Gamification Technique and Checklist-Based and Technique For Inspection of Interaction Models: Additional Support

Adriana Lopes[1], Anna Marques[1], Williamson Silva[1], Simone Diniz Junqueira Barbosa[2], Tayana Conte[1]

[1] USES – Grupo de Usabilidade e Engenharia de Software
Universidade Federal do Amazonas (UFAM) Manaus, AM – Brazil
{adriana,anna.beatriz,williamson.silva,tayana}@ufam.edu.br

[2] Semiotic Engineering Research Group
PUC-Rio, Rio de Janeiro, RJ – Brazil.
simone@inf.puc-rio.br

USES Technical Report
RT-USES-2017-0021
November, 2017

Institute of Computing (IComp)
Federal University of Amazonas (UFAM)
Manaus, Amazonas 69077-000

ABSTRACT

This technical report describes the verification items developed for the diferente interaction modeling languages. We developed these items with the purpose of evaluating the consistency of the interaction models with the system user scenario. These items can be instantiated for other interaction modeling languages, as long as the elements have the same purpose in the interaction modeling. In addition, this technical report describes the material used to support the experimental studies of the two inspection techniques for different interaction models, called MoLVERIC Cards (MCards) and MoLVERIC Check (MCheck). MCards employs gamification elements to motivate professionals during the inspection. MCheck uses a checklist to guide the inspection.

1. INTERACTION MODELS

Different techniques support user-centered design, such as the creation of personas, task modeling, and prototypes (Paula, Barbosa & Lucena, 2005). However, users often encounter problems using interactive systems (Rogers, Sharp & Preece, 2015). In this context, Beaudouin-Lafon (2004) argues that one of the ways to improve the quality of interactive systems is to shift the focus from interface design to interaction design.

Interaction modeling refers to a set of principles, rules and properties that guide interface design and can be used by designers and developers to create interactive systems (BeaudouinLafon, 2004). According to Meixner, Paternò and Vanderdonckt (2011), interaction models can include a high level of abstraction for the development of interactive systems. Several languages for interaction modeling have been proposed (Paula, Lucena & Barbosa, 2005; López-Jaquero & Montero, 2007; Kim & Yoon, 2005). The OCD (Operation and Control Diagram) provides the representation of the interaction in terms of operations that can be performed by the user in system (Kim & Yoon, 2005). CTDM (Comprehensive Task Dialog Modeling) can be used to specify tasks, dialogues, and information about the system domain (López-Jaquero & Montero, 2007). MoLIC (Modeling Language for Interaction as a Conversation) allows the representation of the user-system interaction, where the designer can correct possible breakdowns in the communication from the designer to the user (Paula, Barbosa & Lucena, 2005). objectives in modeling the interaction with OCD, CTDM, and MoLIC.

Among the OCD, CTDM and MoLIC languages, MoLIC allows detailing the actions of the user and the system with associated contents, while OCD allows only the representation of the action that is performed, without the identification of who performs it. CTMD does not allow representing details regarding the actions that occur in the user-system interaction. MoLIC allows representing other alternatives of interaction, while OCD and CTDM do not have elements with this characteristic. We also identified more reports about experimental studies of MoLIC in the design of interactive systems (Sangiorgi & Barbosa, 2009; Silva, Martins Netto & Barbosa, 2005) compared to OCD and CTDM. For this reason, we chose th MoLIC language in this work to explore the use of interaction models.

2. VERIFICATION ITEMS DEVELOPED FOR THE DIFFERENT

We developed these items with the purpose of evaluating the consistency of the interaction models with the system user scenario. We find that it is possible to support the

identification of defects regarding user objectives through the elements: Scene (MoLIC), Operation (OCD) and Tasks and subtasks (CTDM).

We identify the following defects:

D1 – User objectives that are not represented (Omission);

D2 – User objectives inconsistent with the requirements (Inconsistency);

D3 – User objectives absent from the context of the requirements (Extraneous Information);

D4 – Incorrect descriptions defined in the model (Incorrect Fact);

D5 – Ambiguous User Objectives (Ambiguity); and

D5 – Different user goals cause ambiguity due to similar description (Ambiguity).

From these defects, we developed the following verification items:

• Are all user goals, described in the requirements/scenario information, represented in the interaction models? If not, report as an Omission defect - **Developed based on D1.**

• Are there user goals inconsistent with scenario requirements/information? If so, report it as an Inconsistency defect - **Developed based on D2**.

• Are there user goals that are not in the context of scenario requirements/information? If this is the case, report it as an Extraneous Information defect - **Developed based on D3**.

• User goals can be read as “At this time, you (user) can (or should) <verb + objects>?” For example: The user objective to register a student can be read as: “At this time, you (user) can (or should) Register student”. If not, report as an Incorrect Fact - **Developed based on D4**.

• Is it possible to get different interpretations in reading each user goals? If so, report as Ambiguity - **Developed based on D5**.

• Are there similar scenes? If so, also report as Ambiguity - **Developed based on D5**.

We find that it is possible to support the identification of defects regarding the user objectives that indicate how the user-system interaction occurs in the elements: Transition Utterance (MoLIC), System’s States and Responses (OCD) and Transitions (CTDM). We identify the following defects:

D6 - Direction of the arrows are incorrect (Incorrect Fact);

D7 - Incorrect arrows (Incorrect Fact);

D8 – Lack of arrows when necessary (Omission);

D9 - Arrows with content outside the context of requirements (Extraneous Information);

D10 - Arrows with content inconsistent with the context of the requirements

(Inconsistency);

D11 - Arrows with ambiguous content (Ambiguity);

D12 - Enunciator omitted (Omission);

D13 - Incorrect statement (Incorrect Fact);

D14 - Objectives of the user without the necessary arrows (Omission).

For the D13 and D14, we developed them specifically for MoLIC. From these defects, we developed the following verification items:

• Is the direction of the arrows correct in relation to the scenario requirements /information? If it is not, report it as an Incorrect Fact defect- **Developed on the basis of D6**.

• Are the correct arrows used? If it is not, report it as an Incorrect Fact defect – **Developed on the basis of D7**.

• Do the arrows represent necessary content? If not, report as an Omission defect - **Developed based on D8**.

• Is the content of the speeches in the context of the scenario requirements / information? If not, report as a Extraneous Information defect - **Developed on the basis of D9**.

• Is the content of the speech consistent with the requirements / information in the scenario? If not, report as an Inconsistency defect - **Developed based on D10**.

• Does the content of the speech provide multiple interpretations? If so, report as na Ambiguity defect - **Developed based on D11.**

• In the case of MoLIC - Do the utterances use the "u:" or "d:" enunciator? If not, report as an Omission - **Developed based on D12**.

• In the case of MoLIC - Was the correct speech enunciator used? Being "u:" for user and "d:" for designer. If not, report as an Incorrect Fact defect - **Developed based on D13**.

• Are there any omissions between the scenes? If this is the case, report as an Omission defect - **Developed based on D14**.

We find that it is possible to support the identification of defects for the elements used to indicate the next goal of the user from a particular action, as System Process (MoLIC), Memory Header (OCD) and Transition Labels: Start and Error (CTDM). We identify the following defects:

D15 - Lack of use of element to interpret user action (Omission).

D16 - Improper use of interaction for the result of the system process (Incorrect Fact).

D17 - Lack of feedback to the user during the system process (Omission).

D18 - Failure to provide user rupture recovery during interaction (Omission).

From these defects, we developed the following verification items:

• Was element used to interpret a required user action in case of system feedback? If no, report as an Omission defect- **Developed based on D15**.

• After processing the system, are the appropriate responses used? If not, report it as na Incorrect Fact defect - **Developed based on D16**.

• Has feedback been used on the system's processing, in moments like downloading files? If not, report as an Omission defect - **Developed based on D17**.

• Have rupture recovery been used for the user? If not, report it as an Omission defect - **Developed based on D18**.

We find that it is possible to support the identification of defects for the elements used to indicate the beginning and end of a certain action, such as Opening Point and Closing Point (MoLIC) and Initial State and Final State (CTDM). We identify the following defects:

D19 - Lack of use of the elements to demonstrate the beginning and end of the interaction (Omission).

D20 - Inappropriate use of elements to represent other objectives (Incorrect Fact).

From these defects, we developed the following verification items:

• Have the elements been used to demonstrate the beginning and end of the user-system

interaction? If not, report as an Omission defect - **Developed based on D19**.

• Are the elements that represent the beginning and end of the user-system interaction

used correctly? If not, report it as an Incorrect Fact defect - **Developed based on D20**.

We find that it is possible to support the identification of defects for the elements used to indicate the as the interaction may occur in relation to a given user goal, such as Signs, Utterances and Dialogues (MoLIC); and State Header (OCD). We identify the following defects:

D21 - Lack of use of the elements to demonstrate how the interaction should occur in relation to the objectives that are accessed in the system and the responses of the system (Omission).

D22 - Lack of use of the elements to demonstrate how the interaction should occur in relation to the objectives that are accessed in the system and the system responses (Incorrect Fact).

From these defects, we developed the following verification items:

• The elements used to indicate how the interaction should occur in relation to the goals that are accessed in the system? If not, report as an Omission defect - Developed based on D21.

• Do the elements used to indicate how the interaction should occur in relation to the user's objectives have the appropriate answers? If not, report as an Omission defect - Developed based on D21.

• Do the elements that represent how the interaction should occur in relation to the user's objectives have been used correctly? If it is not, report as an Incorrect Fact defect - Developed based on D22.

We find that it is possible to support the identification of defects for the elements used to indicate the opportunity to change the objective in the interaction at any time, being the Ubiquid Access (MoLIC). We identified the following defects:

D23 - Inconsistent use of element in relation to requirements for user action (Inconsistency).

D24 - Incorrect use of the element to demonstrate how the user can achieve other goals at any time in the system (Incorrect Fact);

From these defects, we developed the following verification items:

• Is the element representing the opportunity for the user to change the goal in the interaction at any time used consistently with the requirements? If not, report as na Inconsistency defect - **Developed based on D23.**

• Is the element representing the opportunity for the user to change the goal in the interaction at any time used correctly? If it is not, report as an Incorrect Fact defect - **Developed based on D24.**

From these verification items, inspection techniques can be developed for OCD, CTDM and MoLIC interaction models. However, we note that these verification items can be adapted to other interaction modeling languages that have elements for the same purpose.

3. FORMS FOR THE PARTICIPANTS TO REPORT THE IDENTIFIED DEFECTS

- Forms for the participants to report the identified defects with the MCards (used in the Feasibility study and Crossover study)

Form of Defects – MoLVERIC Cards

NAME: _____ Starting time: _____ Finishing time: _____

Type of Defect	Points and Code	Description of Defects

- Forms for the participants to report the identified defects with the MCheck (used in the Feasibility study and Crossover study)

Form of Defects – MoLVERIC Check

NAME: _____ Starting time: _____ Finishing time:

Type of Defect	Description of Defects

REFERENCES

Barbosa, S. D. J., & Paula, M. G. (2003). Designing and Evaluating Interaction as Conversation: a Modeling Language based on Semiotic Engineering. Proceedings of the 10th Interactive Systems. Design, Specification, and Verification Workshop (DSV-IS), 16–33.

Beaudouin-Lafon, M. (2004). Designing Interaction, not Interfaces. Proceedings of the Working Conference on Advanced Visual Interfaces (AVI '04), 15-22.

Kim, H., & Yoon, W. (2005). Supporting the Cognitive Process of User Interface Design With Reusable Design Cases. International Journal Human-Computer Studies, 62(4), 457 - 486.

López-Jaquero, V., & Montero, F. (2007). Comprehensive Task and Dialog Modelling. Proceedings of the 12th international Conference on Human-computer Interaction: Interaction Design and Usability (HCI'07), 1149-1158.

Meixner, G., Paternò, F. & Vanderdonckt, J (2011) Past, Present, and Future of Model-Based User Interface Development. i-com – Journal of Interactive Media, 10(3), 2-11.

Paula, M. G., Barbosa, S. D. J., & Lucena, C. J. P. (2005). Conveying Human-Computer Interaction Concerns to Software Engineers through an Interaction Model. Proceedings of the 2005 Latin American Conference on Human-Computer Interaction (CLIHC '05), 109-119.

Rogers, Y., Sharp, H., & Preece, J. (2015). Interaction Design: Beyond Human-Computer Interaction. (4th ed.). John Wiley & Sons.

Sangiorgi, U., & Barbosa, S. (2009). MoLIC Designer: Towards Computational Support to HCI Design with MoLIC. Proceedings of the Symposium on Engineering Interactive Computing Systems, 303-308.

Silva, B., Martins Netto, O., & Barbosa, S. (2005). Promoting a Separation of Concerns via Closely Related Interaction and Presentation Models. Proceedings of the CLIHC - El Congreso Latinoamericano de la Interacción Humano-Computadora, 170-181.