Tackling the Requirements Jigsaw Puzzle

Maria Pinto-Albuquerque
School of Technology and Architecture,
ISCTE-IUL (Inst. Univ. Lisboa), Portugal, and
School of Computing and Communications,
Lancaster University, UK
maria.albuquerque@iscte.pt

Awais Rashid
School of Computing and Communications,
Lancaster University, UK
marash@comp.lancs.ac.uk

***Abstract*—A key challenge during stakeholder meetings is that of presenting the requirements and conflicts to stakeholders in a way that fosters co-responsibility and co-ownership regarding the conflicts and their resolution. In this paper, we propose a jigsaw puzzle metaphor to make identified conflicts explicit as well as an associated method to utilise this metaphor during stakeholder meetings. The metaphor provides an easy to understand language for stakeholders from otherwise diverse backgrounds. It enables stakeholders to work with a well-understood concept - that of building a system from misshapen pieces. These characteristics foster communication and team work, which improve commitment of stakeholders in co-authoring of requirements and co-responsibility in conflict handling. The gamification of conflict resolution also promotes a relaxed environment, which in turn improves team cooperation and creativity. Our experience in three user studies demonstrates that the jigsaw puzzle indeed improves such co-responsibility and co-ownership when compared with typical text-based representations of requirements.**

***Index Terms*—Requirements, conflict, creativity, game, jigsaw puzzle, stakeholders, team work, communication, metaphor, visualization.**

I. Introduction

Although brainstorming, one of the most popular creativity techniques used for requirements identification, dates back to 1935, the use of creativity techniques in RE is still under investigated [25]. In this paper, we focus on fostering creativity with regards to resolving requirements conflicts during stakeholder meetings. Such meetings often take place between analysts and customer stakeholders to inspect and/or negotiate conflicts in requirements, and to update requirements as needed. A key issue during such meetings is that of explicitly presenting the requirements and the conflicts to the stakeholders in a way that fosters co-responsibility and co-ownership.

Fostering such co-responsibility and co-ownership is challenging for a variety of reasons. Firstly, though methods have been developed to provide decision support for identifying and handling ambiguity and inconsistency, these methods "code" such information (that depicts the conflict) in a technical formalism. Stakeholders, however, often possess heterogeneous backgrounds. Current formalisms (e.g. [6, 11, 19, 27]) are not readily accessible to users with non-computing backgrounds (and even different cultural backgrounds in current multi-national settings can pose a challenge in this regard). Secondly, studies [20] have shown that analogical reasoning with unfamiliar domain classes is difficult without prior learning. Finally, RE is not only an analytical task but also a creative one. The use of too many analogies during stakeholder meetings can hamper analogical reasoning [21, 22] and hence creativity.

Of course, requirements conflicts arise for a variety of reasons. On the one hand, the vast majority of requirements documentation is described using natural language [24]. As natural language is inherently imperfect [9, 23], requirements may contain conflicts due to incompleteness, ambiguity, or simply the definition of the same feature in two incompatible ways (inconsistency) [3]. On the other hand, stakeholders often have different interests. These interests/desires may be technically impossible, may place other requirements at risk or lead to increased costs [33]. In this paper, we are not concerned with the root causes of conflicts. Instead, our focus is on how to present these conflicts to stakeholders during consultation meetings in a fashion that enables them to collectively recognize the challenges posed by the conflicts and the obstacles that have the potential to compromise the project as a whole, and its quality.

It has been argued that a lack of specialist notation improves communication in multi-disciplinary contexts [26]. Given the typical diversity of stakeholder backgrounds, we therefore focus on communicating conflicts through a "language" that is easy to understand and relate with. We propose a jigsaw puzzle game to make identified conflicts explicit. In this jigsaw puzzle game, each puzzle piece represents a requirement. When the requirement text contains conflicts with other requirements, the respective puzzle pieces almost fit together but not perfectly. The jigsaw puzzle metaphor provides an easy to understand language for stakeholders from otherwise diverse backgrounds. Furthermore, it enables stakeholders to work with a well-understood concept – that of building a system from misshapen pieces. These characteristics foster communication and team work, which improve commitment of stakeholders in co-authoring of requirements and co-responsibility in conflict handling. The gamification of conflict resolution also promotes a relaxed environment, which in turn improves team cooperation and creativity.

Our experience in three different user studies shows that the jigsaw puzzle indeed fosters such team work and communication compared with the use of typical textual presentations. It also improves the commitment of stakeholders in the co-authoring of requirements and co-responsibility in conflict handling.

The novel contributions of this work are as follows:

1) A visual metaphor in the form of a jigsaw puzzle to communicate requirements conflicts to stakeholders. Though work exits on use of information visualisation [10, 2, 36], games [18, 31] and creativity techniques [22, 25] during requirements engineering, such a metaphor and game has not been considered to date. Given the easy to understand and gaming nature of the metaphor, there is little or no need to introduce the metaphor (as we found in our user studies). The meeting itself is a game, a jigsaw puzzle, which is at the same time the conflict communication mechanism and the means to foster discussions regarding conflict resolution strategies.

2) Our use of a jigsaw puzzle metaphor leads to a separation of the processing of the information about the conflicts from the issue of communicating those conflicts. This separation is crucial to permit a user-centred design of the communication of the requirements and its conflicts. Such a separation, though critical, has not been proposed to date.

The reminder of this paper is organized as follows: Section II summarizes the related research. Section III describes the jigsaw puzzle game, while Section IV explains how the jigsaw puzzle should be used in the stakeholder consultation meetings. Section V presents the evaluation of the approach. Section VI discusses what the potential impact of the insights from the jigsaw puzzle game hold for the state-of-the-art in stakeholder consultation meetings and the interactions between requirements engineers and stakeholders, as well as the future work we envision. Finally, Section VII presents the conclusions.

II. Related Work

A. Requirements Elicitation and Creativity

Maiden et al. have developed a RE process, RESCUE [22], which incorporates creativity workshops to foster creative thinking to discover requirements. These creativity workshops use several techniques to encourage creativity, including brainstorming and analogical reasoning [20, 21, 22]. Their work demonstrates that: 1) Workshop participants found it difficult to exploit the analogies (e.g., between air traffic management and textile and music domains) [20, 21]; and that strategies are needed to make analogical reasoning more effective [22]. 2) Brainstorming generated more creative ideas than analogical reasoning, and it was more cost-effective and easier to use. However the ideas obtained with analogical reasoning were described in more detail than the brainstormed ideas [21].

The jigsaw puzzle game aims to address the issue of analogies that are difficult and costly to explain and learn. It provides two easy to understand visual analogies. One is the analogy between a jigsaw puzzle and a system both built out of pieces to create a desired object. This analogy reinforces the view of RE as creative task, not only analytical one. The other analogy employed uses a misshapen graphical visualization of a puzzle piece to represent that there is a problem.

Games are often used at the beginning of elicitation meetings, to induce a fun and relaxed environment, and promote creativity [22]. In this approach there is no need to introduce a game; the meeting itself is a game, a jigsaw puzzle, which is also the communication mechanism.

B. Visual Communication Metaphors in RE

Gotel et al. [14] envisioned the potential synergy between Information Visualization and SE Visualization to provide good metaphors for software development tools, enabling effective communication in software development. Visual metaphors have, for a long time, been used to represent information in SE. In the last fifteen years the predominant visualizations in the area of RE are either associated with UML diagrams or i* goal models [14]. Diagrams use geometric shapes and geometric relations to build a visual language with defined syntax and an associated semantics. These visualizations present apprehension problems (i.e., the structure and content of visualization is not readily perceived and comprehended [35]). Thus, they need to be learned. Field studies [5] confirm that the visualization’s efficacy in diagrams is dependent on the user’s previous experience and visual literacy, which can present a barrier for communication and work among multidisciplinary user profiles, and thus between requirements engineers and stakeholders. Some more sophisticated visual metaphors have been recently developed to visualize software (e.g. [10, 2, 36]), but with different goals compared to this work. Boccuzzo et al. [4] describe an interesting usage of the concept of well-shaped graphical visualization to represent that the corresponding artifact is well designed. Their work shows that the metaphor “language” can be used to express information (quality aspects) about the artifacts that are being represented. Boccuzzo et al.’s work inspired us to develop the analogy that uses a misshapen graphical visualization of a puzzle piece to represent that there is a problem.

C. Inconsistency and Ambiguity Handling

Inconsistency identification has been recognized as a RE problem for many years. Viewpoints and Inconsistency Management (VIM) [13, 28, 11] and the Non-Functional Requirements Framework (NFRF) [6] provide specification formalisms that explicitly represent and manage relationships between the artifacts (ViewPoints, Softgoals respectively) that represent the knowledge about the system to be built. However, one needs to be knowledgeable in the formalism (and these are quite complex) to be able to manage the conflict relationships. The inconsistency detection tools (e.g. [30]) developed under the umbrella of aspect-oriented textual requirements approaches that use semantics-based decomposition present the output as a list of possible inconsistency situations. These tools, however, do not address the communication goals (in particular with non-SE stakeholders), which are the focus of the jigsaw puzzle game.

A number of techniques also exist to reduce the level of ambiguity in requirements (see e.g., [3]). Our approach aims to use the outputs of ambiguity and inconsistency detection tools as input to build the jigsaw puzzle game, making it a communication mechanism accessible to all kinds of stakeholders, thus promoting their participation and aiming at co-responsibility in requirements. A contribution of our work is heuristics to identify ambiguities and inconsistencies that may need to be flagged using the jigsaw puzzle game (see sections III.A and III.B).

III. JIGSAW PUZZLE GAME

In the jigsaw puzzle metaphor each puzzle piece represents a requirement. When the requirement's text potentially leads to conflicts with other requirements, the respective puzzle pieces have a matching edge that almost fits but not perfectly. The background of each jigsaw puzzle piece contains part of a picture. When all the pieces of a puzzle are assembled correctly each piece contributes to a bigger complete image, just like in commonly used jigsaw puzzles. The jigsaw puzzle game uses the cues commonly used in jigsaw puzzle pieces: pieces that have straight borders, and thus belong to corners and jigsaw borders; interlocking shapes that match pairs of pieces; and an image in the background, with each piece containing a part of it. Another cue is the orientation of the text of the requirements.

In order to demonstrate an example, we utilise the Non-functional requirements (NFRs) from the Crisis Management Systems (CMS) requirements documentation [17]. We note that this is a case study available in literature and any inconsistencies and ambiguities in the requirements exist in the original documentation and have not been introduced for the purposes of demonstrating the approach in this paper.

Figure 1 shows a picture of the four pieces of such a jigsaw puzzle representing the NFRs Availability, Reliability, Real-time, and Accuracy.

Figure 2 and 3 present the text of the Real-time and the Availability requirements, respectively. The text written in a jigsaw piece representing a requirement does not capture the full text as in the requirements documentation. Instead the goal is to capture essential elements in particular due to the need to improve readability. Some unnecessary text is cut out, some words are abbreviated, the text is displayed in list mode, and upper case letters are used to stress the "topic" of each requirement.

The interlocking shapes between the pieces in Figure 1 do not match perfectly, depicting conflicts between the requirements, represented by the pieces. For instance the bad fitting between the Availability piece and the Real-time piece represents the following inconsistency:

The Real-time requirement has the word expressions: "receive and update…information", "communication of information", and "retrieve any stored information". For these situations to happen, the system must be available, thus they are related to the following word expressions of Availability: "in operation except for a maximum downtime" and "recover…upon failure". The possible inconsistency situation resides in the following. If the system is allowed a downtime for 2 hours (as allowed by the Availability requirement), how can it guarantee the Real-time requirement, i.e., "receive and update…information…at intervals not exceeding 30 seconds", "the delay in communication of information…shall not exceed 500 milliseconds", and "retrieve any stored information with a maximum delay of 500 milliseconds".

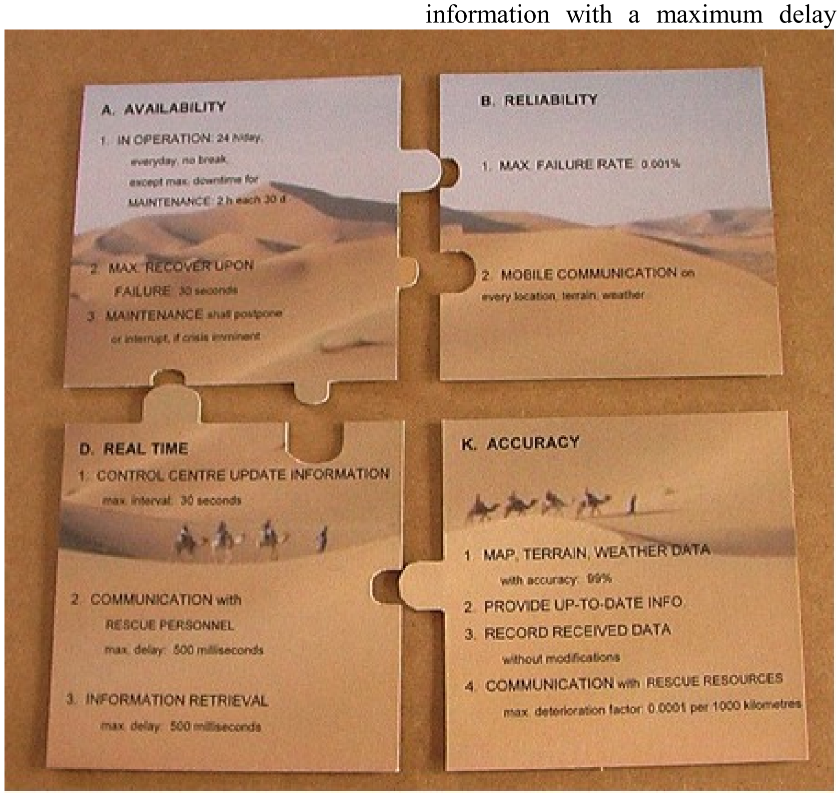

Fig. 1 The jigsaw puzzle for the requirements Availability, Reliability, Real-time, and Accuracy of the CMS case study

The same type of question can be posed in relation to the second phrase of the Availability requirement, which states: “The system shall recover in a maximum of 30 seconds upon failure”. But, thus if it may be up to 30 seconds down how can the system guarantee what the Real-time requirement demands?

• Real-time

1. The control centre shall receive and update the following information on an on-going crisis at intervals not exceeding 30 seconds: resources deployed; civilian casualties; crisis management personnel casualties; location of super observer; crisis perimeter; location of rescue teams on crisis site; level of emissions from crisis site; estimated time of arrival (ETA) of rescue teams on crisis site.
2. The delay in communication of information between control centre and rescue personnel as well as amongst rescue personnel shall not exceed 500 milliseconds.
3. The system shall be able to retrieve any stored information with a maximum delay of 500 milliseconds.

Fig 2. Text for the Real-time requirement [17]

• Availability

1. The system shall be in operation 24 hours a day, everyday, without break, throughout the year except for a maximum downtime of 2 hours every 30 days for maintenance.
2. The system shall recover in a maximum of 30 seconds upon failure.
3. Maintenance shall be postponed or interrupted if a crisis is imminent without affecting the systems capabilities.

Fig 3. Text for the Availability requirement [17]

A jigsaw puzzle game, such as the one in Figure 1, can be prepared using a machine but, for the purpose of this research it was just prepared by hand (though the pieces were generated using software implemented in MATLAB, they are printed and cut manually). In the longer term such a process can be fully automated though we note that, in consultation meetings with stakeholders, typically, requirements engineers want to focus on a small number of requirements (say 4 to 6) [33]. Therefore, the jigsaw puzzle needn't contain a large number of pieces so as not to detract from the key conflicts that should be focus of discussion. Other permutations are possible such as using touchscreen interfaces or augmented reality though they would lose some of the tactile and physical manipulation aspects of the jigsaw puzzle that were welcomed by users during our studies.

In order to generate the bad-fitting pieces it is necessary to identify the most pertinent or most problematic conflicts that should be the focus of the meeting. As we noted above, there is a significant body of work on ambiguity and inconsistency identification as well as tools, which can be used for the purpose. We, therefore, build on this body of work (as well as our own insights and experience) to provide suitable heuristics to identify sources of ambiguity or inconsistency that may need to be flagged using the jigsaw puzzle. We focus on both ambiguity and inconsistency identification heuristics. This is because the definition of ambiguity as being a property of both the text and the interpretations held by a group of readers of the text [37] enables to establish a connection between ambiguity and relations among requirements with unknown influence but with a potential to be an inconsistency. In such cases, the analysis by the stakeholders of the two pieces of the SRS (software requirements specification) to decide if there is an inconsistency promotes the clarification of the interpretation each person gives to the parts of the SRS in appreciation. This clarification of the interpretation(s) is what is needed to resolve the ambiguity, or to boost the discussion on the ambiguity itself. It can happen the other way round, i.e., when analyzing ambiguity in two parts of the SRS, the resolution of the ambiguity may clarify if certain relations between the requirements in those two parts of the SRS are inconsistent.

As our focus is on resolving conflicts amongst NFRs, often the heuristics below advise the search for word expressions referring to qualities or breaking of qualities. The domain of qualities to be considered for such search can be taken from catalogues of qualities. Such catalogues have been developed in the NFRF [6], with developments like the work of Cysneiros et al. [8]. These catalogues may also be standards such as the ISO/IEC 25010:2011 [16]. For each heuristic, we provide both the heuristic and an example. It is important to notice that the final selection of the sets of requirements, to be used in a group consultation meeting, belongs to the team of requirements engineers undertaking the RE process. The heuristics are a means to support the engineers' work in identifying the requirements that should be the focus of the jigsaw puzzle game.

A. Identification of the Most Problematic Ambiguities

We propose the following heuristics to identify ambiguity[1]. They draw upon the definition of an unambiguous SRS [15] , and the classification of ambiguity proposed by Berry et al. [3], and applies them to the domain of NFRs. Ambiguity is classified into: lexical, syntactic (or structural), and semantic.

1) Lexical Ambiguity Heuristic: Berry et al. define: “lexical ambiguity occurs when a word has several meanings” [3].

Heuristic - To find ambiguity in a SRS:

1. search for:
   - word expressions that belong to a catalogue of qualities (or their synonyms); e.g.: accurate; or
   - word expressions meaning a violation of quality (including antonyms of the qualities); e.g.: failure; and
2. check whether the word expression may have more than one meaning, enabling more than one interpretation.

Example. “Are control centre and system, referred in Real-time requirement, the same or different entities?” It is not clear if the expressions ‘control centre’, and ‘system’ refer to the same entity. There are at least two possible interpretations: one that ‘control centre’ and ‘system’ are the same entity; and the other that they are different entities. Such pairs of words constitute ambiguity cases quite often related with inconsistency.

[1] “A SRS is unambiguous if, and only if, every requirement stated therein has only one interpretation” [15].

2) Syntactic Ambiguity Heuristic: Berry et al. define: “syntactic ambiguity, also called structural ambiguity, occurs when a given sequence of words can be given more than one grammatical structure, and each has a different meaning” [3].
Heuristic - To find ambiguity in a SRS:

1. search for:
   - sequence of words referring to a quality from a catalogue (or its synonyms); or
   - sequence of words referring to a violation of a quality from a catalogue (including antonyms of the qualities); and
2. check whether that sequence of words can be given more than one grammatical structure, each having a different meaning, enabling more than one interpretation on how that quality or breaking of quality has to be realized in the system.

Example. “Is this a list of five types of interaction modes, or just three?” The CAS system SRS [1] describes: “SaaS applications offered through the Internet are usually supporting different interaction modes including classic page-oriented HTML GUIs, rich internet GUIs (e.g., AJAX) as well as access channels for mobile users such as WAP, data replication for offline use or speech control”. Different interpretations are possible depending on diverse assumed grammatical structures. It is possible to interpret that the phrase lists five “types of interaction modes”: a) classic page-oriented HTML GUIs, b) rich internet GUIs, c) channels for mobile users such as WAP, d) data replication for offline use, e) data replication for speech control. Or that there are just three types of interaction modes: a) classic page-oriented HTML GUIs, b) rich internet GUIs, and c) channels for mobile users.

3) Semantic Ambiguity Heuristic: Berry et al. define: “semantic ambiguity occurs when a sentence has more than one way of reading it within its context although it contains no lexical or syntactic ambiguity” [3]. We adopt the definition of context by Berry et al: “The SRS context comprises the language context (i.e. the sentences before and after the sentence in which the quality word expression occurs) and the context beyond the language (i.e. the situation, the background knowledge, and expectations of the speaker or hearer and the writer or reader)” [3].

Berry et al. define: “semantic ambiguity occurs when a sentence has more than one way of reading it within its context although it contains no lexical or syntactic ambiguity” [3]. We adopt the definition of context by Berry et al: “The SRS context comprises the language context (i.e. the sentences before and after the sentence in which the quality word expression occurs) and the context beyond the language (i.e. the situation, the background knowledge, and expectations of the speaker or hearer and the writer or reader)” [3].

Heuristic - To find ambiguity in a SRS:

1. search for:
   - word expressions that belong to a catalogue of qualities (or their synonyms); or
   - word expressions meaning a violation of a quality (including antonyms of the qualities); and
2. check whether the sentences using these word expressions have more than one way of reading it within the SRS context, enabling more than one interpretation on how that quality or breaking of quality has to be realized in the system.

Example. “Is downtime for maintenance in Availability requirement to be accounted for failure rate referred in Reliability requirement?” The first sentences of the Availability (see Figure 3) and Reliability requirements from the CMS document show an example of semantic ambiguity. Considering that each of these sentences is part of the context of the other; and considering also the background knowledge and the expectations of the readers, there are two ways of reading the following two sentences. From the Availability requirement: “1. The system shall be in operation 24 hours a day, everyday, without break, throughout the year except for a maximum downtime of 2 hours every 30 days for maintenance”, and from the Reliability requirement: “1. The system shall not exceed a maximum failure rate of 0.001%”. These two ways of reading are: 1) what is required in the second sentence concerning the ‘maximum failure rate of the system’ has to take into consideration the ‘maximum downtime’ allowed in the first sentence or; 2) these concepts are not related to each other. The ambiguity can be expressed with the question: should ‘downtime for maintenance’ be included in the situations considered for the calculation of ‘failure rate’, and thus the times allowed for ‘downtime’ have to be accounted in the ‘failure rate’ value; or ‘downtime … for maintenance’ is not to be accounted for in the ‘failure rate’?

B. Identification of the Most Problematic Inconsistencies

We propose the following heuristics to identify inconsistencies. They draw upon the definition of an internally consistent SRS[2], and our own insights inspired from existing RE approaches namely: Preview [34], and the NFRF[6].

1) Synonymous/Antonymous Quality Inconsistency Heuristic: When searching for pairs of NFRs defining features (qualities, in this case) of the system in an incompatible way, one should search for two phrases in the SRS referring to the same quality. Picking a quality ‘X’ referred with the word expression ‘X1’, other reference to the quality ‘X’ can be done using a synonym of ‘X1’; or in a negative form, using an antonym of ‘X1’. Preview [34] uses the concept of “focus” to find probable conflicts. Viewpoints whose foci intersect are the most likely sources of conflict. In fact these viewpoints with intersecting foci are the ones that impose requirements on the same system components or features, and thus the ones where conflicts are more likely to appear. When for a quality ‘X’ referred with a word expression ‘X1’, an antonym or synonym (say ‘X2’) of ‘X1’ is found in another part of the SRS, these two parts of the SRS are describing the same quality (i.e. impose requirements on the same features) and

[2]Using the IEEE Std. 830-1998 [15] and the definition of contradiction by Meyer [23] as a basis, we define that: A SRS is internally consistent if, and only if, no subset of individual requirements described in it define a feature of the system in an incompatible way.

thus they may conflict. In the Preview approach such a pair ('X1','X2') would belong to two viewpoints that have intersecting foci, and thus would be selected as probable conflicts.

Heuristic - To find an inconsistency situation in a SRS:

1. search two word expressions from a catalogue of qualities (or violation of qualities), which are synonym or antonym of each other, and
2. check whether their descriptions are incompatible with each other.

Example. "Something required concerning up-to-date information in Real-time requirement and nothing in Accuracy requirement". The possible inconsistency situation has to do with the fact that the Real-time requirement describes that "The control centre shall receive and update the following information … at intervals not exceeding 30 seconds." Thus it offers some detail about the "update of information". But the Accuracy requirement just says: "The system shall provide up-to-date information…" The issue is: "are these requirements compatible?" and "How can they be made compatible?"

2) Actions Operationalizing Quality Inconsistency Heuristic: In computer science and engineering it is known that an abstract concept such as a quality (e.g. Real-time) has to be operationalized in less abstract concepts usually described by actions (e.g. receive and update information). The description of the abstract concept and its operationalizations should be compatible. In Preview it is sometimes helpful to consider the decomposition of a viewpoint into sub-viewpoints. This is advised in the presence of conflict among the requirements of a viewpoint, especially if the sources of the requirements have imperfectly matched foci. This heuristic (as the Quality dependency inconsistency heuristic described below) aims to identify these cases of conflict among requirements of a viewpoint.

Heuristic - To find an inconsistency situation in a SRS:

1. search two word expressions of actions describing the operationalization of the same quality (from a quality catalogue), and
2. check whether these descriptions are incompatible with each other.

Example. "Real-time description requiring times possibly incompatible." In the Real-time requirement, described in Figure 2, it is questionable if the time intervals and limits described are compatible among them.

3) Quality Dependency Inconsistency Heuristic: It is pertinent to organize system qualities in hierarchies, with more general qualities at the top and more specific ones at lower levels of the hierarchy. 'Availability' is certainly a quality belonging to the top level. If a system is not available it is useless to discuss other qualities. At a lower level of abstraction it is useful (although arguable on how to do it best) to organize, for instance, in a tree below 'Performance': 'speed', 'efficiency', 'resource consumption', 'throughput', 'response time'. Such hierarchies can be found in standards (e.g. [16]). The NFRF [6] and its developments propose such hierarchies through catalogues that organize previously accumulated design knowledge per type of NFR (e.g. a catalogue about security, another about performance). In particular, NFRF catalogues support the discovery of inconsistencies when there are qualities that require other qualities.

Heuristic - To find an inconsistency situation in a SRS:

1. search two word expressions from a catalogue of qualities (or violation of qualities), and one of these qualities requires the fulfillment of the other quality, and
2. check whether their descriptions are incompatible with each other.

Example. "Real-time dependency on Availability incompatible." This example was explained at the beginning of Section III.

IV. Jigsaw Puzzle Game in Action

The goal of the consultation meeting is to use the jigsaw puzzle game to analyze conflicts in the requirements presented, the ultimate goal being to handle these conflicts, e.g. resolve or defer the resolution. The participants in the meeting will be the requirements engineers and stakeholders chosen according to the requirements and conflicts under analysis. The requirements engineers are the meeting facilitators.

The consultation meetings should be organized according to the following guidelines. These guidelines follow the established practice in RE [33], but were modified according to specific goals, namely to promote cooperation and team building, to improve communication and co-responsibility, as well as to induce creativity. These guidelines were derived from, and tested in the evaluation sessions, and are as follows:

1. The participants are asked to perform group work to assemble the jigsaw puzzle. This is arranged to be easy and quick. The goal is to promote a game playing context, "break the initial ice", and call the participants' attention to focus on the same "object", thus fostering cooperation and communication.
2. While assembling the jigsaw puzzle participants discover that there is a way the pieces fit but not perfectly. The fact that the pieces don't fit means that the way the requirements are written contains conflicts, which do not allow such a system to be built.
3. Once the puzzle is assembled, the participants are asked to read the text in the pieces, and scan what could be the possible sources of conflict. The goal is to get the participants to scan visually for the conflicts, instead of the conflicts being readout: this is important to improve the sense that the conflicts are everyone's problem and require cooperation and co-responsibility. Participants are encouraged to allow themselves to be questioned by the information the pieces convey, and to freely speak about any comments, and questions that came to their minds, as well as, to interact with other participants. Obviously, if the participants have difficulty in trying to identify the conflict cases, the facilitators can guide them.
4. Upon the discovery of a possible conflict, the group is challenged to analyze and discuss that conflict (including if it is really a conflict) and achieve a consensus on how to

handle it, for instance, additional; information that could resolve the conflict, rewording the requirements text (say, to remove ambiguity), or defer the resolution, etc. This promotes cooperation and team work among stakeholders.

V. EVALUATION

A. Goals

The goal of the evaluation was to test the following hypothesis, concerning the use of the jigsaw puzzle game during stakeholder consultation meetings:

1) *Hypothesis H1:* The use the jigsaw puzzle game promotes a relaxed environment and thus team cooperation and creativity.

The following two hypotheses act as second level hypotheses that help to contribute to validate H1:

2) *Hypothesis H2:* The effectiveness in communication and handling of requirements conflicts is increased, when compared with the same tasks performed with a text presentation.
3) *Hypothesis H3:* Team work and communication are fostered, improving commitment of stakeholders in co-authoring of requirements and co-responsibility in handling of conflict.

B. Research Methodology

The ideal research methodology would have been a series of case studies, since the context is expected to play a role in meetings between engineers and stakeholders [12]. As the access to real consultation meetings required by case studies was not possible, the experiments emulating "real-life" meetings were chosen as the best possible evaluation method. The evaluation was done through three such experiments, with a mixed philosophical stance: positivist and constructivist [12]. The experiments were confirmatory since they were used to test the hypotheses stated above. The experiments were also exploratory because they were used to understand the capabilities and problems of the proposed metaphor and method of working with it, leading to improvements.

C. Experiments Setup

The 1st experiment had 1 session, it was mainly exploratory and the 5 participants had RE background. The 2nd experiment consisted of 3 sessions and the 3rd experiment had 2 sessions. Every session was attended by a group of 3 to 4 participants, where at least one participant was an RE expert, while the others were non-software engineers, and at least one had no RE knowledge to emulate the role of external stakeholder. In the 3rd experiment the diversity in participants' background was extended with the inclusion of 1 participant with management background in each of the sessions. The 2nd experiment introduced a control group, asking the participants to analyze 2 different systems: one with a jigsaw puzzle presentation, the other with textual documentation. In the 3rd experiment, the design of the information given to the control group was improved, i.e., the "same amount and type" of information was given for the system described in text as for the one presented through the jigsaw puzzle. This was done by introducing in the text presentation phrases saying "Detected conflicts between requirement X and requirement Y", exactly for the same situations that were signaled in the jigsaw puzzle with interlocking shapes. The SRS used were: the CMS [17] (the requirements Availability, Reliability, Real-time, and Accuracy); the Health-Watcher system (HW), which is a web-based system to manage health-related complaints [32] (the requirements Availability, Performance, Security, and Standards); and the CAS system, which is a customer relationship management application [1] (the requirements Availability, Security, Multi-channel access, and Accurate and Up-to-date information). A more detailed description of the experiments and their results can be found in [29].

D. Results

Concerning the hypothesis H1, it was observed that the fun side of assembling the jigsaw puzzle clearly contributed to a more informal and relaxed environment. Participants had fun and enjoyed the sessions. They cooperated as a team with open-mind and looked for ideas to solve the problems. Some tried to assemble the jigsaw puzzle pieces with the picture up-side down, which indicates they had their minds open to do things in an unusual way.

Concerning the hypothesis H2, the experiments' results demonstrate that the jigsaw puzzle game exhibits strong potential to increase the effectiveness in communicating and handling conflict in stakeholder meetings, when compared to a textual presentation. The most significant result sustaining this statement is that all participants (17), except one (who showed no preference), preferred to work with the jigsaw puzzle than with the textual presentation. In experiment 1 (where 5 of the participants were knowledgeable in RE and thus able to evaluate the comparison of the proposed approach against text) the participants concluded: "the jigsaw puzzle-based presentation is really good in helping identifying problems and conflicts". In experiment 1 the participants were able to discover all four conflicts the puzzle intended to make explicit, for the CMS SRS. In experiment 1 there was no comparison with a textual version. In experiment 2 the participants discovered all three conflicts in the CMS puzzle, but none in the textual version of the CMS (which had no visual cues for conflicts). In experiment 3 the participants discovered three of the four conflicts in the CMS puzzle, but just one in the textual version of the CMS (with visual cues for conflicts). It is also significant that a number of participants were aware of and explained the jigsaw puzzle features that make them prefer to work with it. These features are as follows:

- The text presented in the jigsaw puzzle pieces is simpler and easier to read, there are keywords, smaller sentences, the letters are bigger, the "important" information is bullet-pointed and some is highlighted. Both the relevant topics and the relations between pieces are easier to spot. Some aspects are accentuated, enabling the users to think if, for instance, the numbers (shown in more than one piece for the same or related aspects) should match up;
- The jigsaw puzzle is colorful;

- Users can move the jigsaw puzzle pieces making it a more flexible presentation to work with;
- It promotes group work: everyone is focused at the same time on only one object;
- By presenting the requirements relationships through the interlocking shapes, users understand (visually) that requirements have impact on each other.

Concerning hypothesis H3, it was clearly observed that electrical engineers, and managers with no SE background, did not have any problem in understanding both the task to perform and the information that was conveyed with the jigsaw puzzle game: that there were problems to solve. Beginning the meeting with assembling a jigsaw puzzle also contributed to making everyone feel at the "same level". The contributions, in terms of inconsistencies and ambiguities identified as well as possible modes of handling them, came from participants in spite of their backgrounds. The easiness in participate as well as to perceive the message that there were problems enabled participants to behave co-responsibly and to contribute to the requirements specification. This demonstrates co-responsibility and co-authoring.

E. Insights and Discussion

The way the puzzle pieces are laid down imposes/proposes an order to scan the possible conflicts. In all cases users tended to analyze the relation of the first piece they picked (usually the top left piece) with the ones that were adjacent. The question raised was: is this a good or bad effect? Participants' opinion was that the way the puzzle pieces are disposed allows controlling the focus of the discussion. According to participants' opinion, the structure provided by the puzzle, though on the one hand promotes analysis in a certain order, on the other it does not limit people to find imperfections among pieces that are not directly connected. This opinion was supported by the results collected, since participants reported conflicts among pieces that were neither directly connected nor had badly fitting interlocking shapes. The discovery of conflicts not represented is normal since the jigsaw puzzle aims to represent just a subset of conflicts. As indicated by the participants' feedback, the use of different shapes (and geometrical structures) is an interesting area for further exploration.

As previously explained there is a connection between ambiguity and relations among requirements with unknown influence but with a potential to be an inconsistency. In such cases, the analysis of the two pieces to decide if there is an inconsistency promotes the clarification of the interpretation each person gives to the parts of the SRS in appreciation. This clarification of the interpretation(s) is what is needed to resolve the ambiguity. The 2nd and 3rd experiments provided insight that in such cases it might not be needed to have a visual representation for the ambiguity. It appears to be sufficient to provide a visualization for the associated inconsistency. There is ambiguity in the word 'up-to-date' in the Accuracy requirement as it does not describe what is meant by "up-to-date information". In those experiments there was no visualization for ambiguity in the word 'up-to-date'. However, the participants discovered (due to the jigsaw puzzle shape metaphor) both the ambiguity and the associated inconsistency "Something required concerning up-to-date information in Real-time requirement and nothing in Accuracy requirement" (see Section III.B.1).

The number of conflicts well communicated and thus identified, as well as the suggestions for their handling can improve even more significantly, with an interactive digital support. Several participants suggested having the jigsaw puzzle game as a digital interactive set up. In itself just the fact that there is digital support (other than pure text) for analysis and rationale descriptions (e.g. comments, arrows relating requirements) can bring enormous benefits: these analyses and rationale are not lost (are recorded), and can easily be distributed. If the digital support also provides intelligent real-time interaction, enabling that the user changes, for instance, the text of a requirement and the interlocking shapes are redesigned according to a recalculation of possibility of conflicts, then we can say the (digitally supported) tool is enabling us to move towards minimization of conflicts. Furthermore, if this progress in work is automatically recorded, its effects are visually observable by the users involved and can be discussed "now", instead of in the next meeting. This would represent a huge improvement in handling of requirements conflicts. One interesting possibility would be to have 3D jigsaw puzzles to enable visualization of a bigger number of relationships among requirements, and possibly with several levels of abstraction. Some participants wished to retain the physical pieces: they did not want to lose the possibility to touch the pieces physically, neither the initial challenge of putting the pieces together physically and realize how they are physically linked. These participants accepted to work with the digital jigsaw puzzle, but they would like to have also the physical form.

F. Threats to Validity

The evaluation used was qualitative and "perhaps more" constructivist than positivist [12]. We deemed applicable the following strategies [7] for improving validity of constructivist research: 1) clarify bias, 2) rich, detailed descriptions and 3) report discrepant information. Concerning the clarification of bias (1), the following situations may have introduced bias by the researchers: a) in the experiment preparation, when formatting the text in the pieces, researchers may have assumed implicitly one of the interpretations for the grammatical structure of a phrase that has syntactic ambiguity. This was mitigated by the use of heuristics described in Section III.A.2. In the future the use of ambiguity detection tools can help to format the requirements text with minimum bias; b) during the experiment, as the investigators also acted as facilitators, there was the possibility, that they conducted the participants to the solution; c) when reporting and analyzing the results. Concerning the report of the experiments' results we sought to produce detailed descriptions (2) as well as to report discrepant information, should it be the case (3). Strategies 2) and 3) were aimed to reduce researchers' bias (1c).

With regard to construct validity (positivist view) we already acknowledged that the best methodology to evaluate this approach would have been a case study, but this was not possible. Anyhow the experiments performed as emulations of real meetings between stakeholders and engineers provided a reasonable approximation of the context of a real meeting, and it is plausible that results obtained would be confirmed in real meetings. We acknowledge that what was observed and described concerning the ability of the proposed approach to foster team work in groups having elements with different backgrounds, improving both cooperation, and co-responsibility, provides a conclusion with limitations. In fact as the evaluation was not done with real stakeholders the participants did not have a priori any reason to create an undesirable environment. In the 3rd experiment the quality of the real meeting emulation was improved through the introduction of participants with management background.

Concerning internal validity due to confounding factors (positivist view) [12], familiarity and learning were avoided using, for each session, two different systems. The system presented with the jigsaw puzzle was different from the one presented in text. Still concerning confounding factors, after the 2nd experiment, the hypothesis was raised that participants were not given the same information (treatment) while working with the jigsaw puzzle, compared to working with the textual documentation. The difference was that the information about conflicts was provided (as jigsaw puzzle cues) when the requirements were presented through the jigsaw puzzle, but was not provided in the textual documentation. This difference was addressed in the 3rd experiment, where conflict information was provided as part of the textual presentation as well. Another possible confounding factor detected when analysing the 2nd experiment, was the tiredness effect in the participants. In the 2nd experiment the 1st system was always presented with the jigsaw puzzle game, and the 2nd using the textual representation. The participants' preference for analyzing conflicts in requirements, with textual versus jigsaw puzzle game, could very well be biased in favor of the jigsaw puzzle, just because when participants worked with the text presentation (always in the second half of a two hour session) they were already too tired. To deal with this in the 3rd experiment, in one session the text description was utilized first and vice versa in the other session. One other possible confounding factor is that the examples worked might not be comparable, and in particular the HW system could be considered simpler than the CMS, i.e., having requirements easier to examine. It is difficult to ensure that the requirements documentation for two systems are "similar" in "handling difficulty".

VI. Discussion and Future Work

Our user evaluation provides two key insights with regards to the state-of-the-art in stakeholder consultation meetings:

Firstly, the user experiments offer insights into the relevance and richness of the jigsaw puzzle metaphor as an adequate communication means to discuss requirements with stakeholders. This communication mechanism is adequate and relevant because it makes the conflicts in requirements explicit through an easily understandable language. These characteristics together with the gaming nature of this language improve co-ownership, co-responsibility, and creativity. This insight is strongly supported by the improved effectiveness in communication and handling of requirements conflicts, as well by the reactions of the users who easily engaged in team work with a co-responsibility and creative attitude.

Secondly, the search for such a communication mechanism, to make requirements conflicts explicit, brought into the conclusion that it is crucial to separate the processing of the information about the conflict from the issue of communicating that conflict. This separation is essential to permit a user-centered design of the communication of conflicts in RE.

Concerning future work, the possibility to have the jigsaw puzzle pieces as a digital interactive real-time mechanism was keenly advocated by the participants. We do believe it would bring enormous potential to the jigsaw puzzle game. However, we are equally mindful that the tactile, physical nature of the pieces was also highlighted a very positive aspect of the game. Any digital variant would need to preserve this tactile sensory experience. Certainly future work should include more experiments: with real stakeholders and software engineers in a system under development.

VII. Conclusion

We propose a jigsaw puzzle game and a method to effectively address the challenge of presenting the requirements and their conflicts to stakeholders, in a way that fosters co-responsibility and co-ownership regarding the conflicts and their resolution. The jigsaw puzzle metaphor is a novel contribution in the area of visual metaphors, and in particular as a communication mechanism to make conflicts explicit during stakeholder consultation meetings. The novelty of this metaphor resides in: 1) being an easy to understand language for users from non-computing backgrounds; 2) providing easy to understand analogies with well-understood concepts such as building a system out of pieces, and misshapen pieces representing that the system cannot be built with such faulty pieces; and 3) its gaming nature.

The evaluation performed demonstrates that the jigsaw puzzle game does increase effectiveness of communication and handling of conflicts in requirements, when compared with textual presentation of requirements. It fosters team work and communication, and improves commitment of stakeholders in co-authoring of requirements and co-responsibility in resolution of conflict. As the proposed solution builds upon a game (the jigsaw puzzle), it promotes a relaxed environment and thus creativity.

The separation of the processing of the information about the conflicts from the issue of communicating those conflicts is a core design principle of the jigsaw puzzle game. Such a separation is crucial to permit a user-centered design of communication of conflicts in RE and can act as a stepping stone for future research in this regard.

References

[1] D. Ayed, T. Genssler, Dynamic Variability in complex, Adaptive systems, Deliv. D6.1 DiVA EC project, 2009.

[2] M. Balzer, A. Noack, O. Deussen, and C. Lewerentz, "Software landscapes: Visualizing the structure of large software systems", Symposium on Visualization, pp. 261–266, 2004.

[3] D.M. Berry, E. Kamsties, and M.M. Krieger, From contract drafting to software specification: Linguistic sources of ambiguity, https://cs.uwaterloo.ca/~dberry/handbook/ambiguityHandbook.pdf, November, 2003.

[4] S. Boccuzzo, and H.C. Gall, "Cocoviz: Towards cognitive software visualization", IEEE Int. Workshop on Visualizing Soft. for Understanding and Analysis, 2007.

[5] S. Bresciani, and M. Eppler, "The risks of visualization. A classification of disadvantages associated with graphic representations of information", ICA Working Paper #1/2008, University of Lugano (USI), 2008.

[6] L. Chung, B.A. Nixon, E. Yu, and J. Mylopoulos, Non-Functional Requirements in Software Engineering, Kluwer Academic Publishers, 2000.

[7] J.W. Creswell, Research design: Qualitative, quantitative and mixed methods approaches, 2nd Edition, Sage, 2002.

[8] L.M. Cysneiros, E. Yu, and J.C.S.P. Leite, "Cataloguing non functional requirements as softgoals networks", Workshop on RE for Adaptable Architectures, 11th IEEE Int. Conf. Requirements Engineering, pp. 13-20, 2003.

[9] J.S. Davis, "Identification of errors in software requirements through use of automated requirements tools", Information and Software Technology, 31(9), pp. 472-476, 1989.

[10] S. Diehl, Software Visualization, Springer, 2007.

[11] S. Easterbrook, and B. Nuseibeh, "Using ViewPoints for inconsistency management", Software Engineering Journal, 11, pp. 31-43, BCS/IEE Press, 1996.

[12] S.M. Easterbrook, J. Singer, M. Storey, and D. Damian, "Selecting empirical methods for software engineering research", in Guide to Advanced Empirical Software Engineering, F. Shull, and J. Singer, Eds. Springer, 2007.

[13] A. Finkelstein, J. Kramer, B. Nuseibeh, L. Finkelstein, and M. Goedicke, "Viewpoints: A framework for integrating multiple perspectives in system development", Int. Journal of Soft. Eng. and Knowledge Eng., 2(1), pp. 31-58, 1992.

[14] O.C.Z. Gotel, F.T. Marchese, and S.J. Morris, "The potential synergy between information visualization and software engineering visualization", 12th International Conference on Information Visualization, 2008.

[15] IEEE-SA Standards Board, IEEE Recommended Practice for Soft. Req. Spec., IEEE Std. 830-1998, IEEE, 1998.

[16] International Standard Organization, ISO/IEC 25010:2011: Systems and software engineering -- Systems and software Quality Requirements and Evaluation (SQuaRE) -- System and software quality models, ISO, 2011.

[17] J. Kienzle, N. Guelfi, and S. Mustafiz, Crisis Management Systems: A case study for Aspect-Oriented Modeling, http://www.cs.mcgill.ca/~joerg/taosd/TAOSD/TAOSD.html, May, 2009.

[18] E. Knauss, K. Schneider, and K. Stapel, "A game for taking requirements engineering more seriously", 3rd Int. Workshop on Multimedia and Enjoyable Req. Eng., 2008.

[19] J. Lee, J. Kuo, N. Hsueh, and Y. Fanjiang, "Trade-off requirement engineering", in Soft. Eng. with Comp. Intelligence, J. Lee, Ed. Springer, pp. 51-72, 2003.

[20] N. Maiden, A. Gizikis, and S. Robertson, "Provoking creativity: imagine what your requirements could be like", IEEE Software, 21(5), pp. 68-75, 2004.

[21] N. Maiden, and S. Robertson, "Integrating creativity into requirements processes: experiences with an air traffic management system", 13th IEEE Int. Conf. RE, 2005.

[22] N. Maiden, C. Ncube, and S. Robertson, "Can requirements be creative? Experiences with an enhanced air space management system", Proc. ICSE, 2007.

[23] B. Meyer, "On formalism in specifications", IEEE Software, 2(1), pp. 6-26, 1985.

[24] L. Mich, M. Franch, and P.N. Inverardi, "Market research for requirements analysis using linguistic tools", Requirements Engineering, 9(1), pp. 40-56, 2004.

[25] L. Mich, C. Anesi, and D.M. Berry, "Applying a pragmatics-based creativity-fostering technique to requirements elicitation", Requirements Engineering, 10(4), pp. 262-275, 2005.

[26] N. Millard, P. Lynch, and K. Tracey, K.; "Child's play: using techniques developed to elicit requirements from children with adults", 3rd Int. Conf. RE, pp. 66-73, 1998.

[27] J. Noppen, J., Imperfect Information in Software Design Processes, PhD Thesis, Enschede, The Netherlands, 2007.

[28] B. Nuseibeh, A Multiple Perspective-Framework for Method Integration, PhD Thesis, Department of Computing, Imperial College, London, UK, 1994.

[29] M. Pinto-Albuquerque, Communicating Conflict and Ambiguity in Requirements Engineering, PhD Thesis, Lancaster University, UK, 2013.

[30] A. Sardinha, R. Chitchyan, N. Weston, P. Greenwood, and A. Rashid, "EA-Analyzer: automating conflict detection in aspect-oriented requirements", 24th IEEE/ACM Int. Conference on Automated Software Engineering, 2009.

[31] R. Smith, O. Gotel, "Gameplay to introduce and reinforce requirements engineering practices", 16th IEEE Int. Conf. on Requirements Engineering, pp. 95-104, 2008.

[32] S. Soares, P. Borba, and E. Laureano, "Distribution and persistence as aspects", Software: Practice and Experience, 36 (7):711–759, 2006.

[33] I. Sommerville, and P. Sawyer, Requirements Engineering: A Good Practice Guide, Wiley, 1997.

[34] I. Sommerville, P. and Sawyer, "Viewpoints: principles, problems and a practical approach to requirements engineering", Annals of Soft. Eng., 3, pp. 101-130, 1997.

[35] B. Tversky, J.B. Morrison, and M. Betrancourt, "Animation: can it facilitate?", International Journal of Human Computer Systems, 57, pp. 247-262, 2002.

[36] R. Wettel, and M. Lanza, "Program comprehension through software habitability", 15th Int. Conference on Program Comprehension, pp. 231 - 240, 2007.

[37] H. Yang, A. de Roeck, V. Gervasi, A. Willis, and B. Nuseibeh, "Analysing anaphoric ambiguity in natural language requirements", Requirements Engineering, 16(3), pp. 163-189, 2011.