Technical Report
Combining semiotic and cognitive perspectives to evaluate software interaction design

Cognitive Dimensions of Notation framework and
Metacommunication Template analysis
Collected Data - IBM RSA

Juliana Jansen Ferreira, Clarisse Sieckenius de Souza
SERG/Informática – PUC-Rio
R. Marquês de São Vicente 225 Rio de Janeiro – RJ, Brasil
[jferreira, clarisse]@inf.puc-rio.br

Renato Cerqueira
IBM Research Brazil
Avenida Pasteur 138 - Rio de Janeiro - RJ, Brasil
rcerq@br.ibm.com

Abstract

We have developed a Combined Semiotic-Cognitive method (CSCmethod) to evaluate software interaction design because we believe that an inspection combining both perspectives provides more comprehensive and powerful results compared to when we use semiotic and cognitive evaluation methods separately. Originally, the CSCmethod combined the Semiotic Inspection Method (SIM), which evaluates the communicability of interaction design, and an analysis based on the CDN framework (CDNf), which evaluates design usability. The original combination produced insightful results regarding both usability and communicability issues. Recently, however, we have been asking ourselves further questions. For example, does the order of execution matter? Could a "light version" of SIM be used in the combined method? This report presents a detailed account of the data and the evaluation process, using what we called "SIM light", instead of the complete version of the method. We evaluated a popular software modeling tool, namely IBM RSA. The report includes the complete set of *raw data* used in the combined evaluation, which is also the one used for the analysis reported in a paper published in the proceedings of the13th Brazilian Symposium on Human Factors in Computing Systems (IHC 2014).(Ferreira et al., 2014)

Summary

1. Introduction

The combination of usability-driven and communicability-driven perspectives enables a more comprehensive inspection of relevant interactive aspects of software artifacts, as well as the relations between them. For this study, we selected the Cognitive Dimension of Notation framework (CDNf) (Green and Blackwell, 1998) (Blackwell and Green, 2003) as our usability-driven inspection tool, and the Metacommunication Template (MetacommT) as our communicability-driven one. The latter is a basic common step in two much more elaborated methods proposed by Semiotic Engineering, namely the Semiotic Inspection Method and the Communicability Evaluation Method. (de Souza and Leitão, 2009)

The evaluation proposed and took the *tool-notation-people triplet* as a reference, from the preparation to the final analysis phases. In the preparation phase, we selected the software *tool* to be inspected, the "T" part of the triplet. At this early stage, the tool is not fully characterized. The "N" and "P" parts are likewise only loosely defined, in general terms (*e. g.* "N" is UML, "P" represents the requirement engineers). Together the three parts (T, N and P) define the *triplet baseline* for the inspection. Later in the analysis phase, we will have learned more about the "T", "P" and "N". Thus, we can then build a complete characterization of the triplet for a specific inspection scenario that directs the entire study.

The inspected tool was IBM RSA, a popular modeling tool in software development. In the next sections we provide details of the evaluation study's settings and procedures, as well as the cognitive and semiotic evaluation techniques stemming from CDNf and Semiotic Engineering methods.

2. The Semiotic-Cognitive combined evaluation method

In this section we provide details about CDNf and the use of the metacommunication template (MetacommT), which is a fundamental piece of the Semiotic Inspection Method and the Communicability Evaluation Method.

2.1. The Cognitive Dimension of Notation framework (CDNf)

The Cognitive Dimension of Notation framework is defined as set of design principles for creating or evaluating notations, user interfaces and programming languages used with information artifacts (Blackwell and Green, 2003). They provide a common vocabulary for discussing many cognitive factors of such representation systems. Their aim is to improve the quality of discussions and decisions in design and evaluation activities. (Green and Blackwell, 1998) There are fourteen dimensions in the CDN framework as shown in Table 1.

Table 1. List of CDN (Blackwell and Green, 2003)

Cognitive Dimension of Notation	Description
Abstraction	Types and availability of abstraction mechanisms
Closeness of mapping	Closeness of representation to domain
Consistency	Similar semantics are expressed in similar syntactic forms
Diffuseness	Verbosity of language
Error-proneness	The notation invites mistakes and the system gives little protection
Hard mental operations	High demand on cognitive resources
Hidden dependencies	Important links between entities are not visible
Premature commitment	Constraints on the order of doing things
Progressive evaluation	Work-to-date can be checked at any time
Provisionality	Degree of commitment to actions or marks
Role-expressiveness	The purpose of an entity is readily inferred
Secondary notation	Extra information in means other than formal syntax
Viscosity	Resistance to change
Visibility	Ability to view components easily

There are several steps to be taken when applying Cognitive Dimensions analysis to a system's design:

(1) get to know your system;

(2) decide what the user will be doing with the notation;

(3) choose some representative tasks;

(4) for each step in each task, ask whether the user can choose where to start; how a mistake will be corrected; what if there are second thoughts; what abstractions are being used; and so on, for all dimensions. These steps will generate an *observed profile*; and

(5) compare the observed profile with the ideal profile for that type of activity.

2.2. Metacommunication Template (MetacommT)

MetacommT is used to guide communicability investigation, from either a sender's or a receiver's perspective, by providing an abstract and logically articulated representation of the designers' discourse about their understating of who the users are (what needs and preferences they have, what may be their goals and expectations, their abilities, their knowledge, etc.) as well as the design decisions and choices that ultimately express their design intent (and its expected value for the targeted users). (de Souza and Leitão, 2009) The content and structure of MetacommT is:

"Here is my understanding of who you are, what I've learned you want or need to do, in which preferred ways, and why. This is the system that I have, therefore, designed for you, and this is the way you can or should use it in order to fulfill a range of purposes that fall within this vision." (de Souza, 2005)

The first person of discourse in the template (referred to as "I", "my") stands for the designer, and the second (referred to as "you", "your") stands for the user. The third person in discourse (referred to as "it", "this") stands for the system, represented by its interface. Together, they characterize the participants in an elaborate metacommunication process that takes place during human-computer interaction: user and designer.

3. The Evaluation methodology

We performed four steps (Figure 1) to execute the combined evaluation, using the TNP triplet as reference thought out all the steps:

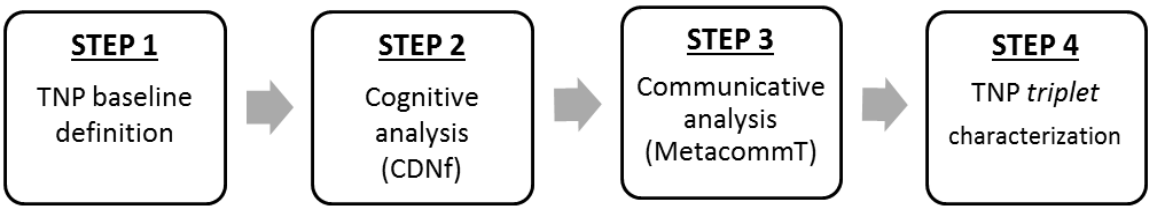


Figure 1. Combined evaluation with TNP triplet

3.1. TNP baseline definition

The TNP baseline is the reference for the entire evaluation. SIM (de Souza and Leitão, 2009), like most other evaluation methods, takes a baseline configuration in the "preparation phase", which may be implicit in evaluation scenarios and tasks or explicit as an evaluation component in itself. Since the TNP baseline will guide our evaluation, we highlight its importance by making it a separate step of our combined method.

3.1.1. Selected tool and modeling notation

We chose IBM Rational System Architect (IBM RSA)[1] as the tool for our evaluation study. IBM RSA is a well-known software modeling tool selected because it features as one of the top modeling tools in the Gartner Magic Quadrant of Enterprise Architecture tools[2] used by the software engineering industry.

In order to focus the inspection on the modeling activity itself, the necessary steps to establish a "project" or "environment", which the modeling tool requires, were not part of the evaluation scenario. Thus, we started from projects or folders which we assumed had been previously created with IBM RSA.

We selected the Unified Modeling Language (UML) (OMG, 2014) as the modeling notation for the inspection because UML is the most used modeling notation in software development projects. Regarding the type of model to work with, we chose

[1] IBM Rational System Architect (IBM RSA) trial 8.0.4, https://www.ibm.com/developerworks/rational/products/rsa

[2] Gartner Group - https://www.gartner.com/doc/2601526

the activity model notation because it helps create a kind of "boundary model" between the business and the technology portions of the system being developed. It can be used to model more abstract activities (like interactions between system and users or between systems), as well as to model specific object parts of the system itself, showing how they change along the process and over time.

3.1.2. User profile

The user profile defined for this study is that of an active professional in business modeling, requirements modeling and conceptual modeling. However, we assumed that this user *had never used the selected tool* and *had little experience with UML activity models* in real projects. All he knew about them was what he had learned in professional training and education some time ago

3.1.3. Evaluation scenario

The selected scenario describes a commonly encountered situation in a development project, that of a new *IT professional joining a software development team and having to collaborate with others right away*.

The scenario professional joins the IT team of a company to *work in the development project of a new HR system*. This professional will participate in the specification phase of the project, *interacting with the client and other IT team members* while defining the business domain related to the software under development.

The professional *is not familiar with the modeling tool used by his new colleagues*, so the team manager gives him a special task with which he will get acquainted with the tool. He should build an activity model for a "Vacation request process", using previously collected data with annotations. The model is to be used in conversations with clients as well as with other IT team members.

The final model should represent the sequence of activities of the "Vacation process request" for a given corporation, including the following sequence of tasks:

1. The employee fills out a vacation request form and submits it to the HR system.
2. The HR system checks if there are conflicting vacation dates scheduled for the same period. Conflicts arise when two or more people ask to be absent from the company at the same time. Normally people who work in the same sector of a company are each other's backup (peers).

3.a. If there are no conflicting vacation dates, the HR system sends the vacation request to the Manager to review and approve it.

3.b. If there are conflicting vacation dates, the HR system informs the conflict to the employee who can then propose new dates to avoid the conflict or , alternative, to negotiate the desired vacation period with his colleagues.

- If an employee wants to negotiate dates with a colleague, he sends him or her a message through the system, asking about alternative dates.
- The peer can accept or reject this proposal.
- The manager receives the new vacation request and reviews it.

4.a. If the manager agrees with the request without any comments or adjustments, the manager approves the request. Then the HR system notifies the HR vacation handler that an approved vacation needs scheduling and also notify the employee about the approved vacation request.

4.b. If the manager does not agree with the request, he writes a comment on the request form and sends it back to the employee for review.

3.2. Cognitive analysis with CDNf

In this step, while executing the proposed task of the evaluation scenario, we carry out the cognitive analysis by inspecting the interface and supported interactions of IBM RSA. During the inspection, we identified a number of HCI issues and associated each one of them with cognitive dimensions, according to the observed characteristics. We also used the TNP triplet baseline as reference for the identification of such issues, relating them to "T", "N", "P" or a combination of these factors. The evidence collected to illustrate each one of the issues is presented in the section 4, below.

3.3. Semiotic analysis with MetacommT

In this step, we filled out the Metacommunication template (MetacommT) taking into consideration what we learned during the previous step, with the CDNf analysis. Then, we carried out an analysis of the designer-to-user metacommunication through interface and interaction signs in order to fill out the details of the metacommunication template proposed by Semiotic Engineering (see on page 5). We divided MetacommT in four segments as a means to categorize the issues identified during the metacommunication inspection:

1. "Here is my understanding of who you are, …"
2. "…what I've learned you want or need to do, in which preferred ways, and why…."

3. "…This is the system that I have, therefore, designed for you, and this is the way you can or should use it…"
4. "…in order to fulfill a range of purposes that fall within this vision."

3.4. TNP triplet characterization

The TNP triplet characterization is the step in which we resort to the TNP triplet baseline and take it as the "ideal situation" in terms of which to discuss the issues identified during the inspections. We discuss each part of the triplet in each and every context where a cognitive or a semiotic issue was detected in previous steps of the combined evaluation procedure. We relate each issue to one or more of the factors (T, N or P) or to one or more of the relations among them. For example, if the tool *misleads* the user in his or her use of the notation, this issue has to do with T, N and P. If the tool resizes and redraws model elements, without any explicit request from the user, and this leads to user *disorientation*, this issue has to do with T and P, only.

4. Evidence collected during evaluation

In this section, we present the raw data collected during the combined evaluation. Notice that, as is the case with interpretive methods in general, in our proposed method *data* is "constructed' by the analyst. In other words, it is the analyst's interpretation of a contingent configuration of his or her object of interest that determines whether this is a piece of *data* for investigation. For example, the resizing illustration mentioned in the last paragraph of the previous section *may or may not* constitute a piece of evidence in the analyst's view. It is his or her judgment about the causes/consequences of the resizing for the user that will ultimately tell if this will count as 'raw data' for ulterior steps of analysis.

In this report we presented all the pieces of evidence, and related issues, which we identified throughout the combined cognitive-semiotic evaluation. In the paper where we discussed this research (Ferreira et al. 2014), for lack of available space in conference paper format, we mentioned only a few of the most evocative issues.

4.1. Decision node connected to itself

IBM RSA allows for an activity model to have a decision node connected to itself (see Figure 2). This notational configuration refers to a loop that can never be solved, because there isn't any action to define where the activity flow will go after the decision is made. This association does not represent a proper model configuration, since no activity or action is assigned to the system (it doesn't *act* in any way).

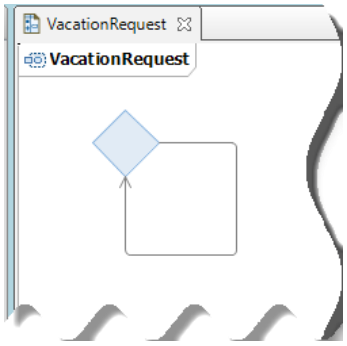


Figure 2. A decision node associated with itself

CDNf notes:	Although this feature relates to an important cognitive dimension in usability terms, namely **Progressive evaluation** (the ability to produce incomplete representations that are incrementally composed along the modeling process, in our case), it does so in such a way that a usability problem is entailed by the interaction designers' solution. Because there is no distinction between provisional representations to be further specified and definitive representations that express the modeler's final decision, this IBM RSA feature is also associated with another cognitive dimension of notations, namely **Error-proneness.** The notation invites mistakes and the system gives little protection against them. IBM RSA allows the user to use the UML notation to build connections that do not have a meaning considering a model as a whole. A decision node connected to itself characterizes a loop that does not have a solution. The user building the model ("P") can be misled by "T", while using "N".
MetacommT notes:	**"Here is my understanding of who you are…"** *"I understand that you have a fair amount of experience with modeling and the UML activity model notation…"* IBM RSA designers somehow excuse themselves for *not providing* explanations further information about UML.

	Therefore, we can assume that in the designer's view, this is not a critical need of the users (or else they should have included this matter in their tool). **"…what I've learned you want or need to do, in which preferred ways, and why."** *You may wish to build models incrementally, which means that some intermediary stages in the process may be obviously wrong from a UML semantics perspective.* **"…This is the system that I have therefore designed for you, and this is the way you can or should use it…".** *"Therefore, the system doesn't check the semantics of the model all the time and trust you to be able to track where the evolving model needs to be elaborated further in order to be semantically correct. We provide a semantic verification, but it is not complete. Only a subset of connections can be checked, and you will find this by trial and error. "*

4.2. Initial node directly connected to a final node

In IBM RSA an activity model can be represented with an initial node directly connected to a final node (see Figure 3). This connection does not have a proper meaning, considering that the user cannot build an activity model without any activity.

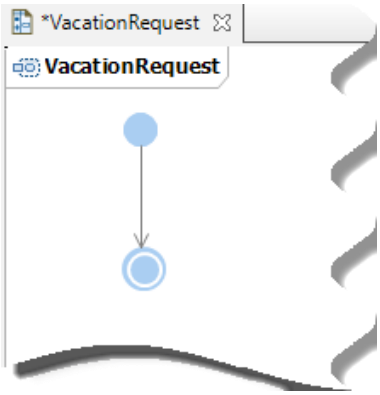


Figure 3. Initial node connected to a final node

CDNf notes:	The CDN **Progressive evaluation**, the ability to produce incomplete representations that are incrementally composed along the modeling process, in our case, and CDN **Error-proneness**, which means that the notation invites mistakes and the system gives little protection against them, can be associated with this issue. This is another situation similar to the previous issue described. Considering the user profile, someone who does not have much experience with the UML

	notation, the "progressive evaluation" feature may lead to "Error-proneness".
MetacommT notes:	**"Here is my understanding of who you are…"** *"I understand that you have a fair amount of experience with modeling and the UML activity model notation…"* IBM RSA designers somehow excuse themselves for not providing explanations further information about UML. Therefore, we can assume once again that in the designer's view this is not a critical need of the users (or else they should have included this matter in their tool) **"…what I've learned you want or need to do, in which preferred ways, and why."** *You may wish to build models incrementally, which means that some intermediary stages in the process may be obviously wrong from a UML semantics perspective.* **"…This is the system that I have therefore designed for you, and this is the way you can or should use it…".** *"Therefore, the system doesn't check the semantics of the model all the time and trust you to be able to track where the evolving model needs to be elaborated further in order to be semantically correct. We provide a semantic verification, but it is not complete. Only a subset of connections can be checked, and you will find this by trial and error. "*

4.3. A 'too quick' notification about an incorrect connection

A prohibition sign (⃠) quickly flashes when the user tries to make an incorrect association between elements (see Figure 4). If the user is not paying close attention, or is moving the mouse rapidly across the interface display, the feedback about the mistake may not be noticed. As a result, he may be left wondering why the association is not being done.

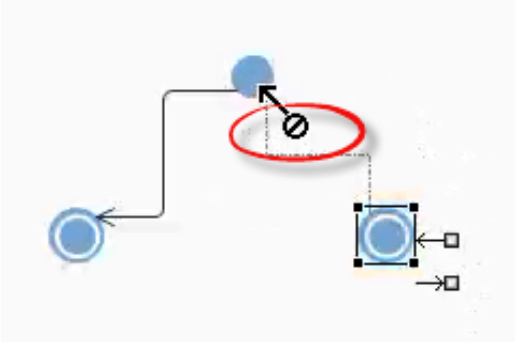


Figure 4. Prohibition sign while trying to connect a final to a initial node

CDNf notes:	This characteristic is related to two cognitive dimensions in CDNf: **Visibility** (the visual feedback is not visible enough) and **Diffuseness**, which refers to how many symbols (or how large a representational space) the notation requires to express meaningful content. Only that one quick sign indicates to the user that what he is trying to do is not allowed by the tool.
MetacommT notes:	**"…what I've learned you want or need to do, in which preferred ways, and why."** *"… you need to build activity models with some help to prevent incorrect connections. And you are constantly paying close attention to all screen states in the system's interface."* **"…This is the system that I have therefore designed for you, and this is the way you can or should use it…".** *"Therefore, the system will help you by not allowing you to make some incorrect connections. Since you clearly just slipped-up about the incorrect connection, I don't need make a big deal about it. You will quickly notice what you are doing wrong."*

4.4. Visual ambiguity

In Figure 5 we show two different elements with the same visual representation: Decision and Merge nodes. UML actually assigns the same representation to these two elements, but we would expect that a modeling tool (designed to support modelers) should use the opportunity to call the users' attention to the problem *visually*. According to UML, the behavior of Decision and Merge nodes is different: only one flow can arrive at a Decision node and only one flow can come out of a Merge node. The user

will only realize that distinction if he clicks on each element to see its properties. IBM RSA also communicates that distinction by implementing the "one flow rule" mentioned before, but the way this communication is performed, the user might not understand the difference. If the user tries to add a 2nd flow arriving at a Decision node, IBM RSA will create a flow leaving the decision node. It is like saying: "My dear user, I believe you made a mistake! What you really wanted was to add a flow leaving the Decision node. Therefore, I changed the direction of the flow you added." The message seems fine, but IBM RSA does not really notify the user, it simply changes the flow direction and this action looks just as quick as the prohibition sign in the previous issue. The user could easily miss the "message".

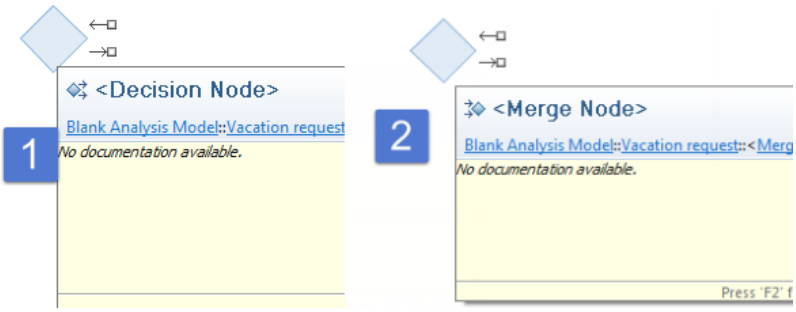


Figure 5. Decision and Merge nodes

CDNf notes:	We can see two different elements with the same visual representation: merge and decision nodes. This can be associated with CDN **Visibility.** Although this is a problem with UML, the designer could have used this opportunity to call the users' attention to the problem *visually*. According to UML, the behavior of 'Decision' and 'Merge' nodes is different in that only one flow can arrive at a decision node and only one flow can come out of a merge node. IBM RSA helps the user by not letting him exceed the limit of flows. However the way it communicates this to the user (by not allowing in or out connections depending on which is the case) may be very confusing to the user who realizes that some connections are not allowed, but misses the reason why this is happening. This issue can also be related to another CDN, ***Hard mental operation*** (a high demand on cognitive resources for the user to understand the notation system) if we consider that for proper interaction the user must be able to handle visual ambiguities in the notation and keep mental track of which is the case at every instance of occurrence.
MetacommT	**"…what I've learned you want or need to do, in which preferred ways, and why."**

notes:	*"… you need some help to prevent incorrect connections while building your models.* **"…This is the system that I have therefore designed for you, and this is the way you can or should use it…".** *"Therefore, the system will help you by correcting some incorrect connections that you may try to execute. Since you clearly just slipped-up about the incorrect connection, I don't need to alert you about it. The system makes it all ok and you can go on modeling."*

4.5. Modeling or drawing tool?

IBM RSA provides the usual resources that tools dealing with visual representations typically incorporate (*e. g.* arrange and align forms), as well as other drawing functions, such as sketched and geometric shapes (see Figure 6). Likewise, IBM RSA implements *some* (although not *all*) of the UML constraints for the activity model. The tool explicitly provides a function called "UML model validation" (see Figure 7).

In its help content, IBM RSA states that it "supports UML 2.2" and refers the reader to the OMG website for more details (Figure 8). In other words, IBM RSA decides to hand over the task of providing notational definitions and explanations to the organization that maintains UML (OMG).

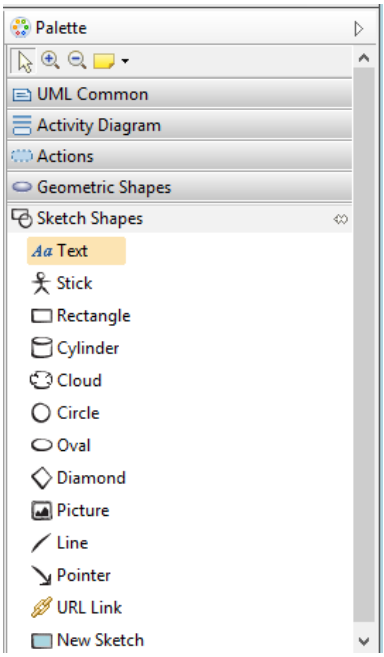


Figure 6. IBM RSA Palette

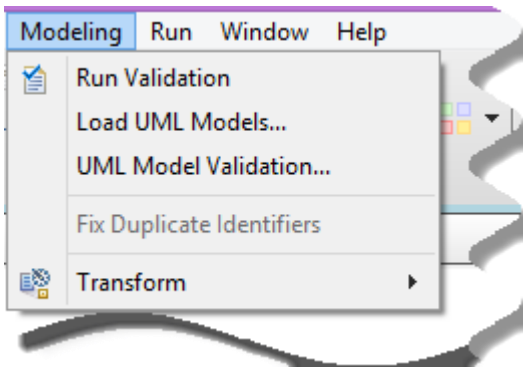


Figure 7. IBM RSA UML validation

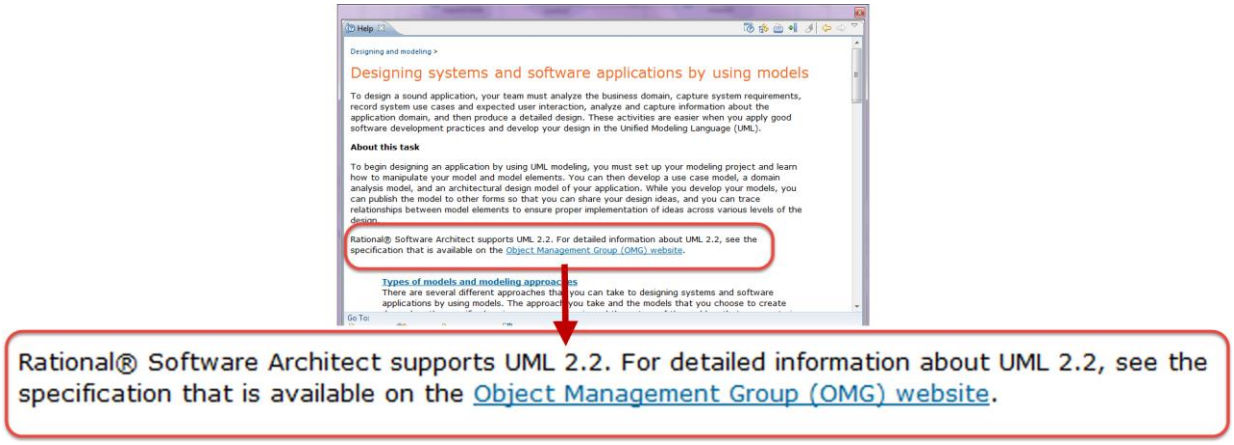


Figure 8. IBM RSA help item about OMG UML specification

CDNf notes:	Having an unexperienced UML notation user in mind, the freedom to switch between modeling and drawing functions while building a model may make it difficult to identify potential problems regarding the final model's compliance with UML standards. The CDN **Hard mental operations** (A notation can make things complex or difficult to work out in your head, by making inordinate demands on working memory, or requiring deeply nested goal structures.) is a cognitive characteristic related to this issue. The user building the model will need to have some previous knowledge of the notation in order to build a "correct" UML model that other people can understand and use in subsequent stages of software development.
MetacommT notes:	**"…what I've learned you want or need to do, in which preferred ways, and why."** *"You may wish to build models incrementally, which means that some intermediary stages in the process may be obviously wrong from a UML semantics perspective. We, therefore, don't check the semantics of the model all the time, and trust you to be able to track where the evolving model needs to be elaborated further in order to be semantically correct …"* The designer transfers the responsibility about UML specification to OMG, indicating that it only "supports UML 2.2", but further information is someone else's responsibility, as shown in Figure 8. **"…This is the system that I have therefore designed for you, and this is the way you can or should use it…".** *"So, you can build a visual representation of the model and invoke a semantic verification function to check the correctness of your model at any time. The semantic*

	verification is not complete, however. Only a subset of connections can be checked, and you will find this by trial and error. ...”

4.6. Visual distinction between modeling and drawing elements

As discussed and shown in the previous item, IBM RSA provides modeling and drawing resources to users. The elements have different goals and also distinct appearances, which is a positive feature. Some elements may have similar forms, as shown in Figure 9, but the perceivable differences in form help the user realize that *there must be differences* in their meaning or use, as shown in Figure 10. This difference might be a good way to alert the user about the availability of different kinds of representations which are actually provided by IBM RSA. We are, in this case, pointing at an emerging *rhetoric* of representations, which the tool might more actively support, that is, communicate more explicitly to the user when and why use one type of representation and not other(s).

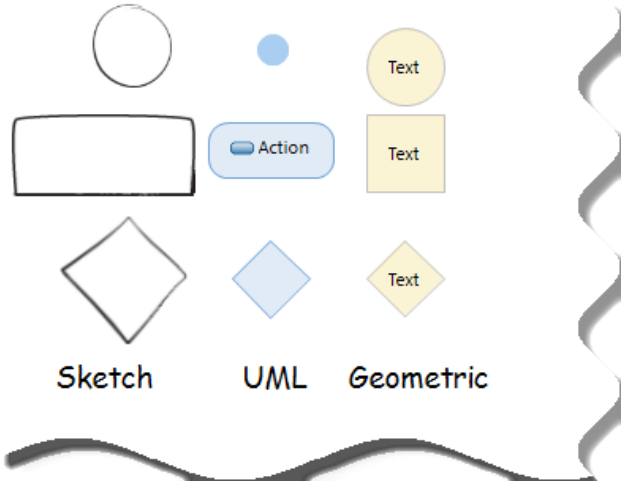


Figure 9. Sketch, UML and Geometric elements

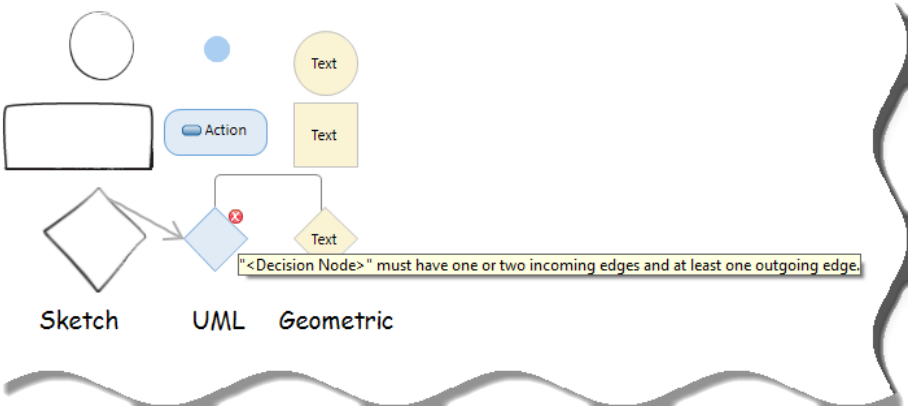


Figure 10. Different kind of elements connected, but caught by UML validation

CDNf notes:	The **CDN Consistency**, which states that similar semantics should be expressed in similar syntactic forms, can be related to this evidence. The distinct appearance of each element provides the user with enough information to realize that IBM RSA deals with different (rhetorical) kinds of elements. On the other hand, because the user can associate elements from different (rhetorical) types, we can also relate the **CDN Error-proneness** with this piece of evidence. Of course this may be a pragmatic error, like writing a letter that uses highly formal speech and street jargon in the same piece of communication. IBM RSA allows the user to connect various kinds of notational elements and save the model as a (supposedly valid) UML activity model. If, by his own initiative, the user subsequently runs the UML model validation, IBM RSA will issue some alerts for semantically questionable parts of the model, but not for connections between elements from different (rhetorical) kinds. In Figure 10, the drawn edge (oblique arrow shape joining a sketched diamond and a UML decision node may have been meant, by the unexperienced user, as an incoming edge. However, according to the error message we see in Figure 10, there are no incoming edges to that decision node.
MetacommT notes:	**"…what I've learned you want or need to do, in which preferred ways, and why."** *"You may wish to build models incrementally, which means that some intermediary stages in the process may be obviously wrong from a UML semantics perspective. To this end, you may wish to use different styles of shapes, with which you annotate things that you must take care of in more detail as you make progress in the modeling task."* **"…This is the system that I have, therefore, designed for**

	you, and this is the way you can or should use it…" *"Since you may need some intermediary representation of the UML activity model, we allow you to use different types of representation (geometric, UML, sketch. Some connections between elements of different types are allowed, but some UML constraints are still been checked while modeling and you will find this by trial and error. …"*

4.7. Activity model without activity

IBM RSA allows the user to build a "valid" activity model without any activity node in it, as shown in Figure 11. The UML specification (OMG, 2014) states that an activity model must have at least one activity node. However, IBM RSA allows the user to save that model *as is* without any warning. More importantly, the tool provides a "UML model validation" function that does not indicate any problem with the model, as shown in the encircled area of Figure 11:

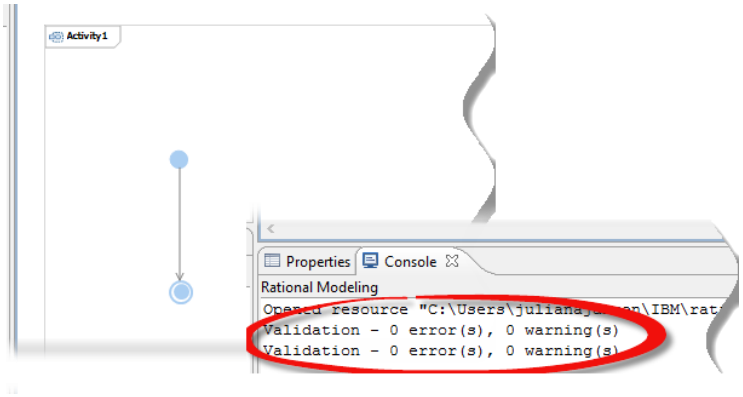


Figure 11. Activity model without activity

CDNf notes:	Issue referring to ***Error-proneness,*** which means that the notation invites mistakes and the system gives little protection. IBM RSA does not provide any help for users during modeling activity regarding the validity of the activity model being built.
MetacommT notes:	**"…what I've learned you want or need to do, in which preferred ways, and why."** *"You may wish to build models incrementally, which means that some intermediary stages in the process may be obviously wrong from a UML semantics perspective."* **"…This is the system that I have, therefore, designed for**

	you, and this is the way you can or should use it…" *"Therefore, the system will not highlight every semantic mistake, or even warn you about it. You can build your model as you wish, until you think it is finished and correct."*

4.8. Element size and element text

When an activity model is being built in IBM RSA, the size of elements added to the model are changed according to the length of text (element name) typed in by the user as shown in Figure 12 and in Figure 13. As a result, after adding text, the user generally needs to resize the corresponding element in order to have a better layout configuration of the elements in the model. We looked for a toggle option to resize elements or not, which would prevent such 'hypercorrections' to be automatically performed by the system, but couldn't find any. Not even a broader search in the Internet returned anything suggesting that this is a usability problem in IBM RSA.

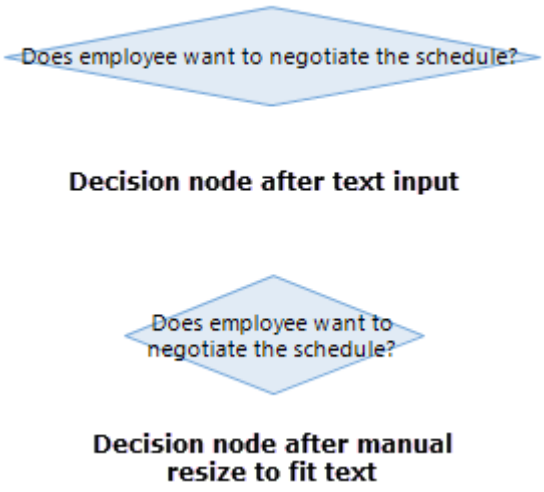


Figure 12. Decision node resided according to the text

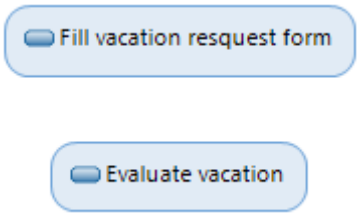


Figure 13. Action nodes with different sizes

CDNf notes:	The complicated process to resize each element to rearrange it in the model can be associated with the CDN **Viscosity**, which is related to resistance to change. A viscous system needs many user actions to accomplish one goal. The user needs to resize each added element to fit the model. The problem with auto-resize always 'on' is that a model's

	layout can get visually unbalanced and even difficult to read, challenging users to recognize wide geometric shapes (like the diamond in Figure 12) as having the same meaning as other smaller instances, not to mention the problem of spreading the model area way out of the user's viewport.
MetacommT notes:	**"…what I've learned you want or need to do, in which preferred ways, and why."** *"… by connecting UML activity elements using the mouse to build a model and the keyboard to assign a name to each element."* **…This is the system that I have, therefore, designed for you, and this is the way you can or should use it…"** *"Therefore, we will enlarge every element added to the model so its whole name is displayed. If you want to change the size, you have to adjust each one, after adding it to the model."*

4.9. Confusing help content – Interface language vs. local language

IBM RSA help content displays content in English and in Portuguese all mixed up, as shown in Figure 14. There isn't any consistent language configuration in the Help system, as far as we know. We suppose that some of the content is related to the country where the user is located, but not all technical content is translated. Therefore, most of it is displayed in English. However, some localization algorithm seems to be in place.

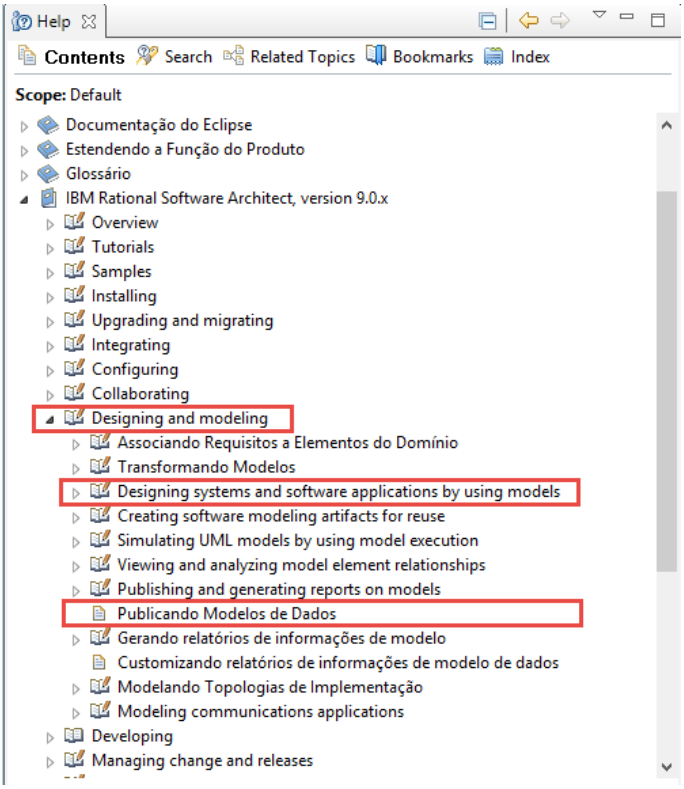


Figure 14. English and Portuguese content mixed up on the same help item

CDNf notes:	This situation can be related to CDN **Hidden dependencies,** which stands that important links between entities are not visible. The user does not know the relation between help language and tool language.
MetacommT notes:	**"Here is my understanding of who you are…"** *"I understand that you want to use your local language whenever possible, but that you don't mind using the foreign language used in the original software interface itself."* **…This is the system that I have, therefore, designed for you, and this is the way you can or should use it…"** *"I have, therefore, left some parts of the system in the original design language (English), whereas others are being progressively translated into other languages for localized experience. Sometimes the content will be in the original design language (English), sometimes in your local language."*

4.10. Confusing help content - Path does not exist!

Another confusing situation with help content is the indication of a path to a menu item that does not exist (shown in Figure 15). The option indicated on the help is not available on the IBM RSA menu. The function option can be found on another menu item as showed in Figure 15.

Another confusing situation with help content is the indication of menu items that do not exist (shown in Figure 15). The option indicated in the help instruction is not available in the corresponding IBM RSA menu. However, the option can be found in another menu item as shown in Figure 16. This inconsistency between online Help content and interface version is unfortunately not rare, suggesting that agile help update and validation checkers are yet to be developed to guarantee that users can *trust* online instructions for how to use systems.

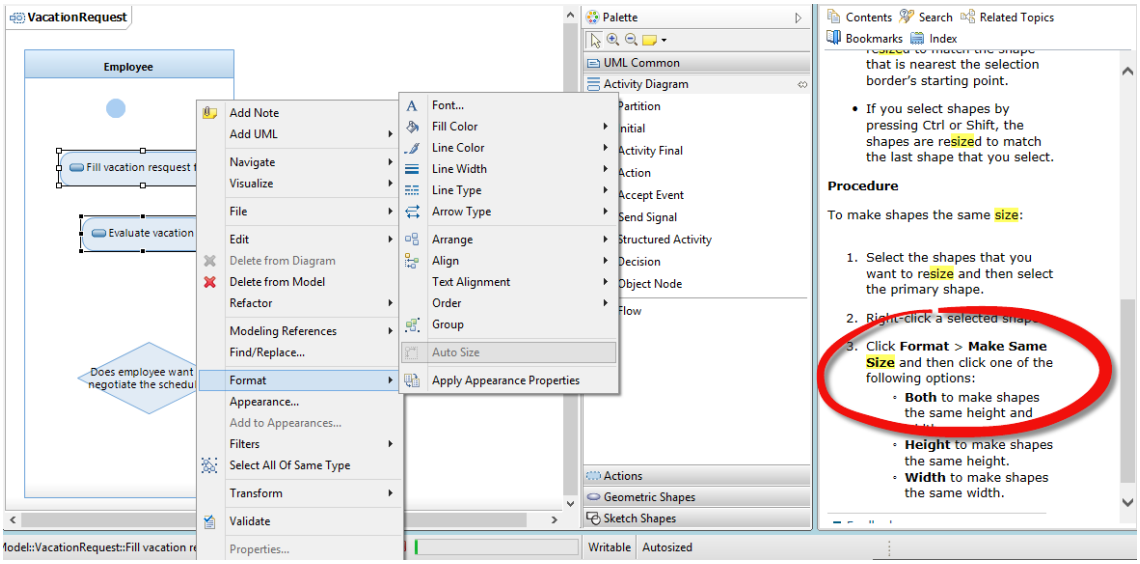


Figure 15. Option indicated on the Help content does not exist on IBM RSA

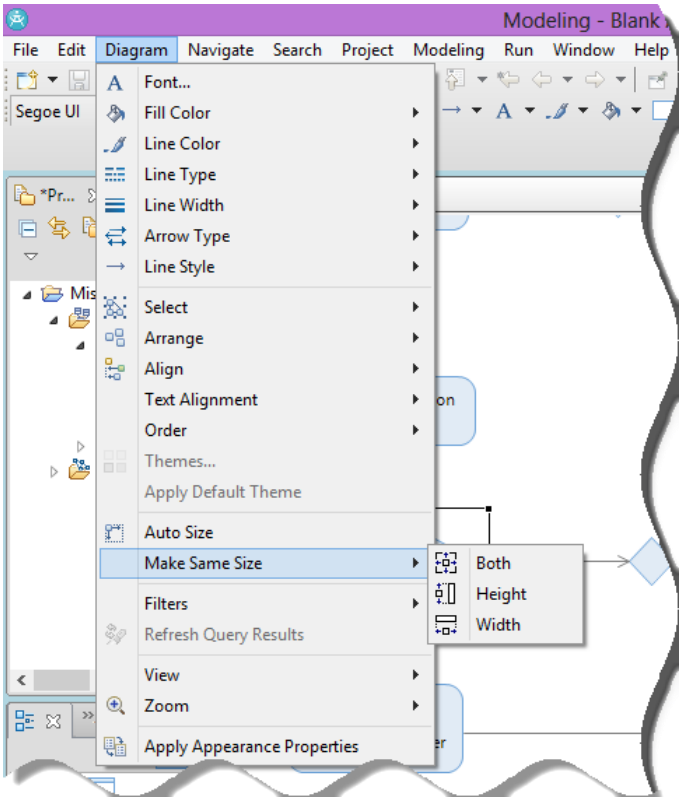


Figure 16. "Make same size" function on another place different from indicated by Help

CDNf notes:	The lacking of the menu item indicated in Figure 15 does not relate to a CDN itself. It can be a version error, but it also illustrates the hidden dependency issues of help and the tool itself.
MetacommT notes:	**"Here is my understanding of who you are…"** *"I understand that you do not really need to use the help to perform your modeling tasks."* **…This is the system that I have, therefore, designed for you, and this is the way you can or should use it…"** *"Therefore, the help content might show some discrepancies with the current tool's version, like showing a function on a different menu item. We are constantly changing! So, the help content can be outdated."*

4.11. Objects need constant refitting

The activity model uses the partition element to represent the responsible for each action modeled. The partition is represented in UML by a delimited area of the model, where some elements can be disposed. All elements need to be on a partition. During modeling activity, the user needs to be constantly stretching the partition to get a

new element on it, or to fit an element that was already there, but needed some replacement. As showed in Figure 17, the action object is not all in the "Manager" partition, some of it is outside the model.

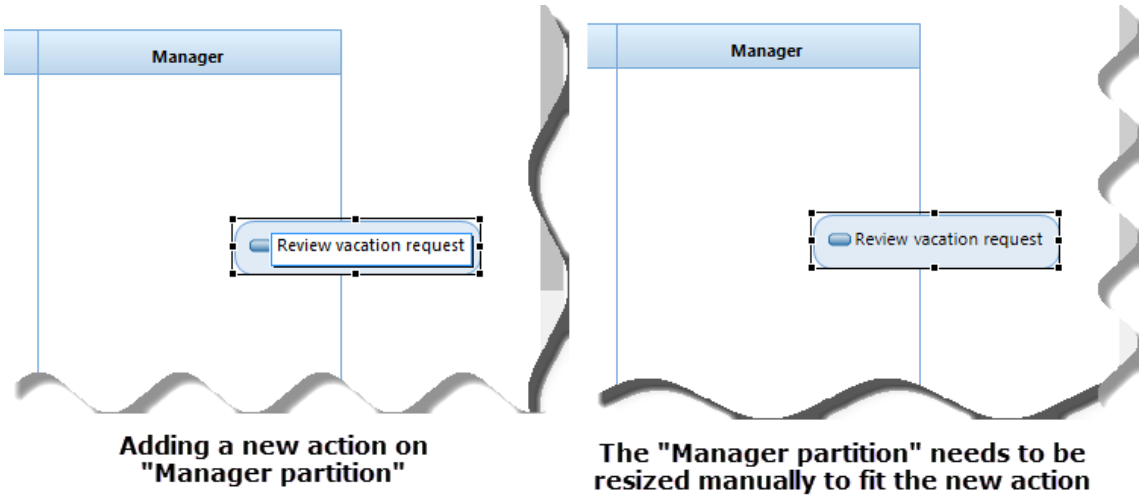


Figure 17. Partition does not resize itself once one of its elements gets bigger

CDNf notes:	This situation can be related to CDN **Viscosity**, which is the resistance to change. A viscous system needs many user actions to accomplish one goal. Sometimes the user needs to get more space in a partition, he needs to stretch the partition them add or move elements.
MetacommT notes:	**"…what I've learned you want or need to do, in which preferred ways, and why."** *"… by connecting UML activity elements using the mouse to build a model and the keyboard to assign a name to each element…"* **…This is the system that I have, therefore, designed for you, and this is the way you can or should use it…"** *"We let you add as many elements as you want to the partition element, but sometimes you might need to resize it to make room for more elements."*

4.12. The Metacommunication template filled up

The Metacommunication template was filled up as presented below:

"I understand that you have a fair amount of experience with modeling and the UML activity model notation. I understand that you do not really need to use the help to perform your modeling tasks. If you eventually do need to use the help, you want to use your local language whenever possible, but that you don't mind using the foreign language used in the original software interface itself. You build models by connecting UML activity elements using the mouse and the keyboard to assign a name to each element. You may wish to build models incrementally, which means that some intermediary stages in the process may be obviously wrong from a UML semantics perspective. You might need some help to prevent incorrect connections, and you are constantly paying close attention to all screen states in the system's interface. To build intermediary models, you may wish to use different styles of shapes, with which you annotate things that you must take care of in more detail as you make progress in the modeling task. Therefore, the system doesn't check the semantics of the model all the time and trust you to be able to track where the evolving model needs to be elaborated further in order to be semantically correct. We provide a semantic verification, but it is not complete. Only a subset of connections can be checked, and you will find this by trial and error. We will help you by not allowing you to make some incorrect connections. Since you might slip-up about the incorrect connection sometimes, we don't make a big deal about it. I believe you will quickly notice what you are doing wrong. Sometimes, we will help you by correcting some incorrect connections that you may try to execute. I don't need to alert you about it. We correct it and you can go on modeling. We provide the semantic verification that can be invoked at any time, although it is not a complete verification. In the UML activity model, we allow you to use different types of representations (geometric, UML, sketchings). Some connections between elements of different types are allowed, but some UML constraints are still being checked as you go on modeling, and you will find this by trial and error. We will not highlight every semantic mistake, or even warn you about it. You can build your model as you wish, until you think it is finished and correct. We will enlarge every element added to the model so its whole name is displayed. If you want to change the size, you have to adjust each one, after adding it to the model. We left some

parts of the system in the original design language (English), whereas others are being progressively translated into other languages for localized experience. Sometimes the content will be in the original design language (English), sometimes in your local language. The help content might show some discrepancies with the current tool's version, like showing a function on a different menu item. We are constantly changing. So, the help content can be outdated. We let you add as many elements as you want to the partition element, but sometimes you might need to resize it to make room for more elements. We believe that you want to use the system to build UML activity models that you can store, print, publish, revise, transform into code or discard along the software development process."

5. TNP triplet characterization

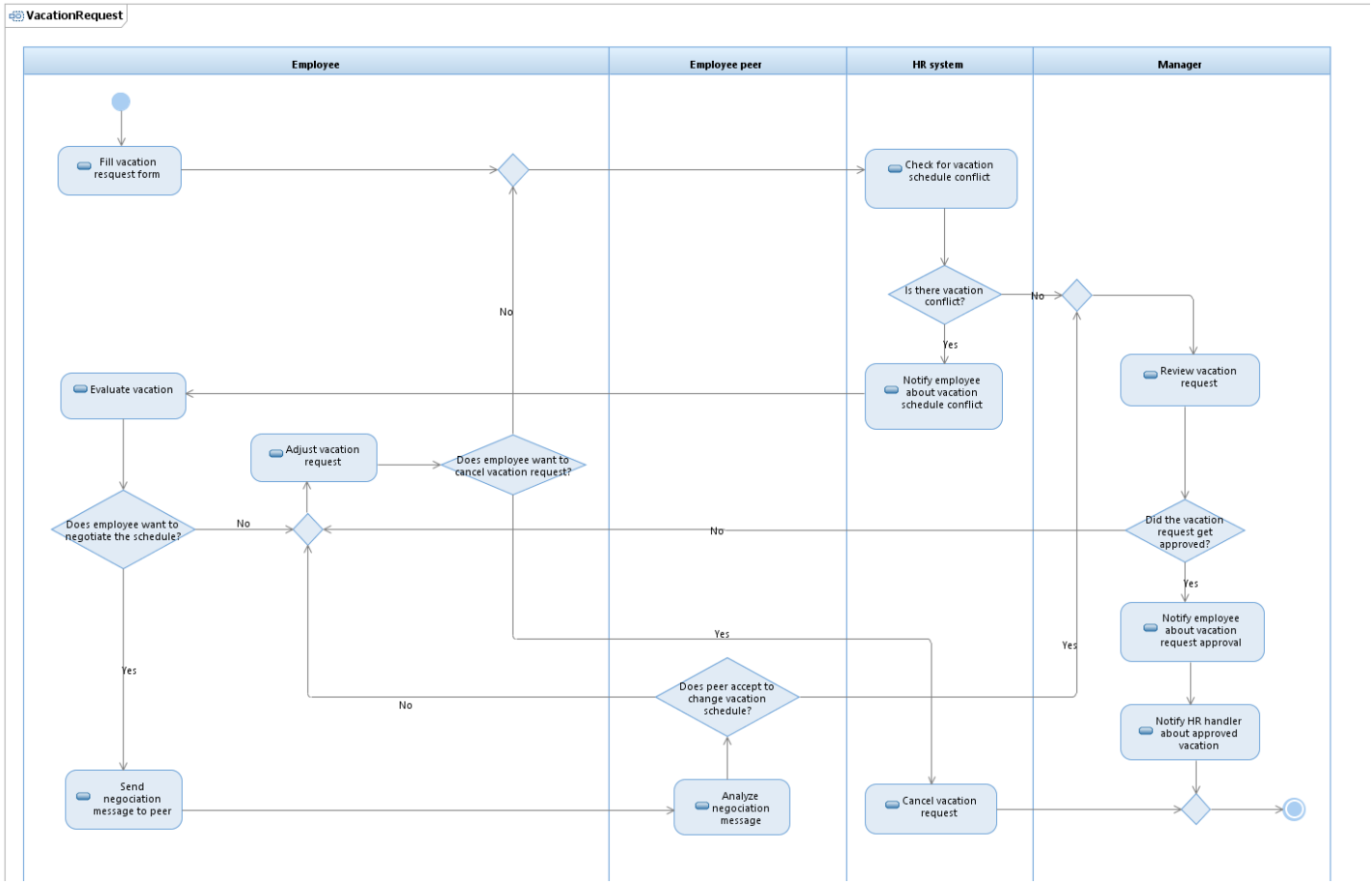
In this step, we were able to define some interesting relations among T-N-P factors, considering them in pairs or even all of the factors combined. The complete TNL triplet characterization, the conclusions and discussion about the combined evaluation can be found in the published paper. (Ferreira et al., 2014)

- "T" allows the user to build models using the "N", but the model built may not have a proper meaning as far as the UML activity notation semantics is concerned. The user building the model ("P") can be misled by "T", while using the "N".
- "T" does not provide proper feedback information to "P" that "N" is being mistakenly used. This is a very common usability problem in a number of applications, not only modeling tools.
- By preventing the user from making an incorrect connection in a UML activity model, IBM RSA helps the modeler to achieve better results ("T-N-P"). It also tells us something about "T-P", similar to the feedback example, as well as about "N", regarding the ambiguity in UML notation. Because it provides a description of the element once it is selected, IBM-RSA also supports the "T-N" relation, clarifying the meaning of the notation for the people who use it.

- "T" is not handling "N" properly and may mislead "P" into building an activity model using the specifications of "N" that does not make any sense to other "P" in the process.
- With the MetacommT, the "P' part of the triplet considers the designer as well as the user of the tool. By providing some degree of modeling liberty to users ("Pu"), the designer ("Pd") indicates that he believes in the users' capacity to be critical about constraints of the model under construction. "Pd" provides some validation features, but "Pu" is the ultimate responsible for model "correctness".
- The indirect mention to other "P" in the development team when the designers communicate that models can be printed, published (for sharing purposes) or transformed into code, but the apparent inconsistency between this and the fact that the final model representation does not clearly communicate which parts are semantically verified, and which parts aren't (or cannot be, and why).

6. UML activity model produced

The following figure presents the UML activity model produced on the evaluation task:



References

1. Blackwell, A., Green, T.: Notational systems–the cognitive dimensions of notations framework. HCI Models Theories and Frameworks Toward a Multidisciplinary Science (2003), 103–134

2. de Souza, C. S., Leitão, C. F., Prates, R. O., da Silva, E. J.: The semiotic inspection method. In Proc. IHC2006, ACM. (2006), 148-157.

3. de Souza, C. S., Leitão, C. F.: Semiotic engineering methods for scientific research in HCI. Princeton: NJ. Morgan & Claypool. (2009)

4. de Souza, C. S., Prates, R. O., Barbosa, S. D. J.: A method for evaluating software communicability. In Proc. IHC1999, (1999) 17-19.

5. de Souza, C.S.: The Semiotic Engineering of Human–Computer Interaction. Cambridge, MA. The MIT Press. (2005)

6. Ferreira, J. J., de Souza, C. S., Cerqueira, R. Characterizing the Tool-notation-people Triplet in Software Modeling Tasks. In Proceeding of the 13th Brazilian Symposium on Human Factors in Computing Systems (IHC' 2014), Brazilian Computer Society, Foz do Iguaçu, Brazil, 2013. p. 31-40.

7. Green, T.; Blackwell A.: Cognitive dimensions of information artifacts: a tutorial. BCS HCI Conference (1998)

8. OMG, Unified Modeling Language (UML), V2.4.1- http://www.omg.org/spec/UML/2.4.1/Superstructure/PDF. Accessed in August, 2014.