Improving Requirements Engineering through Goal-oriented Models and Tools: Feedback from a Large Industrial Deployment

Christophe Ponsard[1] and Robert Darimont[2]
[1]*CETIC Research Centre, Gosselies, Belgium*
[2]*Respect-IT SA, Louvain-la-Neuve, Belgium*

Keywords: Requirements Engineering, Model-driven Engineering, Specification by Example, Case Study, Toolchain Integration.

Abstract: Nowadays, mastering the requirements phase is still challenging for companies of any size and often impacts the quality, delay or cost of the delivered software system. While smaller companies may suffer from maturity, resource or tooling problems, larger companies have to cope with the larger size, complexity and cross-dependencies between their projects. This paper reports about the work carried out over the past three years to address such challenges within Huawei, a very large Chinese company active worldwide in the high-tech and telecommunication sectors, with the help of experts from the requirements engineering community. We show how goal-oriented requirements engineering (GORE) is able to provide a strong foundation to support the evolution of requirements engineering practices and also in connection with related processes such as business analysis, technical specification and testing. We also report about our experience in developing adequate tool support to achieve successful industrial adoption and address team-work, scalability and toolchain integration needs. Although anchored in a specific case, most of the reported issues are shared by many companies in many domains. To further abstract away from our case, we also formulate some "Chinese wisdom" learned, identify useful strategies for successful technology transfer and point further research challenges.

1 INTRODUCTION

It is an established fact that software development projects are difficult to lead to success, i.e. finish within time, budget and scope. Regular surveys conducted over the past 20 years by the Standish Group have constantly shown that about half of the projects are challenged on at least one of those dimensions (Group, 2015). Although projects failure rate is lower than in the past, it is still the case for 1 project out of 5. Another fact is that those failures happen in every country and regardless of the company size (small vs large), its kind of activity (commercial, nonprofit, governmental), and to its status or reputation. Such surveys have also been replicated and consolidated by others with similar observations (Hughes et al., 2015; Majchrowski et al., 2015).

A recurring observation in such surveys is that requirements related problems consistently rank among the top 3 reasons of project failure. More specific cause mainly relate to the requirements completeness, the lack of customer involvement, unrealistic requirements and requirements change management. The difficulty to master Requirements Engineering (RE) is because it is intrinsically a difficult task which requires to cope with multiple stakeholders, explore the informal problem world to produce a structured consistent set of requirements, deal with implicit or hidden need and assumptions, deal with different levels of granularity, priorities, etc (van Lamsweerde, 2009).

Huawei makes no exception to this rule and has started an RE improvement process in 2013. The first step was to get in touch with the international requirements engineering community in the context of the annual conference which was taking place in Brasil. It quickly became apparent that Goal-Oriented Requirements Engineering (GORE) could bring many benefits to many of Huawei's needs: progressively refining requirements, structuring them, keeping a clear rationale as the requirements are transferred across teams, identifying clear acceptance criteria, etc. Another key point for the company in the selection of a method was the availability of tool support meeting high industry requirements in terms of maturity, scalability and tool-chain integration. This led to the combined use of the KAOS methods (Dardenne et al., 1993; van

Ponsard, C. and Darimont, R.
Improving Requirements Engineering through Goal-oriented Models and Tools: Feedback from a Large Industrial Deployment.
DOI: 10.5220/0006462503720381
In *Proceedings of the 12th International Conference on Software Technologies (ICSOFT 2017)*, pages 372-381
ISBN: 978-989-758-262-2

Lamsweerde, 2009) together with the Objectiver tool (Respect-IT, 2005).

This paper is structured as follows. Section 2 gives some background on the company, its requirements practices and some challenges, both at the method and tool levels. It also highlights the value-base drivers underlying all process improvement. Section 3 details the rationale and expected benefit related to the choice of a GORE strategy. Sections 4 and 5 detail how GORE was deployed, respectively at the method and tool levels. Section 6 draws some lessons learned that can be useful for other companies having similar needs. It also highlights some research challenges for the research community. Finally, Section 7 concludes and also highlights some next steps in the RE improvement roadmap.

2 COMPANY BACKGROUND AND ANALYSIS PRACTICES

Enriching Life Through Communication
- Huawei's motto

2.1 Company Presentation

Huawei Technologies Co. Ltd. is a Chinese multinational company active in networking and telecommunication equipment. It is headquartered in Shenzhen, Guangdong. The company was founded in 1987 and started by manufacturing phone switches. It has since then expanded its business to telecommunications networks and communications devices for the consumer market (e.g. smartphones). It is also providing operational and consulting services to enterprises both inside and outside of China. Since 2012, it is the largest telecommunications equipment manufacturer in the world. Its products and services have been deployed in more than 140 countries and it currently serves 45 of the world's 50 largest telecom operators. The company is constantly growing from 22,000 employees in 2002, 40,000 in 2006 to more than 170,000 nowadays.

The company invests over 10 percent of its annual sales revenue into R&D and more than 45 percent of its employees are engaged in R&D activities. Huawei cooperates with global innovators through its 16 R&D centres and 36 Joint Innovation Centres around the world, including the US, Germany, Sweden, the UK, Belgium, France, Italy, Russia, India and China. This strategy pays off: in the past few years, Huawei has regularly ranked within the most innovative companies (Fast Company, 2016).

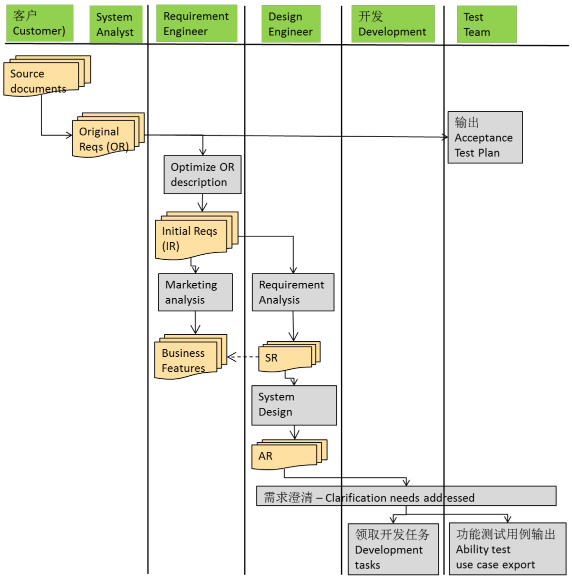

Figure 1: Overview of the RE Process at Huawei.

2.2 Requirements Engineering Process

Huawei's requirements engineering practices are based on domain modelling, analysis of the current solution (i.e. existing product), use case analysis and UML modelling. The global process is depicted in Figure 1 and is composed of the following steps:

- **Elicitation:** The business department collects all the requirements from the customer. User stories (Cohn, 2004) are used as practical means at this stage "As a <customer_type>, I need <some_feature> in order to reach <some_goal>". Those requirements are called "Original Requirements" (ORs for short in the rest of this paper). They usually target a whole system or a complete solution design. The identification of a satisfaction criteria is also very important for the later acceptance.
- **Analysis:** Together with the business department, the product team reviews and refines the ORs. They are categorised both at the functional and non-functional levels. There is also some alignment with requirements shared across products (product line perspective). Those refined requirements are called "Initial Requirements" (IRs). They are also prioritized and form the baseline to develop the product specification.
- **Specification:** IRs are transferred to the relevant R&D department. This department is in charge of analysing the IRs from a technical point of view and produce the System Requirements document (SR). Before introducing new methods, this phase

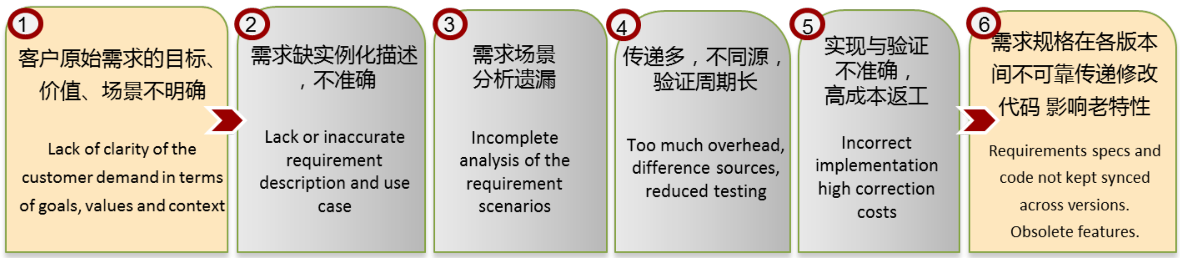

Figure 2: Identification and articulation of main RE challenges.

mainly relied on domain analysis and use cases.

- **Business Marketing:** In parallel with the specification, marketing and design team still work on the IRs and detail them in terms of feature properties. This phase is conducted in interaction with customers and target business issues. The properties that are selling points for customers are clearly identified as well as how they can be controlled. Traceability between SR at feature level is maintained for this.
- **Design:** Based on the SR, the development department produces a detailed design including allocation of functions to modules and clear specification of their interface. The resulting requirements are called "Allocated requirements" (AR). Those are implemented.
- **Acceptance Test:** ORs are also transferred to the test team for the production of acceptance tests plans. Later on, the test team will also receive the detailed AR to test.

2.3 Main Requirements Engineering Challenges

Trying to diagnose why projects fail to fulfil their requirements at the expected level of satisfaction is difficult. Typically, there is no unique cause but rather an accumulation of flaws and those tend to only surface when the code is produced. Figure 2 shows a typical chain reported by Huawei but encountered in many projects: it starts by customers having unclear demand, then requirements are kept too vague with a lack of supporting use cases, the validation becomes then difficult and results in high correction costs. Huawei is addressing RE challenges in the scope of a value-chain approach of the whole software engineering lifecycle. RE plays an important role because it helps to understand the problem and where the value will be generated. High quality requirements will also result in easier testing and acceptance. We detail hereafter those two key aspects.

Improving the RE Process Effectiveness. The current process supports synchronisation points between successive roles of the development chain to get some refined requirements (from marketing) or to ask for clarification (at developer level). However the global process is quite heavyweight and does not cope easily with agile development for different reasons: the application domain requires quite rigorous testing and cannot cope with the delivery of partial products on which iterations can be carried out. The team size and the possible geographic distribution also do not favour using such methods. Of course, agile is not ruled out but the main requirement is to be able to raise and keep the quality of the requirements along the whole chain and detect flaws as early as possible. While improving the process, it is also important to avoid introducing over-complexity and match it with the Huawei context in order to reduce the training effort, have a good learning curve and do not overload people with too advanced concepts. This can be summarised by the following method requirements:

- Better structuring mechanisms to organise requirements
- Support for specific categories defined at Huawei
- Guidance and diagnostics about requirements quality
- Limit the complexity of the method

Improving the RE process Efficiency. It is also important to get the most out of the effort invested in the requirements activities by more systematically and more automatically exploiting the resulting requirements artefacts in other software development steps. For example, for testing, tests plans can be generated from the requirements by relying on domain specific languages like Gherkins under the form of "Given-When-Then" and can also directly refer to User Stories (Wynne and Hellesoy, 2012; Adzic, 2011). A crucial enabler for this is the availability of strong tooling with key characteristics such as:

- Efficient support for large specifications
- Ability to cope with multiple concurrent users
- Navigation across multiple versions
- Very good usability (visual feedback, shortcuts,...)

- Requirements import, export and transformation
- Flexible toolchain integration though web-services
- Reduced installation and maintenance effort (e.g. SaaS)

3 WHY INTRODUCING GOALS?

Flowers in a mirror and the moon in water
(Chinese proverb)

3.1 GORE vs Other Methods

Requirements Engineering is concerned with the elicitation, specification, documentation, validation and management of the objectives, functionalities, qualities, and constraints a software-based system should meet within some organizational or physical setting. From the needs identified in the previous section, the focus is on the specification phase while providing good connections with the other phases. Different methodologies can be considered:

- Simple template-based methods such as the Volere template (Atlantic Guild, 2014) or use-cases (Cockburn, 2000). Such templates provide a sound structure mostly for capturing and documenting requirements. They guide the user in the systematic documentation of each requirement (including originating stakeholders, purpose, agents, scenarios description, satisfaction criteria, etc). They also provide some sound structure to present requirements both at functional and non-functional levels. Although limited to structured text, they can be combined with some modelling notations. Although the templates provide some means to detect incompleteness, they have limited structuring mechanisms able to relate requirements and provide deeper quality checks.
- Methods based on standard modelling paradigms such as UML or SysML (OMG, 2011; OMG, 2012). Such methods rely on semi-formal models and are essentially design-oriented but do offer some support for requirements through specific diagrams such as the Use Case diagram (which should be associated with a use case template) and sequence diagrams (to capture detailed) scenarios. SysML also provides some tree-structured requirements diagrams that can be seen as a very simplified version of goal models.
- Goal-oriented methodologies such as KAOS (Dardenne et al., 1993), i*(Yu and Mylopoulos, 1997), URN/GRL (International Telecommunication Union, 2012; Amyot and Mussbacher, 2011). Such methods rely on the central notion of goal which is an objective the system under consideration should achieve. Goal formulations thus refer to intended properties to be ensured. Goals can be of functional or non-functional nature and may be formulated at different levels of abstraction, ranging from high-level, strategic concerns (such as "Provide ubiquitous communication" for a smartphone) to lower-level, technical concerns (such as "Support 4G communication protocol"). They can be refined from higher level to lower levels using AND-OR graphs. An AND-refinement link relates a goal to a set of subgoals called refinement and meaning that the parent goal can be satisfied by satisfying all subgoals in the refinement. A goal node can be OR-refined into multiple AND-refinements, each of them forms a different alternative (or variant) for achieving the parent goal (van Lamsweerde, 2001; van Lamsweerde, 2009).

Table 1 provides a high-level comparison of those three main classes of methods against the different RE phases while considering the level of tool support that can be achieved and the complexity to use the method.

Table 1: Comparison of some RE methods.

Needs	Templates	UML/SysML	GORE
Elicitation	Static categories	Use-case based	Goal-driven (top-down, bottom-up)
Structuring	Structured text	Use Case, Req. trees	Goal models
Validation	Template, check-list	Simple checks	Deeper semantic checks
Documentation	Document based	Model export report generation	Powerful generation (document, spreadsheets)
Toolchain integration	Document based	Simple export	Service integration
Complexity	Simple structure	Standard Modelling skills	Specific Modelling

GORE meets the needs to a best level despite a bigger complexity that needs to be addresses adequately in terms of method adaptation and deployment. GORE is rather complementary than a pure alternative: e.g., templates can be used both for elicitation and documents generation and modelling is used as part of the structuring process.

3.2 GORE Method Integration

GORE is no silver bullet, but it is being considered in a larger perspective that goes beyond the pure requirements phase. The main synergies considered with other activities are the following:

- **Traceability to Source Documents.** This phase is performed quite informally using text documents based on different company templates. Identifying goals and requirements in such documents cannot be automated but requires to identify and encode them manually in the RE tool. In order to ease that work, it is convenient to be able to "tag" the concept directly from the requirements document and automate its creation inside the goal model. In addition, a traceability link can be kept both inside the document and the model, so any change can be detected. It also eases the navigation between the source documents and the model.
- **Illustrating Goals with Specification by Example.** Goals are statements that can be rather abstract. In conjunction with goal, Huawei is also considering "Specification by Example" (SBE), a collaborative approach for defining requirements and business-oriented functional tests (Adzic, 2011). This approach is generally applied in the context of agile projects with a behavioural-driven development (BDD).
- **Behavioural Driven Development and Testing.** BDD is a software development process which relies on Test-Driven Development (TDD) but which complements it with a better connection with the problem domain (i.e. linking them with the goals). It is thus able to better cope with acceptance testing. BDD relies on the use of a simple domain specific language (DSL) using natural language constructs. In the scope of Huawei, Gerkins is used (Cresswell, 2011). The language supports both user stories (including stakeholder, functionality and goal) and scenarios specified using "Given-When-Then" constructs. BDD is usually supported by specific tooling frameworks, like Cucumber (Wynne and Hellesoy, 2012) enabling to automate the testing.

4 HOW TO DEPLOY A GORE METHODOLOGY?

A journey of a thousand miles begins with a single step.
(Chinese proverb)

4.1 Time Schedule

Introducing a GORE was fostered and led by the tool technology department together with the R&D tools and test platform design department. This team is in charge of conducting the change management approach company wide. The global timeline and milestones are the following:

- July 2013-November 2013: Initial contact at RE'13, two-day workshop in Belgium for in depth review of KAOS method and tool support. Visit of the business unit and R&D units.
- December 2013 - October 2014: Methodology introduction and positioning within the company: training sessions in R&D sites across China, test of the method and tool by trained users.
- October 2014 - March 2015: First adaptation and integration round. Minor additions to the KAOS meta-model were introduced and the tool was coupled to an existing web-based RE tool.
- April 2015 - July 2015: Second adaptation and integration round. Introduction of Huawei specific concepts, support for the SBE method and connection with the new BDD-based platform.
- August-October 2015: Third integration step: in-text identification and traceability from upstream document
- November 2015-June 2016: Fourth deployment phase: web-based collaborative version of the tooling and more powerful integration within the Huawei toolchain
- July 2016-December 2016: Improving scalability and achieving high-availability.

4.2 Introducing GORE

Introducing GORE within Huawei was carried out based on the following means:

- Reference material: books, tutorials (e-learning), exercises
- On-site training sessions using a complex case study
- Deeper training of the core team (training the trainers)
- Specification of a small-sized pilot case study

4.3 Method Adaptation

The complete KAOS meta-model is very complete and is actually composed of four sub-models shown in Figure 3

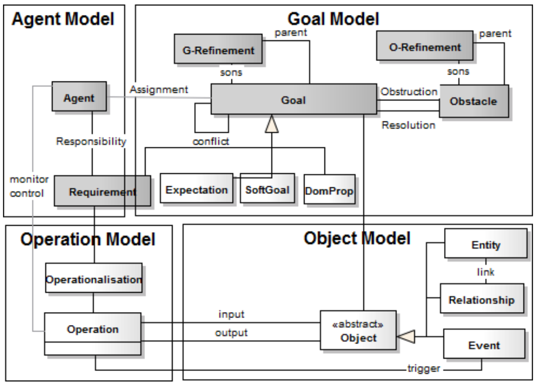

Figure 3: KAOS Meta-Model (simplified in grey).

- The goal model captures goals refinement structure (and possibly related obstacles and their refinement too)
- The object model describes the relevant domain vocabulary to express the goals
- The agent model takes care of assigning requirements to agents in a realisable way
- The operation model details the work an agent has to perform to reach the goals under its responsibility

After the first experiments within the company, it quickly appeared that deploying the standard GORE modelling was not the most effective way to proceed because:

- The full meta-model is rather complex and it is important to focus on the core parts on which the requirements engineer have to focus, i.e. goals and responsibilities.
- Object modelling is redundant with domain modelling techniques already used.
- KAOS operation modelling is actually not required as specification are detailed at a later SR step with SBE.
- some Additions were also needed for specialised Huawei requirements types, e.g. OR, IR, AR.

Globally this resulted in the simpler meta-model highlighted in grey in Figure 3. This also reduces the training effort and softens the learning curve.

4.4 Method Integration with SBE

Specification by Example specifications are used to describe goals in a structured way, using Gerkins syntax (Cresswell, 2011). The description is composed of a user story and scenarios that consistently relate with the information provided in the goal model.

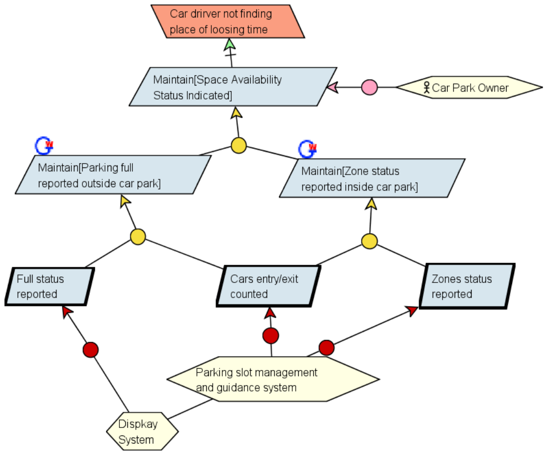

Figure 4: Availability sub-goal for a car park.

The User Story Part is formulated under the following form: "As a <stakeholder>, I need <current_goal> in order to <some_rationale>". The <current_goal> corresponds to the goal to which the description is attached. For example, Figure 5 corresponds to the goal $Maintain[ZoneStatusReportedinsideCarPark]$ shown in Figure 4. The <some_rationale> part can refer to a higher level goal or obstacle. In our example, it refers to an obstacle that the goal is avoiding, i.e. the car driver not finding any place or having problems to locate it. Finally the stakeholder <car_park_owner> can also be associated to its goal in the diagram.

The Scenario Part is composed of one or several scenarios expressed using a Given-When-Then form that is close to KAOS notion of Pre-/Trigger-/Post-conditions used in KAOS operation but with a structured text syntax. In this case, we specified two scenarios corresponding to the two status of a zone: full or not. The precondition in both scenarios is that the zones are under monitoring by a specific agent able to count the spaces. The action trigger is the status change (car leaving or entering) that will result in displaying the number of free spaces or telling the zone is full.

5 HOW TO PROVIDE EFFICIENT GORE TOOL SUPPORT ?

A needle is not sharp at both ends
(Chinese proverb)

Providing efficient tool support is a major success factor for deploying GORE. It is reflected by the importance of the successive integration steps described

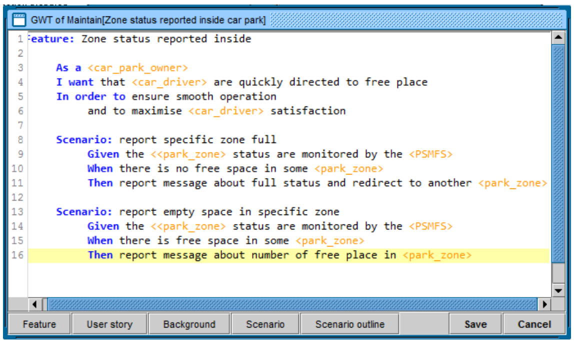

Figure 5: GWT specification for availability in a car park.

in the previous sections that progressively covered a full set of industrial requirements described in more details in this section. Besides ensuring functional GORE support, there are many non-functional tooling requirements that play a key role in industrial adoption such as usability, team work, versioning, scalability, and seamless integration. The rest of this section summarises the main tool development steps of the developed GORE support.

5.1 Coping with Method Adaptation

Implementing meta-model changes is not a big issue if we can rely on a meta-model based tool. In early stages, we relied on the Desktop version of the tool. It easily supports the evolution of its meta-model because it is actually a meta-case tool. In a later stage, the underlying database was evolved towards a more scalable technology based on the Eclipse Modelling Framework (EMF) (Steinberg et al., 2009). This also supports meta-model extension although existing projects have not the capability to automatically adapt to a new proposed meta-model.

5.2 Tight Integration with Text Editor

In order to cope with integration with text-based document, it was necessary to strengthen the support, especially by supporting commercial text processor (Microsoft Word) in addition to the pre-existing Open/Libre Office support. In order to support a fully flexible and usable integration of text processor with GORE modelling, we developed some innovative features such as ability to tag concepts from the text document, to locate them in the model from the document and even to rename them, as shown in Figure 6. The document is considered as part of the model but can also be exported and reimported while preserving the existing links.

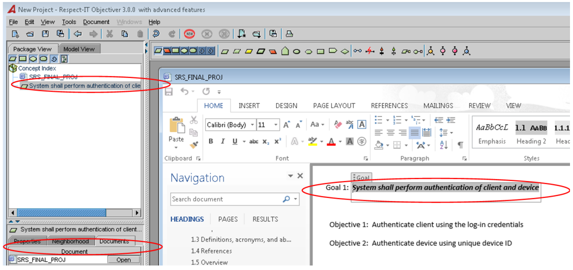

Figure 6: Integration with text processor (Desktop Edition).

5.3 Versioning Management

Model versioning is an important feature in order to keep track of the model evolution. The implementation evolved from a simple database supporting a single baseline to a fully historised model supporting multiple baselines. Access to concepts history can be useful in order to undo some changes in its own session or to understand possible interactions with other users as the current collaboration model was left opportunistic. Baseline are more important steps in the work as they define milestones in the work, possibly agreed amongst the analysis team. In order to ease the work, a powerful model comparison feature is available both in the Desktop and SaaS editions of the tool.

5.4 Usability Issues

Usability is a very important issue for adoption. In addition to the state-of-the-art layout proposed for modellers, a few important features were specifically introduced for the Huawei deployment such as visual decorators on the goal tree enabling to quickly visualise important requirements attributes like Huawei categories and GWT markers. The later decorator also plays the role of shortcut for editing it. In addition the support of the Chinese character set and the internationalisation of the user interface for Chinese was of course considered.

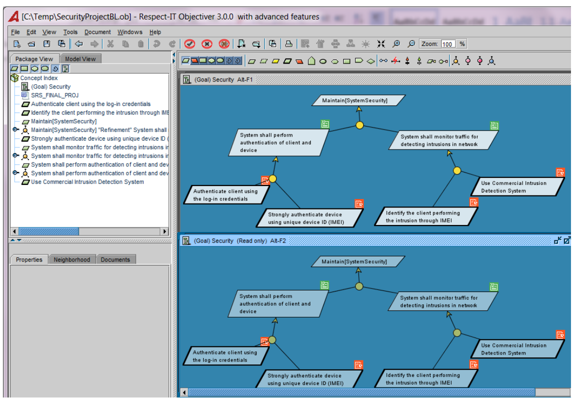

Figure 7: Version comparison feature (Desktop Edition).

5.5 Scalability

Scalability is an important requirement. Given the size of the company, the load may grow to several hundreds of users although there are only currently less than 30 active users. This motivated the move to a robust database back-end. Moreover, the application can easily be distributed across different servers based on project pools as projects are not expected to host large number of concurrent users.

5.6 Ease of Integration

Toolchain integration has started from simple export mechanisms and then evolved towards tight integration with specific software such as text processors as described in a previous section. A more fundamental evolution was to present the GORE tool as a series of services available for use over the company intra-net. This comes at two different levels.

First, at the model level, a clean REST API is directly available to perform CRUD (Create/Read/Update/Delete) operations both on the model elements and on model representations inside diagrams, baselines, etc. This allows third-party tools to directly push or query requirements inside the tool while previous import and export actions had to be initiated from the tool. This greatly helps to automate the requirements workflow described in Figure 1.

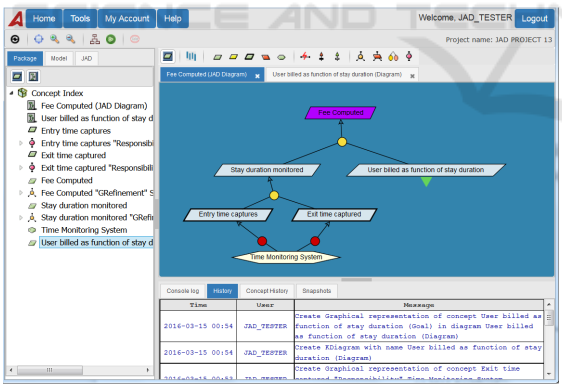

Figure 8: Modular web-based user interface.

Second, at the user interface level, the tool is available as a modular web-client. It can be used quite the same way as a standard modelling tool but, besides this, specific components can also easily be embedded inside project management platforms and dashboards, possibly limited to read-only mode. Those two kinds of interfaces are respectively depicted in Figure 8 and 9.

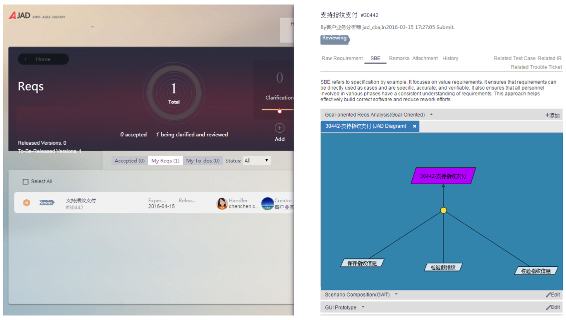

Figure 9: Sample of GORE component integration within a third party tool interface.

6 KEY GUIDING PRINCIPLES

To know the road ahead, ask those coming back.
(Chinese proverb)

Yet anchored in the Huawei context, we were attentive to formulate our feedback independently of the specific flavour of the GORE method and tooling support. Hence it has already a good potential of reuse by other companies having similar needs. In this section, we take an additional step back to formulate a series of useful guidelines and principles summarising the followed approach.

"One Method does not fit all Needs" is an important lesson learned. Rather than trying to rely on a complex goal model potentially overlapping with other notations, like the SBE specification and the KAOS operation models, we decided to use a simplified goal focusing on key GORE concepts which is easier to introduce and which can be integrated with more specific methods such as Behaviour Driven Development in our case.

The **"Keep It Simple Stupid"** principle was often a major driver. Both to keep the method and tool complexity low. This guided us to find our way across challenging problems mixing different features such as multiple users, concurrent editing, versioning support. However, simple does not mean over-simplistic: the idea is always to define a path that can cope with many wished features at short term and not exclude more at longer term. With respect to this, we were also careful selecting mature and evolvable underlying components, e.g. the model repository.

Combining RE Practices and Agile Tool Development. The global collaboration is based on definition of clear objectives, first at business level (as values to achieve) and in terms of method and tooling requirements to cover. A GORE approach was used for setting up the global set of prioritised requirements and define a roadmap, typically on a bi-annual basis and is also generally organised as a one week intensive physical workshop. This forms the backlog forms on which an agile tool development approach is then carried out based on teams both in Europe and China. Important milestones are typically set every month and the common sprint duration is one week.

Open and Evolutive Solutions. Chinese companies are fast paced with constant tooling evolution. In any long term project, the software environment will change for sure and it is thus important to build on solutions that have a big adaptation capability. At some point, it may also be necessary to consider starting on new grounds and let a past solution aside. In our case, at some point we moved from a desktop-centric tooling to a web-based tooling because desktop centric tooling could not cope with key needs of scalability, flexible multi-user and ease of deployment/integration/maintenance. A major improvement in the process was the definition of powerful integration interfaces for third party tools.

Paying Attention to Usability was a major concern. Software engineering tools should pay great attention about an efficient way to communicate all relevant information to the user. They should also provide efficient ways to conduct frequent operations by minimising the number of steps to trigger them and enable some form of batch processing.

7 CONCLUSION

In this paper, we reported on the ongoing process at Huawei to deploy Goal-Oriented Requirements Engineering in cooperation with other methods with an important focus on building a globally efficient and effective toolchain. This joint work is a cross-fertilisation at different levels. It was a meeting point between industrial needs and researchers views, as well as a transcontinental cooperation between Westerners and Easterners who could gather different views on requirements engineering and the whole software development process. This rich collaboration was the source of many innovative work, often triggered by the need to provide simple yet powerful tools based on sound methodological grounds. So, we believe the reported work can be valuable both for the industrial point of view and for triggering more applied research as we tried to abstract away from the specific case and identify different areas of concern both at the method and tool levels.

Based on the work achieved, strong foundations are available, most of work ahead related to completing integration with specific tools, support specific interoperability standard but also less technical yet very important tasks such as the preparation of model templates, tutorials, guidelines that will ease the transition to the new platform and foster its adoption by users. So far, the approach has been applied on a few pilots. Hence, it is not yet possible to quantify the gain in quality and productivity of introducing GORE at a large scale. The deployment phase is expected to progressively roll out across this year and will offer the opportunity to gather valuable material about the benefits and limits of the approach.

ACKNOWLEDGEMENT

Warm thanks to Zhang Wei, Liu Zhipeng, Yang Yang (Davin), Lin Zhi Qiang (James), Wang Jianan (John) for their great support for the method and tool deployment at Huawei Limited in Shenzhen and across China.

REFERENCES

Adzic, G. (2011). *Specification by Example: How Successful Teams Deliver the Right Software*. Manning Publications Co., Greenwich, CT, USA, 1st edition.

Amyot, D. and Mussbacher, G. (2011). User requirements notation: The first ten years, the next ten years. *JSW*, 6(5):747–768.

Atlantic Guild (2014). Volere Requirements Specification Template - Edition 17. http://www.volere.co.uk/template.htm.

Cockburn, A. (2000). *Writing Effective Use Cases*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1st edition.

Cohn, M. (2004). *User Stories Applied: For Agile Software Development*. Addison Wesley Longman Publishing Co., Inc., Redwood City, CA, USA.

Cresswell, J. (2011). Gherkins. https://github.com/ cucumber/cucumber/wiki/Gherkin.

Dardenne, A., van Lamsweerde, A., and Fickas, S. (1993). Goal-directed requirements acquisition. *Sci. Comput. Program.*, 20(1-2):3–50.

Fast Company (2016). Huawei Ranked Among the World's 50 Most Innovative Companies. http://www.fastcompany.com/company/huawei.

Group, S. (2015). 2015 CHAOS Report. http://www. standishgroup.com.

Hughes, D. L. et al. (2015). *Success and Failure of IS/IT Projects: A State of the Art Analysis and Future Directions*. Springer Int. Publishing.

International Telecommunication Union (2012). Recommendation Z.151 (10/12), User Requirements Notation (URN) Language Def.

Majchrowski, A., Ponsard, C., Saadaoui, S., Flamand, J., and Deprez, J.-C. (2015). Software Development Practices in Small Entities : an ISO29110-based Survey. In *Proc. EuroAsiaSPI 2015*.

OMG (2011). Unified Modelling Languages. http://www.omg.org/spec/UML.

OMG (2012). SysML. http://www.omgsysml.org/.

Respect-IT (2005). The Objectiver Goal-Oriented Requirements Engineering Tool. http://www.objectiver.com.

Steinberg, D., Budinsky, F., Paternostro, M., and Merks, E. (2009). *EMF: Eclipse Modeling Framework 2.0*. Addison-Wesley Professional, 2nd edition.

van Lamsweerde, A. (2001). Goal-oriented requirements engineering: A guided tour. In *Proc. of the Fifth IEEE Int. Symposium on Req. Eng.* IEEE Computer Society.

van Lamsweerde, A. (2009). *Requirements Engineering: From System Goals to UML Models to Software Specifications*. Wiley.

Wynne, M. and Hellesoy, A. (2012). *The cucumber book*. The pragmatic programmers. Dallas, Tex. Pragmatic Bookshelf.

Yu, E. S. K. and Mylopoulos, J. (1997). Enterprise modelling for business redesign: The i* framework. *SIGGROUP Bull.*, 18(1):59–63.