Privacy and Security in Multi-modal User Interface Modeling for Social Media

Conference Paper · October 2011

DOI: 10.1109/PASSAT/SocialCom.2011.49 · Source: DBLP

Mohamed Bourimi
DB Systel

Ricardo Tesoriero
University of Castilla-La Mancha

Fatih Karatas
Universität Siegen

2011 IEEE International Conference on Privacy, Security, Risk, and Trust, and IEEE International Conference on Social Computing

Privacy and Security in Multi-modal User Interface Modeling for Social Media

Mohamed Bourimi*, Ricardo Tesoriero†, Pedro G. Villanueva†, Fatih Karatas*, Philipp Schwarte*
*IT Security Management Institute
University of Siegen
Siegen, Germany
Email: bourimi;karatas;schwarte@wiwi.uni-siegen.de
†Computing Systems Department
University of Castilla-La Mancha
Albacete, Spain
Email: ricardo.tesoriero;pedro.gonzalez@uclm.es

***Abstract*—This paper addresses privacy and security issues regarding the modeling of multi-modal user interfaces for social media applications. The proposed approach describes how privacy and security concerns are modeled from the user interface perspective, and how this model is related to a four layer conceptual framework for developing multi-modal and multi-platform user interfaces. The approach also explains how to adapt these models to the development of social media applications. Finally, we use this proposal to model the SocialTV case of study as an example of a social media application to show its feasibility.**

***Index Terms*—Usability; user interface design and modeling; privacy and security; awareness and affordance; SocialTV;**

I. Introduction

Social media is not confined to desktop computers; actually, most social applications, such as Facebook, Twitter, etc. are used in different hardware and software platforms. Due to the importance of social media in peoples life, the accessibility of this information through different modalities (vocal, gestural, tangible, etc.) is an important issue to be addressed by social media applications in order to reach disabled people.

Usually, most of security and privacy concerns are boarded in terms of data access. However, from the UI perspective, there are some issues that are not related to the domain model, but they have a great impact on the UI design. The awareness and affordance are two issues that are related to the UI and are independent of the domain model of the application.

For instance, the resource sharing is a common situation in social media applications. Actually, some museums allow virtual and physical visitors to access multi-media resources such as, documents, music, audio, video, etc. Although from the domain model perspective, the resource sharing can be reduced to a permission assignment problem; from the UI perspective, it can be solved in different ways.

A basic approach to solve the problem is exposing visitors a complete list of all the resources that are available in the media library. If a visitor tries to access a resource that is not allowed to access (i.e. some resources are Pay per View), he/she get an error notification.

Although from the security perspective the solution seems to be enough because the visitor is not able to access a restricted resource; from the UI perspective it is not enough because the visitor did not perceive any affordance or awareness about forbidden resources.

An alternative approach taking into account the affordance of the UI may list only the resources that the visitor is allowed to access, the rest of the resources remain invisible. Although this approach is the same from the security perspective; it is more attractive from the UI design perspective because visitors are not induced to make mistakes.

A more sophisticated solution may involve awareness where visitors are able to see all the resources in the media library, but there is a distinction between those that are able to be accessed from those that are not. Again from the security perspective the solution only requires a small modification that introduces the attributes that can be observed from resources that are not accessible. However form the UI perspective, this solution provides awareness and affordance.

So far we have presented three different approaches to solve the resource access problem that are inherent to the UI view of the system. As result of this analysis we can conclude that the security and privacy issues are not only related to the domain model, they also should be addressed at the UI. Thus, this article addresses the modeling of security and privacy concerns from the UI perspective.

To carry out this task we have based our work on the models defined by the UsiXML [1] User Interface Description Language (UIDL) that allows designers to develop multimodal and multi-platform UIs. Thus, this paper proposes an extension to the language that is used to model security and privacy issues on the system.

The article starts by addressing related work and describing the UsiXML UIDL. Then, it exposes the extension for UsiXML to model privacy and security issues. Afterwards, we explain how to model the SocialTV application using these models in order to expose the relationship between UI models and security and privacy models. Finally, we expose conclusions and future work.

978-0-7695-4578-3/11 $26.00 © 2011 IEEE
DOI

IEEE computer society

II. Related Work

Since usability is a prerequisite for security and privacy, it is part of a major effort to balance and improve their design in the UI by considering usability aspects. One of the most disregarded and critical topics of computer security has been and still is, the understanding of the interplay between usability and security [2]. From the security perspective, large amounts of scattered personal data lead to information overload, disorientation and loss of efficiency. This often results e.g. in not using security options offered by the application. From the usability perspective, UIs should address privacy and security since mostly end-users have e.g. to balance functionality and their security as well as privacy preferences by using several mechanisms built into the system [3]. In [2], Karat et al. argue that although many Human-Computer Interaction (HCI) techniques are general, there are unique aspects in the design of security and privacy systems and mention that:

- End-users are primary task oriented and security and privacy are seen as complementing and are not the main goal. Even though end-users would like that security and privacy mechanisms to be as transparent as possible, they also want to be in control of the situation and understand what is happening (e.g. by visualization means in the UI),
- Security and privacy mechanisms are historically designed with a highly technically trained end-user in mind where the reality shows that end-users have limited skills according to the respective scenario,
- Usability issues can have a higher negative impact for security and privacy applications than for any other kinds of systems, and
- End-users have to be able to easily adapt security and privacy solutions to accommodate changes.

Especially in the context of social media centered settings, there is a need for some degree of user's information disclosure in order to achieve the intended collaboration social goals [4]. According to Shneiderman et al., ”an extrapolation of current trends leads to the suggestion that most computerbased tasks will become collaborative because just as most work environments have social aspects” [3]. Social interaction is mostly supported by different categories of collaborative and social software/hardware solutions. These solutions have to fulfill multi-user requirements (e.g., social, context, and workspace awareness) and are consequently characterized by complex scenarios supporting these requirements in the respective domain. Often, this complexity is reflected in the user interface (UI) [5].

A very deep analysis of privacy-related literature with this respect considering different perspectives (e.g., Computer-Supported Cooperative/Collaborative Work, HCI, psychological, and sociological perspectives etc.) can be found in [6]. Based on further literature there, Boyle cites that users are often cautious about how the system handles their privacy/security and are afraid that their mistakes will affect their reputation. Furthermore, according to the context, people do not necessarily want to reveal details about their current work tasks to other people. This is often in conflict with the role of awareness provision which intends to involve knowledge about various things as who is in the social media environment, what is s/he working on, and what s/he is doing [7]. Related to this, Boyle and Greenberg state in [8] that there are privacy and awareness trade-offs; and generally two problems are associated with providing awareness: (1) privacy violations and (2) user disruption.

Many researchers from research areas cited above generally agree on that security and privacy issues arise due to the way systems are designed, implemented, and deployed. However, contemporary related work does not address modeling security and privacy by designing UIs earlier in the (interaction) design process. Security and usability research for developing usable (psychologically acceptable) security mechanisms is a young research field, which depends on the context in which those mechanisms have to be used [2]. We argue that related work leave therefore room for considerable improvement of how such systems can support an usable and secure user experience by modeling as we show in the following. The novelty of our work consists in answering the question: How to link together usability and security in UI elements by means of Meta models?

III. The UsiXML UIDL

The goal of this work is the definition of a model that copes with the introduction of security and privacy issues into the UI modeling. To introduce these issues in a concern independent way, the best alternative is using the Cameleon Reference Framework (CRF) [9] to develop UIs. The Figure 1 presents a simplified version of the framework showing the development process divided into 4 steps according to the level of abstraction of the UI being addressed:

- The **Tasks & Concepts (T&C)** layer describes tasks to be carried out by users, and the domain-oriented concepts required to perform these tasks.
- The **Abstract UI (AUI)** layer defines the abstract containers and the individual components [10] that will represent the artifacts on the UI. Containers are used to group subtasks according to various criteria (e.g., task model structural patterns, cognitive load analysis, and the identification of semantic relationships). Individual component represents an artifact that describes the behavior of a UI component in a modal-independent way (navigation, task performance, etc.). Thus, the AUI model abstracts the CUI model with respect to the interaction modality.
- The **Concrete UI (CUI)** concretizes an AUI for a given context of use. It uses Concrete Interaction Objects (CIOs) [11] to define the widget layouts and the interface navigation. It abstracts a FUI into a UI definition that is independent of any computing platform. The CUI can also be considered as a reification of an AUI and an abstraction of the FUI with respect to the platform.
- The **Final UI (FUI)** represents the operational UI running on a particular computing platform either by interpreta-

tion (e.g., through a Web browser) or by execution (e.g., after code compilation).

Thus, security and privacy issues can be modeled independently from the UI and then can be related to each layer accordingly.

There are many UIDL candidates that follow the CRF. For instance: MariaXML[1], UIML[2], UsiXML[3], or XIML[4]. We argue that we could select any of these alternatives, but we have chosen the UsiXML [1] because it presents a set of models that can be described in terms of the MOF[5] language allowing the definition of model-to-model (M2M) transformation and model-to-text (M2T) transformations that enable the definition of a MDA where the Security and Privacy model can be easily introduced to the architecture as new concerns.

The following set of models is defined in UsiXML to support the conceptual modeling of UIs, and to describe UIs at various levels of abstractions:

- The **taskModel** describes the task as viewed by the end user interacting with the system. It represents decomposition of tasks into subtasks, and temporal relationships among tasks. CTT [12] can be used as a task model.
- The **domainModel** describes the classes of objects manipulated by a user while interacting with a system. Typically, it could be a UML class diagram, or an entity-relationship-attribute model.
- The **mappingModel** contains a series of related mappings between models, or elements of models. It gathers the inter-model relationships that are semantically related (reification, abstraction and translation)
- The **contextModel** describes three aspects of the applications context of use where an end user is carrying out an interactive task using a specific computing platform on a surrounding environment.
- The **auiModel** describes the UI at the abstract level as previously defined.
- The **cuiModel** describes the UI at the concrete level as previously defined.
- The **contextModel** describes three aspects of the implementation: the user, the environment and the platform of the application. Thus, the context model wraps the other models to specify the deployment characteristics of the application.

IV. Security and Privacy Models

This section exposes how to model security and privacy concerns, as well as how these models are related to the four layers of the model defined by the UsiXML approach.

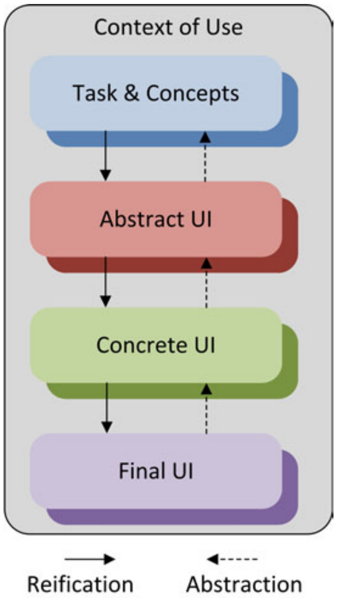

Fig. 1. Four layer conceptual framework

A. Security model and Privacy models

There are various security and/or privacy models with different degrees of abstraction. However, many of these models suffer from lack of applicability in our context, which consists in supporting security and privacy requirements modeling in the UI at different abstraction levels at the same time. Most of conceptual security and privacy models are focused on technical issues/level. Because of this, we base in this work on adapting the PriS approach [13] to our purposes in this work. The PriS conceptual model (s. Figure 2) targets at earlier addressing privacy and security requirements (e.g., authentication, authorization, anonymity etc.) in the system design, which corresponds to our model-driven approach.

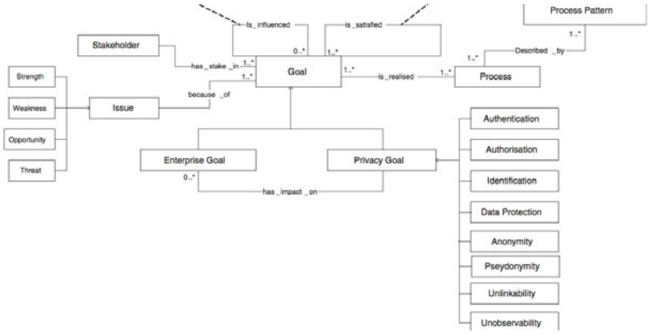

Fig. 2. PriS conceptual model (from [13])

The main strength of PriS is that both, security- as well as privacy-oriented technologies are considered.

The limitation of mapping security solutions by also considering (competing) privacy goals in the system design is

[1] MariaXML: http://giove.isti.cnr.it/tools/Mariae/
[2] UIML: http://www.uiml.org
[3] UsiXML: http://www.usixml.org
[4] XIML: http://www.ximl.org
[5] The Metaobject Facility: http://www.omg.org/mof/

addressed.

Our PriS-oriented security metamodel is depicted in Figure 3. There we took into account the most relevant security goals we need for our scenarios for now (e.g., authentication, authorization, anonymity etc.).

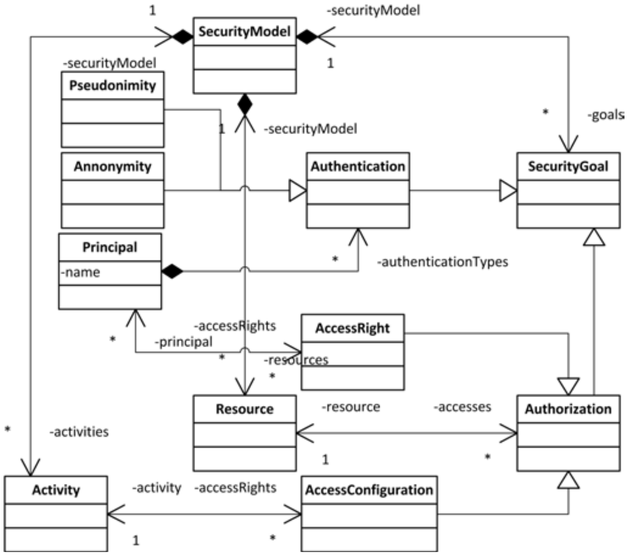

Fig. 3. PriS-oriented security metamodel

B. Mapping model

The mapping metamodel for our extension related the Task metaclass defined at the Task model of UsiXML to the Activity metaclass of the Security metamodel we have defined in the previous section.

Besides, it relates the Class metaclass of the domain metamodel to the Resource metaclass of the security metamodel.

Thus, security and privacy issues are integrated into the structure and behavior of the system.

The traceability among the rest of the models is warranted by definition because of the Cameleon Conceptual Framework.

V. The SocialTV Case of Study

This section describes the SocialTV scenario that needs usability as well as privacy and security consideration. It also defines the domain, task, security, privacy, and abstract and concrete user interface models; and how privacy and security models are related to them.

A. The SocialTV scenario

SocialTV mainly targets the integration of software/hardware to support social activities for either synchronous real-time interaction schemes or those of asynchronous nature. For synchronous real-time interaction, a good example is allowing chatting with friends or family while watching the same TV content. Asynchronous interaction for instance is mostly represented e.g., via comments or ratings related to a TV episode or movie. Thus the SocialTV presents a completely different scenario where new multi-user requirements related to media consumption have to be fulfilled while considering usability and privacy aspects.

Based on previous work we performed involving many experts and end-users [14], a list of usability and security related requirements were identified. The SocialTV prototype used in the lab and field trials was based on the extension of the Web-based cooperative system CURE [15] to support TV and video content diffusion capabilities. The end-users were able to benefit from collaborative functionalities while interacting with other users during a TV program. For instance group chat and private, instant messaging, and leaving comments in wiki pages were supported. With respect to privacy, the prototype allowed for centralized as well as decentralized interaction social settings as described in [16]. In summary, the identified (change) requirements with respect to the designed solution

- should reflect realistic SocialTV situations (R1).
- allow for flexible parallel interaction of the involved people (R2).
- be flexible in terms of costs emerging from adaptations to new situations and tests (R3).
- should thereby support secure and privacy-preserving interaction (R4).

To fulfill our requirements R1-R4 we extended the WallShare [17] platform. WallShare provides users with a collaborative multi-pointer shared desktop that is projected on a wall, or displayed on a big screen (R1, R2). Pointers are controlled through mobile devices, such as PDAs, smartphones, tablet PCs, etc. using dragging gestures over the mobile device screen (R2). Here, the mobile devices act as remote controls that can be adapted easily to new situations (R3). The system allows users to upload, or download resources to/from the shared desktop using a pointer that is controlled from users mobile devices by the means of gestures over the screen by considering access rights according to the Idemix anonymous credential system (R1-R4). Idemix [18] is a proof-based credential system, which could support such access to SocialTV adult content by only providing a proof, so that the person who is asking for such access has only to proof that he/she is over 18 years. In contrast to other credential systems, Idemix does not send attributes, which could lead to linkability (R4).

From the interactive perspective, according to the taxonomy described in [19], WallShare is a distributed user interface (DUI) represented by a multi-display ecosystem composed by Inch and Perch scale size displays that define a few-few social interaction relationship among users. From the collaborative perspective the system provides users with face-to-face collaboration, in the same space, at the same time) (R3).

Figure 4 shows a prototype realized by extending the WallShare platform. The Figure shows people navigating in a SocialTV environment by using their own mobile devices as remote controls. However, the system allows other groups to see the same content with this group together which was required in our case to better study privacy aspects (e.g., listening to people's impressions while carrying out the tests).

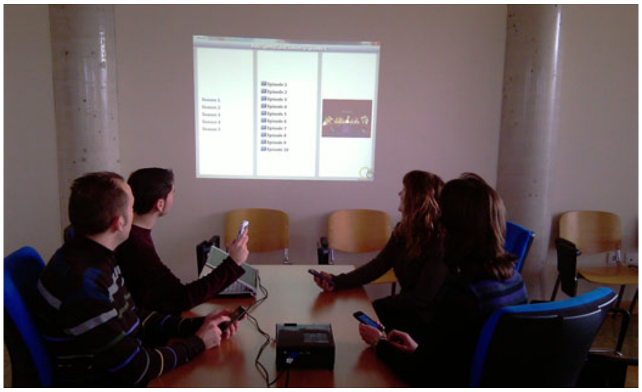

Fig. 4. SocialTV Scenario

The description of the application is detailed in the next sections.

B. Tasks & Concepts

The Task & Concepts layer of the framework is defined by two models: the domain model and the task model. While the domain model describes the concepts that will be manipulated by the SocialTV UI, the task models describe the interactions that will be carried out by the users of the system.

The notation we used to describe the domain model is the UML Class diagram. The domain model depicted in Figure 5 describes the most relevant parts of the SocialTV system regarding security and privacy issues. The **SocialTV System** is composed by a set of **Resources** and **Users**. While Users are defined by 4 attributes: **id**, **name**, **birthdate**, **subscription**, **Resources** are described in terms of **name**, **ageRating** and **Subscription**. Note that although the attributes defined by **Resource** and **User** are related (**subscription**, **birthdate** and **ageRating**) there is no explicit relationship between users and resources. This relationship is defined in a separate model leveraging the level of reuse in the system.

The notation we used to describe the task model is CTT. The task model depicted in Figure 6 describes the most relevant parts of the SocialTV system regarding security and privacy issues. Thus, a **SocialTVProcess** allows users to **NavigateOnResources** and **ManipulateVideo**. While the navigation task allows users to select a video by browsing on categories, the manipulation task allows users to control the video reproduction. As result of the analysis of this layer of models we can conclude that the domain model defines the data model of the system and the task model defines the behavior of the interaction system improving the decoupling of concerns at this level of abstraction.

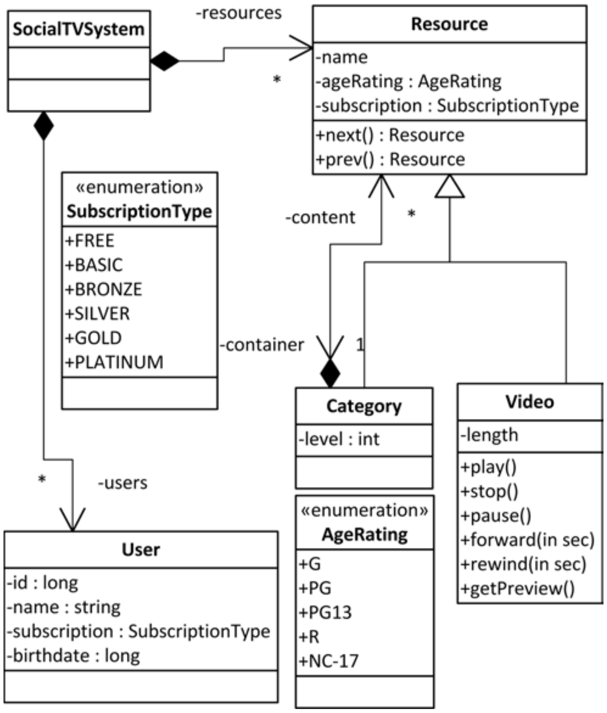

Fig. 5. SocialTV domain model

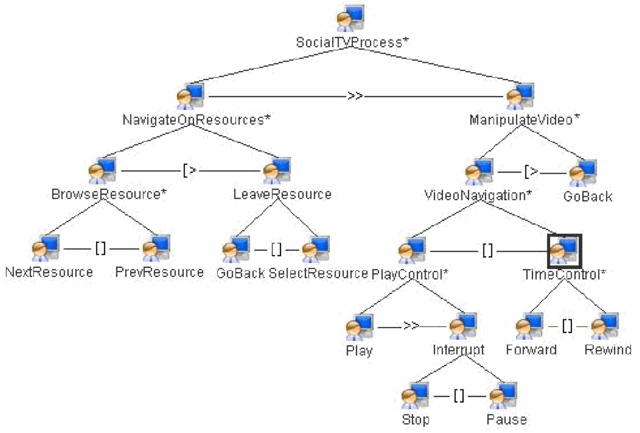

Fig. 6. SocialTV task model

C. Privacy and Security

Figure 7 shows the security model for our SocialTV scenario addressed in this work based on the metamodel we presented in Figure 3.

In this case, we use Idemix for both, anonymous authentication as well as authorization (access rights configuration and enforcement).

D. Abstract User Interface

The abstract user interface model describes the UI in a model independent way. For instance, the same model can be used to describe a Graphical User Interface GUI or a Vocal User Interface. The AUI is described in terms of containers and components that are characterized by facets (input, output control and navigation).

The SocialTV AUI model is depicted in Figure 8 where we can see that the **SocialTVUI** is composed by two main containers: the **ResourceBrowser** and the **VideoPlayer**. The

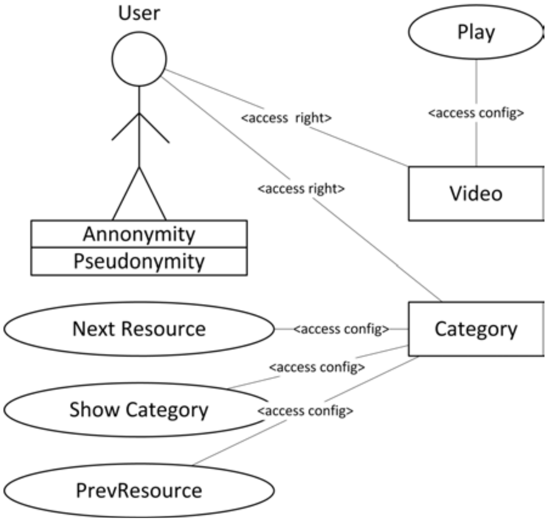

Fig. 7. SocialTV security model

first one is in charge of providing SocialTV users the ability to browse information. The second one is responsible for controlling the Video player. As the SocialTV system is defined as DUI, we added two containers on both, the **ResourceBrowser** and the **VideoPlayer**, representing the **Wall** and the **Mobile** device.

From the video player perspective, the **VideoPlayerWall** is composed by the **Viewer** output component that is in charge of playing the video on the shared surface; and the **VideoPlayerMobile** that is composed by the following set of control components: **Play**, **Stop**, **Pause**, **Forward**, **Rewind** and **GoBack** to control the video playback. And from the resource browser perspective, the **ResourceBrowserWall** is composed by 3 output components: **Name**, **Rating** and **Subscription**, to show resource information on the shared display; and the **ResourceBrowserMobile** that is composed by two navigation components (**Next** and **Previous**) to navigate among categories sharing the same parent category, and two control components (**Back** and **Select**) to select a category or go back to the parent category, if any.

Thus, input, output, navigation and control facets of components can be linked to the security and privacy models in order to represent the customization of the UI components behavior.

E. Concrete and Final User Interfaces

Due to space reasons, the Concrete UI model will be exposed jointly with the Final UI model because, from the security and privacy perspective, there is no conceptual difference between them.

The Final UI for the SocialTV system is defined as a Distributed User Interface (DUI) [20], [21], [22] based on the Model View Controller architecture where most of the view components are displayed on the shared screen (see Figure 10), and the controller components are located into the mobile devices (see Figure 9).

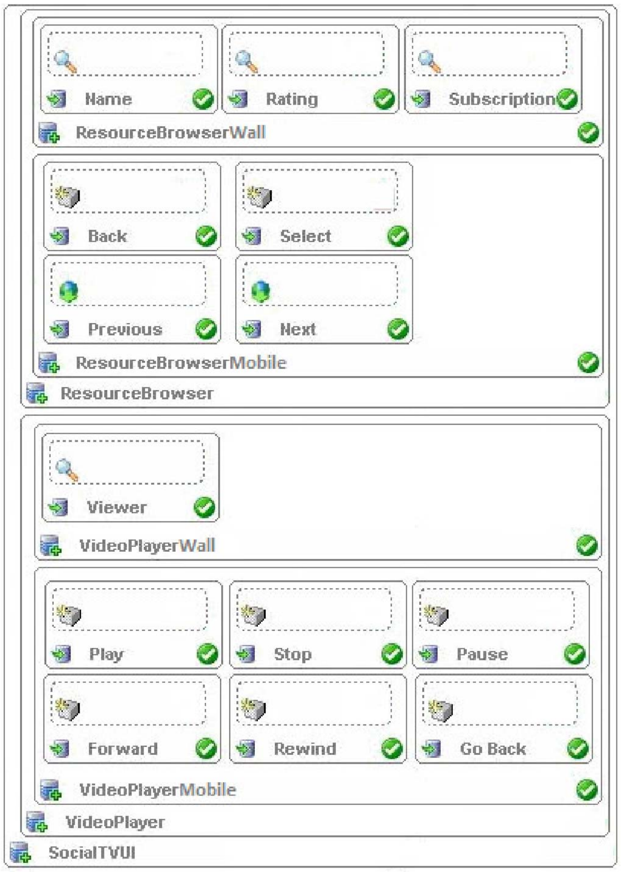

Fig. 8. SocialTV abstract user interface model

To show how security and privacy issues are related to the CUI, we will define two users with different characteristics and explain how the presence of users in the system affects the UI. In this case, for the sake of simplicity we will only show the effects on the shared display.

Suppose that we define two users: John (25 years old, silver subscription) and Peter (7 years old, no subscription).

When Peter joins a SocialTV session, the shared screen shows the upper screen of the Figure 10.

Let us analyze why the system is showing this screen. It shows the Winnie the Pooh category because it is able to perceive that Peter is 7 years old and he is alone in the session. Besides, the system shows that Season 1 is the only season he is able to navigate because he has no subscription.

Now suppose that John joins the session. Thus, Peter is not alone anymore, and the system shows the lower screen on Figure 10.

Let us analyze why the system updates the screen. First, PG-13 movies can be watched by Peter because an adult is with him. Besides, as John has a Silver subscription they are able to see more seasons.

The UIs depicted on Figure 10 shows both UI properties that are linked to two UI issues: affordance and awareness. The affordance is expressed by showing the categories that are available according to the age of the user/s; and by showing the upgrade option on seasons that may be accessible if the user performs a subscription upgrade. On the other hand, the awareness is expressed by disabling seasons.

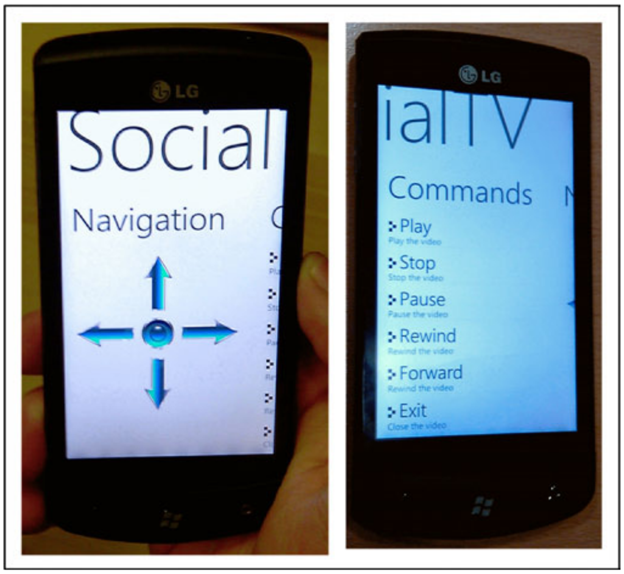

Fig. 9. SocialTV final user interface model (mobile device part)

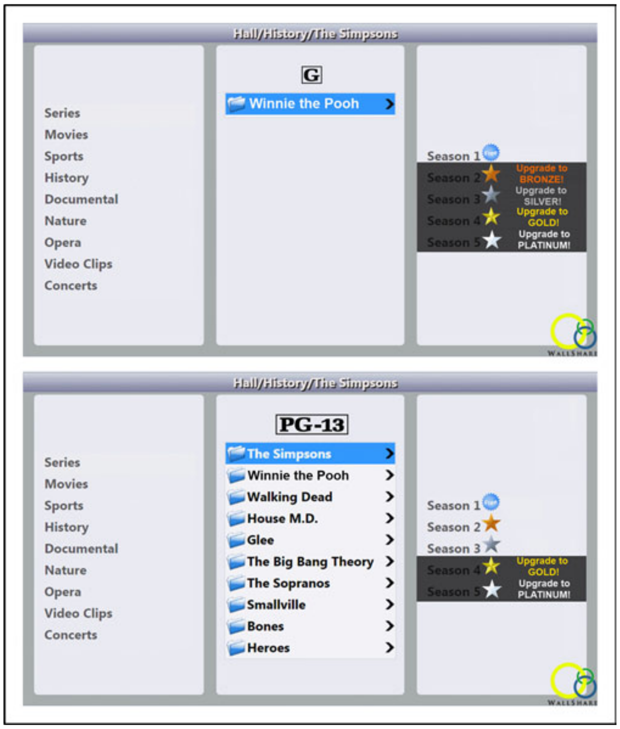

Fig. 10. SocialTV abstact user interface model (shared display part)

F. Mapping

The mapping model is straightforward among elements at different models. It is captured by the Mapping model where we link the Resource at the Security model to the Resource defined at the domain model. Besides, we have to link the Task to Activities.

VI. Conclusion and Future Work

In this paper, we presented a novel approach that takes together the consideration of UI and security metamodels for the development of multi-platform and multi-modal UIs. We suggested an extension of the UsiXML UIDL for addressing security issues in the UI metamodeling by presenting a premature security metamodel for addressing security goals. Both are linked together within a mapping model. For instance, for explicit authentication a password UI string field should not show the password and permit copy&paste. Another example is carrying out in other steps by using anonymous credential systems such as Idemix in the background without any interaction at the level of the UI.

This work goes beyond the contributions of related work since (1) approaches in the UI metamodeling and security research areas focus more on specific solutions in their own fields, and (2) the consideration of merging results from both fields is still not mature at metamodel level. With this approach, we hope to have carried out the first step in this direction.

Future work will target refining the security metamodel towards covering more privacy and security goals like unlinkability, unobservability, etc. As an approach to ensure fulfilling of multi-lateral security and usability as well as other competing non-functional requirements at the level of development lifecycle, the AFFINE methodology [23] will be applied (and enhanced, if necessary) along the further lifecycle of this approach. We hope from this to reach better consideration of non-functional requirements trade-offs in general in the whole development process.

Acknowledgment

This research has been partially supported by the Spanish CDTI research project CENIT-2008-1019 and the regional projects with reference PPII10-0300-4174 and PII2C09-0185-1030.

References

[1] Q. Limbourg, J. Vanderdonckt, B. Michotte, L. Bouillon, M. Florins, and D. Trevisan, “Usixml: A user interface description language for context-sensitive user interfaces,” in *In proceddings of the ACM AVI’2004 Workshop Developing User Interfaces With XML: Advances On User Interface Description Languages*. Press, 2004, pp. 55–62.

[2] L. Cranor and S. Garfinkel, *Security and Usability*. O’Reilly Media, Inc., 2005.

[3] B. Shneiderman, C. Plaisant, M. Cohen, and S. Jacobs, *Designing the user interface: strategies for effective human-computer interaction*, 5th ed. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2009.

[4] L. Palen and P. Dourish, “Unpacking ”privacy” for a networked world,” in *CHI ’03: Proceedings of the SIGCHI conference on Human factors in computing systems*. New York, NY, USA: ACM Press, 2003, pp. 129–136.

[5] S. Lukosch and M. Bourimi, "Towards an enhanced adaptability and usability of web-based collaborative systems," *International Journal of Cooperative Information Systems, Special Issue on 'Design, Implementation of Groupware*, pp. 467–494, 2008.

[6] M. Boyle, C. Neustaedter, and S. Greenberg, "Privacy factors in video-based media spaces," in *n Media Space: 20+ Years of Mediated Life*, S. Harrision, Ed. Springer, 2008, pp. 99–124.

[7] C. Gutwin, S. Greenberg, and M. Roseman, "Workspace awareness in real-time distributed groupware: Framework, widgets, and evaluation," in *Proceedings of HCI on People and Computers XI*. London, UK: Springer-Verlag, 1996, pp. 281–298.

[8] M. Boyle and S. Greenberg, "The language of privacy: Learning from video media space analysis and design," *ACM Trans. Comput.-Hum. Interact.*, vol. 12, no. 2, pp. 328–370, 2005.

[9] G. Calvary, J. Coutaz, D. Thevenin, Q. Limbourg, L. Bouillon, and J. Vanderdonckt, "A unifying reference framework for multi-target user interfaces," *Interacting With Computers*, vol. 15, pp. 289–308, 2003.

[10] Q. Limbourg, J. Vanderdonckt, B. Michotte, L. Bouillon, and V. Lopez-Jaquero, "Usixml: A language supporting multi-path development of user interfaces," in *Engineering Human Computer Interaction and Interactive Systems*, ser. Lecture Notes in Computer Science, R. Bastide, P. Palanque, and J. Roth, Eds. Springer Berlin / Heidelberg, 2005, vol. 3425, pp. 134–135.

[11] J. M. Vanderdonckt and F. Bodart, "Encapsulating knowledge for intelligent automatic interaction objects selection," in *Proceedings of the INTERACT '93 and CHI '93 conference on Human factors in computing systems*, ser. CHI '93. New York, NY, USA: ACM, 1993, pp. 424–429.

[12] F. Paterno, *Model-Based Design and Evaluation of Interactive Applications*, 1st ed. London, UK: Springer-Verlag, 1999.

[13] C. Kalloniatis, E. Kavakli, and S. Gritzalis, "Addressing privacy requirements in system design: the pris method," *Requir. Eng.*, vol. 13, no. 3, pp. 241–255, 2008.

[14] M. Bourimi, D. el Diehn I. Abou-Tair, D. Kesdogan, T. Barth, and K. Höfke, "Evaluating potentials of Internet and Web based SocialTV in the light of privacy," 2010, first International Workshop on Privacy Aspects of Social Web and Cloud Computing (PASWeb-2010), 2010 (IN PRESS).

[15] M. Bourimi, S. Lukosch, and F. Kuehnel, "Leveraging visual tailoring and synchronous awareness in web-based collaborative systems," in *CRIWG*, ser. Lecture Notes in Computer Science, J. M. Haake, S. F. Ochoa, and A. Cechich, Eds., vol. 4715. Springer, 2007, pp. 40–55.

[16] M. Bourimi, F. Kühnel, J. M. Haake, D. el Diehn I. Abou-Tair, and D. Kesdogan, "Tailoring collaboration according privacy needs in real-identity collaborative systems," in *CRIWG*, 2009, pp. 110–125.

[17] P. G. Villanueva, R. Tesoriero, and J. A. Gallud, "Multi-pointer and collaborative system for mobile devices," in *Proceedings of the 12th international conference on Human computer interaction with mobile devices and services*, ser. MobileHCI '10. New York, NY, USA: ACM, 2010, pp. 435–438.

[18] J. Camenisch and E. Van Herreweghen, "Design and implementation of the idemix anonymous credential system," in *Proceedings of the 9th ACM conference on Computer and communications security*, ser. CCS '02. New York, NY, USA: ACM, 2002, pp. 21–30.

[19] L. Terrenghi, A. Quigley, and A. Dix, "A taxonomy for and analysis of multi-person-display ecosystems," *Personal Ubiquitous Comput.*, vol. 13, pp. 583–598, November 2009.

[20] A. Larsson and E. Berglund, "Programming ubiquitous software applications: requirments for distributed user interface," in *Proceedings of The Sixteenth International Conference on Software Engineering and Knowledge Engineering (SEKE04)'*, 2004.

[21] K. Luyten, J. V. den Bergh, C. Vandervelpen, and K. Coninx, "Designing distributed user interfaces for ambient intelligent environments using models and simulations," *Computers & Graphics*, vol. 30, no. 5, pp. 702–713, 2006.

[22] K. Luyten and K. Coninx, "Distributed user interface elements to support smart interaction spaces," in *Proceedings of the Seventh IEEE International Symposium on Multimedia*. Washington, DC, USA: IEEE Computer Society, 2005, pp. 277–286.

[23] M. Bourimi, T. Barth, J. Haake, B. Ueberschär, and D. Kesdogan, "AFFINE for enforcing earlier consideration of nfrs and human factors when building socio-technical systems following agile methodologies," in *Human-Centred Software Engineering*, ser. Lecture Notes in Computer Science, R. Bernhaupt, P. Forbrig, J. Gulliksen, and M. Lárusdóttir, Eds. Springer Berlin / Heidelberg, 2010, vol. 6409, pp. 182–189.