Icon Representations in Supporting Requirements Elicitation Process

Sukanya KHANOM[1], Anneli HEIMBÜRGER and Tommi KÄRKKÄINEN
University of Jyväskylä, Department of Mathematical Information Technology, Finland

Abstract. This paper considers the difficulties faced by the stakeholders in general requirements engineering (RE). These difficulties range from the complexity of requirements gathering to requirements presentation. Affordable visualization techniques have been widely implemented to support the requirements engineering community. However, no universal characteristics that could be associated with requirements completion have been identified so far. The research focus of this paper is driven by the above considerations to introduce the icon-based language comprising a set of icon notations, syntactic and semantics. Icon-based language would support the requirements engineering tasks that normally executed by stakeholders and provide a visual modelling language to unify the requirement activities. Research approach is recapitulate, firstly, by identifying the requirements engineering artefact, secondly by refining the icon artefact, and thirdly, by integrating those two artefacts by means of requirements engineering process. The result aimed at to make communications more interactive and manageable by facilitating the exchange of information and to promote global understanding in any requirements development context across cultural and national boundaries.

Keywords. Icon-based language, requirements development, requirements elicitation, stakeholder.

Introduction

As requirements engineering (RE) is theoretical and communication nature, it makes stakeholders difficult and challenge to carry out, especially in intercultural communication (across different geographical boundaries). Inappropriate performing requirements activity is the leading cause of software failure syndrome [1]. Research attempts have been done to develop the computer-intensive system which primarily aims at fostering people to communicate to each other. Several of those are purposed to the field of RE such as UML and goal-oriented models. However, one of the considerable deficiencies is the fact that the proposed techniques require skill and knowledge to achieve the tasks of each technical-rich technique [2, 3]. Misinterpretation and misconception are other two challenges when using the available modelling techniques. The reason might be because in RE process, practitioners possess a variety of backgrounds and knowledge.

Visualization modelling languages are recommended for discovering, specifying and reviewing the requirements in the software development. For instance, the use of diagrams in UML appears to convey information more effectively to nontechnical

[1] Corresponding Author.

stakeholders than does natural language approaches [3]. Nevertheless, the utilization of abstract shapes whose meanings are articulately conventional results in the comprehensive difficulty. The empirical evidences reveal that icons that their meaning can be perceived straightforwardly and easy learnt enables to enhance RE visual modelling language [2]. Icons are varied, depending on the purpose, from icon that are manipulated to operate devices to icons that are in the public places, airports, hotels, maps and traffic signs. Many of these icons have been designed to provide the same information about particular conditions or instructions to people everywhere [4, 5, 6]. Icons have also been accepted successfully in the human-computer interaction HCI. But amazingly, only a small amount of research has been addressed concerning the icon applications in the RE domain [2]. This lack of attention to the iconic adaptation aspects might be the case that the existing methods for analysing visual representations are less mature than the methods for analyzing mathematical representation.

The aim of the current research is to propose an uncomplicated visual modelling language that emphasizes on these two key contexts: *RE world* and *Icon World*. The *RE world* is signified to the requirements activities such as elicitation tasks that involve stakeholders' knowledge domain that contributes significantly to the successful RE practice. Broadly speaking, requirements elicitation serves as a bridge to communicate and discover the real world needs of the users, customers and other stakeholders who are affected by a software system [1]. *Icon world* refers to a set of icons produced as a visual sentence to support the defined RE world. The construction of visual sentence is formulated from visual vocabulary, syntactic and semantics.

The focus of this paper is on elicitation phase. Other phases are analysis, negotiation, specification and validation. The remainder of the paper is organized as follows: Section 1 reviews related work on visualization in the context of RD; Section 2 describes theoretical framework in research methodology and system model of the research; Section 3 introduces icon-based approach in requirements elicitation and implementation details of the prototype to be developed; an evaluation pattern is described in Section 4; and final section is reserved for discussion and future work.

1. Related Work: Visualization in the Context of Requirements Development

Several RE techniques have been presented with the purpose of overcoming RE issues. Traditional processes and procedures in the realm of RE are rapidly changing due to advancements in visual techniques. An example of these visualization approaches is the use case diagram, which is widely used in elicitation and modelling activities, as it makes untangling of the gap between business and IT stakeholders possible [1, 7]. Another visualization mechanism is the goal-oriented model in which goals of a system are identified to aid the requirements process elicitation, analysis and negotiation, documentation and validation [8]. A recent process, known as the agile approach, has been widely accepted as "rapid" and "adaptable" to software development. The approach emphasizes iterative development, continuously getting feedback from customers throughout the software development life cycle (SDLC) to capture the system's expected behaviours. Instead of stating formal requirements or the design process as other models do, in agile approach, it is more flexible to make change to the requirements, even late in the development.

Many RE visualization tools that enhance collaboration came into being as thousands of system developers and other stakeholders turned to software visualization

to simplify their jobs. Computer-Aided Software Engineering (CASE) is one of the visual tools that support collaborative features by using a computer-assisted method to manage and control the development of software. The CASE tool for RE is limited to two areas: 1) requirements analysis utilizing structured methods and 2) requirements management. Unfortunately, the critical areas of requirements elicitation, understanding and negotiation are not properly supported by CASE tools [9].

Although various techniques and tools have emerged and accommodated by the RE process, there is still a demand for RE process improvement. The literature review summary in Figure 1 indicates that UML and Scenario have gained enormous popularity as promising vehicles for this research area. Another pervasively adapted visualization classification is the Diagram Flow or Chart [10, 11, 12]. Additionally, Callele [13] has suggested exploiting "physualization" physical manipulation of visualization entities like stickers, transparency, marker, and sketchpad as building block in RE. As can be seen, considerable efforts have been done in this field, yet a variety of problems – inadequate understanding, scope change and requirements volatility – are still left over to be resolved. Because of that, an icon-based language can be brought into play to ameliorate RE contexts.

Visualization Categories	Publication Papers			
	2008	2009	2010	2011
Symbol/Pictogram/Graphic	[K. K. Birmrah]	[F. Perez] , [K. Siau]	[A. R. Oliverira], [F. Ricca]	[E. R. Luna]
Flow/Graph/Chart	[K. Schneider]	[K. Wnuk], [L. Kof]	[S. Supakkul], [T. Ugai]	[E. R. Luna], [L. Santos]
Rational Notation/Ontology			[C. Martinie], [W. F. Tichy]	[A. Rago]
UML	[I.Cardei], [F. Meziane], [D. Lubke]	[M.Kamalrudin], [K. Siau]	[J. Helming]	[F. Yang-Turner], [M. El-Attar], [F. Solms], [M. Kamalrudin]
Scenario		[G. Gabrysiak, [S. Uchitel], [N. Seyff]	[S. Supakkul], [G. Mussbacher], [A. S. Penim]	[M. Alawairdhi], [A. Sutcliffe], [E. R. Luna], [G. Atladottir], [G. Mussbacher]
Tabular	[B. Palyaga]			
Physical and Multimedia			[D. Callele]	[K. Pitula]
Goal-Model	[C. Rohleder]		[J. Horkoff], [D. L. Moddy]	

Figure 1. Visualization contribution in each category from 2008-2011.

2. Theoretical Foundation

The research vision is to develop visual sets of icons for supporting multifaceted stakeholder environments such as users, developers, requirement engineers, business analysts and project managers in requirements development context. The following research methodology and system model are encircled to reinforce the vision:

2.1. Research Question and Methodology

This paper urged by the research problem that requirements are typically not the product of a single individual's work but rather are the result of many parties'

articulated need over different communities. There have negatively impact on misinterpretation in requirements gathering, negotiation, validation and specification when cross-functional stakeholder groups specify requirements across cultural, language and time zone boundaries. The framework encapsulated in the above key concepts can provide the answer to the question of how the tasks of software developers and other stakeholders in the RE process can be supported by icon-based language to handle interpretation circumstance and to offer easy-learnt modelling.

The overall icon-based language concept is developed using a design science research method. The concept is equipped with a support mechanism for icon-based language development and is designed to carry out the following three principal phases: theoretical study, design a proposed solution, and empirical evaluation.

- Phase1- Problems and objected identification: it is essential for this work to identify problems existed in the whole RE and to understand the real interest of stakeholders. This phase can be accomplished by a literature review of previous research and of a collection of available icons.
- Phase2- Design solution: the development of the icon-based language is designed corresponding to three main steps: - identifying RE world, defining icon world and integrating of those two worlds by carrying out RE process.
- Phase3- Empirical evaluation: it is a necessary procedure in order to maintain stable services and user satisfactions. Empirical evaluation on the defined concept to validate whether the icon-based language facilitates the real RE practice. Two practitioner groups will be assigned to be participating evaluation: - the first group player is students in the RE course offered by the University of Jyväskylä and - the second group is software companies in Thailand and Finland. Usability testing will be used as evolution methods.

2.2. System Model

In Figure 2, the model is grounded upon a theoretical framework for SDLC, the RD process, system context and stakeholder facets refined by [1]. It is directed towards the construction of a new way to represent icon-based language in the RE process (elicitation, analysis and negotiation, validation, documentation and management). The delivered system endeavouring to accomplish stakeholders' daily tasks is relied heavily on structures of roles, processes and responsibilities. The system differs from a traditional approach like the Waterfall in terms of its presentation and iterative process. With the Waterfall model, each phase is assigned to a different team. The model is allowed only a single run through the full process and has a textual display, whereas our methodology works jointly with the stakeholders along all the activities required by the iconic protocol. Requirements activities are maintained through whole cycles since they can be revisited and reconsidered in individual iterations [1]. Besides that, human principal concerns are included as important elements of system design to reduce effort exploited, to provide interaction experience to the user and to prevent disastrous human errors. Other utmost critical components are version and status control by which - version number is increased automatically when certain changes occur, - changes and old versions are always be available, - a requirement can be changed back to any previous state anytime, and – status progress of each requirement and that of the project must be acknowledged in every state.

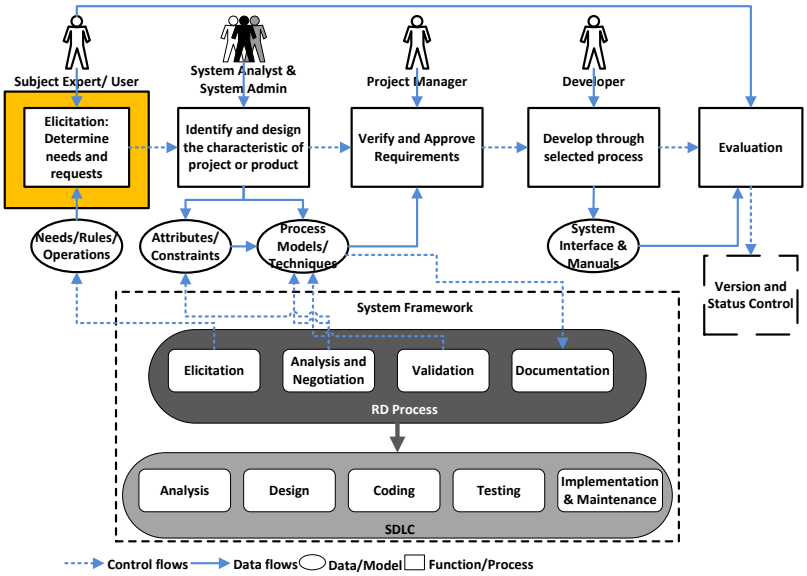

Figure 2. A scope of the system model for Icon-based Language in requirements development process. This paper focuses on the left hand side of the diagram (elicitation tasks). The system framework components, RD process and SDLC, are outlined from the existing theories.

3. A Proposed Concept of Icon-based Language in the RE Context

In this section we illustrate the proposed concept of icon-based language by presenting its three main parts: defining RE world, defining icon world and integrating those two worlds by means of RE process.

3.1. RE Artefact

The scope of RE world has been skeletonized into two main catalogues, namely, TaxonomyModel and RequirementModel.

3.2.1. TaxonomyModel

The taxonomy meta-model illustrated in Figure 3 has been modeled for general characteristics of requirement elicitation. It contains a set of potential tasks, actors, and linkable elements. The model is characterized by name and may contain different kinds of tasks types (see Figure 4). Underneath the task class, it may contain requirements taxonomy: functional requirements, non-functional requirements, business requirements and business rules. A linkable element abstracts the model element that can be linked to other taxonomy types such as analysis tasks. Each elicitation task has its relevant stakeholders' role and responsibility. For instance, domain understanding is relation to business team and system analyst. From that reasons, it is necessary to record some information about individual stakeholder (actorID, name, and description).

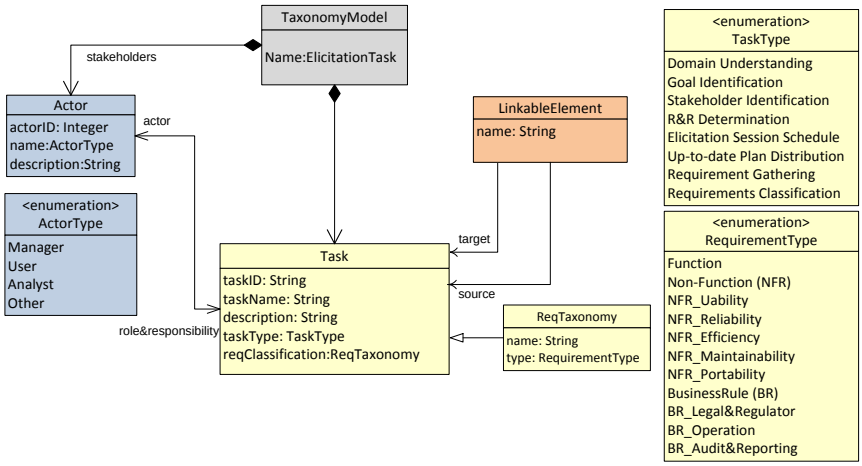

Figure 3. The context of requirements engineering world (taxonomy) meta-model

Establishing different types of requirements elicitation tasks in a project helps the team members to engage in consensus activities that will consistently yield results and to communicate more clearly. The eight artefacts can be found at the decomposition of the elicitation process denoted in Figure 4. The diagram explains the role of the tasks in the requirements elicitation life cycle. *Domain understanding* process studies the problems of the system to be developed, its contexts, environments, expectations of stakeholders and the overall impacts. As an outcome of this process, *goal identification* provides a complete description of the needed functionality, behaviour of the system and *relevant stakeholders* in charge of the development. The workflow model produces the *roles and responsibilities* of all stakeholders in the project. *Session scheduling* is designed to establish the plan and the agenda for capturing requirements in accordance with the expertise needed for each domain. *Up-to-date plan distributing* provides the stakeholders with the latest RE process, demonstrations, features, and elicited comments or suggestions. *Requirements gathering* process turns the abstracts demands into real-world goals and specifies a system that is best-geared to those goals. *Requirements classifying* procedure formalizes the elicited requirements to a coherent cluster of business rules and business, functional and non-functional requirements.

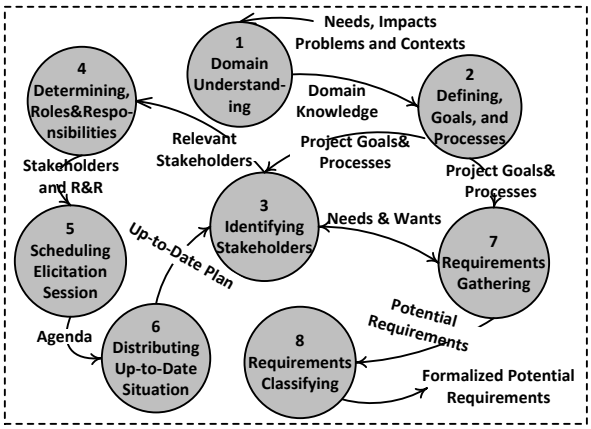

Figure 4. Identification of attributes for requirement elicitation tasks

3.2.1. RequirementModel

The Requirement meta-model depicted in Figure 5 has been defined to model general characteristics of requirement artifact. It contains a set of related requirements and actors. All requirements specifications are stored in a model to promote requirements reusability. The model is characterized by name and may contain three different kinds of requirements: business, functional and non-functional requirements. Business requirements are high-level requirements that reflect a goal or vision of the organization that the system must accomplish. Under business requirements, it may contain functional requirements that present a behavior of a system under specific condition. Or, it enables to contain non-functional requirements that represent a quality attribute in which the system must have. These attributes are not system features, yet they do encourage how the functionality of the system is developed. Non-functional requirements usually cope with some aspect of usability, security, availability, or maintainability. A requirement may be attached with business rules such as a law, policy, standard or procedure, which restrict or constrain the certain degree of freedom in delivering a solution.

A Requirement is characteristics of requirements artifacts that have a unique identifier (ID property), a name, a description, highLevelGoal (linking to BusinessReq), a priority (taking values from PriorityType enumeration), a status (determining the requirements life cycle values from StatusType enumeration), and a number of changes (NoOfChange). BusinessReq contained by a Requirement class relies on a unique identifier (ID), type (selecting one or more of the values pre-defined in a series of RequirementType enumeration), and constraint (grasping the value from BusinessRule enumeration). RequirementType sets have been identified on the basis of ISO/IEC 9216: Quality characteristics and guidelines for their use [14]. Each requirement is proposed by Actor. It is important to keep information about individual actor (actorID, name, and description) for further inquiry.

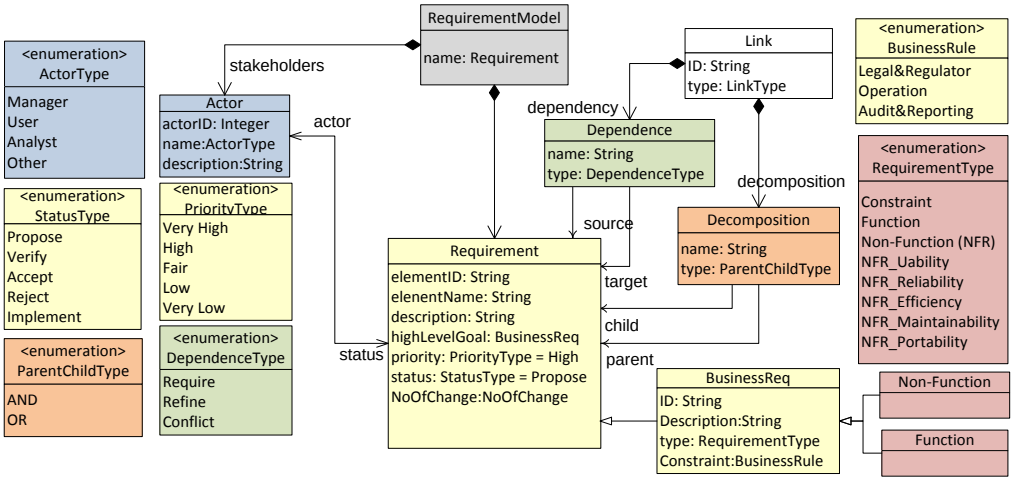

Figure 5. The context of requirements engineering world (Requirement) meta-model

3.2. Icon Artefact

The early stage of iconic implementation is illustrated in this section. To make feasible visual sentences supporting requirements elicitation process, there compose of three basic components: visual vocabulary, visual syntax and semantics.

3.2.1. Visual Vocabulary

Accordingly two types of icon-based language meta-model (Taxonomy and Requirement), visual vocabularies are, then, have been delineated into two series, one for Taxonomy and another one for Requirement. Supportive Taxonomy icons are statistical and individual, that is, there are eight icon set for eight elicitation tasks whereas supportive Requirement icons are varied accordingly to their purpose and function. There are twenty four constructions comprising of four element groups (actor, status, priority and number of change) and five relationships (three dependency types and two parent-child types) as depicted in Figure 6a and 6b. In total, there are thirty four distinct iconic symbols: thirty symbols (eight for Taxonomy and twenty two for Requirement), two logical signs and two link types.

3.2.2. Visual Syntax

In the current research, visual syntax is designed based on the principle theories in [2, 15]. To maintain clarity of icon notation, we enact one restriction, that is, no more than the three different signs are combined into single signification. Two types of visual syntax or grammatical rules have been designed to support Taxonomy and Requirement in RE world.

3.2.1.1. Visual Syntax for Taxonomy Type (the Elicitation Tasks)

- Each visual notation can contain at least one of these two components: a verb or action (e.g. "to read") and object (e.g. "book"). Both verb and object are portrayed graphically, using pictorial, sign and symbol representation without label caption surrounding. To realize the action, the verbs of each task are represented either with concrete images of the equipment tools or signs. The presenters of a tool "to set up or determine", and a funnel "to filter are for instance. Furthermore, arrows are introduced and combined with other sets of icons to add a novel interpretation. To avoid ambiguity of arrow's semantics, we give its meaning for a specific set of actions that carry out almost the same result such as "send", "become", "distribute", and "contribute".
- Orientation: the icons located above the other can be interpreted that they are the influencer of or have impact to the remainders in a specific visual notation as well as they must only be a kind of verb representative. One object is able to be doubled or tripled to reach another meaning of group, classification or category, for example, duplicating a document and making overlap of duplicated objects, it generates new meaning of category.
- Color: the icons representing Taxonomy type are completely abstraction therefore icons are proposed without color composition because to bring color into designing for multicultural stakeholders, it needs more cultural learning and experiencing. Color in different cultures has different meanings and

irreplaceable therefore it is a significant element, not only for human-computer-interface but also for all forms of international synergy

3.2.1.2. Visual Syntax for Requirement Type

- Actor: stick figures would be used to represent actors because they are globally interpreted for the representation of people. Variations of stick figures could help reader to distinguish the different types of actors. For instance, a smile stick can be representative of user, and a stick with sunglass and magnifying glass can be referred to system analyst.
- Node Type: goal symbol representing business requirements is composed of a football surrounding with a goal which is a sport metaphor. A normal rectangle usually accessible in almost all of diagrams determines a Function requirement. In contrast, rectangle used again for Non-function, but it is supplemented by black triangle plus an incorrect mark at the bottom left corner. For the constraint, a triangle shape to exemplify the warning meaning as in traffic sign is planted with the image of chains to signify the restriction.
- Dependency Link: different lines and arrows are exploited to represent require, refine, and conflict relationships.
- Priority Type: the different vehicles are used to represent the priority disgusted from their velocity: aircraft for "Very High", train for "High", car for "Fair", bicycle for "Low" and baby carriage for "Very Low".
- Number of Change (NoOfChange): the measurement toolbar is adapted to represent the number of requirements change occurred. Lightness is brought into consideration in illustrating the critical situation of change: the lightest illustrates zero change while the darkness is demonstrated when reaching to the max value of change.
- Status Type: different behaviors or interaction of people can used to represent status states, for example, when one person sends something to another one it can implied as Propose.
- Parent-Child Type: the standard logical signs of AND and OR gates are used to represent the relationship between a parent and a child requirement.

3.2.2. Semantics

The semantics of an icon-based language is represented in the meta-model (See Figure 3 and Figure 5). Each syntactic creature is arranged to some semantic construct. In the same fashion as syntactic definition, two groups of semantics are for Taxonomy (Figure 6a) and Requirement (Figure 6b) in the icon-based language meta-model.

Icon/Visual Sentence	Semantics	Icon/Visual Sentence	Semantics
	Domain understanding		Elicitation Session Scheduling
	Goal Identification		Up-to-date Plan Distribution
	Stakeholder Identification		Requirement Gathering
	R&R Determination		Requirements Classification

Figure 6a. The semantics of icon-based language in the requirements engineering Taxonomy type

Icon/Visual Sentence	Semantics	Icon/Visual Sentence	Semantics
	Manager or Management Team		"Very low" for lowest priority and severity: Quantitative value = 1
	Business Team		"Require": requirement R1 requires requirements R2 if R1 if fulfilled only when R2 is fulfilled. R2 can be acted as a pre-condition of R1.
	System Analyst or Requirements Engineer		"Refine": requirement R1 refines requirements R2 if R1 is derived from R2 by intensifying more detail on it.
	Other actor, it can be specified by modeler		"Conflict": Requirement R1 conflicts with requirements R2 if the negation of R1 can be inferred from R2 or the removing of R1 no longer results in R2 and vice versa.
	A Goal is a condition or state of affairs in the world that actor would like complete.		"Propose" is used for specific the reasoned element or requirement that has been requested by stakeholders in the first step.
	A Function provides operations and captures the intended behavior of the target system to meet the needs stated in the desires and goals.		"Accept" is done after analyzing and estimating A requirement for its impact and being committed by the key stakeholders to implement.
	A Non-function is used to represent design rational of qualitative requirements.		"Implement" refers to the code that implements the requirement has been designed, written and tested
	A Constraint such as business rules law, policy, or procedure which restricts the certain degree of freedom in delivering a solution.		"Verify" Is about the correct functioning of the implemented requirement that has been confirmed in the integrated system
	"Very high" for specifying the highest and urgent importance: Quantitative value = 5		"Reject" denotes the requirement that has been proposed but it is not planned for implementation in any upcoming release.
	"High" for high importance with not much urgent: Quantitative value = 4		"AND" refers to the relationship of a parent requirement and child requirement when a parent requirement need a child requirement to justify.
	"Fair" for a level of moderation in between High and Low: Quantitative value = 3		"OR" happens when a parent requirement could be either fullfiled or not fullfiled by a child requirement
	"Low" for some non-null importance: Quantitative value = 2		Number of Change, the value will start from minimum of zero and maximum of any configurable number

Figure 6b. The semantics of icon-based language in the requirements engineering Requirement type

3.3. The Integration of RE and Icon Worlds

The design of an icon-based language in the RE process requires the specification for each phase, the goal to accomplish it, and the notations and possible modelling they can use. The integration of RE artefact and icon artefact can be carried out by means of RE process started by creating scope and vision document, use case scenarios and ended up with the requirements specification for further development. Our concept focuses on the modelling of visual notations providing the means for RE stakeholders to cooperate with icon-based language in the context of RE shown in Figure 7. First of all, we need to derive the icon library of a visual notation being designed. After analysing and defining the requirements elicitation context, icons are necessarily produced to simplify that context. For each icon notation that has to be conducted, we must generate a final Icon Library. The Icon Library contains the series of iconic symbols and visual sentences symbolizing the icon notation, and their semantic interpretation. During the Iconic Syntax Modelling, we refine the specification of the iconic symbols according to the attribute-based representation approach. In Semantics Generator stage, we analyse the semantics aspects of the icon-based language model notation and designate extra semantics attributes for its visual symbol. The entire stages in icon-based language design are iterative which means that if tests reveal usage drawbacks, we might decide to review the Icon Library, Grammatical Rule and Semantics Library, as well as, to replicate the usability testing in the next version.

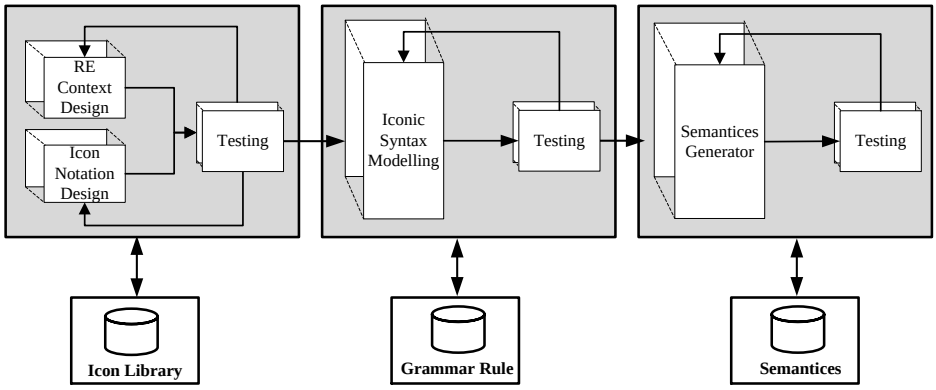

Figure 7. The process for modelling the icon-based language

4. Evaluation Plan for Icon-based Language in Requirements Elicitation

The empirical evaluation need to be setting up for testing the concept of icon-based language. It includes the formulation of questionnaires and test case scenarios. The best way to reach the diverse participants is a web-based icon test. By having a survey dispatches on the Internet, it can possibly grasp the information from any person in any location that has access to the Internet. Different types of survey questions will be asked to validate if the define concept could be support RE stakeholders in real practice. Following three questions deal with the efficiency of iconic communication in relation to icon usability.

Icon meaning: we give the permission to the subject and map the icons and their meaning from two pertinent lists: one of icons and another one of meanings (portrayed in Figure 8).
Icon Construction: we have the pre-defined set of icons and ask the subjects to create iconic sentences from those provided icons.
Requirements category: we arrange a specific set textual requirement statements that refer to requirements taxonomy and ask the subjects which category does each statement belong to?

Figure 8. An example of icon meaning scenarior.

4.1. Target Audience Group 1: Students in Requirements Engineering Course

Students of the RE course of the Department of Mathematical Information Technology at the University of Jyväskylä will be the first group assigned to participate in an empirical evaluation session. Being aligned to the selection criteria, i.e., those practitioners should be naïve, amateurs and have multicultural characteristics, students are outstanding candidates.

4.2. Target Audience Group 2: Software companies in Finland and Thailand

The second group to appraise the proposed series of icons will be software companies in Finland and Thailand. The first condition for selecting candidates is that they have to have multicultural backgrounds. A half of the selected companies should meet the second limitation that the RD process must be defined and used regularly in their software development. The rest have to conform to the third restriction that they do not define the RD process as a standard practice in the development life cycle.

5. Conclusion

The fundamental problem of misinterpretation of and distorted information about requirements is happened because usually, user needs may not be clearly expressed initially in the requirements. And the developer or requirements analyst may make some incorrect assumptions based on this ambiguity. Motivated by the misinterpretation and communication problems of RE in universal software development, visualization comes as an outstanding technique for increasing software quality and efficiency. With an intuitive process through icon representations, those mistaken assumptions would be handled faster and corrected sooner. The fact that a strong RE process can mitigate risks when implementing software development drives this research to more ambitious directions. The contribution of this paper is to bring into attention the capabilities of a cross-border communication system that avoids the deployment of words and depends exclusively on a set of pictorial icons that are easy to handle and provide an alternative visual modelling language to execute requirements activities. Human-computer interaction is likely to benefit from this work as well.

One interesting topic for future research would be to extend the icons to software requirements analysis context. The topic can be seamlessly expanded from the current work but with some altogether different questions asked, for example: How to transform requirements submitted by users into a form of visualization? Can icons function totally independently without having any text for help? Do we need an icon dictionary or a grammatical structure to support pure icon representation?

References

[1] Pohl, K. Requirements engineering fundamentals Principles and Techniques, Springer, Berlin, 2010.

[2] Moody, D. L., Heymans, P., and Matulevicius, R. Improving the Effectiveness of Visual Representations in Requirements Engineering: An Evaluation of i* Visual Syntax, 17th IEEE International Requirements Engineering Conference, pp 71 - 180, 2009.

[3] Moody D. L. The "Physics" of Notations: Towards a Scientific Basis for Constructing Visual Notations in Software Engineering, IEEE Tran. on Software Engineering, vol. 35, pp. 756-778, 2009.

[4] Khanom, S., Heimbürger A. and Karkkainen, T. Icon-based Language in Requirements Development, 22nd European Japanese Conference on Information Modeling and Knowledge Bases, 2012.

[5] Heimbürger, A., Kiyoki, Y. and Ylikotila, T. Communication Across Cultures in the Context of Multicultural Software Development, Reports of the Department of MIT. No. C 1/2011. 64 p. 2011.

[6] Heimbürger, A., Kiyoki, Y. and Kohtala, S. Intelligent Icons for Cross-Cultural Knowledge Searching, 21st European-Japanese Conference on Information Modelling and Knowledge Bases, Estonia, 2011.

[7] Issa, A. A. and AlAli. A. I. Automated requirements engineering: use case patterns-driven approach, IET Software, Vol. 5, Iss. 3, pp. 287–303, 2011.

[8] Harvey, G. Requirements Engineering: State-of-the-art Report, Defense Research and Development Canada, 2010.

[9] Kotonya, G. and Sommerville, I. Engineering, Processes and Techniques, John Wiley & Sons, 2002.

[10] Ugai, T., Hayashi, S. and Saeki, M. Visualizing Stakeholder Concerns with Anchored Map, Fifth International Workshop on Requirements Engineering Visualization (REV 2010), 2010.

[11] Oliverira, A. R., Araujo, J. and Amaral, V. The VisualAORE DSL, Fifth International Workshop on Requirements Engineering Visualization (REV 2010), 2010.

[12] Helming, J., Koegel, et al Towards a Unified Requirements Modeling Language, Fifth International Workshop on Requirements Engineering Visualization (REV 2010), 2010.

[13] Callele, D. Physualization: Going Beyond Paper Prototype, Fifth International Workshop on Requirements Engineering Visualization (REV 2010), 2010.

[14] ISO/IEC. ISO/IEC 9126: Information technology - Software Product Evaluation - Quality characteristics and guidelines for their use, International Organization for Standardisation, 1991.

[15] Aykin, N. Usability and Internationalization of Information Technology, 2005.