

Adding Semantic Relations among Design Patterns

Marcos Alexandre Rose Silva and Junia Coutinho Anacleto

Department of Computing, Federal University of São Carlos, Washigton Luis km 235, São Carlos, São Paulo, Brazil

Keywords: Design Patterns, Design Pattern Formalism, Pattern Language, Semantic Relations, Pattern Relationships.

Abstract: Design patterns have been used to support design decisions to solve recurring design problems adopting the successful solutions stated in design patterns. One of the main characteristics of design patterns is to allow the patterns' content understanding because they are written using a common language, i.e., not specialized, and they bring examples to support the comprehension of the solutions. On the other hand, to understand the correlation among these design patterns, usually organized through nodes and edges as in a graph, is not a simple task. In this context, this paper presents a semantic approach, based on how humans organize their knowledge, to connect design patterns and define those relationships according to our intellectual structure and function. A feasibility study, described here, shows evidences that semantic relations allow organizing patterns to support the comprehension of patterns connections, as well as, the name of these relations are able to express their meaning.

1 INTRODUCTION

According to Borches (2001) design patterns contain the essence of successful solutions to recurring design problems in a certain context. The concept of design patterns came from an architect called Alexander who started observing and formalizing problems and their successful solutions (Alexander et al., 1977). Alexander et al., (1977) decided to formalize and register them in order to share their knowledge and experience with their colleagues, as well as, to communicate with their clients. That's why they used a common language and examples. These knowledge and experience registered through problems and solutions were named patterns.

Patterns were well accepted by architect communities because they noticed patterns as useful way to share knowledge and to describe successful solutions, i.e., solutions applied many times with satisfactory results (Alexander et al., 1977). According Michael et al., (1998), patterns started been used in Software Engineering (SE) area, and then Human-Computer Interaction (HCI).

HCI researchers and professionals started formalizing and writing patterns considering their knowledge, experience and observation about design of interfaces (Michael et al., 1998). Figure 1 illustrates a part of HCI design pattern written by Montero, who has experience on web sites design (Montero et al., 2002).

Each pattern has some required information as name of patterns – main idea of pattern; problem – specific problem that the pattern is meant to solve solution – main message of pattern, it presents the solution to the problem; and some optional information as examples – images, schemes, etc., to illustrate the solution and facilitate its understanding (Fincher et al., 2003).


<p>Name: Polyglot</p> <p>Context: The power of the Web is in its universality.</p> <p>Problem: How can the user do a useful use of the Web site and access information at your own pace?</p> <p>Forces: The user wants easy access to information; The user has little or no incentive to spend time learning technical details.</p> <p>Solution: Speak user's language is "design for all". Kids, older or disabled people can visit our Web site and universal design techniques can be applied in the design of Web site and his services. Information should be provided of a suitable manner by considering several kinds of peoples and technical features and by using Polite language.</p> <p>Examples:</p> 
--

Figure 1: Part of Montero's pattern (Montero et al., 2002).

Figure 1 represents one from the twenty-three patterns formalized by Montero. These patterns are organized through nodes, with names of patterns, and edges to represent the relationships to each other, as in a graph. When patterns are weakly related to each other, e.g., through categories, is considered a pattern collection. When there is a well structured collection of patterns that rely to each other, as Montero's patterns, is considered a pattern language (Coplien, 1998).

It is possible to scan the graph and find the pattern, which best describes the overall scope of the project or the problem that needs to be solved, i.e., design decisions are made by selecting and instantiating appropriate patterns, and composing them together (Coplien, 1998).

In contrast, it is not a trivial task to understand these relationships among patterns because edges indicate relationships among patterns but they do not represent the meaning of these relationships. In this context, this paper presents semantic relations to organize patterns and define relationships according human intellectual structure and function. Our hypothesis is - organizing patterns as human intellectual structure and inserting semantic relations can support the comprehension of pattern connections.

2 RELATED WORK

Kruschitz et al., (2010) investigated 21 pattern languages and collections to analyze the design patterns relationships variety. They describe that there is no consensus on how patterns should be organized and categorized in order to provide appropriate information to produce good design. They also discuss that the most of authors are using the Alexandrian form, nodes and edges, possibly because it was the first form used to encapsulate design knowledge. Because of that, pattern authors have transferred the Alexandrian pattern structure to software engineering.

According to Kruschitz et al., (2010) some authors are using *association*, *aggregation* and *specialization*, from software engineering concepts, to define relationships among patterns. *Association* – pattern x has an unspecific connection between pattern y; *Aggregation* – pattern x is frequently used together with pattern y; *Specialization* – pattern x add more attributes to pattern y, i.e., a pattern inherits the attributes from another pattern, as well as, add new ones to fulfill the purpose of pattern. These authors also pointed out another relationship

called *anti-association*, which means that pattern x and a pattern y must not be used together.

Considering software engineering context, Gamma et al., (1994) defined the relationships among design patterns considering object-oriented concepts. Therefore, there is a classification of design patterns according two criteria: jurisdiction (class, object, compound) and characterization (creational, structural, behavioral). Because of that, there are relationships called *is implemented using* – pattern x is implemented using pattern y; *similar in constructing object structures* – pattern x is similar in constructing object structures as pattern y; *often builds a object* – pattern x often builds a pattern y object, among others.

In order to improve the comprehensibility of the relationships defined by Gamma, Zimmer (1995) proposed a classification of the relationships which helps in understanding the similarities among the relationships. He defined three types or classifications: *uses in its solutions* – pattern x uses the design pattern in its solution. Thus, the solution of pattern y represents one part of the solution of pattern x; *is similar to* – pattern x and pattern y address a similar kind of problem but not a similar kind of solution; *can be combined with* – patterns x and y are somehow similar, but it is difficult to state it more precisely.

Conte et al., (2002) also defined relationships considering software engineering context. For example, they used stereotypes for use cases, as Uses and Extend, existing at UML (Unified Modeling Language) to define relationships among patterns. *Uses* – pattern x uses a pattern y; *Refine* – pattern x refines pattern y, i.e., one must be a specialization of another; *Requires* – pattern x is required in pattern y; *Alternative* – pattern x is an alternative of a pattern y, i.e., they have the same context and problem but not the same solution.

Girardi et al., (2006) described about OntoPattern, an ontology that represents knowledge about how patterns are described and about the relationships defined by Conte et al., (2002). Considering UML context, Bottoni et al., (2010) use the formalizing of the UML diagrams to organize patterns, for example, they use sequence diagram and structure diagram.

Taking into consideration architecture context, Kumar et al., (2010) applied architecture level techniques at pattern level to derive the DDTM (Design Decision Topology Model) of a pattern. According to these authors, this representation enriches pattern descriptions and helps to analyze quality requirement traceability, as well as

relationships amongst patterns. The relationships defined are: *Is-Duplicate-of* – patterns x and y provide same solution to same problem; *Is-an-Alternative-to* – patterns x and y solve the same problem, but propose different choices; *Comprises* – pattern x uses the pattern y in its solution; *Refines* – patterns x and y address same problem but pattern x provides more refined (with less consequences) solution than y.

Fincher et al., (2003) discuss a possibility to identify which patterns from various authors could refer to patterns in other pattern languages or collections. According to authors, they defined three pre-defined link types to reflect the common ways patterns are structured. These types were inserted in Pattern Language Markup Language (PLML) specification, which is a language that explains how to formalize/write patterns, describing also required information to express the knowledge and experience. The types are: *is-a* – a pattern is the same as, or is an alternative solution to the same problem; *is-contained-by* – a pattern is “smaller” and is used (with others) to instantiate a larger one and; *contains* – the reciprocal of is-contained-by.

Kruschitz (2009) discusses about a framework called XPLML (eXtended Pattern Language Markup Language) to support patterns formalization including these types of relationships among patterns. On the other hand, Janeiro et al., (2010) describe that these three types are not enough to describe more precisely the relationships among design patterns. Because of that, they presented more five types as an extension to these existing types. They defined these types analyzing patterns descriptions and their references to other patterns. The types are: *Used With* – pattern x is frequently used together with pattern y, but they are not hierarchically related; *Similarity* – pattern x has some characteristics similar to the pattern y, i.e., one can be used as an alternative to another; *Realization* – pattern x implements the concepts described by a pattern y; *Enhancement* – pattern x builds upon an pattern y, enhancing its functionalities; *Conflict* – pattern x and a pattern y must not be used together.

According to Fricke et al., (2000) there is not a temporal distinction among patterns' relationships, i.e., there is no information about what patterns should be used firstly or what patterns should be used together. Because of that, they defined a hierarchical organization inserting colors on lines (edges). Red line means that a pattern must be used before another; Black line means that a pattern is a specialization of another and; when there are two or more patterns at the same rectangle means that they

share a common context and they must be used together. According to authors, through colors and direction of the arrow is possible to know the sequence and how the patterns should be used.

White (2012) describes how to visualize design patterns relationships at mobiles. Different lines (edges) show the relationship, solid lines indicate that a related pattern is required, while dashed lines indicate that the related pattern is optional. In the beginning, it is not displayed all patterns at the same time. The main patterns are displayed and others related patterns are displayed considering interaction. For example, when there is a click on a pattern, others connected patterns are displayed with solid or dashed lines.

Investigating the related work, it is possible to identify that these types of relationships can be appropriated to define the relations among patterns from different pattern languages or collections. On the other hand, they are not appropriated to define the relations among patterns from the same pattern languages or collections, as well as, some of them do not have a natural semantic for people, e.g. clients, with no programming and/or software engineering knowledge.

For example, relationships as *association*, *aggregation* and *specialization* come from software engineering concepts. Because of that, they are not understandable for all people (Kruschitz et al., 2010). It is not supposed to people who read patterns know these concepts and others related to software development process (Borchers, 2001; Kruschitz et al., 2010; Welie et al., 2012). It important to highlight that one of the main characteristics of design patterns is to support communication among people at software development process.

Others relationships as *is similar to*, *Alternative*, *Is-an-Alternative-to*, *Is-Duplicate-of*, *is-a*, etc., which mean that patterns solve the same problem with same solution, etc., are not useful for patterns at the same pattern language, because a pattern language is a structured collection of patterns that rely to each other; then, there are not equals patterns and all of them *can be combined with* (*Used With*, *Uses*) each other with no *Conflict* (Coplien et al., 1998).

In this context, this paper presents semantic relations defined by Minsky to be used among patterns in order to connect them at the same way that humans organize their knowledge, as well as, using familiar semantic to be comprehensible for all people.

3 MINSKY’S RELATIONS

Marvin Minsky is a researcher at artificial intelligence, cognitive psychology, computational linguistics, among others areas (Minsky, 1987). He has worked chiefly on imparting to machines the human capacity for commonsense reasoning.

Commonsense is a common knowledge shared by nearly people. This knowledge comes from our social interactions, observations, behaviors, belief, culture, etc., i.e., it is not necessarily scientific knowledge but it can be. For example, some children believe in Santa Claus and they know how to describe him, telling that he is an elder with white beard, red clothes, as well as, he always comes at Christmas to give gifts, etc. Other people believe that some teas help to cure some health diseases and that the Earth revolves around the Sun, etc (Minsky, 1987; Carvalho et al., 2008; Liu et al., 2004).

Marvin Minsky has investigated human intellectual structure and function in order to know how to store this knowledge at computer considering the way that people’s brains do. He intends to make intelligent machines and explore new interface designs with that knowledge (Minsky, 1987; Liu et al., 2004). There are some projects using this investigation to collect and use human commonsense at computer (Carvalho et al., 2008; Liu et al., 2004; Minsky, 1987).

In this paper our focus is not to give intelligence to machine, etc., but it is to investigate and use human intellectual structure and function to organize patterns in order to allow people notice and understand the relationships among them.

Marvin Minsky has defined twenty semantic relations to store and organize concepts or knowledge in a close way to human cognitive structure (Liu et al., 2004). Figure 2 shows a network with concepts related to Santa Claus connected by Minsky’s semantic relations.

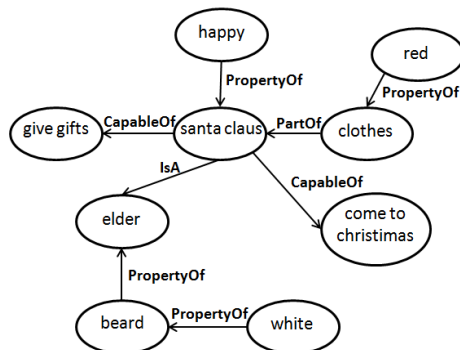


Figure 2: Concepts related to Santa Claus connected by Minsky’s semantic relations.

Considering Figure 2, it is possible to observe that Santa Claus *IsA* elder, *CapableOf* give gifts, one *PartOf* him is his clothes, red can be considered a *PropertyOf* his clothes, etc. There are two main characteristics in these semantic relations: firstly, as described previously, they allow organizing concepts as human cognitive structure and; secondly they were defined using a familiar semantic, i.e., it is not necessary scientific knowledge, as software engineering, etc., to understand the meaning of *IsA*, *PartOf*, *CapableOf*, etc (Liu et al., 2004).

Figure 3 shows part of Minsky’s semantic relations grouped into various thematics. For example, *IsA*, *PropertyOf*, etc., are grouped as Things, because these relations are used to store and organized concepts related to things. *UsedFor* and *CapableOfReceivingAction* are grouped as Functional, etc.

Thematics	Relationships
K-Lines	ConceptuallyRelatedTo
	ThematicKLine
	SuperThematicKLine
Things	IsA
	PropertyOf
	PartOf
	MadeOf
Agents	DefinedAs
	CapableOf
Spatial	LocationOf
Causal	EffectOf
	DesirousEffectOf
Functional	UsedFor
	CapableOfReceivingAction
Affective	MotivationOf
	DesireOf

Figure 3: Minskys’semantic relations (Liu et al., 2004).

These relations had investigated and applied at one pattern language and one pattern collection. The pattern language supports designing of web sites (Montero et al., 2002) and the pattern collection, called Co-authoring patterns, support designing of web educational systems (Anacleto et al., 2013). These patterns were chosen because we have been used Montero’s patterns since 2007 and we had formalized the collection. The results of these investigations and applications are described at the next section.

4 SEMANTIC RELATIONSHIPS FOR PATTERNS

One of the thematics, Figure 3, was not applied. K-lines represent relations to be used when the connection between concepts is not clear. These generic relations are not useful for patterns at the same language and collections, because it is supposed that these patterns are connected, and then

it is necessary to specify clearly these relationships.

Figure 4 illustrates the Co-authoring patterns collection and their relationships considering Minsky’s semantic relations, and after there is Table 1 with explanations about each relation presented.

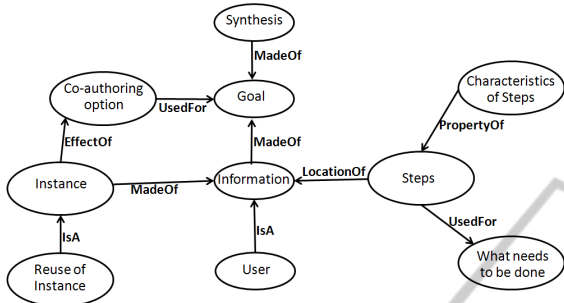


Figure 4: Patterns connected by semantic relations.

Table 1: Minsky’s semantic relations to connect patterns.

Relationships	Explanations and Examples
IsA	<p>pattern x specialize pattern y; they are hierarchically related. Natural Example: Santa Claus IsA Elder Pattern Example: User IsA Information</p> <p>Observation: IsA defined by Minsky and is-a described by Fincher are different because IsA does not mean the same problem and solution. DefinedAs relation described by Minsky has this specific meaning (equal, synonym), but it is not applied at patterns from the same language/collection as described at Section 2.</p>
PropertyOf	<p>pattern x contains properties/characteristics of pattern y. Natural Example: red PropertyOf clothes Pattern Example: Elements PropertyOf Steps</p> <p>Observation: PropertyOf means composition (at software engineering concepts) because there is no sense to apply just a property. For example, it is necessary to know about clothes to identify where to apply the red. This property (red) must be applied with clothes. On the other hand, clothes can be applied without red.</p>

Table 1: Minsky’s semantic relations to connect patterns. (Cont.)

PartOf	<p>pattern x can be determined as part of pattern y. Natural Example: clothes PartOf Santa Claus Pattern Example: Search PartOf HomePage (patterns from Montero)</p> <p>Observation: Each part can have its own properties, etc., i.e., PartOf means aggregation (at software engineering concepts) because there is sense to apply one pattern without another. For example, It is possible to apply clothes without Santa Claus, as well as, Santa Claus without clothes.</p>
MadeOf	<p>pattern x is a subtype of pattern y. pattern x is a product and pattern y is a substance. In other words, pattern x is obtained through a processing from pattern y. Natural Example: bacon MadeOf pork Pattern Example: Synthesis MadeOf Goal (in this case, Synthesis represents a tip of the Goal).</p>
LocationOf	<p>pattern x represents the location of pattern y. Natural Example: in war LocationOf arm Pattern Example: Steps LocationOf Information</p>
EffectOf	<p>pattern x represents a consequence of an action or an event of pattern y. Natural Example: entertainment EffectOf view video Pattern Example: Coauthoring option EffectOf Information (in this case, Coauthoring option allows the insertion of Information).</p>
UsedFor	<p>pattern x specifies a function of pattern y. Natural Example: fireplace UsedFor burn wood Pattern Example: Coauthoring option UsedFor Goal (in this case, Goal contains the explanation about the use of Coauthoring Option).</p>

5 FEASIBILITY STUDY

A feasibility study was done in order to observe the comprehension of relationships among patterns

through three different kinds: 1) Alexandrian form considering just nodes and edges; 2) Conte et al.' Relations; 3) Minsky's semantic relations. The second way was chosen because two of the relations defined by Conte et al., (2002) represent relationships among patterns that complement each other with no conflict. For instance, there are *Uses* and *Requires* to make clear when a pattern can be used or required by another.

At feasibility study, it was necessary to design low-fidelity web educational system prototypes, i.e., design interfaces of systems on papers, considering web pattern language formalized by Montero et al., (2002); or Co-authoring patterns collection formalized by Anacleto et al., (2013) for web educational systems designing.

At feasibility study, there had been 20 participants who attended an optional discipline at university about Human-Computer Interaction (HCI) concepts to design web computer systems and, 5 undergraduates from Pedagogy or Mathematic who accepted the invitation to attend the discipline to support designing low-fidelity web educational system prototypes.

5.1 First Step

Firstly, there was a brief explanation about the feasibility study;

Secondly, the 20 participants filled a pre-questionnaire considering their experience and knowledge about Software Engineering (SE), HCI and their practical experience about design computer systems, etc., and the 5 participants filled a pre-questionnaire considering their uses of computer and internet, as well as their experience using computer on teaching.

Thirdly, the participants were introduced to some HCI concepts as design system, prototype and scenario. This last concept was taught because participants needed to design prototypes considering scenarios that described educational activities, reported by teachers.

During these explanations, the answers of the questionnaires were analyzed in order to divide the participants in homogenous groups. It was interesting to observe that the 20 participants were electrical engineering undergraduates with no knowledge related to ES or HCI concepts but they were interested about designing systems. Everybody had attended one discipline at computer area, Data Base and, some of them had a little experience with web sites development. It was considered as an opportunity to observe how participants from others

areas understand pattern connections.

Five groups were created with 5 participants. In each group, there were: one undergraduate from pedagogy or mathematic, all of them is familiar with computer and internet and, they had used computer games at classroom and they think that "educational games are useful tools"; one undergraduate with web sites development experience and; others.

Fourthly, groups could discuss the concepts introduced and one scenario randomly selected for each group.

5.2 Second Step

Firstly, the groups were introduced to prototype and design patterns concepts. It was explained about pattern language and collection but with no details about the connections among patterns, i.e., with no explanations related to Alexandrian form, Conte et al.' Relations and Minsky's relations.

Secondly, each group accessed a randomly selected pattern language with one kind of relationship. One group with Montero's Patterns

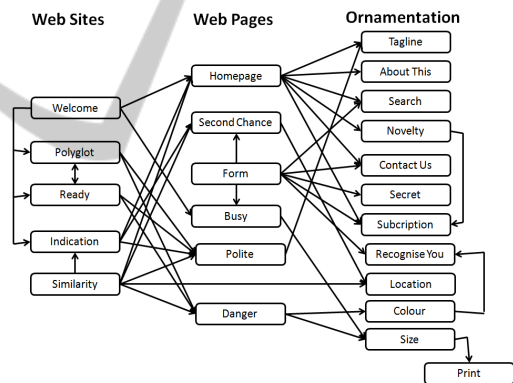


Figure 5: Montero's Patterns with Alexandrian Form (MPAF).

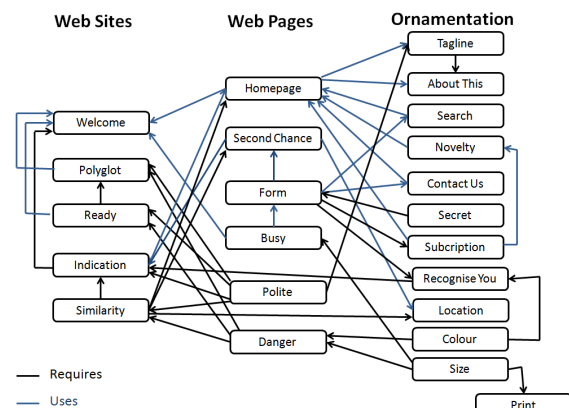


Figure 6: Montero's Patterns with Conte's Relations (MPCR).

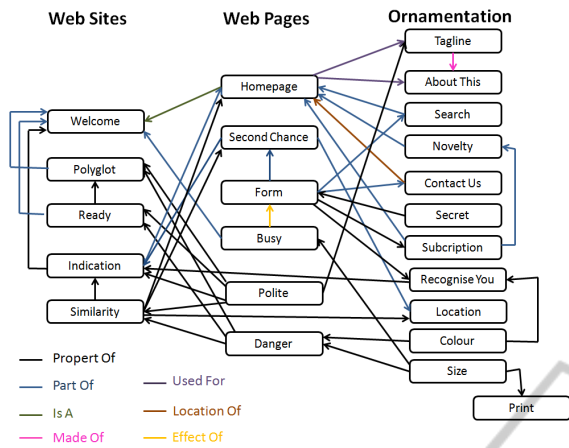


Figure 7: Montero’s Patterns with Minsky’s semantic Relations (MPMR).

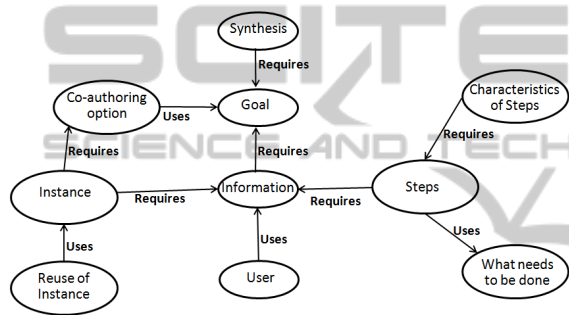


Figure 8: Co-authoring Patterns with Conte’s Relations (CPCR).

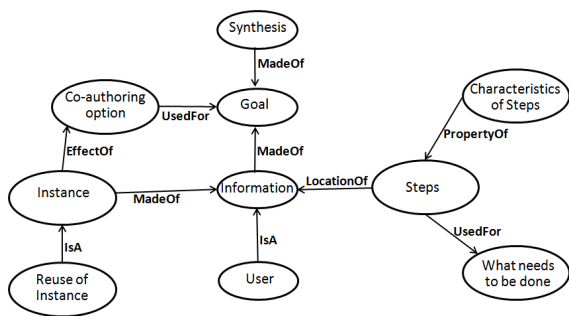


Figure 9: Co-authoring Patterns with Minsky’s semantic Relations (CPMR).

with Alexandrian Form (MPAF), Figure 5; another with Montero’s Patterns with Conte’s Relations (MPCR), Figure 6; another with Montero’s Patterns with Minsky’s semantic Relations (MPMR), Figure 7; another with Co-authoring Patterns with Conte’s Relations (CPCR), Figure 8 and; the last group with Co-authoring Patterns with Minsky’s semantic Relations (CPMR), Figure 9.

Thirdly, each participant of the groups needed to analyse the graph of web patterns or Co-authoring

patterns to answer a question: “Explain what you understand when you see the drawing with patterns”. We did not use words with graph, connections, relations, etc., to avoid any influence at answers.

Answers of this question were analyzed considering Content Analysis methodology to make the categorization, description and interpretation in order to identify the frequency of occurrence of certain terms to observe which information in the graph is used to understand it (Moraes, 1999).

The terms more cited by participants were relations among patterns, direction of the arrows, name of the patterns and the three categories defined by Montero as Web Sites, Web Pages and Ornamentation. For example, one participant, who accessed MPMR, answered “Looking only the direction of the arrows I can understand that a pattern with arrow out is connected to a pattern with arrow in. The tags are very useful and it is possible to understand how the patterns are connected”. In this case, tags were interpreted as Minsky’s semantic relations.

In this answer, the participant wrote about direction of the arrows and relations, and then they were counted at Table 2.

Table 2: Quantity of participants who described some terms in their answers.

Groups	Direction of the arrows	Name of the patterns	Relations among patterns	Montero’s categories
MPMR	2	2	4	1
MPCR	2	1	3	1
MPAF	4	0	no	2
CPMR	1	2	3	no
CPCR	2	3	3	no

Fourthly, each participant of the groups needed to answers some questions considering the graphs, as MPMR, MPCR, etc. They could answer them just browsing the graph and/or clicking on each pattern to read its content.

The questions were created in order to observe if participants could identify each Minsky’s semantic relation to be considered to answer them, as well as, if Conte’s Relations and Alexandrian Form could support identifying the answers. There were questions related to three aspects:

- (1) Interpretation to observe if participants could notice the meaning of the relations, because each question was created considering a Minsky’s relation. For example, questions about where a pattern needs to be, the expected answer is another pattern connected with it through *LocationOf* relation.

(2) Sequence of Use to observe if participants could notice the meaning of the relations and direction of the narrows to identify what pattern should be used firstly.

(3) Obligation to observe if participants could notice the meaning of the relations to identify what patterns should be used together.

Table 3 presents the questions related to Co-authoring patterns with expected answers, as well as, each Minky’s relation to be considered. Table 4 presents this information as well, but related to Montero’s patterns.

Table 3: Questions for Co-authoring Patterns.

Interpretation
Q1 – Which pattern describes where pattern “User” needs to be? Expected Answer: “Steps” – Relation: LocationOf – “User” IsA “Information” : “Information” LocationOf “Steps”.
Q2 – Which pattern describes explanations about the utility of pattern “Steps”? Expected Answer: “What needs to be done” – Relation: “UsedFor”.
Q3 – Which pattern describes what pattern “Information” contains? Expected Answer: “Goal” – Relation: “MadeOf”.
Q4 – Explain the relationship that you understand between patterns “User” and “Information”. Expected Answer: “User” IsA “Information”, “User” is a specialization of “Information” or, “Information” is a generalization of “User” – Relation: “IsA”.
Sequence of Use
Q5 – Which pattern should be used before pattern “Information”? Expected Answer: “Goal” – Sequence of relation “MadeOf”.
Q6 - Considering patterns “Information” and “Goal”. Which pattern must be considered firstly? Expected Answer: “Goal” - Sequence of relation “MadeOf”. “Information” is MadeOf “Goal”, then it needs to be considered before.
Obligation
Q7 – There was no question because Co-authoring patterns do not contain “PartOf” relation.

Figure 10 shows the quantity of answers, from each group, that matched with expected answers. X means no question for the group, for example, there was not Q5 for groups that accessed Montero’s Pattern neither Q7 for groups that accessed Co-authoring patterns.

Table 4: Questions for Montero’s Patterns

Interpretation
Q1 – Which pattern describes where pattern “Contact Us” needs to be? Expected Answer: “Homepage” – Relation: LocationOf.
Q2 – Which pattern describes explanations about the utility of pattern “Homepage”? Expected Answer: “Tagline” and “About this” – Relation: “UsedFor”.
Q3 – Which pattern describes what pattern “Tagline” contains? Expected Answer: “About this” – Relation: “MadeOf”.
Q4 – Explain the relationship that you understand between patterns “Welcome” and “Homepage”. Expected Answer: “Welcome” IsA “Homepage”, “Homepage” is a specialization of “Welcome” or, “Welcome” is a generalization of “Homepage” – Relation: “IsA”.
Sequence of Use
Q5 – There was no question because Montero’s patterns contain just one pair of patterns connected by “MadeOf”.
Q6 - Considering patterns “Tagline” and “About this”. Which pattern must be considered firstly? Expected Answer: “About this” - Sequence of relation “MadeOf”. “Tagline” is MadeOf “About this”, then it needs to be considered before.
Obligation
Q7 – Do pattern “Novelty” must be considered when pattern “Homepage” is used? Expected Answer: No – Relation: “PartOf”.

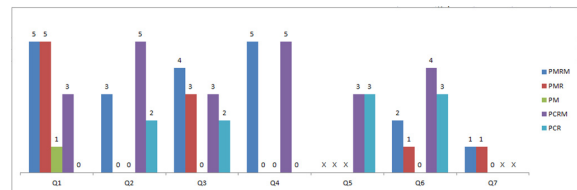


Figure 10: Quantity of expected answers from each group.

6 DISCUSSIONS AND FINAL CONSIDERATIONS

Analyzing participants’ answers about their comprehension of the graphs, shown at Table 2, it is possible to observe that the names of the patterns and arrows among patterns are considered in order to understand them as described as Montero et al., (2012) and Fincher et al., (2003). Name of patterns

were cited by 32% of participants and arrows by 44%, including direction of the arrows.

Considering 25 participants, 52% of them took into consideration the name of the relations among patterns. In contrast, 5 participants or one group did not access pattern with name of the relations, i.e., they accessed just with arrows as Alexandrian Form. Because of that, considering 20 participants, who accessed with Conte's Relations or Minsky's semantic relations, the percentage is 65%.

These results can be considered as evidence that name of the relations are visualized and considered in order to understand the relationships among patterns. Because of that, names on arrows to express the meaning of relationships can be considered as a useful strategy.

Two participants mentioned none of the terms in their answers, e.g., one, who accessed PCR, wrote "It was easy to understand the graph, I could realize the steps that I have to follow"; another, who accessed PMRM, wrote "It is a flowchart that user usually executes to access a site and what s/he hopes to find on it, then this flowchart helps to design interface".

Three participants mentioned their no comprehension of the graph, two who accessed PMR and one who accessed PMRM. For example, one participant from PMR groups wrote "Column Web sites represents the first level, to prepare website. Web Pages and Ornamentation describe more details. Blue arrow represents Uses and its direction means which pattern uses another; the same thing for black arrow that represents Requires, but no legend or explanation about each term makes the comprehension not clear".

Another example from PMRM groups was "It is possible to understand how patterns are connected to each other, but some relations are not clear as User is a Information."

Others participants mentioned what they were seeing, for example, one who accessed PMRM wrote "patterns are divided into three groups and I follow the direction of the arrows to read the legend, e.g., Size is a property of Print and Busy is an effect of Form".

Analyzing Table 2, it is possible to observe that the term Relations among patterns was more cited than others terms in almost all Groups, just CPCR group cited Name of the patterns as many as Relations and MPAF group, with no relations, considered more Direction of the arrows.

These results can also be considered as evidence that relations are visualized and considered to

understand the relationships among patterns, because it was more cited than others terms.

In this context, the next step was to observe if these relations could be interpreted considering their meaning and intention of the use. Because of that, a questionnaire was available to observe if participants could realize the expected relation to answer the questions and, consequently, to answer as expected answer.

Figure 10 presents the quantity of participants from each group who answered as expected answers, e.g., expected answer was "Goal" pattern and participant wrote "Goal". In this case, others answers as "Goal and User", or "Instance" were not considered. It is important to say that there was an "Observation" field after each question to allow participants write anything about their answers.

In general, Alexandrian Form provided less understanding than Conte's Relations or Minsky's semantic relations. In the most of the questions, participants, who accessed Alexandrian Form, needed to click on each pattern to read it in order to guess the relationships among them, but the answers did not represent the correct answers. Usually, participants wrote all patterns connected with the pattern described at the question, for example, Q1 – Which pattern describes where pattern "Contact Us" needs to be?, the most of the answers contained all of the patterns connected with "Contact Us".

Conte's Relations supported some interpretations, for example, considering Q1, all participants, who accessed Monteros' patterns, chosen the right pattern connected with "Contact Us" by *Uses*. On the other hand, the pattern where "Contact Us" needs to be was directly connected with the answer and, its name "Homepage" was a help for participants. Two participants expressed at questionnaire that things are on homepage. In this case, Conte's relations could be a help in this reasoning, because nobody from MPAF, with no relations, described this interpretation.

In contrast, nobody chosen the right answer on Co-authoring patterns because, in this case, it was necessary to understand that a pattern, named "User" is another "Information" that was localized at another "Steps" to answer the question. In this case, it was an evidence that it was necessary semantic among patterns to support this comprehension.

Answers for others questions related to Interpretation as Q2 and Q4 also illustrated that Conte's relations do not represent meanings related to explanation about the use as *UsedFor* and generalization as *IsA*. Two participants from CPCR wrote the expected answer at Q2, but they read the

patterns contents. Nobody wrote the expected answer at Q4, participants wrote answers as “User needs Information”, “User accesses Information”, “they are connected”, “Homepage uses Welcome”, etc.

In contrast, 60% from MPCR group and 40% from CPCR wrote the expected answer at Q3. Nobody did any observation about the answers, but two participants from MPCR and one from CPCR read the patterns before answering, then it is not possible to confirm that they interpreted the meaning of composition from Conte’s relations.

Answers for Q5 and Q6 shown that relations supported identify the sequence of use, because 60% of participants from each Conte’s relation group answers as expected and, nobody read the patterns before. On the other hand, possibly, names of patterns helped the groups who accessed Co-authoring patterns because just 20% of participants, who accessed Montero’s patterns, wrote the expected answer.

Answers for Q7 shown that Conte’s relations and Minsky’s relation do not represent when a pattern can be used or must be used by others through relations. *Uses* and *Requires* did not represent “can be” and “must” for all participants. In this question, all participants who accessed considering Alexandrian Form did not write the expected answer. Three participants wrote observation as “I think that one pattern must be used by another when they are connected”. This answer is not right in all cases, because some patterns can be used with others, but it is not necessary to use them every time together (Montero et al., 2002).

It is important to clarify that there is no answer for Q7 at Co-authoring patterns, because *PartOf* relation was not necessary. Then, it was an example that the all Minsky’s Relations are not necessary in all pattern languages or collections, it is possible to choose some of them according the connections among patterns.

Minsky’s Semantic Relations supported more understanding than others. For example, at Q1, Q2, and Q3, the most of participants could notice that *LocationOf* represents which pattern describes where another one needs to be, as well as, *UsedFor* represents when a pattern explains the utility of another and *MadeOf* represents when a pattern is obtained through a processing from another. Two different participants, who answered as expected, read the pattern before answering; one to answer Q2 and another to answer Q3.

Three participants wrote some observations for Q1 describing about relation, e.g., “There is an

arrow with *LocationOf*” or “It was clear identify the dependency between Contact Us and Homepage through *LocationOf*”. About Q2 two participants, who wrote expected answer, reported “In the beginning, it was not easy to understand” – this participant read the pattern Steps before; another “There is an arrow between Steps and What needs to be done with *UsedFor*”, others participants did not write anything.

Answers for Q4 shown that participants wrote what they were seeing, for example, “User is an Information” but it does not mean that they understood the relation. Two participants wrote at Observation field that “It is possible to understand that Welcome is a concept more abstract” and “User is a kind of information to be inserted”. Even with these two answers, it is not possible to confirm that others participants had the same comprehension.

The most of participants answered as expected at Q5 and Q6. Nobody read the patterns before answering. Two different participants wrote observations. One for Q5 “Goal pattern has an arrow pointed at Information” and another for Q6 “the drawing defines a sequence where Information is made of Goal”. In these cases, direction of the arrows were most important than name of relation to identify the sequence of use, because there are four patterns connected with *MadeOf* at Co-authoring patterns.

Finally, these expected answers were evidences that Minsky’s semantic relations allow organizing patterns to support the comprehension of patterns connections, as well as, the name of Minsky’s relations are able to express their meaning.

As future work, we intend to make more studies with other pattern languages and collections, as well as, to available some information, e.g., legend, about the relations in order to observe how the connections among patterns can be understood when the meaning of relations are known before their use, as well as, define Observation field as required, because it supports knowing participant’s compression of each answer of the questionnaire.

ACKNOWLEDGEMENTS

We thank CAPES, FAPESP and BOEING for partial financial support to this research.

REFERENCES

Alexander, C.; Ishikawa, S.; Silverstein, M.; Jacobson, M.;

- Fiksdahl- King, I.; Angel, S. "A Pattern Language: towns, buildings, construction". New York: Oxford University Press, 1977, 1171p.
- Anacleto, J. C. ; Silva, M. A. R. ; Hernandes, E. C. M. . Co-authoring Proto-patterns to Support on Designing Systems to Be Adequate for Users' Diversity. In: International Conference on Enterprise Information Systems, 2013, Angers. Proceedings of the 15th International Conference on Enterprise Information Systems. Portugal: SCITEPRESS Science and Technology Publications, 2013. v. 1. p. 84-89.
- Borchers, J. O. A Pattern Approach to Interaction Design. John Wiley & Sons, Chichester, UK, 2001, 264p.
- Carvalho, A. F. P. DE ; Anacleto, J. C. ; Zemascarenhas, Silvia Helena . Learning Activities on Health Care Supported by Common Sense Knowledge. In: 23rd ACM Symposium on Applied Computing, 2008, Fortaleza. Proceedings of ACM SAC. New York: ACM Press, 2008. v. 1. p. 1-5.
- Christian Kruschitz and Martin Hitz. 2010. Analyzing the HCI design pattern variety. In Proceedings of the 1st Asian Conference on Pattern Languages of Programs (AsianPLoP '10). ACM, New York, NY, USA, , Article 6 , 8 pages.
- Conte, Mounia Fredj, Ibtissem Hassine, Jean-Pierre Girardin, and Dominique Rieu. 2002. A Tool and a Formalism to Design and Apply Patterns. In Proceedings of the 8th International Conference on Object-Oriented. Information Systems (OOIS '02), Zohra Bellahsene, Dilip Patel, and Colette Rolland (Eds.). Springer-Verlag, London, UK, 135-146.
- E. Gamma, R. Helm, R. Johnson, and J. Vlissides. Design Pattern. Addison-Wesley, To Appear, 1994.
- Fincher, S., & Finlay, J., (2003). CHI 2003 Workshop report: Perspectives on HCI patterns: Concepts and tools (introducing PLML). Interfaces, 56, 26-28. Brooklyn, NY: ACM Press.
- Fricke, A.; Völter, M. Seminars: A Pedagogical Pattern Language about teaching seminars effectively. In: Fifth European Conference on Pattern Languages of Programs, Germany, 2000, pp.1-36.
- Coplien, J. O. Software Design Patterns: Common Questions and Answers. In Linda Rising, editor, The Patterns Handbook: Techniques, Strategies, and Applications, p. 311-320. Cambridge University Press, New York, January 1998.
- Jordan Janeiro, Simone D. J. Barbosa, Thomas Springer, and Alexander Schill. 2010. Semantically relating user interface design patterns. In Proceedings of the 1st International Workshop on Pattern-Driven Engineering of Interactive Computing Systems (PEICS '10). ACM, New York, NY, USA, 40-43.
- Kumar, K.; Prabhakar, T. 2010. Design decision topology model for pattern relationship analysis. In Proceedings of the 1st Asian Conference on Pattern Languages of Programs (AsianPLoP '10). ACM, New York, NY, USA, , Article 3 , 9 pages.
- Kruschitz, C. XPLML: a HCI pattern formalizing and unifying approach. In CHI '09 Extended Abstracts on Human Factors in Computing Systems (CHI EA '09), 2009. ACM, New York, NY, USA, 4117-4122.
- Liu, H.; Singh, P. ConceptNet – a practical commonsense reasoning toll-kit. In: BT Technology Journal, vol. 22, Outubro 2004, pp. 221-226.
- Michael, J.; Lorraine, J. J. Principles for a Usability-Oriented Pattern Language. In Proceedings of the Australasian Conference on Computer Human Interaction (OZCHI '98). IEEE Computer Society, Washington, DC, USA, 1998, pp. 132-140.
- Minsky, M: 'The society of mind', Simon & Schuster, 1987, 336p.
- Montero, F.; Lozano, M.; González, P.; Ramos, I. A First Approach to Design Web Sites By Using Patterns. In: First Nordic conference on Pattern Languages of Programs: VikingPLoP. Hojstrupgard, Denmark, 2002, pp.137-158.
- Moraes, R. Análise de conteúdo. Revista Educação, Porto Alegre, v. 22, n. 37, p. 7-32, 1999.
- Paolo Bottoni, Esther Guerra, and Juan de Lara. 2010. Formalising design and interaction patterns and their relationships. In Proceedings of the 1st International Workshop on Pattern-Driven Engineering of Interactive Computing Systems (PEICS '10). ACM, New York, NY, USA, 32-35.
- Rosario Girardi and Alisson Neres Lindoso. 2006. An ontology-based knowledge base for the representation and reuse of software patterns. SIGSOFT Softw. Eng. Notes 31, 1 (January 2006), 1-6.
- van Welie, M. Web Design Patterns. <http://www.welie.com/patterns/> Acessado em: Mai, 2012.
- Walter Zimmer. 1995. Relationships between design patterns. In Pattern languages of program design, James O. Coplien and Douglas C. Schmidt (Eds.). ACM Press/Addison-Wesley Publishing Co., New York, NY, USA 345-364.
- White, B. Visualizing mobile design pattern relationships. In Proceedings of the 14th international conference on Human-computer interaction with mobile devices and services companion (MobileHCI '12), 2012. ACM, New York, NY, USA, 71-76.