there, when the data is available, which ranges are allowed etc. Many times these problems arise in the client acceptance test."

**Finding 5: The use of rule languages and rule engines is rather exceptional in practice.**

All participants except P1 agreed that no rule language or rule engine was found to be relevant, useful, or essential in practice. The reasons quoted were as follows: (1) the learning effort associated with formal languages, rule engines, and tools; (2) the lack of the "right" people to specify and formalize the rules; (3) the unwillingness of customer to try new things since "they view this as a risk"; (4) the cost of licenses; (5) the resistance to change; (6) the lack of tools (or time to identify the right tools); and (7) the maintenance effort of formal rules which might over-exceed the maintenance effort of source code. P1 stated that a tool (RuleXpress) was piloted during a project to build a repository of rules and maintain them. In his opinion, this was helpful as the rules were more precise.

*C. Usage of Business Rules*

**Finding 6: Business rules are captured and used not only in requirements engineering but also in other project activities.**

There was a general agreement among participants that rules get "discovered" and used in all phases of a project. While some participants (e.g. P2) claimed that everyone used rules without realizing, others (e.g. P6) were more specific in stating that rules were discovered during RE and used in implementation and testing phases. P7 claimed that "many rules were discovered during the implementation, and testing and were used in these phases as well." P9 explained that her team used rules for creating and updating product documentation. She also claimed that product configuration is "rule-driven" and enhancements requests were often about adding new rules. P8 concurred that product management teams look at rules to detect commonalities and "highlight them" for the next implementation.

**Finding 7: If captured correctly, business rules can reduce the complexity of domains and implementations.**

Seven participants (P1, P6, and P7-P11) expressed that the complexity of a domain can be simplified with the definition of business rules. P1 informed that the underwriter team in his project examined the code and found 150 rules implemented. The team then analyzed these rules and identified much overlap and unneeded complexity. They were able to reduce the 150 to only 12 rules. This was a "big saving, due to better business efficiency and easier maintenance of the code." P6 gave an example from the Telecom domain. When an SMS is sent, the system has to decide what message to be sent to whom, when,

and whether it should wait. Decisions about simultaneous notifications to multiple entities need to be made. All these aspects are captured in rules." Handling failure of an event (such as an SMS) is also determined by a set of rules. P11 cited an example from the banking domain. For fixed deposits there are global and local rules regarding periodicity, interest for different products, customer-specificity, and 30X360 rate factors in fixed bonds. All of these get encapsulated "nicely in rules." P7 gave an example of the market infrastructure for the banking domain. User behavior, conformance to current trend, user preferences, and country specifics are increasingly getting encapsulated in rules, similarly to how to dynamically handle peak loads of trade transactions.

**Finding 8: Externalized business rules facilitate sharing knowledge with new team members.**

In the host organization, people frequently move between units and projects. Five participants (P1, P2, P3, P4, P7) mentioned that business rules offer a concise mechanism to communicate existing requirements knowledge to new entrants in a team.

P1 expressed that in every annual review new people joining the project needed a 3-4 weeks of intensive training. After rule documentation, the time was reduced significantly to just 1 week. P3 claimed, "People who join projects later have big problems to read and understand 400 plus pages documents." P4 found that new people joining must understand how the whole system is supposed to work, but have very little time available for this task due to delivery deadlines. If business rules were documented, this process would be easier.

**Finding 9: Externalized business rules facilitate the reuse of domain knowledge across the projects.**

The organization that hosted this study develops thousands of systems over the years in the same domain for different customers across the globe. Naturally, there is a lot of emphasis on domain knowledge management for reuse to achieve productivity and efficiency benefits [8]. Reuse of existing domain knowledge, which was acquired and validated in previous projects allows for a needed quick-start to projects and ensures good quality requirements. This is especially useful, given that the time allotted to requirements is short.

This was mentioned explicitly by two participants (P1, P6) and was implied in the interviews of four (P7, P8, P9, P11). P1 stated that the insurance domain is tightly regulated in the US with small variations across the states. Therefore there is a large potential for reuse across the projects developed for different states. P6 expressed that the reuse of rules depends on similarity of purpose. If a rule is generic enough to be applicable across, it may be used across the domain or company. P7 stated that understanding a complex domain was "inadequate in the absence of articulation of rules." P11 mentioned the bonds auction process had many similarities across the world and the common rules were amenable to reuse. P9 mentioned the regulatory challenges of the financial services domain when it comes to variations regulations in different countries while executing the same operations.

## V. Findings on Requirements Engineering

This section summarizes our findings on RE practices, tools, and problems related to business rule usage (which correspond to RQ3 and RQ4).

### *A. Impact of RE Practices and Tools on Business Rules*

**Finding 10: Business rules are documented differently in service-oriented and product-oriented RE. Product-oriented teams spend more effort to maintain business rules as they are more interested in reuse.**

The first half of participants (P1-P6) reported their RE to be service-oriented while the other half (P7-P11) practice a product-oriented RE. In the service-oriented projects, RE was strongly dependent on customers. P6 claimed, "Requirements are not only given but also *driven* by customers" referring to the process which should be followed or the documentation that should be written, including business rules. P5 said, "Customers decide about requirements. They want to understand what they need." P4 stated, "We focus on business processes of the customer. In the requirements documents we put which steps are parts of the process, what is the input and output of each step, and sometimes the processing rules." P2 claimed, "We cannot change a customer's perspective. It starts with defining the approach, templates, plan, and deployment of most suitable subject matter experts vs. available experts, clients' familiarity of certain tools and methodologies, their existing contract with certain 3rd party vendors for tools with whom there is an ongoing license or contract. All these impact the requirements and the tools."

In the product-oriented RE (telecom and banking domains) participants' teams (i.e. the software vendor) initiated the RE process. When these teams are capturing requirements, their objective is "also to sell as many product features as possible" (P8) and to minimize customization effort of existing product platforms. Product-oriented teams seem to update requirement documents more systematically and try to achieve completeness of requirements and identify generic features that must go into a product line. P7 stated, "We have 45 components in the platform. Each component has it own product specification. Each specification describes the functionality of that component in Use Case format including restrictions and rules. We create Business Requirement documentation specifically to each customer. The extended product specification document is created based on a gap analysis. We use our working software as a source for identifying more product requirements and evolve the product and its documentation." For the service-oriented projects, we did not observe such motivation to update requirements documents. The first priority is to deliver the expected software features on time with the expected quality (P3).

**Finding 11: Customers expect software vendors to have domain knowledge and to "develop" business rules.**

We also observed that the interviewed teams often play the role of domain experts, even when they did not have previous experience with similar projects. For example, P1 stated, "The customer mainly intervenes in the conflict resolution. We do

the "creativity" on what should be developed and the customer decide in case of conflicts." P11 claimed, "Customers are unable to give requirements. They give bits and pieces and we have a full-fledged analysis team to do this."

The participants' teams had to acquire or "improvise" domain knowledge, e.g., by hiring subject matter experts or by studying public knowledge such as standards and laws. We observed this particularly in the insurance and banking domain, where the regulations change frequently and depend on the location of the customers. Even in technology and IT domains, where software development belongs to their core business, customers tend to "outsource" RE to the software vendor, as claimed by P3 and P5.

**Finding 12: There is a trend towards agile projects, which lead to an increased documentation and evolution of rules in natural text documents.**

When asking about processes, participants reported to follow both waterfall and agile methods, with a growing trend toward agile (P1, P3, and P6). Customers are increasingly asking for agile delivery because of (1) the "pressure" to follow industry trends, (2) iterative prototyping that allows more frequently for a "sneak peek" into the progress of the project, and (3) the flexibility for prioritizing differently if necessary. Customers may or may not be knowledgeable on agile, but they are keen that vendors adopt agile practices anyway. P3 stated, "We have followed scrum with sprints of 9-13 days. But within a sprint, we followed a waterfall model. We had 2-3 days to read the document, 4 days to implement the features, and 2-3 days for the testing." P6 stated, "In agile, user story embeds both rules and functionality in a single statement. In waterfall or V model, the Use Case templates may contain separate placeholders such as pre-conditions and post-conditions for capturing rules."

**Finding 13: In industry, office tools are widely used for RE also to capture and maintain rules as part of requirements.**

When asking about the tools, nine participants (all except P1 and P6) reported that no specific tools were used for requirements engineering. In most of the projects, participants' teams used text editors and spreadsheet to capture and manage requirements and rules. When asked about the rationale, P2 and P11 stated, "This is the fastest and easiest way. We use documents and complex spreadsheets since customers are comfortable with them." P2 and P9 claimed similarly "otherwise, we need tool training. This is time consuming, and licenses are expensive." P8 claimed, "We do not use particular tools for capturing requirements and rules, if you know something good, please tell us."

Two participants (P1 and P6) stated that they have used DOORS in a few projects mainly because it supports version control. Other participants used change-tracking mode of Microsoft Word or an extra column in a spreadsheet for version control. P2 claimed, "Spreadsheet did not work well when adjustments were needed." He referred to the change requests that impact various requirement elements. For example, a change in workflow would require changes all the way from screen navigation to the processes and rules that govern the processes and it was not easy to track this using spreadsheet. P1 and P6 said that they have used special tools (including ReqPro and Caliber) only if the customer explicitly asked for it. Otherwise, the team reused templates publicly available on the Internet, and sometimes developed own templates in house.

### *B. RE Problems Affecting Business Rules*

**Finding 14: Requirement practitioners allocate low effort for understanding requirements, in particular for understadning special flows and business rules.**

Six participants (P1, P3, P4, P5, P8, and P10) stated that the time and effort allotted for RE is usually short. This might have a serious impact on the understanding and on the quality of requirements, as well as the planning and the effort estimation.

P3 stated, "Understanding of the requirements is the main thing. This is more important than the documentation. That's why we need all the discussions." Participants claimed that due to time constraints there isn't enough documentation. Sometimes meeting notes are used as a formal requirements document. People who joined the project from the beginning know how it works. People who, joined later have big problems to read and understand "400 pages documents" (P3). P5 claimed, "There are very few available resources to do proper requirements engineering. In one large project, the real development started during testing, since all major requirement changes came by then. We needed nine months implementation after starting the system testing. In a project of nine months, at best 1 month is given for requirements engineering. Teams have rush, both in term of coverage and depth of the requirements."

P8 stated, "At the sign off of the Business Requirements Specifications one customer from North America asked us to share the specifications. When we sent the specifications, we got numerous comments. We had already started programming in parallel. But because of these comments, we had to rework the code and a 6 months project went to 18 months. This happened because the time spent on understanding requirements and business rules was too short. We relied on a lot of *assumptions*, and we did not capture the business rules, the actions, and validations. The customer wanted to know the validation rules but these were undocumented."

When we asked why less time was allotted to RE participants argued that in most of the projects the rollout deadlines are fixed, either because the customer promises a roll out date to the users, a regulatory deadline must be met, or for competition reasons. In contrast, deciding about the vendor is a long process, which results in a late contract award. The time available for software development is hence shortened. The unrealistic deadlines lead to shortening the entire project. The time allocated for RE is often shortened disproportionally. Within RE, typically restrictions, constraints, special cases, and exceptions are either ignored or not captured and understood.

**Finding 15: The lack of domain knowledge in development teams hinders the capturing and usage of business rules as well as reuse across projects.**

Five participants (P1, P2, P4, P6, and P9) explicitly mentioned the lack of domain knowledge as a recurring issue in their

software engineering projects. P4 explained, "In our projects, people who do RE do not necessarily know much about the domain. In one extreme case, even the customer was new to the domain. They approached to us because we had previously implemented a similar system. But unfortunately our people who worked on that project had left already." P2 stated, "The lack of domain knowledge and domain training is a problem while defining and implementing the rules."

We observed that the number of team members who know the domain well is rather small (P1: 80 from a team of 1500, P3: 2 in a team of 30 people, overall about 5%). Teams seem to have an issue with the optimal utilization of subject matter experts. P1 explained the importance of domain knowledge with an example from the insurance domain in the US, which is tightly regulated. On the one hand, domain expertise is needed to deliver software, e.g., taking into consideration the variations across states. On the other hand, there is a lot of potential for reuse across the projects. We also observed similar claims in other domains. For example, P9 stated, "The financial services market infrastructure domain is very challenging. There are new regulations and business rules all the time. We are not privy to the discussions that happen during these changes."

**Finding 16: Identifying the right level of detail in requirements documentation is difficult. Either too much information is captured and the document becomes too large and unreadable or only main flows without business rules and exceptions are captured and the document becomes useless.**

Participants complained that requirements documents do not have the right level of details: either too much information or too little. In particular four participants (P3, P4, P5, P7) claimed that the exceptional flows are not captured in sufficient detail during RE. For example, P5 explained "there is often a narrow view of requirements to just capture *the happy flow*. This often matches the as-is, but not the to-be. We had once, for example, big data migration issues since we had the data structure in place but did not know the workflows and the restrictions on the data." P4 and P9 gave an example of superficial requirements "calculate a rate based on interest rate." When starting the implementation based on this requirements developers encounter many questions and ask the people about the details and the rules behind. Teams often had delays in the implementation schedule, because they did not have the required level of details at the beginning. P3 explained "The customers or the managers show you a screen and ask for estimation, which was correct since the details like the specific scenarios, exceptional flows, and the constraints in particular contexts were unknown."

Surprisingly, participants also complained about too detailed specifications. For instance, P3 complained about the difficulty of finding the right information under time pressure within a 400 pages requirements specification document. P7 stated, "In western countries customers expect high level of requirements details with a very fine-grained change management. In developing countries, organizations are rather small with 20-25 people. For them, over-documentation is overwhelming." When asking for details, we found that the real problem lays in the information identification and the efficient answering of stakeholders' questions due to the static nature of large documents.

**Finding 17: Business rules and exceptions represent a substantial amount of requirement knowledge, which often remains tacit in the mind of people.**

Five participants (P3, P4, P6, P7, P9) complained that relevant requirements knowledge such as *exceptions*, *rules,* and *justifications* remain tacit in the mind of people. For instance, P4 explained, "Many requirements and rules are implicit. Either the customers do not know it (e.g. interfaces of a legacy system) or the customer thinks you should know it anyway (e.g. the response time or the authorization). Participants explained that knowledge "is there" for years leading to many implicit requirements. P3 stated, "There are implicit rules that are not mentioned anywhere. For instance, the validations for similar fields in web forms are described only once."

Three participants (P6, P7, and P9) associated this issue with the personal backgrounds and skills of the involved stakeholders. P6 stated, "Technical experts are not involved in requirements exercises. This leaves huge gaps in understanding and interpreting requirements." P7 proposed, "We have to de-personalize the problems. It depends who does requirements." As a result of tacit knowledge residing with people, the whole RE process becomes heavily dependent on people. Participants claimed that they would like to make this less subjective and less people-dependent, by making the tacit knowledge explicitly available. They argue that the quality of requirements that are backed with externalized knowledge will be higher than when requirements are based on tacit knowledge.

## VI. Discussion

### *A. Implications*

#### *1) Tools for Managing Requirements and Business Rules*

One consistent finding of our study is that requirements tools are rarely used in practice – at least in the host organization. The participants were aware of special RE tools but they used them neither to capture business rules nor other types of requirements knowledge. Given the size of this organization (over 100.000 employees) and the broad spectrum of domains and customers involved, we think that this finding should be taken seriously into consideration by requirements engineering researchers and tool vendors.

Over two decades of tool research and development word processors and spreadsheet tools are predominating RE in the studied projects. As consequence, information is difficult to retrieve and detailed questions cannot be easily answered (Finding 16). A frequently mentioned example in the interviews was of the static requirements document of 400 pages, that no one wants to read or maintain. We wonder: When do we need RE tools at all? When do we need formal means for capturing business rules and other types of requirements knowledge? Why are other software engineering tools, such as bug trackers or IDEs, more established amongst practitioners? To answer these questions, more systematic, broad, and reliable studies of RE practices are needed.

There are pragmatic and fundamental reasons behind this situation. Pragmatic reasons for using spreadsheets and office tools for capturing rules and domain knowledge in general include license costs, training effort, compatibility, and usability issues. Fundamental reasons include the way state-of-the-art requirements tools support capturing, updating, and accessing requirement information especially in large projects. Participants claimed that relevant information is available somewhere. However, it is difficult to get the right information in the right context. Usually stakeholders must read hundreds of pages manually and reason about interdependencies: sometimes having in mind that the requirement document might be not up-to-date. Researchers should study means to automatically mine types of business rules in text and develop question-answering systems to retrieve the required information with the required details depending of the current task and context. This by itself would increase the usefulness of domain knowledge and business rules and provides an incentive to capture and maintain them.

*2) Reuse of Domain Knowledge Through Business Rules*

One main lesson learnt from this study is that nowadays software organizations do not only rely on the asset of the software, technology, and design knowledge, but also more and more on the domain knowledge in general and business rules in particular. We learnt that users and customers expect engineering teams to know the domain very well. And if not, it was not easy for the teams to extract the characteristics of the domain, the restrictions, the constraints, and the exceptions. Customers either assume that domain knowledge is common or they have no detailed knowledge about the domain.

We learned that business rules represent a crucial part of domain knowledge, but unfortunately frequently reside implicit in the mind of subject matter experts. If captured or extracted, business rules would bring large potential for reuse. Business rules can be found as a natural text in any project document from a marketing brochure, to a requirements document, an email, or a source code comment. Extracting and tracing them to other artifacts can assist requirements stakeholders while working on different tasks: from the negotiation of requirements to the planning and testing.

To this end, the first step is to get a common understanding of the types and knowledge included in business rules. In this study, we found that there is no precise common definition of the term business rule amongst practitioners. However, there exist an implicit understanding. All practitioners knew the term and were able to give examples and report on their experience. We identified 27 types of business rules. Some of them were mentioned frequently by the participants. Some types strongly overlap with other types of software engineering knowledge, which can be found, e.g., in API documentation or manuals. This increases the reuse potential of business rules.

Our results let us believe that researchers and tool vendors should focus more on the knowledge and documentation potentials of business rules in addition to the automation and code generation potentials from rule based engines. For instance, business rules can be included as annotations to accelerate understanding of complex documents. The system design can benefit from highlighting rules in a document. Search engines can crawl the various artifacts and leverage business rules to stakeholders through search interfaces. One participant mentioned, “It would be perfect if I could just select a component or a feature and my tool show me the most important business rules which I should take care of.” To realize this vision, the software and requirements engineering research community need to learn from the information retrieval or knowledge management communities.

### *B. Limitations and Threats to Validity*

There are several limitations to the internal and external validity of our results. We are aware that in 90 minutes we can only discuss a fraction of participants’ activities and experience. We might have missed certain types of business rules, usage scenarios, or problems. However, we think that extending the interview time would not fundamentally change the findings since concentration typically decays over time. Moreover, participants were able to give free comments at the end the interviews or contact us afterwards via email or phone.

Another potential threat to the internal validity is that interviewers might have had assumptions and expectations and might have reported only clues affirming these expectations, while ignoring different unexpected statements. To mitigate this threat, all interviews were conducted by at least two interviewers with a rotating minute taker. The minutes were also circulated to the interviewees to validate the results and add clarifications.

Similarly, participants might have behaved differently because they were interviewed, stating what the interviewers “are willing to hear.” This threat cannot be eliminated completely but we addressed it by assuring participants complete anonymity and confidentiality. We also stressed that there was no "right and wrong answers" as we only aimed at documenting the state of practice through their experience and subjective opinions. Moreover, we only report on findings that were observed at least *in two different interview* sessions.

We also asked participants to send example documents that backup important examples and statements. We received these documents from 8 participants. Nevertheless, observing the participants during their daily work will lead to complimentary, or possibly different results and additional evidence.

Finally, our sample size with 11 participants is rather small. Because we did not study a random sample, that is representative of the entire target population of stakeholders, it is difficult to generalize from these findings. But, our study was designed to be exploratory, qualitative rather than representative, quantitative. It has a strong degree of realism as it was conducted with real industrial experts.

Within the host organization, we were unable to draw representative samples from all employees due to the coordination and interviewing effort. However, we sampled exclusively experienced participants, all experts within their teams. The distribution of participants includes different roles, application domains, projects, and technologies – representing a wide range of potential participants. We purposefully recruited at least two people from each participating domain (insurance, banking, telecom, and industry). This gives us some

confidence that the results have a medium degree of generalizability – at least within the target organization and similar outsourcing companies and IT service providers. The quantifications reported in this paper (e.g. the number of observations) should be interpreted carefully. The number of observations and statements might seem low at the first glance. We think however that all reported statements are important since participants brought them by their own as opposed to surveys were options are given to participants. Nevertheless, checking our findings and quantifying the results would require conducting a broad online survey and content analysis studies with representative samples.

## VII. Conclusion

In this paper, we studied how business rules are being used in requirements engineering practice. We identified 17 findings describing the types of business rules, the way they are captured and maintained, as well as problems in managing and using them. Experienced, interviewed employees of a large software vendor and IT service provider agreed that capturing and managing business rules is important and would impact not only requirements engineering activities but also other activities such as testing, documentation, and planning. However, neither the current RE tools, nor processes give room to easily retrieve business rules, trace them to other project artifacts, or reuse them in future projects within the same domain or for the same customer. To support practitioners dealing with these challenges the research community should move from exploratory to explanatory and evaluative research, focusing on systematically identifying, mining, and (re)using business rules as a main part of domain knowledge.

## Acknowledgment

The TCS Research Sabbatical program enabled this collaboration. We are very grateful for Preethu Rose for the support with conducting and summarizing the interviews as well as all participants for their effort and feedback.

## References

[1] S. Ali, B. Soh, and J. Lai. Rule extraction methodology by using XML for business rules documentation. In *3rd IEEE International Conference on Industrial Informatics,* 2005.

[2] J. Becker, C. Ahrendt, A. Coners, B. Weiß, and A. Winkelmann. Modeling and Analysis of Business Process Compliance. In *Governance and Sustainability in information systems managing the transfer and diffusion of IT*. IFIP Advances in Information and Communication Technology, Vol. 366. Springer 2011.

[3] H. Boley, B. Grosof, and S. Tabet. RuleML Tutorial. The RuleML Initiative, 2005. Last visited Jun. 2014 http://www.ruleml.org/papers/tutorial-ruleml.html

[4] T. Breaux, and A.I. Antón. Mining Rule Semantics to Understand Legislative Compliance. In *WPES '05 Proceedings of the 2005 ACM workshop on Privacy in the electronic society*, pp. 51-54, ACM, 2005.

[5] S. Ceri and P. Fraternale. Designing Database Applications with Objects and Rules, The IDEA Methodology. Addison and Wesley, 1997.

[6] M. Döhring, B. Zimmermann, and E. Godehardt. Extended Workflow Flexibility Using Rule-Based Adaptation Patterns With Eventing Semantics. In *40. GI Jahrestagung Informatik 2010, Service Science*, LNI, GI, pp.216-226, 2010.

[7] S. Ghaisas and N. Ajmeri. Knowledge-Assisted Ontology-Based Requirements Evolution. In W. Maalej and A. K. Thurimella (ed.) *Managing Requirements Knowledge* pp. 143-167, Springer 2013.

[8] E. Gottesdiener. Business RULES - Show Power, Promise. EBG Consulting, Inc. Application Development Trends 4 pp.3.1997.

[9] D. Hay and K. Healy. Defining Business Rules - What are they Really? The Business Rule Group, 2000.

[10] H. Herbst, G. Knolmayer, T. Myrach, and M. Schlesinger. The Specification of BusinessRules: A Comparison of selected Methodologies. In A.A. Verijn-Stuart, T.-W. Olle (ed.) *Methods and Associated Tools for the Information System Life Cycle*. pp. 29-46, Elsevier 1994.

[11] I. Horrocks, P.F. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, and M. Dean. SWRL: A Semantic Web Rule Language Combining OWL and RuleML. W3C Member Submission 2004. http://www.w3.org/Submission/SWRL/

[12] H. Mahgoub, D. Rösner, N. Ismail, and F. Torkey. A Text Mining Technique Using Association Rules Extraction. In *International Journal of Computational Intelligence*, Vol. 4 Issue 1, p21. 2008.

[13] C. Potts. Let the Business Drive IT Strategy. In *cio.com* (online) Sep. 2007. Last visited Jun. 2014. http://www.cio.com/article/135202/Let_the_Business_Drive_IT_Strategy

[14] O. Resch. Six Views on the Business Rule Management System. In *E-Journal of Practical Business Research*. 11/2010.

[15] D. Rosca, S. Greenspan, M. Feblowitz, and C. Wild. A Decision Making Methodology in Support of the Business Rules Life Cycle. In *Proceedings of 3rd IEEE International Symposium on Requirements Engineering (RE'97)*, IEEE Computer Society Press, pp. 236-246, 1997.

[16] R. Ross. The Business Rule Book: Classifying, Defining and Modeling Rules. Database ResearchGroup, Boston, MA, 2nd edition, 1997

[17] SAP Solution Brief. Centrally Compose, Execute And Maintain Business Rules. Last visited on Jun. 2014. http://scn.sap.com/docs/DOC-4570

[18] P. G. Sellfridge, R. C. Waters, and E. Chikofssky. Challenges To The Field Of Reverse Engineering. In *Proceedings of Working Conference on Reverse Engineering, WCRE '93*, IEEE Computer Society Press, pp.144-150.1993.

[19] S. Spreeuwenberg, J. Von Grondelle, R. Heller, and G. Grijzen. Using CNL Techniques And Pattern Sentences To Involve Domain Experts In Modeling. *Lecture Notes in Computer Science*, V. 7175, pp.175-193, 2012.

[20] P. Vincent. OCEB White paper on Business Rules, Decisions and PRR. OMG 2008. Last visited Jun. 2014. http://www.omg.org/oceb/OCEB-WhitePaperforPRR1.1.pdf

[21] M. Weiden, L. Hermans, G. Schreiber, and S. Van der Zee. Classification and Representation of Business Rules, Universiteit van Amsterdam, 2002.

[22] M. Zoet, J. Versendaal, P. Ravesteyn, and R. Welke. Alignment of Business Process Management and Business Rules. In *European Conference on Information Systems (ECIS)*, 2011.