



PANDA: prototyping using annotation and decision analysis

Jean-Luc Hak, Marco Winckler, David Navarre

► To cite this version:

Jean-Luc Hak, Marco Winckler, David Navarre. PANDA: prototyping using annotation and decision analysis. 8th ACM SIGCHI conference Engineering Interactive Computing Systems (EICS 2016), Jun 2016, Brussels, Belgium. pp. 171-176. hal-01712526

HAL Id: hal-01712526

<https://hal.science/hal-01712526>

Submitted on 19 Feb 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Open Archive TOULOUSE Archive Ouverte (OATAO)

OATAO is an open access repository that collects the work of Toulouse researchers and makes it freely available over the web where possible.

This is an author-deposited version published in : <http://oatao.univ-toulouse.fr/>
Eprints ID : 18847

The contribution was presented at EICS 2016 :
<http://eics.acm.org/2016/>

To cite this version : Hak, Jean-Luc and Winckler, Marco Antonio and Navarre, David *PANDA: prototyping using annotation and decision analysis*. (2016) In: 8th ACM SIGCHI conference Engineering Interactive Computing Systems (EICS 2016), 21 June 2016 - 24 June 2016 (Brussels, Belgium).

Any correspondence concerning this service should be sent to the repository administrator: staff-oatao@listes-diff.inp-toulouse.fr

PANDA: Prototyping using ANnotation and Decision Analysis

Jean-Luc Hak, Marco Winckler and David Navarre

Institute of Research in Informatics of Toulouse (IRIT), Interactive Critical Systems (ICS) team
University of Toulouse
31062 Toulouse Cedex, France
{Jean-luc.hak, winckler, navarre}@irit.fr

ABSTRACT

Prototyping is a core activity in User-Centered Design (UCD) process and is aimed at supporting iterations of design ideas until all users' requirements are met. Although many dedicated prototyping tools exist, we have found out that most of them lack features for the traceability of information that can be useful for driving the evolution of prototypes. In this paper, we presents a prototyping tool called PANDA which has been specifically conceived to investigate features for dealing with the evolution of prototypes. Herein we present a view at glance about the tool a more specifically annotations mechanisms that can be used for recording design choices, new requirements, fixing design typos, and so on.

Author Keywords

User interface prototyping; User-centered design; Human computer interaction; Annotation

INTRODUCTION

Prototypes feature a concrete (yet partial) representation of aspects of an interactive system and they can be used to explore many design alternatives before implementing the final product [2]. Prototypes are aimed to evolve incrementally toward products that better suit users' needs. For that, prototypes should provide a user interface description (covering the behavior, presentation and/or properties) that might have divers uses (such as assess design alternatives) along an iterative process. Moreover, as we shall see in Figure 1, prototyping cannot be dissociated from tool support because features provided by the tools determine how prototypes evolve into a final product.

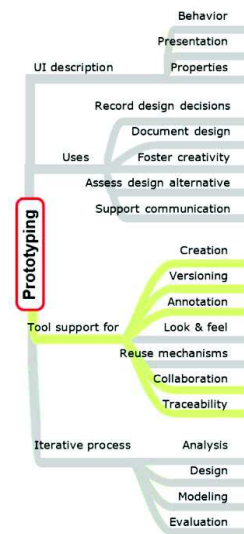


Figure 1. Dimensions covered by the prototyping activity.

Our research is concerned by the underlying process that drives prototyping activities including cycles of design and evaluation, communication support, and management of many prototypes versions. Despite the fact these activities are not directly related to the production of the actual prototype we suggest that they might have positive side-effects such as allowing to record a justification of design choices, linking the prototype with other artefacts used during the development processes and improving the communication among the design team. In order to investigate our hypothesis, we have developed a prototyping tool support called PANDA which stands for “Prototyping using ANnotation and Decision Analysis”. Herein we present a view of this tool and more specifically the annotations mechanisms that can be used for recording design choices, new requirements, fixing design typos, and so on. The paper starts with an overview of the state of the art and then the tool is illustrated with brief case study. The last part of the paper present the discussion and future work.

EXISTING PROTOTYPING TOOLS

There is a vast literature about prototyping tools which has inspired the development of commercial tools. We have screened publications about prototyping environments published from 1988 to 2014 in the main HCI conferences and 113 commercial tools available on the web. The

preliminary results available at Figure 2 and Figure 3 enable a brief comparison of characteristics present in existing tools, how they have evolved and what are the gaps that can provide insights for future research and development.

We can observe in the literature three main moments of interest in the history of prototyping tools, as shown by Figure 3. The first one, before 1995, coincides with the emergence of UIMS tools. The first tools mainly treated of high-fidelity prototypes, using mostly design elements from the final interface, and being strongly dependent on the hardware. The second period (1995-2005) is characterized by tools that start to support the life cycle of prototypes development and the emergence of function to support low-fidelity prototyping through sketching, the most emblematic ones being SILK and DENIM. This period was followed by an increasing interest in other ways to prototype interfaces and the inclusion of behaviour modelling which can be roughly dated after 2005, leading up to a third and more recent period with a substantial increase of commercial tools from 2007 to now.

| Feature | Percentage |
|---|------------|
| Annotations | 41.54% |
| Behavior Specification | 56.15% |
| Collaborative Working | 21.54% |
| Non-Programming Skills | 67.69% |
| Pen-Based Interaction | 5.38% |
| Preview Mode | 52.31% |
| Reuse Mechanism | 63.08% |
| Scenario Management | 8.46% |
| Support for Code Generation | 21.54% |
| Support for UCD interactive development | 6.92% |
| Support for Usability testing | 5.38% |
| Version Control | 19.23% |
| Widgets | 60.77% |

Figure 2: Features supported by prototyping tool

Figure 2 presents a finer analysis of features supported by prototyping tools. As we shall see, currently there is a lot of tools dealing with the creation of prototypes but only a few of them cover all the dimensions associated to the prototyping activity as described in Figure 1. Moreover, none

of them consider the prototype as an evolving artefact in the UCD cycle that should always match with users' needs [8]. More specifically existing prototyping tools lack of support for some features that we consider key for helping developers to make prototypes to evolve:

- **Annotations:** prototypes often contain information that is not embedded to the design but that can be represented through annotations [5, 8]. Annotations are important because they act as memos for changes that should be taken into account in the next iterations with the prototype. Moreover, several studies demonstrated that annotations could support effectively collaborative activities ranging from a better understanding of a document (in our case an electronic document representing a UI prototype) by reducing the time to read a document to a better collaboration, coordination and communication among stakeholders [3, 4, 11]. Indeed, annotations can be used to emphasize and retain important information over time [9]. Moreover, they can also be used as an easy way to actively engage people on a document [7]. Whilst annotations are not new their implementation is unequal depending on the tool. For example, Balsamiq allows the creation of markups annotations that can be showed/hidden, NinjaMock or UXPin allow the creation of discussion thread through a "comment and reply" system. We assume that as far as annotations are necessary for supporting the communication around the design, they should be attached to the prototype artefact somehow. Moreover, tools should be flexible to represent diverse styles of annotations from textual comments to references for external files.
- **Version control:** versioning is supported by less than 1/5 of tools, mainly web-based tools such as Alouka or HotGloo. The version management system were limited on those tools to a list of backup versions of the prototype that were stored on the cloud and only one tool (Concept.Ly) among the tools reviewed allowed the comparison of two versions of a prototype side by side.

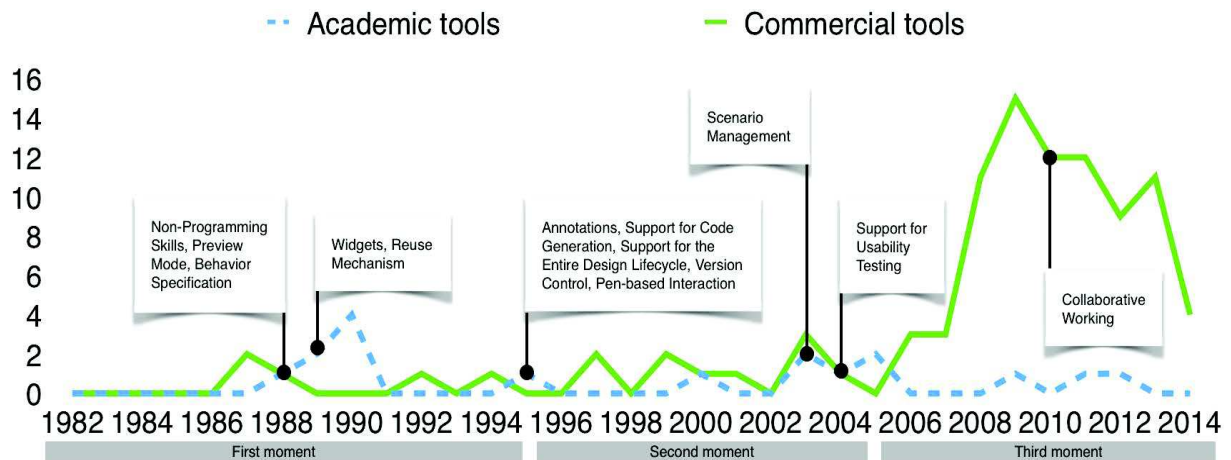


Figure 3. Milestones for the development of prototyping tools.

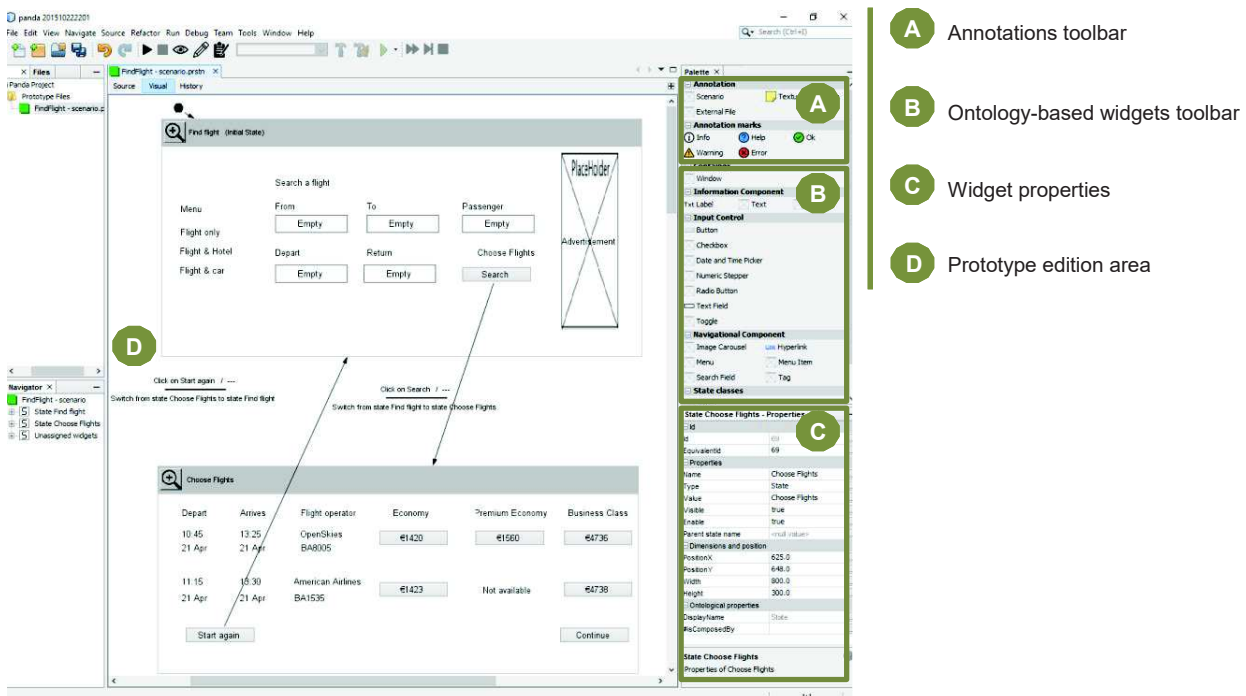


Figure 4. Interface of PANDA

- **Traceability of design choices:** none of the tools provides explicit mechanism for dealing with argumentation and traceability of design choices over prototypes. We would expect to be able to document the discussions made over the prototypes because they might contain useful information for the next iterations. However, we can still note that annotations used as a discussion thread can be used to trace modifications on a prototype that were discussed.

OVERVIEW OF THE TOOL PANDA

In this section we present the tool PANDA and how it can help designers to build prototypes and annotate them. We assume that annotations contain information that should be taken into account in the next iterations with the prototype. For the sake of simplicity we are following a single cycle of development process including the construction of an initial design, description of prototype behavior and placement of annotations. Therefore, functions for supporting versioning and traceability of design choices will not be presented here but they will be further discussed in the conclusion of the paper.

The presentation of PANDA in this section is based in on the construction of a prototype aimed at representing a simple booking flight Web application. We assume that at this point, the prototype we want to represent only covers the initial tasks of a booking activity which include searching for a flight and selecting one of the flight available.

Our flight booking prototype is depicted at Figure 4 which present the prototyping environment proposed by PANDA.

This environment is composed of different areas in which users have to interact to create a prototype.

PANDA was coded in Java using the framework Netbeans. It is part of the tool suite CIRCUS [1] which encompasses other tools for engineering interactive systems such as PetShop, SWCEditor, HAMSTERS and DREAMER.

Creation of an Initial Prototype

Like many existing tools, PANDA relies on a library of widgets that can be used to compose the design. For the creation of a prototype, we just need to drag and drop widgets available from the toolbar shown at Figure 4.B. The user places the different parts of the prototype in the drawing area (see Figure 4.D) to create a first representation of the initial user interface of the prototype. That toolbar contains widgets for describing the presentation aspect of the user interface (for example, containers, buttons, form fields, etc.), elements for annotating the prototype and elements for specifying the behavior (the so-called state classes).

One of the particularities of the tool PANDA is the fact that the toolbar is dynamic and it is automatically instantiated according to a domain specific Ontology. The underlying idea is that, according to the application domain of the prototype we want to build, developers will need different sets of widgets. Using an ontology to create a prototype provides several advantages for example, it can embed a specific semantic meaning for every element used to specify the prototype, the association of elements in the design can be constrained by rules specified in the Ontology and the properties associated to the elements can be formally tested.

Therefore, the first thing we should do when PANDA is launched is to import an external file containing an Ontology described in OWL. Once that file is loaded the toolbar of widgets is automatically generated based on the classes Figure 4.B and properties Figure 4.C defined in the Ontology. In our simple case study for flight booking we are instantiating an Ontology that contains Web-based widgets.

Specification of the Prototype Behavior

PANDA allows the description of both the presentation and the behavior of user interface prototypes. The description of the behavior is driven by executable automata that can be also used to simulate the execution of prototype. Figure 5 illustrate how automata are visualized in PANDA. A state describes the different widgets that are displayed by the system like in Figure 4.D and is represented by a rectangle with a grey header. The transitions between two states are represented with arrows Figure 4.D and Figure 5 and are defined by the triplet $\frac{\text{Event} / (\text{Condition})}{\text{Actions}}$. In the detailed view Figure 4.D, we can note which widgets can trigger the transition while the automaton view Figure 5 gives an overview of the behavior of the prototype. In the center of each states is the name of the corresponding state.

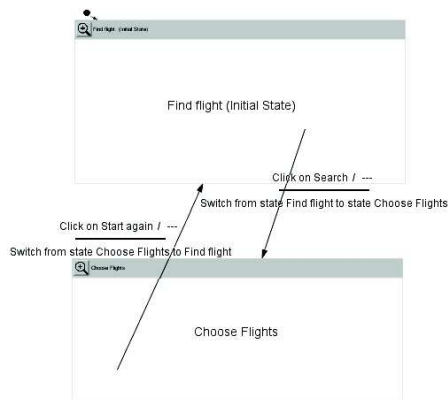


Figure 5. Specification of the prototype behavior

In our case study, we have associated a specific behavior to the two Web pages that constitute the design of the Web site: namely the “Find flight” page (which is also the initial state of that automata) and “Choose flights”. In the first state, a user enter his search criteria and in the second state, the system makes a research based on the search criteria and displays the matching results.

The automata describing the behavior of the prototype can be executed making the widgets interactive. In our example, when the state “Find flight” is active, clicking on the button “Search” will activate and display the state “Choose flights”. Though, the interactions in PANDA are on a high level of abstraction since it does not process data and the transitions handle only left-click events on widgets.

Placement of Annotations

We assume that once the prototype has been created, it can be presented to users or during meetings for discussions and

evaluation. It can be evaluated on both the representation of the dialogue and the presentation regarding the users’ needs, the client’s constraints and the requirements. The persons who participate to the evaluation and discussions can edit the prototype to express themselves through annotations or modifications of the prototype. The communication around the prototype can be worth noting since valuable information can emerge. For instance, we can note data on users’ tasks, feedbacks, suggestion of modifications, ideas or problem reports. Thus annotations can be effectively used as a support of communication and collaboration since they add information and refers to a concrete artefact (e.g. a widget, a window), they support the memorization, the planning and can be used as a support to draw the attention for future analysis [11]. PANDA proposes five types of annotation that are illustrated in Figure 6, as follows:

Graphic Marks (Figure 6.A) feature icons used to place hotspots over a prototype. The meaning of the iconic marks is arbitrary and they are meant to support quick recognition of areas that needs further attention and/or development. So far, we propose 5 types of marks: Info, Help, Ok, Warning and Error. They can be self-sufficient or they can be completed with another annotation to explicit the details.

Simple Texts (Figure 6.B) are, as the name indicates, textual annotations that can contain any kind of comment, question or suggestion of improvement.

Extended Texts (Figure 6.C) are similar to Simple texts but they can be used to create discussion threads around a subject. Indeed, doing so might promote collaboration and argumentation about a matter on the prototype. Extended Text annotations include a voting system so that we can associate a weight for comments that corresponds to the results of a discussions, for example with the users and/or clients. Indeed, those factor might have an influence on the decisions that could be made based on the annotations in later iterations of prototyping. We suggest that extended text can also be used as a planning tool featuring a checklist. However, other features should be included to fully support the planning like the description of the task, the deadline, the person in charge of the task [12] and the status of the advancement of the task.

Scenarios (Figure 6.D) are a type of annotation that refers to a list of tasks that can be performed with a prototype. A first use for this annotation is in replacement for the formal description of the behavior. Scenarios in PANDA are structured text. A scenario starts with an initial state and the keyword “Given”. Then a set of steps are described in each following line. A step define an action to perform. This action is described with a verb (e.g. “I click on”) and a target (e.g. “Search”). An action can be completed with data (e.g. “I type “Paris” in the field “From””). Lastly, the scenario ends with a final action made by the system indicating a final state.

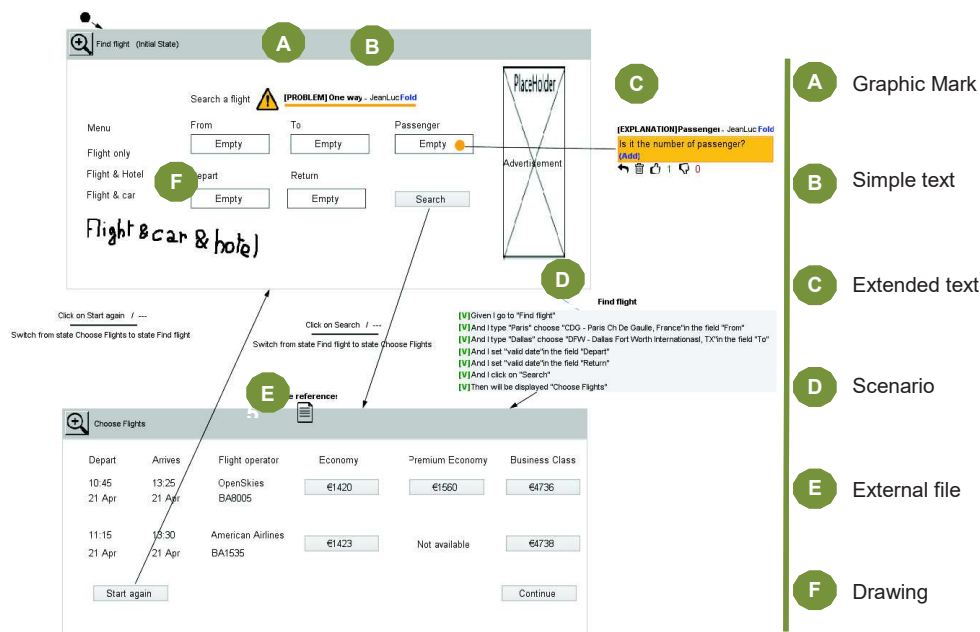


Figure 6. Overview of annotations supported by PANDA

External Files (Figure 6.E) are used to enclose any kind of document located on the computer directly on the prototype and put it where it is relevant. For instance, it can be an audio file, a video from a recorded usage of the prototype or a specification document. By clicking on the annotation, the Operating System opens the file associated.

Drawing annotations (Figure 6.F) are the more polyvalent since it allows the annotator to create any freehand drawing to represent anything. On top of that, annotating with a pen is less cognitively demanding than with a mouse and a keyboard [7] and those annotations stand out from the underlying document making them easier to find. A drawing can be used to show, illustrate an idea, write something or emphasize elements by underlining, circling similarly to annotation on paper.

All of these annotation can be placed on the prototype or next to it. The proximity of the annotation is one way to establish its connection with its target [6] like we have done for the graphic mark annotation and the simple text in Figure 6. However, we also developed an anchoring system that allows to create bullets to represent explicitly the target of the annotation as illustrated in Figure 6 with the extended text annotation.

Once annotations have been created, their management should not be neglected. Indeed, over time, the number of annotation can grow significantly and thus, making their exploitation harder. Moreover, transient annotations [6] are not relevant during the whole design process (e.g. a modification request handled or a planned task done) but it might be interesting to keep a record of it to understand the progress of the design process. As a consequence, annotations should be updated to take into account their status of advancement.

The management of annotations can also be facilitated thanks to the indexation of annotation. So we decided to store metadata of annotations like the last modification date, the author's name, the type of annotation, a tag etc.

DISCUSSION AND FUTURE WORK

This paper presents some preliminary results about the development of the prototyping tool PANDA. It is interesting to notice that PANDA feature some unique characteristic such as the support for the creation of interactive prototypes that include a formal description of the behavior, a large set of annotations types and a domain specific Ontology.

Indeed, prototypes created with PANDA are interactive thanks to automata (which specifies the behavior) and scenarios (which describe the sequences of steps to be executed). Scenarios is one of the many types of annotations supported by PANDA.

Annotations are well-suited for the collaborative work [11, 12]. As S. Bringay et al. described in [5], annotations can relate to several targets simultaneously. As for prototypes, annotations can not only be related to one object but they can also be associated to different parts of the prototype, and they can be on the automata or the widgets of the prototype. Once the prototype is completed, it can be annotated in various ways to encourage communication around the prototype and the collaboration.

The construction of prototypes based on an Ontology presents many advantages. First of all it determines the semantic of elements of the design. Moreover, by creating a domain/platform independent prototype, PANDA can help prototypes to evolve to other platforms. We suggest that by using transformation rules and manual fine tuning, it would

be possible to convert a user interface prototype described in using an Ontology into another by cascading the changes in the user interface as described in [10].

PANDA is still in development and existing features can be extended. As for annotations, they are limited in our tool at the moment, further customizable options are planned like the opacity, stroke color, width to enable highlighting for drawing annotations. The management of annotations could be useful for filtering annotations, sorting them, archiving them or simply process them. So, an indexation system based on metadata could be implemented. This indexation can be done from several point of view [12].

Future development on PANDA will address the traceability of design choices. Indeed, once a prototype has been evaluated, modifications can be made based on feedbacks collected and some feedback can be stored with annotations. At that point, choices can be made to reject or adopt suggestion of modification, to solve identified problems or to implement additional features.

In order to ensure the traceability of the evolution of the prototype, we will be examining version control features to store the different versions of the prototypes which includes successive versions and alternative versions of a prototype. Alternative versions are used to test different options of design and represent a potential evolution of the prototype while successive versions shows tested and validated choices.

We will also examine different ways to represent and visualize the information that could be integrated for the evolution of the prototype like important decisions and relationship between different versions in a timeline for instance since a prototype can be abandoned, merged with another or put on hold.

After that, we expect to initiate user tests on real case studies in the long term in order to observe if our contribution is useful for the prototyping process once PANDA will be usable enough. As for existing features, they will be used as a first material to perform empirical studies with designers and developers.

REFERENCES

1. Eric Barboni, Jean-François Ladry, David Navarre, Philippe Palanque, and Marco Winckler. 2010. Beyond modelling: an integrated environment supporting co-execution of tasks and systems models. In *Proceedings of the 2nd ACM EICS '10*. ACM, New York, NY, USA, 165-174.
2. Michel Beaudouin-Lafon and Wendy Mackay. 2002. Prototyping tools and techniques. In *The human-computer interaction handbook*, Julie A. Jacko and Andrew Sears (Eds.). L. Erlbaum Associates Inc., Hillsdale, NJ, USA 1006-1031.
3. Jean-François Boujut, Françoise Darses, Sylvie Guibert. Etude des annotations en situation collaborative de conception mécanique. Pascal Salembier, Manuel Zacklad. Annotations dans les documents pour l'action, Lavoisier, pp.127-152, 2006. <hal-00290132>
4. Ugo Braga Sangiorgi, François Beuven, Jean Vanderdonckt: User interface design by collaborative sketching. Conference on Designing Interactive Systems 2012: 378-387
5. Sandra Bringay, C. Barry, J. Charley. Annotations: A new type of document in the Electronic Health Record. Paper presented at the 2nd International Conference on Document Research and Development in Sciences, arts and business: DOCAM 2004, University of California, Berkeley, Etats-Unis, octobre 2004
6. Catherine C. Marshall. 1998. Toward an ecology of hypertext annotation. In Proceedings of the ninth ACM conference on Hypertext and hypermedia : links, objects, time and space---structure in hypermedia systems: links, objects, time and space---structure in hypermedia systems (HYPERTEXT '98). ACM, New York, NY, USA, 40-49.
7. Kenton O'Hara, Sellen A. A comparison of reading paper and on-line documents. In: Proceedings of the SIGCHI conference on human factors in computing systems, CHI '97. New York, NY, USA: ACM; 1997. p. 335-42
8. Thiago Silva, Jean-Luc Hak, Marco Winckler. A Review of Milestones in the History of GUI Prototyping Tools. Workshop on User Experience and User-Centered Development Processes 2015 (IFIP WG 13.2). p:1-12. University of Bamberg Press. Sanderson R, Ciccicarese P, Van de Sompel H: Designing the W3C open annotation data model. In Proc 5th Annu ACM Web Sci Conf - WebSci '13. New York, New York, USA: ACM Press; 2013:366-375.
9. Michele L. Simpson, S.L. Nist, Textbook annotation: an effective and efficient study strategy for college students J Read, 34 (2) (1990), pp. 122-129 <http://dx.doi.org/10.2307/40032053>
10. Marco Winckler, Vanderdonckt, J., Trindade, F., Stanciulescu, A. Cascading Dialog Modeling with UsiXML. International Workshop on the Design, Verification and Specification of Interactive Systems (DSVIS'2008). Kingston, Ontario, Canada, July 16-18 2008. Springer LNCS 5136. pp. 121-135
11. Manuel Zacklad. Annotation : attention, association, contribution. Annotations dans les Documents pour l'Action, Hermes science publications, pp.29-46, 2007. <sic 00180781>
12. Manuel Zacklad, Lewkowicz M., Boujut J-F., Darses F., Et Détienne F. (2003), Formes et gestion des annotations numériques collectives en ingénierie collaborative, in Actes des journées Ingénierie des Connaissances 2003, Laval