Transitions: A Crossmedia Interaction Relevant Aspect

João Paulo Delgado Preti  
Departamento de Informática  
Instituto Federal de MT, Brazil  
joao.preti@cba.ifmt.edu.br

Flavio Miyamaru  
Escola Politécnica  
Universidade de São Paulo, Brazil  
flavio.miyamaru@poli.usp.br

Lucia Vilela Leite Filgueiras  
Escola Politécnica  
Universidade de São Paulo, Brazil  
lfilguei@usp.br

Abstract

The allocation of Interactive narrative across several devices characterizes a crossmedia interaction; a phenomenon that occurs naturally as the number and availability of devices to support interaction grows, even though there is still only ad hoc technological support for designing such systems. This paper focuses on transition, a relevant aspect of crossmedia interactions, presents experiments made and a conceptual framework to support it.

1. Introduction

The distributed user interaction in different devices can be observed today as a natural phenomenon. A study of the use of information and communication technology by young users (from 10 to 18 y-o) organized by Bringué and Sábada [3] concluded that these users perform multitasking over devices, using television, games and Internet navigation simultaneously.

The integrated use of multiple device communication has been studied by the Distributed User Interface (DUI) research field [7] and has also received attention regarding the design of communication professionals. In this community, distributed interaction has been explored under the concept of crossmedia storytelling. A narrative distributed across different media is defined as crossmedia. We have adopted the following definition for this: “The collaborative support of multiple media to delivering a single story or theme, in which the storyline directs the receiver from one medium to the next, according to each medium’s strength to the dialogue” [9].

A thorough discussion about the similarities and differences between the concepts of crossmedia from the communications community design and DUIS is not in the scope of this paper. There are, doubtless, points of intersection between the two research fields.

Briefly, crossmedia interaction design is associated to the distribution of the interaction narrative across devices, exploiting their different capabilities; on the other hand, research on DUI has been oriented towards the distribution of graphic user interface.

In this paper, we address the aspect that is common to both fields: interaction happening in more than one device, be it sequentially or simultaneously. Navigating from one device to another is a relevant aspect for distributed interaction that can inherit insights from the study of crossmedia applications. In this paper, we set the foundations of crossmedia interaction design and on DUI applications to characterize these interactions, which we name “transitions” and present a framework that supports the development of applications using transitions.

Our results come from a three-year research project funded by Fapesp and Microsoft Research to understand and to develop distributed interaction. We believe that the current state-of-the-art does not allow for standardization of transitions. There is still no “winning design” or toolkits that can be used for distributed interaction. There are very few experiences involving development frameworks and middlewares. We also believe that the lack of technological support for applications to conveniently explore the potential of DUIS synergy has limited the potential of commercial crossmedia applications and marketing campaigns. We expect our discussion about transitions to help the development of better tools.

In order to present our ideas, we organize this paper in three sections. First, we explore the transition concept. Then, we present a compilation of transition properties. Finally, these properties originate a set of requirements for a framework that supports them.

2. Transitions

A transition is defined here as the bridge that allows the interaction flow to migrate from one device to another in a DUI application. The concept of transitions is implicit in several works on DUI.
A close look at transitions should consider two different points of view – one is the interaction design perspective, which focuses on communication abilities of transitions: what makes a good transition, how perceivable, persuasive and seamless is the navigation from one device to another, which alternatives provide better user experiences; the other is the engineering perspective, which concerns the mechanisms for implementing transitions, which is the focus of this paper.

2.1. Transitions as narrative hooks


In order to present transitions under the communication perspective, one should see the interaction flow as a narrative, part of which is built by the designer, part of which is written by the users’ experience.

Crossmedia interaction design is about defining which the best medium for a certain part of the story is, or which devices have the best attributes to support interaction. Henry Jenkins [12], who prefers the term transmedia, says “transmedia storytelling represents a process where integral elements of a fiction get dispersed systematically across multiple delivery channels for the purpose of creating a unified and coordinated entertainment experience. Ideally, each medium makes it own unique contribution to the unfolding of the story. So, for example, in The Matrix franchise, key bits of information are conveyed through three live action films, a series of shorts, two collections of comic book stories, and several video games. There is not one single source or text where one can turn to gain all the information needed to understand the Matrix universe.”

Jenkins’ example helps to illustrate the communication perspective of transitions. Linking narrative steps, in order to keep the audience interest, is usual in storytelling. Transitions are here seen as hooks for interaction narrative.

Because crossmedia has been used in games, TV series, movies and advertising, persuasion and fun are often embedded into crossmedia interaction design. Transitions are thus proposed as a challenge: finding the next step to the narrative. The innovation and surprise characteristics attract users that enjoy new interactive media. For instance, a game was developed by IPerG project aiming at destroying a virus that was spread in the university campus, in which mobile phone, touch screens, GPS locators and laptops had to be used [14].

However, in some DUI projects, the designer may not be able to count on the pleasure of discovering how to design an effective distribution strategy. When a service, application or story has its content spread by different devices, the designer must persuade the user to move from one device to another in order to pursue his or her intent. The ability of being persuasive can be seen as an important characteristic of transitions.

Transitions depend on semantic and cognitive issues, as well as on hardware and software resources. While a simple sentence such as “Call number NN and receive the best price offer!” or a newspaper advertisement “Visit our website: www.somename.com!” can effectively serve as a transition if the user gets the phone and dials the number or sits at the computer and enters the website; other applications that involve gestures or the use of cameras may be challenging if users are not aware of what to do. A transitions detailed study has to include the semiotics research area, analyzing culture codes and message meanings for different people. Intelligibility and perceivability are also important attributes.

2.2. Transitions as technological resources

Designers should create transitions to direct the user to the best navigation flow, indicating the most appropriate media for each stage and activity to achieve the system purpose.

From the technological point of view, transitions inherit from the familiar hyperlink, which is the transition element for hypertext documents. When a user activates the hyperlink, the document referenced is fetched and opened. Hyperlinks added a new dimension to previous linear content organization, creating a new navigation style and allowing non-linear narrative, in which the user can choose the path among the documents that can be accessed. Transitions extend hyperlinks bridging content in different devices.

However, unlike hyperlinks, transitions are not supported by any particular protocol; transitions are, so far, ad hoc applications that somehow connect devices in a DUI.

In this paper, we define transition mechanisms as extensions of the cyberspace language for cross-device navigation. Researchers have developed and embedded several different transition mechanisms in their
applications. A bidimensional code in a magazine that can be captured by a cell phone camera is an example of a mechanism that transfers the user’s focus from the printed medium to a digital one. In Seifried’s [22] work, the command that transfers the selected object from the user’s tabletop to the board is an example of transition. In Broll’s and his colleagues’ work [4], transitions are implemented using RFID and NFC-based technologies.

From the diversity of technological devices, the reader can infer that different technological platforms are needed to support transitions. Even though the technological convergence may have simplified the engineering task, there are different protocols and services that can be required to integrate several transitions in one application.

The distributed interaction developer should be able to deal with many technologies, such as those presented in the list that follows.

- Email servers capable of sending and receiving automatic messages;
- Voice portals that can deliver voice messages;
- Short Message Service Centers that manage to send and to receive short text and multimedia messages;
- Gateways that integrate mobile telephone networks with other internet platforms;
- iTV protocols for broadcasting and for interaction that can be different from one country to another;
- Interactive Voice Response systems that interact with users by accepting voice or keypad input;
- Tools for developing desktop and mobile website and embedded applications;
- IPTV middleware;
- Instant messaging routers and APIs;
- Gesture recognition systems;
- RFID and NFC tags.

The list is short; in fact, there are many other options and new products arrive at a fast pace. There is no easy integration of the many technologies. It is hard for the DUI application developer to keep updated with the many options. For the communication designer, who should use the technology to design new communication systems, many of these technologies remain untamed.

3. Transition mechanism properties

Based on former studies and on the analysis of existing DUI applications, we compiled a set of properties that characterize transitions mechanisms, which are described in the following sections so that the concept is clearly understood.

3.1. Cross-device movement

An essential property of transitions is the ability to move interaction from one device to another. As Blumendorf et al. [2] observe, in their elaboration of the concept of DUIs, DUI elements can change dynamically at runtime (e.g. moving from speech to graphical output or transferring controls to a different user). Transition mechanisms are responsible for implementing these movements across devices.

3.2. Temporal coordination

Transition mechanisms must organize interaction distribution as well as handling UI events and procedures. Agreeing with Demeure et al. 4C framework for DUIs [5], coordination implies harmonizing media, efforts, and so forth, to perform a task.

Blumendorf et al. [2] advocate that DUIs aims to provide the best possible interaction capabilities to users by utilizing multiple devices and by considering the current context of use (user, platform and environment). Therefore, DUIs have to support the use of different interaction devices at the same time (simultaneity) and/or the use of different interaction devices over time (sequentiality) by one or several users. Thus, within the concept of coordination, we find that transitions may present continuity, synchronism and scheduling properties.

According to López-Espin et al. [15], continuity is the ability of providing a continuous and consistent user experience without confusion, conserving the state of the interaction. In order to provide continuity, the transition mechanism must be able to convey information on the status of user interaction.

More sophisticated coordination is achieved with the synchronism property, which is the ability of different UI elements to be managed in the same instant of time in different platforms.

Synchronism is related to the fact that the transition may require both, the origin and destination media, to play together in order to create the desired experience. A synchronous transition is triggered in the origin medium, which activates the destination medium, synchronizes both, and delivers the message. In turn, an asynchronous transition liberates the origin medium once the narrative has been handed by the destination medium.

Synchronism requires a transition that lasts over time. For instance, Zollner et al. [25] present ZOIL, a framework for distributed user interfaces. It distributes the user interface by synchronizing the data model of the shared visual workspace while others use protocols...
at the "pixel level" to synchronize UIs visual appearance.

Moreover, transitions can be immediate or scheduled since changes in context may occur. Immediate transitions reach the destination medium as soon as it is available; scheduled transitions have one more control data element that is the time to activate the destination medium. Scheduled transitions have to consider that the user may not be present at the time the transaction is triggered.

3.3. Call to action

Transition mechanisms must implement some form of user invitation to instigate the medium change. According to Dena [6], this invitation is the “call to action” (CTA), in which the designer addresses the user and encourages him or her to look for the next narrative step somewhere else. The term is inspired in the marketing definition of the expression that convinces the consumer to buy a product. CTA is basically equivalent to the hyperlink label.

A CTA also accounts for implicitly or explicitly describing how and when to act. The verbosity needed in the CTA is associated with the assumption of how much the user is aware of the required actions to change/move from one device to another. One example is the use of 2D codes that require special applications installed in cell phones. Because they are not widely known in our country, codes often are accompanied by reader download instructions.

Despite its importance to transition, CTA is not a mandatory attribute. A transition can be automatic or spontaneous instead of CTA–driven.

3.4. Data transportation support

Transitions must be able to carry data that allow interaction flow from one media to another. Data can be of at least two types:
- User or medium identification, that helps applications control the transition (for instance, distributing the interaction to my mobile);
- User session related data, the interaction context in the medium must be shared with the next medium to avoid redefinition.

Marve DUI framework in Fröberg et al. [10] present an interesting example of the data transportation property of transitions - it provides a high level support for transporting components between different devices at runtime.

3.5. Engagement

Some transitions require active user engagement, i.e., some human action is needed to depart from the original medium and resume interaction in the destination medium. These are manual transitions, because media transposition depends on users to find the device, access the medium and provide all the data required to go on.

Automated, “seamless” transitions are more interesting, because no user intervention is needed to manage medium exchange: no user action should be required for handling device availability, transporting data and communicating device readiness.

The fact that the transition is manual or automated brings in the characterization of the technological infrastructure to support device exchange. All device transitions, be they manual or automatic, need some technological support to be accomplished. This attribute characterizes the software application that must run to exchange a device.

3.6. Feedback

An essential usability attribute, transitions must provide feedback. In case of automated transitions, the user must be informed about failures and success - the application is not aware if the user’s attention is in the origin device or in the destination one.

3.7. Privacy

Some transitions may have a private device as the origin medium and thus, the narrative is observed by one user alone, but they may have a public access medium as their destination. This can happen, for instance, when migrating an interaction from a person’s cell phone to a TV set in the living room. This kind of transition requires some special privacy considerations because even though the transition is seamless, the user must be aware of the moment the transition will take place and authorize the exhibition of the data in the destination medium. The user may be concerned not only with the disclosure of personal data but also with the fact that data could be mistakenly transferred to some other public device the audience of which cannot be controlled.

4. Framework for supporting transitions

In this section, we advocate that a technological framework integrating transition technologies and offering designers a simple interface for using the
transition mechanisms in their applications could leverage the concept of DUI.

A framework is a software architecture that provides reuse of solutions. According to Johnson [5], a framework is a skeleton of an application that can be customized by the application developer. The literature points to the existence of some crossmedia and DUI frameworks.

AXMEDIS [18] is an architectural framework, composed by several tools so as to reduce the effort of content production for distribution and integration over different media. This research project was concluded in 2008 and the literature about it provides no explicit mention of transition mechanisms, even though they seem necessary for integration purposes.

Elmqvist [8] presents Munin, a Java peer-to-peer (P2P) distributed user interface middleware. The toolkit is based on two main components: a shared and replicated associative memory, and a common event channel, where peers can produce and consume real-time events (for input and synchronization). The purpose of the toolkit is to enable developers to quickly and easily build ubiquitous visualization spaces where multiple surfaces, devices, displays, and modalities for input and output are combined into a single, coherent space for visual reasoning.

Manca et al. [16] present the MARIA language, which provides an abstract and a set of concrete languages. The concrete languages refer to different platforms, characterized by their interaction modalities (graphical desktop, graphical mobile with touch, vocal, graphical and vocal combination). Thus, a distributed concrete user interface should be able to indicate how the user interface elements are distributed among devices, which can belong to such various platforms.

Melchior et al. [17] defined distribution primitives for DUI, but all of them are related to widget migration. Primitives related to user interaction transition were not considered.

Bardram et al. [1] propose a paradigm called Activity-Based Computing (ABC) as an approach to form, to manage, and to use interactive workspaces. Their approach applies ABC to distributed Multi-Display Environments (dMDE), i.e., environments with multiple devices and displays distributed across several spaces inside a large building. They have defined essential principles: Activity-Centered, Activity Suspend and Resume, Activity Roaming, Activity Adaptation, Activity Sharing, and Context-Awareness. Concerns about transition were slightly perceived in the Activity Roaming principle, but its implementation was restricted to just persist and restore contextual information.

Solutions are verified to exist for content composition, adaptation, management and distribution, but none of those known to us focused on transitions.

4.1. Requirements for a framework supporting transitions

In a previous work of our research group, we identified transition characteristics by analyzing twenty-five market and academic crossmedia applications samples, which included movies, TV series, games, advertisement campaigns, magazines, newspapers, iTV, Web and mobile applications. We extracted the transitions from these samples and worked on their description. A C# framework entitled X-Gov was also created to allow the development of government crossmedia applications [9].

To verify the framework requirements, scenarios were defined and prototypes were developed. These scenarios were based on daily life contexts that can benefit by using multiple devices to perform the tasks required. Six experiments were performed:

4.1.1. Documents for issuing passports. This scenario is about the documents required by the Brazilian Federal Police. The prototype was about a common scenario of informing the documents possessed; rather than writing or printing the list of the web version, a mobile version was proposed in which the user can tick the documents at hand. Regarding the transition aspect, this prototype can illustrate two mechanisms: 2DBarcode and SMS.

4.1.2. On-line photographer. The prototype idea is having the user take a picture using his/her own computer in which a photographer shoots a distance photo. In this scenario, the user accesses a website that requests his/her phone number that once informed, the user immediately receives a call, while the image of the user’s webcam appears automatically on the computer screen. The user hears a recording on the phone requesting him to get correctly positioned in front of the webcam. After a while, the user hears a clicking sound on the phone, indicating that his photo was taken by the webcam, displaying it on a separate page. This prototype was designed to evaluate two devices concomitantly used.

4.1.3. TV marketing. This prototype was developed to evaluate the use of click-to-call with TV as source device. The usage scenario is an application for iTV in which a cell phone carrier promotion is presented for international calls. The application provides a
transition from click-to-call in which the user can enter the desired phone number to automatically receive the call.

4.1.4. Check-in by cell phone. This prototype consists of a mobile application in which the user can confirm his/her presence on the flight. The prototype also includes an airline paper ticket with a 2D barcode that directs the user to the flight confirmation page. This prototype was useful to compare two different kind of 2DBarcode: QRCode and MSTag.

4.1.5. Goods output statement. In this system, a user has a self-service kiosk available to fill out and print the temporary goods declaration withdrawal. This procedure may generate queues to use the kiosk. To expedite service, the system has a mobile web application in which the user can speed up the form completion; when the kiosk is released, it is possible to upload the form to the kiosk, without loosing what has already been filled. This transition was made using push mechanisms.

4.1.6. Tourism digital panel. The tourism panel announces a tour guide in Lisbon that can be accessed directly by phone. The guide presents tourist spots information such as Lisbon Oceanarium, Belém Tower, São Jorge Castle, among others. The user can also see pictures of these sights on the phone and select them to view on the panel.

The survey target audience were users who have control over digital media, users that have no qualms about using digital devices such as phones, computers and televisions. The participants’ selection included people between 18 and 25 years. These users were familiar with SMS, e-mail daily use, playing games, social networking sites and watching videos on the Internet.

The test environment consisted of the following equipment:

- Computers: 1 for the evaluator, 1 for the user and 1 to simulate a digital TV;
- Phones: 1 Nokia N95 and 1 IPhone;
- Television: a 26 inch one;

Usability evaluation was made by metrics of:

- Effectiveness: transition completion, number of attempts and number of requests for help;
- Productivity: time to accomplish the task and time to perform the transition;
- Satisfaction: user’s perceived ease of use, user’s perception of time taken to complete the transition, user’s perception of information clearness and whether the user likes to use the transition.

These transition experiments helped us refine the requirements of the framework; after that, a proof of concept of children school enrollment was also developed, using PC, TV and cell phone as interactive devices. Through these experiments, a minimum set of requirements were elicited to support transitions, as follows.

The framework should provide mechanisms for supporting transitions to and from the most used communication channels.

The framework has to present the crossmedia developer with a set of predefined transitions that links the most used media: websites, email, telephone, cell phones, voice portals, SMS, instant messaging, iDTV. This starter kit would help developers to quickly and effortlessly deploy crossmedia applications, at the same time as it would cover a large spectrum of applications. Integration of platforms allows for the development of new functionalities, such as synchronous control of presentations.

The framework should provide reusable interfaces of transition components that hide implementation details while allowing designers to customize the user’s interface.

Hiding technological details accelerates developers’ learning curve. Customization of the presentation layer aims to facilitate the integration of the transitions into the designed narrative.

The framework should provide resources to automate transition mechanisms as much as possible.

Seamless transitions reduce the amount of user effort in typing parameters and controlling the availability of devices.

The framework support to transitions should present application related data upon users’ request.

URLs, addresses, telephone numbers, email addresses are defined as application-related data and should be hidden from users in seamless transitions. However, in order to recover from error conditions and to allow manual transitions, the framework must provide resources for informing users about transitions data.

The framework should provide resources to inform users in case some error prevents the automated transition mechanisms from working properly.

User’s feedback should be provided by the framework to help users control the interaction flow.

The framework should provide users with resources to undo an automated transition.

Users should be able to step back in the narrative flow.

The framework should be flexible to support the addition of new crossmedia transitions.
As new communication resources appear, they can be incorporated into the crossmedia framework, without compromising existing solutions.

**The framework should keep track of each user’s devices throughout the transaction.**

Keeping track of which devices each user is using allows the crossmedia framework to push content into the most suitable medium.

**The framework should recognize its users.**

Keeping track of users’ identity (or identification, at least) allows annotating preferences and provides support for customization.

**The framework should keep track of users’ navigation across media.**

The DUI application manager should have access to users’ navigation history. He or she should be able to know how many users have been exposed to CTAs; how many have tried to accomplish the transition and how many have succeeded.

One of the benefits of the transition framework is its ability to follow users. An important metrics for distributed application is the conversion rate, that is, the number of users that, having been exposed to the call-to-action, accomplished the migration to the other device. This metric can be used in commercial environments to assess the effectiveness of the DUI approach.

The requirements listed were detected based on a survey research, on usability heuristics and on the research group’s experience.

5. **Transitions middleware architecture**

The creation of the crossmedia framework was a starting point to define and to understand some requirements of a distributed interaction and to understand the need of establishing a transition concept.

The literature survey was also important to understand that many approaches seem to be feasible for addressing the deployment of UIs in a diversity of devices in a ubiquitous environment. However, they do not exploit all the possibilities provided in these environments [23].

This research helped us realize the need of an infrastructure that supports the development of applications, such as our own framework, or applications that need to conduct a distributed interaction without the framework, applications that do not mean to extend a .Net platform or that do not mean to inherit government requirements, for instance. The needed infrastructure was designed as a middleware extension of a Service Oriented Architecture (SOA).

Middleware consists of reusable functionality that offers solutions to frequently encountered problems such as heterogeneity, interoperability, security, dependability, etc. This functionality is offered either by the core of a middleware infrastructure, or by complementary services. The former mediates the interaction between distributed objects, while the latter deal with complementary issues [24]. Our goal is to define these complementary services in order to help construct a planned and organized distributed interaction.

Therefore, this section presents a service-oriented architecture that assists in the development of applications that need to coordinate distributed interaction between multiple devices. Distributed Interaction Support Service (DISS) is a service that intermediates communication between devices, as in Figure 1. DISS specifies a set of necessary services to allow the coordination of multiple devices sequentially or simultaneously and implements a subset of the framework requirements stated in the previous section of this paper, as follow functional requirements (FR).

- **FR1** - The architecture should allow restricted access to one or more devices to a particular user;
- **FR2** - The architecture must allow the use of the services of one or more devices by multiple users;
- **FR3** - The architecture should allow more than one device to be used by the same user simultaneously and synchronously;
- **FR4** - The architecture should allow interactive devices to be interchanged in order to perform a task;
- **FR5** - The architecture should allow a different user that started the process to continue a task;
- **FR6** - The architecture must convert content not supported by a particular device;
- **FR7** - The architecture should allow more than one interaction modality to be used to interact with a particular service;
- **FR8** - The architecture should allow adding new metadata to the set of device information;
- **FR9** - The architecture should allow devices to be grouped or classified according to users’ interest;
- **FR10** - The architecture should allow ad hoc devices discovery;
- **FR11** - The architecture should facilitate device testing communication.
Devices providing its resources through services will be the basis for the next generation of device communication. Nevertheless, the integration of services is hampered by a number of factors, reliability of the network signal, delays in communication and limited processing resources.

SOA technologies for devices emerged as UPnP and Jini, but a promising technology for compatibility with Web Services (WS) is proposed by OASIS through Device Profile for Web Service (DPWS) specification.

DISS has DPWS specification as an architecture base for discovery and communication between devices. As seen in Figure 1, all the devices in the distributed environment must have DPWS specification implemented. The DPWS protocol stack can be seen in Figure 2.

5.1. DISS structure

Even though DPWS stack provides specifications for service description, location, security and events, it does not include some features needed for a distributed interaction scenario. For this reason, DISS existence becomes necessary, its rationale being described in the following paragraphs.

DISS is composed of the services shown in Figure 3.

5.1.1. Interaction modality mapping service. Responsible for mapping the interaction modality with the interface of a remote service a given application wants to interact with.

5.1.2. Device management service. Responsible for registering desired devices, allowing grouping, sorting and adding information, such as location.

5.1.3. Context support service. Responsible for managing the user context, persisting objects stated on the server and client’s side, regardless of the number or type of devices.

5.1.4. User management service. Responsible for users’ registration and device association.

5.1.5. Content conversion service. Responsible for content conversion, in case the device selected does not support the content received.

5.1.6. Transition support service. Responsible for ensuring that, during the execution of a task, users and devices can be interchanged [20].

5.1.7. Default graphics user interface service. Service that provides a default GUI to represent the target device.

Table 1 presents each functional requirement and its services attendant.
Table 1. Services that meet the requirements

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Related Service</th>
</tr>
</thead>
<tbody>
<tr>
<td>FR1</td>
<td>IMMS, UMS, DMS</td>
</tr>
<tr>
<td>FR2</td>
<td>CSS, UMS</td>
</tr>
<tr>
<td>FR3</td>
<td>IMMS, CSS</td>
</tr>
<tr>
<td>FR4</td>
<td>TSS</td>
</tr>
<tr>
<td>FR5</td>
<td>TSS, UMS</td>
</tr>
<tr>
<td>FR6</td>
<td>CCS</td>
</tr>
<tr>
<td>FR7</td>
<td>IMMS</td>
</tr>
<tr>
<td>FR8</td>
<td>DMS</td>
</tr>
<tr>
<td>FR9</td>
<td>DMS</td>
</tr>
<tr>
<td>FR10</td>
<td>DPWS</td>
</tr>
<tr>
<td>FR11</td>
<td>DGIS</td>
</tr>
</tbody>
</table>

In our middleware, TSS is responsible for allowing users to continue their tasks in another device or transfer the task to another user to continue the task that was being performed. Two types of transitions need to be supported in the middleware.

The first one is transition between devices, which direct users to another device in order to take the required steps in pursuit of their goals. For example, a scenario in which the user wants to continue the TV interaction on the phone could occur via QR Code, even though this strategy is not sufficient to transmit the entire interaction state the user performed on TV.

The second one is transition between users, when it is necessary for another user to carry out a task already begun, in order to use the steps already taken in pursuit of a goal. This service is responsible for migrating a user session to another, keeping the original state of the interactions previously performed. This service also needs to know the history of actions performed and place the new user in the last step performed by the former user.

A deeper look into transitions should consider two aspects: one from the perspective of communication, about what constitutes a good transition: how noticeable, persuasive, is integrated a navigation device to another, the other is the technological perspective, which satisfies the mechanisms to implement the transitions.

From the technological point of view, several platforms are currently required to support transitions. Although the technological convergence has simplified the engineering task, different protocols and services are necessary to integrate different types of transition into an application.

6. Conclusion

We understand that the most important research action is towards transition standardization. We found many different mechanisms that work as transitions. We believe that this research can help programmers to develop applications with a distributed interaction nature.

However, from the interaction design point of view, a common language for DUI transitions is still missing. We advocate that this language should be based on a deeper user research to develop an intuitive set of seamless transitions.

7. References


