International Journal of User-System Interaction 14(2) 2021, 47-67 © MatrixRom

A Profile and Design Space for Characterizing User Interface Adaptation

Víctor López-Jaquero[1], Vivian Genaro Motti[2], Francisco Montero[1], Pascual González López[1], and Nicolas Burny[3]

[1]Laboratory of User Interaction and Software Engineering (LoUISE), Computer Science Dept., University of Castilla-La Mancha, Campus Universitario, 02071 Albacete, Spain

[2]Department of Information Sciences and Technology (IST), Volgenau School of Engineering, George Mason University, Engineering Building, Rm. 5350, Fairfax Campus, USA

[3]Louvain Research Institute in Management and Organizations (LouRIM), Université catholique de Louvain, Place des Doyens, 1 – B-1348 Louvain-la-Neuve, Belgium

E-mail: VictorManuel.Lopez@uclm.es, fmontero@dsi.uclm.es, vmotti@gmu.edu, Pascual.Gonzalez@uclm.es, nicolas.burny@uclouvain.be

Abstract. To better characterize the adaptation process of a user interface, we introduce an adaptation profile and a design space based on the seven adaptation stages defined in the GISATIE life-cycle: goals, initiative, specification, application, transition, interpretation, and evaluation. The adaptation profile expresses who is responsible for ensuring each adaptation cycle: one or several end users, one or several machine agents, one or many third parties, and any combination of the former. The adaptation design space expresses seven key dimensions along which adaptation can be decided and designed: autonomy level, granularity level, task resuming granularity, user interface deployment, technological space coverage, user feedback, and modality. Some examples are included to illustrate how to use this profile and design space for two systems ensuring user interface adaptation to some extent.

Keywords: adaptation profile, design space, user interface adaptation

DOI: 10.37789/ijusi.2021.14.2.1

1. Introduction

Since its inception in the late eighties (Browne et al., 1986; Cockton, 1988; Totterdell and Rautenbach, 1990), user interface (UI) adaptation (Jameson, 2003) has often be justified by the desire to see the UI adapted to the end user and not the end user forced to become adapted to the UI. While this goal remains always for the ultimate benefit of the end user, adaptation received a continuous attention since then. UI adaptation consists in changing parts or whole of its components in order to better fit the requirements, the needs, and the wishes of the end user, either taken in isolation or in groups. Depending on who is controlling the adaptation (Browne et al., 1986; Dieterich et al., 1993), it has been classified according to two categories: *adaptability* occurs when the end-user can adapt the UI, while *adaptivity* occurs when the system has the capability to adapt the UI. *Mixed-initiative* adaptation (Horvitz, 1999) exists when both, the end user and the system, cooperate towards adapting the UI. Adaptivity, although more expensive to develop, has demonstrated several benefits (Lavie and Meyer, 2010) and it is used in a wide range of domains of human activity.

Nowadays, the most widely accepted understanding of the adaptation process still comes from Dieterich's survey on adaptation techniques (Dieterich et al., 1993), despite it being produced in 1993. Besides its age, Dieterich's taxonomy suffers from several shortcomings: it is constrained to a single entity (e.g., the user and the system) in each stage of the adaptation process, it does not handle explicit collaboration during the different adaptation stages and it is restricted only to the execution part of the adaptation. While this survey identified five types of agents involved in the life-cycle (i.e., designer, system administrator, local expert, user, and system), we do not know who is involved in which adaptation stage as defined in the GISATIE life-cycle inspired by ISATINE (López-Jaquero et al., 2007). Some other issues in the adaptation process, such as how the adaptation is specified, were left out of the framework (Motti and Vanderdonckt, 2013).

To fill these gaps, this paper introduces a design space for UI adaptation to help the designers and developers during the design of adaptation capabilities of applications and to compare existing ones. This design space is complemented with an adaptation profile that can be used as a quick reference to document the coverage and actors involved at each stage of the adaptation process. A definition of these actors is now part of several recent systems, such as self-adaptive UIs obtained by model-driven engineering (Ygitbas et al., 2020) and the conceptual reference framework for intelligent UI adaptation (Abrahão et al. , 2021).

2. A Profile and Design Space for Adaptation

Within adaptation, there is a wide range of possible situations to be considered. Many of these situations have been investigated since the first systems supporting UI adaptation, such as adaptable dialogues (Cockton, 1988), UI adapt (Totterdell & Tautenbach, 1990), MASHA (Rosaci & Sarné, 2006) until the most recent ones, such as MALAI (Blouin & Beaudoux, 2010), UX-AI (Hussain et al., 2020), and self-adaptive UIs (Ygitbas et al., 2020).

The range covered by the different instantiations of the framework are described in terms of the stakeholders involved in each adaptation stage proposed in the framework and how these stakeholders carry out each stage or substep. To represent both things, a characterization of the system's adaptation capabilities is provided in terms of a profile, where all the stakeholders involved in each stage of the adaptation process are graphically shown (see Figure) and a design space composed of seven semi-axis, each one representing a different dimension considered in the life-cycle (see Figure). Next, both the profile and the design space will be further described.

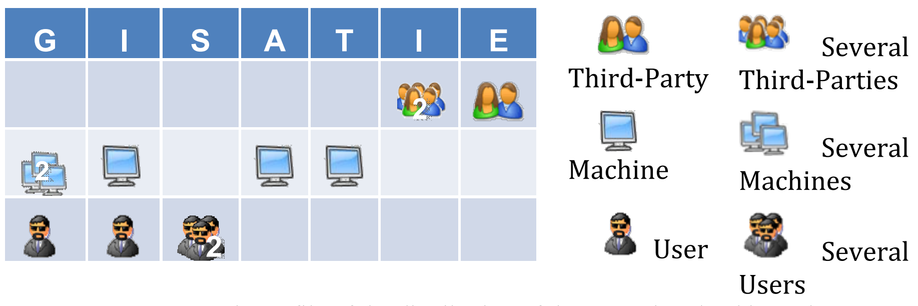


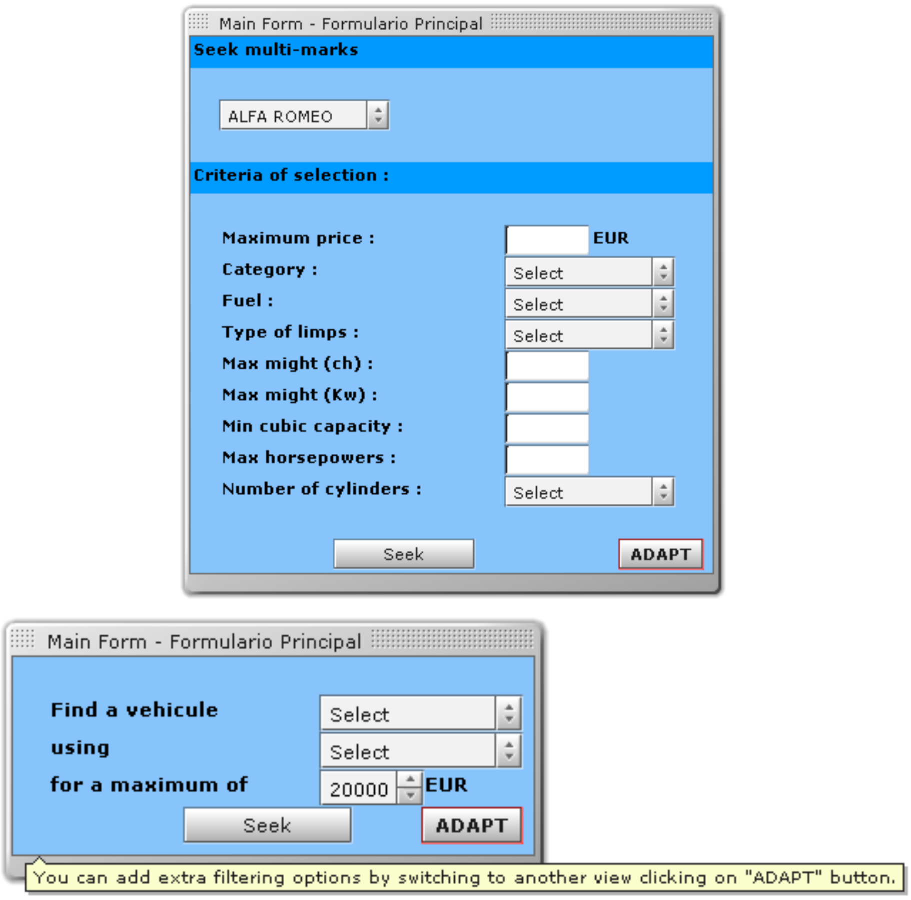
Figure 1. An example profile of the distribution of the actors involved in each stage.

2.1 A GISATIE profile for quick reference

To create a more compact description of an adaptive or adaptable application a profile for UI adaptation is defined (see Figure) based on the seven stages of UI adaptation defined in our life-cycle: **G**oals, **I**nitiative, **S**pecification, **A**pplication, **T**ransition, **I**nterpretation, and **E**valuation. With this profile an adaptive application can be described in terms of the actors involved in each stage of GISATIE framework. The columns define the seven stages of adaptation as described in GISATIE, and the rows define respectively the involvement of third-party(s), machine(s) and user(s). Numbers can be added on top of the icons illustrating the exact amount of entities of that type involved. When a user is involved in a stage then a user icon should be placed in the column of the corresponding stage at the third row. Likewise, if a machine is involved in the stage a computer icon is placed for the corresponding stage in the second row. Finally, if a third-party is involved in a specific stage, an icon will be placed in the corresponding column at the first row. This representation is called the profile of the application with adaptation capabilities and can be used as a quick reference to quickly find out the entities involved at each stage in the adaptation process as well as those stages not supported (by checking the columns with no entity). Furthermore, this profile can also be used to discover the need for coordination between some entities at a specific stage. i.e., in Figure for the *Initiative* stage two entities (a machine and a user) can perform that stage. Therefore, some coordination between both entities will be probably needed. On the other hand, when several actors of the same type are involved in a stage, some kind of coordination will be probably required as well.

Thus, when designing adaptation facilities, this aspect is something important to think about too (Bunt et al., 2004). For example, in Figure for the first stage Goals (denoted by G) both the user and two system actors can be involved. By providing this application profile, any application with adaptation capabilities can be described in terms of the stakeholders in each stage of the adaptation process in a simple and compact way. Thus, it can be used as a quick reference to discuss about an application with adaptation capabilities.

Figure 2 depicts a user interface for searching a second-hand car. In this example, the user with the help of other users selected the UI adaptation (hence, two users in the S column in Figure 1) that is further applied by the machine (hence, a machine in the A column of Figure 1). The resulting adapted user interface is shown afterwards. Between the two UIs, there is no real transition (hence, a machine in the T column of Figure 1), but an animated transition could be ensured between them (Dessart et al., 2012). The two users who combined their efforts to selected the UI adaptation then interpret the results (hence, two users in the I column of Figure 1). But only the single end-user evaluate whether this adaptation corresponds the initial goals (hence, one user in the E column of Figure 1).



Main Form - Formulario Principal

Seek multi-marks

ALFA ROMEO

Criteria of selection :

Maximum price :	EUR
Category :	Select
Fuel :	Select
Type of limps :	Select
Max might (ch) :	
Max might (Kw) :	
Min cubic capacity :	
Max horsepowers :	
Number of cylinders :	Select

Seek ADAPT

Main Form - Formulario Principal

Find a vehicule	Select
using	Select
for a maximum of	20000 EUR

Seek ADAPT

You can add extra filtering options by switching to another view clicking on "ADAPT" button.

Figure 2. An example profile of the distribution of the actors involved in each stage.

2.2 A Design Space for Adaptation

A design space, by generating a space of alternatives, defines the possibilities for the development of an application regarding multiple dimensions. With an explicit representation of the possible options, it is possible to use the design space before the implementation of a project, in order to be aware of and identify alternatives for the design, and after the implementation of a project to analyze its exploration of the possibilities and to compare it with other projects. The Design space can guide stakeholders to make better decisions during a project, by selecting relevant aspects of the dimensions according to the project goal. And the later analysis permits stakeholders to evaluate and extend the project development. Besides, during the development life cycle the stakeholders can use a design space to update the requirements and to identify alternatives for them in case of necessity. In addition to this, a design space can aid to communicate design decisions by allowing their documentation, to review (verify the features available), and to compare applications.

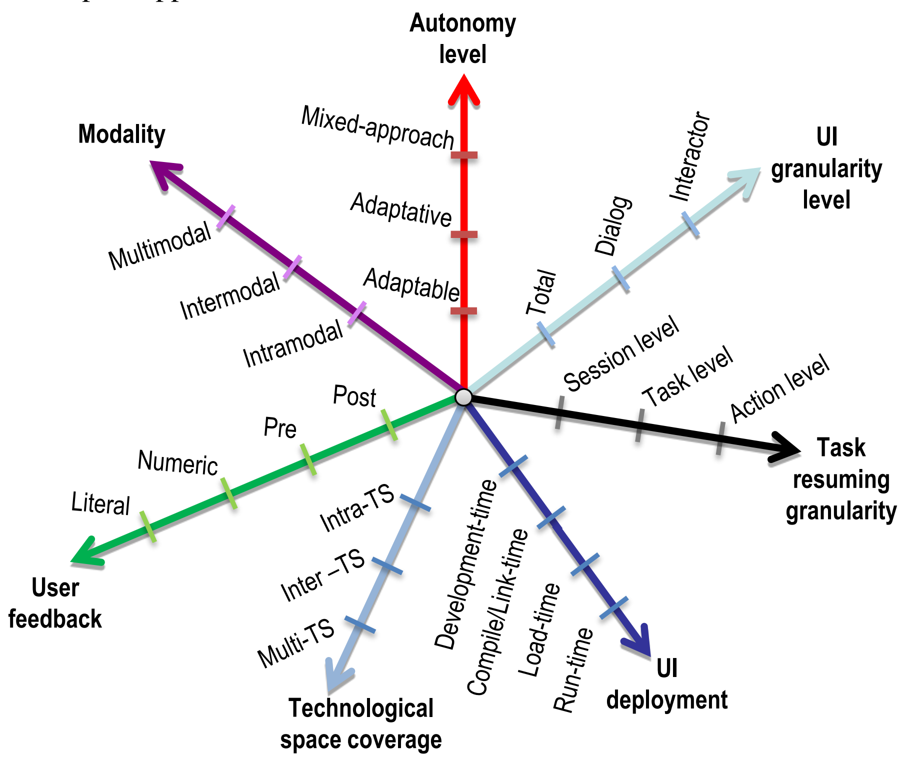


Figure 3. A graphical representation of our design space for adaptation.

The main challenge in the definition of a design space consists in precisely defining dimensions that describe most of the applications, and define them in significative granularity levels, i.e., in a way that accommodates most of the decisions, making it still feasible to locate and identify the possible decisions. Thus, being as much extensive as possible and precise, but at the same time not overdetailed. We elaborate a design space that characterizes the most important design dimensions of user interface adaptation. This design space is composed of seven dimensions, namely *modality, autonomy level, ui granularity, task resuming granularity, ui deployment, technological space coverage and user feedback.* Next, all these dimensions are explained in detail, including some examples and a discussion for each one.

2.2.1 Autonomy level

It corresponds to the degree of autonomy of the adaptation is performed. The choices available in this dimension are: *Designed* means that the system does not support adaptation. *Adaptable* means that the system relies on the user for triggering and executing the adaptation. *Adaptive* means that the application can carry out adaptation autonomously. Lastly, *mixed-approach* systems go one step beyond, since they include both adaptable and adaptive behaviors, that is, both the user and the system can make decisions during the adaptation.

Designed systems are not desirable at all, and should be avoided. Neglecting the consideration of user preferences will probably make the application fail. Nevertheless, it is easy to find plenty of adaptable systems currently. For example, almost every graphical user interface for desktop computers exhibit an adaptable behavior to support the user in changing the background image, the fonts, colors, etc. *Adaptive* systems are not as common as *adaptable*, however some popular applications, such as Microsoft Office 2013, include adaptive features, for instance when the size of the window of the application is changed. *Mixed-approach* systems are harder to find. In MAYA (López-Jaquero et al., 2008; 2009), some mixed-approach features are included that support the user in contributing to several adaptation stages, including initiative, interpretation or evaluation. As we move from designed to mixed-approach adaptation the risk of making wrong decisions is increased, since the autonomy of the system raises. The complexity of the design of the adaptation capabilities will increase, as well as the expected benefits. Nevertheless, the mixed-approach can provide interesting benefits to the user, especially when the system is not sure enough about what to do and delegates to the user for making the decision.

2.2.2 UI granularity level

This dimension describes the UI level of granularity that the system can target in the adaptations provided. *Interactor* means that the adaptations can change some properties in the interactors (e.g., size or color) or even replace it with another one; *Dialog* corresponds to changing part of the interface, such as a container; and *Total* level corresponds to those adaptations that target the entire application.

There are many examples of systems that work at the *Interactor* level. Most systems that support the customizations of the look&feel of the user interface work at this level supporting changing the colors, the fonts, etc. Nowadays it is also common to have some support additionally at the *Dialog* level. A common adaptation feature in many systems working at this level of the UI is moving, adding or removing toolbars, as in Adobe Acrobat Reader. And an example for the *Total* level can be found in those applications supporting switching the theme of the application, as in Microsoft Office. This granularity level is important, in the sense, that if the adaptation engine can change everything, from the interactors to the total level, the adaptation designer can produce almost any adaptations. Nevertheless, as the granularity increases, a greater disruption can be produced in the user interface, since larger regions of the user interface change.

2.2.3 Task resuming granularity

When an adaptation is applied, the user will have to resume was he was doing. If the user is forced to quit the application and restart it for the work to be resumed, then the status recover it at the *Session* level. On the other hand, if it can affect only the current *Task* the user was doing, and forcing it to be restarted, or if it can affect only one single *Action* of the user (more fine-grained adaptation). The more fine-grained status recovery we provide to the user the less disruption will be provoked. Status recovery should be carefully designed, since if the user is not able to resume his work, he might get disoriented and reject the adaptation. This stage is closely related to Transition stage is GISATIE adaptation framework.

2.2.4 UI deployment

To specify when an adaptation state is carried out we use the four dimensions proposed for Compositional Adaptation in (McKinley, 2004), namely *Development-time*, *Compile/Link-time*, *Load-time* and *Run-time*. The adapted user interface is deployed at development-time when it is done while the system is being created. However, an adaptation is deployed at Compile/Link time when it is carried out when the system is compiled and linked to produce the executables that will run the adaptation stage management. Finally, the adapted user interface is deployed at run-time when the adaptation is made on-the-fly while the application is running.

Adaptations deployed at development-time are static and therefore exhibit some limitations regarding their use in dynamic contexts of use. There are some examples where the adaptations are deployed at compile/link time. i.e., in (Schlee and Vanderdonckt, 2004) a calculator application is adapted at compile-time offering the user a set of options to decide what to include or exclude in the application, and in (Sendin et al., 2008) the adaptations are deployed at linking-time by using an approach based on Aspects Programming. There are many examples available in the literature that support the deployment of the adaptations at run-time, for example, Microsoft Office deploys the adaptations at run-time when we resize the window.

Run-time is in principle the most powerful approach for UI deployment, since it allows the user to get the adaptation as soon as the need for adaptation is detected. Furthermore, it is also more dynamic, in the sense that it supports more sophisticated approaches in adaptation that can react more quickly to context of use changes. This dimension of our design space is also closely related to the specification of the Application stage in GISATIE.

2.2.5 Technological space coverage

A *Technological Space* (TS) (Kurtev et. al., 2002) is a working context with a set of associated concepts, body of knowledge, tools, required skills, and possibilities. It is often associated to a given user community with a shared know-how, educational support, common literature and even workshop and conference regular meetings. In the context of this paper we consider a technological space as a set of related hardware and software resources induced by a particular software development method. For instance, if a transformation approach is used then a technological space may consist of the target platform along with a transformation engine, a model editor, a meta-model editor and supporting technologies such as EMF, GMF, eCORE, or mwith model transformations (Aquino et al., 2010) performed on user interfaces specified with a User Interface Description Language, such as UsiXML (Limbourg et al., 2004). These models and their corresponding meta-models are typically gathered in an ontology for describing UIs uniformly (Furtado et al., 2001). In some cases, the computing platform might unambiguously determine the software resources, i.e., for iPhone / iPad hardware, only one operating system and a single development kit. The different steps in this dimension are the following (sorted by increasing order of sophistication):

- *Intra-TS*: it means that the technological space before and after adaptation are identical. In other words, the adaptation process leaves the TS unchanged.
- *Inter-TS*: when the technological space is changed after the adaptation
- *Multi-TS*: when there are alternative options for technological spaces considered and provided with the adaptation process.

2.2.6 User feedback

It corresponds to the ability of the user to provide her perspective over the adaptation performed. *None* when the user has no 'voice'; *Post-acceptance* when the adaptation is first performed and then the user is allowed to accept or reject it; *Pre-acceptance* occurs when the adaptation is proposed to the user before being performed; *Numeric evaluation* occurs when the user is able to evaluate with a numeric criteria the adaptation performed (e.g., with a 7-point Likert scale); and *Literal evaluation* occurs when the users are able to express their opinions about the adaptation being performed literally, i.e. expressing semantic information.

Unfortunately, most systems with adaptation capabilities do not support user feedback. Pre-acceptance is available in those systems that provide a preview of the adaptation. Numeric user feedback can be found in (Arhippainen, 2004). Lastly, an example of literal user feedback is found in Eisenstein et al. (2000), where the user provides some textual feedback that the system interprets. User feedback is at least as important in adaptation as in general user interface design. Even if the adaptation designer made a great effort to design good quality adaptations, it is impossible to foresee the many different situations that can arise during interaction, including the wide range of user preferences. When the user cannot express his opinion about the adaptation it is hard to find out whether the adaptation was good or not. Pre-acceptance will provide a more conservative approach to adaptation, but at the same time it can be cumbersome for the user when the system delegates the decision about the adaptation to the user. Providing a score to the adaptation, either numeric or literal, can be very useful and less intrusive.

2.2.7 Modality

This dimension illustrates the extent to which the adaptation of interaction modality is supported. Since different modalities can be supported in a single user interface (for instance, in a smartphone we have graphical modality, haptic modality when it vibrates or auditory modality to play sounds or input speech) or gestural modality when the UI switches to accepting gestures performed by the end user. *Intra-modality* occurs when the adaptations do not change the modality type (i.e., adapting the textual contents in a web page), *Inter-modality* occurs when the adaptation modality changes from one type to another (i.e., replacing audio feedback with vibration when the device is in silence mode) and Multi-modality occurs when alternative modalities are considered and provided by the application (i.e., the provided adaptation relies on both audio and text to present the contents).

Modality is a very important feature currently with the wide use of mobile devices that support several modalities. Therefore, it should be a very important feature to consider in the design of adaptation for modern sensors, devices, and computing platforms (Calvary et al., 2003).

3. Using the Profile and Design Space

3.1 The MAYA Example

To illustrate how both the profile and the design space can be used, MAYA, the multi-agent system will be used to exemplify the different stages of the adaptation process (López-Jaquero et al, 2008; 2009). How MAYA supports the adaptation stages of GISATIE is characterized on Figure . It helps us in quickly identifying what are the strong and weak points related to the coverage of GISATIE adaptation framework. At first glance it is easy to see that in this case all seven stages are covered in MAYA, but third party involvement can be clearly improved, an thus negotiation between entities would be necessary. Also, *Adaptation* and *Transition* stages cannot be performed by the user. It is also easy to discover that MAYA encourages mixed-initiative, since most stages can be made by either the user or the machine, so that the user is involved in the adaptation process. *Goal* stage can be kept by the machine, and to some extent also by the user (by supporting some adaptability features). *Initiative* stage can be made by either the user or the machine, both can trigger the adaptation process. The *Specification* stage is mostly done by a third-party (designer), but the machine and the user can also decide what rules to apply. This is a strategic stage. Choosing the wrong rule or applying bad rules will yield poor adaptation results for sure. The *Adaptation* stage can only be performed by the machine. This often occurs in most systems, but supporting the user in applying the adaptation could be interesting. i.e., we could imagine an adaptation where the user interface layout will be rearranged and support the user in rearranging the user interface the way he likes. *Transition* stage is made by the machine, it will be the entity in charge of ensuring a smooth transition. We could also imagine the user choosing the way this transition is made. *Interpretation* can be provided by the machine so the user understands why the adaptation was made, or by the user if he choses what adaptation rule to apply, so that the machine can understand why the user did that adaptation. *Evaluation* is twofold: on the one hand the machine performs an automatic evaluation against the goals of adaptation, and on the other hand the user provides feedback by either accepting or rejecting the adaptation applied.

Even though there is a wide coverage of GISATIE stages in MAYA, improving third-party support could be interesting to better integrate the current trend towards the integration of different resources and services from the cloud, being the cloud our third-party provider.

G	I	S	A	T	I	E
		👥				
🖥	🖥	🖥	🖥	🖥	🖥	🖥
👤	👤	👤			👤	👤

👥	*Third-Party*
🖥	*Machine*
👤	*User*

Figure 4. The profile when applied to MAYA.

Figure 5 depicts the design space of GISATIE applied to MAYA. This figure will be explained starting with the vertical axis (*autonomy level)* clockwise. Regarding *autonomy level,* MAYA is considered a *mixed-approach*, since in several stages of the adaptation process either the system or the user can perform this stage. The *ui granularity level* that the adaptations can be applied is at the interactor level, since any property of a user interface widget can be changed (which has been previously specified in the user interface meta-model used). After an adaptation has been applied, the *state recovery granularity* supported is *task level*, because just before the adaptation is applied the current status is stored to be later resumed after the adaptation. Supporting *action level* granularity is quite complex. For example, if the adaptation applied includes widget replacement, the way the task should be done could imply different actions, thus resuming the work just at the action the user was interrupted when the adaptation happened would be impossible. The *UI deployment* in MAYA is made at run-time.

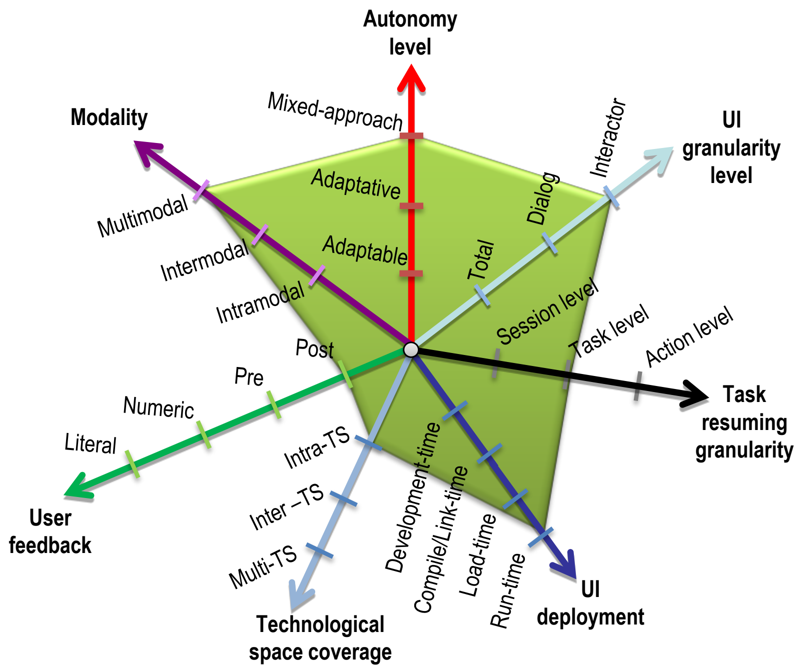
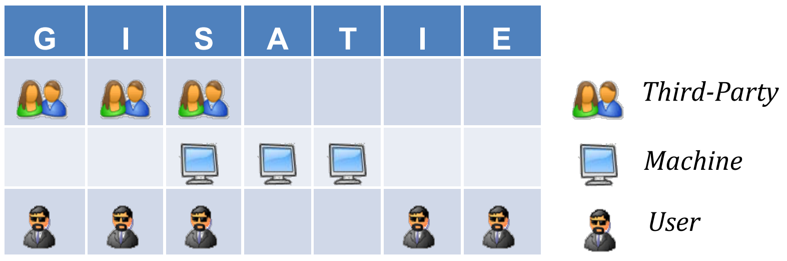


Figure 5. The design space when applied to MAYA.

All the adaptations can be applied while the system is running. The current technological coverage of MAYA is rich Internet interfaces. Although others could be integrated, currently the deployed user interfaces are always rich Internet interfaces. Therefore, currently MAYA should be classified as intra-TS. After an adaptation has been applied the user can provide feedback, for instance by undoing the adaptation and thus showing his disconformity with the adaptation applied. Therefore, the *user feedback* in this case is *post* adaptation. Regarding *modality* dimension, it is *multimodal*, because adaptations can use graphical or audio modality or even both at the same time. With these dimensions we have described some of the most interesting capabilities that our system with adaptation considers. This can be further described by using the adaptation profile to check the coverage of the adaptation process stages and the stakeholder at each stage.

3.2 Using the Design Space and the Profile: The UILAB Example

A second example concerns UILab (Burny & Vanderdonckt, 2021), a workbench for automating the computation of various GUI metrics, such as metrics for aesthetics. UILab does not automate any GUI adaptation per se, but enable the end user or any third party to initiate an evaluation of the GUI layout in order to adapt it, for example to optimize its visual aesthetics. For this purpose, a GUI screenshot or the URL of a wab page can be captured and submitted to GUI metrics evaluation. Consequently, the process is as follows (Figure 6): any end-user or third paty can specify the goals for GUI adaptation by assessing its visual aesthetics. The specifications are encoded into UILAB by declaring which metric(s) should be then applied automatically by the machine (e.g., assess the visual balance of a web page in portrait mode on a high-resolution smartphone). The results are then displayed and positioned with respect to the distribution of the metric related to the context (e.g., how this web page compare to others for this metrics). The results are then under the responsibility of the end-user to be interpreted and executed (e.g., by adapting the layout manually according to the results).



G	I	S	A	T	I	E	
👥	👥	👥					👥 *Third-Party*
		🖥	🖥	🖥			🖥 *Machine*
👤	👤	👤			👤	👤	👤 *User*

Figure 6. The profile when applied to UILAB.

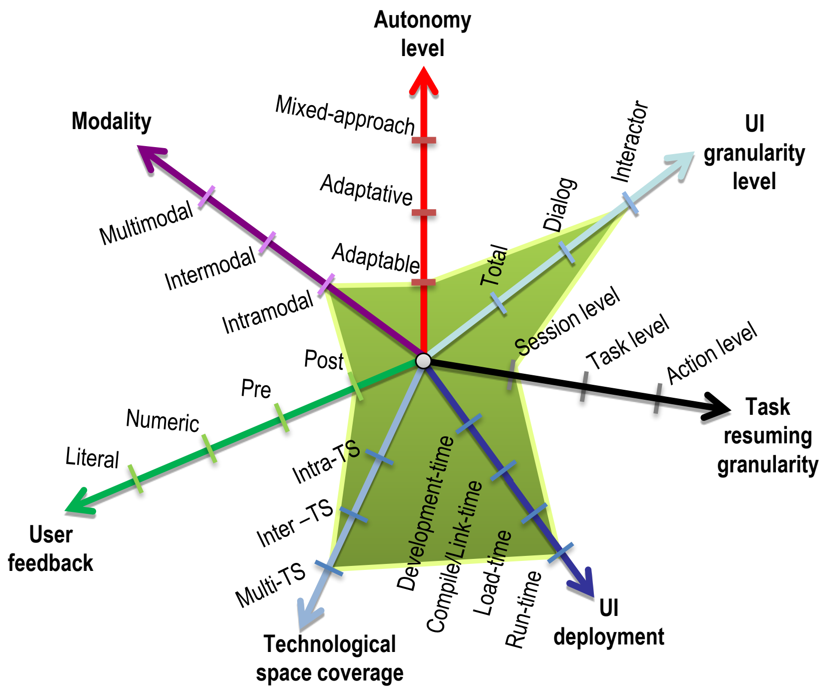


Figure 7. The design space when applied to UILAB.

Figure 7 depicts the design space for a typical GUI aesthetic evaluation performed in UILAB towards adapting it depending on the context of use. The main advantage of UILAB lies in its capability to evaluate the GUI of any application belonging to multiple technological spaces: only a screenshot of the GUI and/or the URL of any web page or web site is required to conduct the evaluation. In the case of a URL, UILAB automatically accesses the web site at run-time in the required configuration (e.g., on a tablet in landspace mode with a certain resolution specified by the end-user), captures the screenshots into a gallery, and performs the automatic computation of metrics on these screenshots. But the limitation of this approach is that only the graphical modality is considered. Other modalities are not supported for the automatic computation of visual metrics and are therefore out of scope. A second advantage is that the computation of visual metrics can be performed at any UI granularity level: from a certin interactor to an entire screen or a series of screens. In the case of low granularity, only some metrics are applicable (e.g., colorimetry for an image). In the case of high granularity, individual metrics for each screen can be computed (e.g., the proportion of the GUI) as well as the consistency of this metrics across screen (e.g., how many variations of the proportion are noticed).

3.3 Discussion

Among the many different benefits that a Design Space provides, Beaudouin-Lafon (2000) remarks three virtues that guide a Design Space definition:

- A *descriptive virtue*: any adaptation technique should be described completely, consistently, and unequivocally based on the design space.
- A *comparative virtue*: each pair of adaptation techniques should be made comparable according to the same criteria defined in the design space. This supports a sound comparative analysis of techniques, and later on, a rigorous benchmarking of techniques. For instance, if we compare two techniques for adaptation on the design space, we will rely on the coverage of the two techniques in order to identify their respective strengths and weaknesses.
- An *exploratory virtue*: all steps of all dimensions of the design space should be explored in order to identify where existing techniques are, where new opportunities are, and where under-explored portions of the design space are. In particular, we will identify uncovered steps and dimensions of UI adaptation, first from an analytic point of view, then according to a technological support.

During the development of our design space we aimed at providing the adaptation developer with an extra tool to support all the three virtues aforementioned. The main challenge in the definition of a design space is to consider all relevant dimensions while keeping the visualization of all the dimensions clear enough. It is important to remark that this design space is extensible, since other axis can be incorporated; flexible, since the axis can be added, removed or further detailed; and unified, because the axis enable an integrated view of all dimensions simultaneously. Other dimensions can also be considered, for instance the precise time when the adaptation will be applied (e.g., design time, linking time, compilation time, run-time), the involvement of users, machines and third parties such as a broker, the user of an underlying UI meta-model to rule the adaptation, the support for seven stages of adaptation, etc. We tried to keep the design space simple enough so it can be readable and easily understood, but a more exhaustive version would be possible. Regarding the adaptation profile, it has been shown useful as a mean to quickly compare the support different systems have of GISATIE and discover what entities are involved at each stage and identify possible coordination needs between entities.

4. Conclusion and Future Work

Adaptation remains a complex topic. When dealing with such a complex topic, the availability of methods, processes, guidelines, etc. helps the designer to create better system with adaptation capabilities. In this paper we aim at contributing in this direction. First, an adaptation framework has been thoroughly described, including all seven stages. For each stage a definition, together with some examples and a discussion is provided. A multi-agent system has been used as a running example throughout the paper to better illustrate GISATIE. Moreover, different stakeholders are considered and described, namely: the user, the machine and a third party. An interesting aspect discussed in GISATIE is how these entities can interact to perform a stage in the adaptation process. Thus, the entities can interact by using negotiation, consultancy, delegation, coopetition or competition. One of the strong points of GISATIE with respect to other adaptation frameworks is supporting more than just the execution part of the adaptation process, that is to say, covering all the stages related to the evaluation of the adaptation. The evaluation part of the adaptation process should be a must for any adaptation designed, since it will provide better adaptations with smooth transitions from the original UI to the adapted one, an understanding of why the adaptation took place and an assessment of how good the adaptation actually was. By designing all the seven stages, adaptations will be more easily understood by the user, reducing the chances that the adaptation applied is rejected.

The design space proposed provides a tool for adaptation designers to document their designed systems, compare them with other system or discover weaknesses or new room for improving the quality of the adaptation designed. This design space consists of seven axes. Even if we tried to cover the most prominent features of adaptation, it could be extended or specialized to fit specific purposes.

The design space is complemented with an adaptation profile that support the documentation of the coverage of the seven stages of GISATIE, considering which stakeholders can be involved in every stage, and even the number of stakeholder of each type. This profile can be used as a quick reference, and it will help in discovering the lack of coverage of any stage of GISATIE in a system with adaptation capabilities. With these three components of GISATIE the adaptation designer has a powerful framework to design the adaptation facilities with some guiding tools to drive the design.

By applying GISATIE framework on our own systems some future work has been envisioned in different areas. One interesting research line is automatic assessment of adaptation goals, especially after automatic UI generation subject to further adaptation (Aquino et al., 2010). So far, we have been working with cognitive models (López-Jaquero et. al, 2008) for this evaluation, but with the availability of plenty of sensors, including aspects such as biofeedback opens incredible possibilities. The use of remote collaborative assessment for goals, making use of collaborative filtering, could probably improve the assessment and help to overcome one limitation of local assessment: the lack of data until the user interacts with the system during a long amount of time. This happens because in most approaches the amount of data required to produce reliable output from the evaluations is quite big, and it takes time to collect it from the use of the system. On the other hand, deciding what adaptation to apply in a given situation can be made by using meta-rules. These meta-rules would decide the strategies that choose the right rules to apply. The specification of meta-rules is something that can provide versatility to the adaptation system. Using machine learning techniques to promotion or demotion of a particular adaptation rule or technique is also a challenge with interesting perspectives.

Acknowledgements

This paper is part of the R+D+i project 2gether (PID2019-108915RB-I00) funded by MCIN/AEI/ 10.13039/501100011033.

References

Abrahão, S., Insfrán, E., Sluÿters, A., Vanderdonckt, J.: 2021. Model-based intelligent user interface adaptation: challenges and future directions. *Softw. Syst. Model.* 20(5), 1335-1349. DOI: https://doi.org/10.1007/s10270-021-00909-7

Aquino, N., Vanderdonckt, J., Pastor, O.: 2010. *Transformation templates: adding flexibility to model-driven engineering of user interfaces*. In: Proc. of ACM Symposium on Applied Computing SAC'2010, ACM Press, New York, pp. 1195–1202. DOI:

https://doi.org/10.1145/1774088.1774340

Arhippainen, L., Rantakokko, T. and Tähti, M.: 2005. *Navigation with an Adaptive Mobile Map-Application: User Experiences of Gesture- and Context-Sensitiveness*. In: Proc. of 2nd Int. Symposium on Ubiquitous Computing Systems, UCS'2004, Tokyo, November 8-9, 2004. Vol. 3598 of Lecture Notes in Computer Science, Springer, Berlin, pp. 62–73. DOI: https://doi.org/10.1007/11526858_6

Beaudouin-Lafon, M.: 2000. *Instrumental interaction: an interaction model for designing post-WIMP user interfaces*. In: Proceedings of ACM Conference on Human Aspects in Computing Systems CHI'2000, The Hague, April 1-6, 2000. ACM Press, New York, pp. 446–453. DOI: https://doi.org/10.1145/332040.332473

Benyon, D. and Murray, D.: 1993. *Developing adaptive systems to fit individual aptitudes*. In: Proc. of the 1st International Conference on Intelligent User Interfaces IUI 1993, Orlando, Florida. ACM Press, New York, NY, pp. 115-121. DOI: https://doi.org/10.1145/169891.169925

Blouin, A., Beaudoux, O.: 2010. Improving modularity and usability of interactive systems with malai. In: Proceedings of the 2nd ACM SIGCHI symposium on engineering interactive computing systems EICS 2010. ACM Press, New York, 2010, pp. 115–124. DOI: https://doi.org/10.1145/1822018.1822037

Browne, D.P., Sharratt, B. and Norman, M.A.: 1986. The formal specification of adaptive user interfaces using Command Language Grammar. In: Proc. of the ACM Conference on Human Aspects in Computing Systems CHI'1986, Boston, April 13-17, 1986. ACM Press, New York, pp. 256–260. DOI: https://doi.org/10.1145/22339.22381

Bunt, A., Conati, C. and McGrenere, J.: 2004. *What role can adaptive support play in an adaptable system?* In: Proceedings of the 9th International Conference on Intelligent User Interfaces, IUI'2004, Funchal, January 13-16, 2004. ACM Press, New York, pp. 117–124. DOI: https://doi.org/10.1145/964442.964465

Burny, N. and Vanderdonckt, J.: 2021. *UiLab, a Workbench for Conducting and Reproducing Experiments in GUI Visual Design*. Proceedings of the ACM on Human-Computer Interaction, Volume 5, Issue EICS, June 2021 Article No.: 196, pp. 1–31. DOI: https://doi.org/10.1145/3457143

Calvary, G., Coutaz, J., Thevenin, D., Limbourg, Q., Bouillon, L., Vanderdonckt, J.: 2003, A Unifying Reference Framework for multi-target user interfaces. *Interact. Comput.* 15(3), pp. 289-308. DOI: https://doi.org/10.1016/S0953-5438(03)00010-9

Cockton, G.: 1988. Some Critical Remarks on Abstractions for Adaptable Dialogue Managers. In: Proc. of Third Conference of the British Computer Society Human-Interactio on People and computers III, February 1988 BCS HCI 1988, pp. 325-343. DOI: https://dl.acm.org/doi/10.5555/43158.43182

Dessart, Ch.-E., Genaro Motti, V., Vanderdonckt, J.: 2012. Animated transitions between user interface views. In: Proc. of ACM Conf. On Advanced Visual Interfaces AVI 2012. ACM Press, New York, pp. 341-348. DOI: https://doi.org/10.1145/2254556.2254623

Dieterich, H., Malinowski, U., Kühme, T.and Schneider-Hufschmidt, M.: 1993. State of the Art in Adaptive User Interfaces. In: Schneider-Hufschmidt, M., Khüme, T., Malinowski,

U. (Eds.), *Adaptive User Interfaces: Principle and Practice*. North Holland, Amsterdam, pp. 13-48.

Eisenstein, J., Vanderdonckt, J., and Puerta, A.R.: 2000. Adapting to mobile contexts with user-interface modeling. In: Proc. of the Third IEEE Workshop on Mobile Computing Systems and Applications WMCSA 2000, IEEE Press, Los Alamitos, pp. 83-92. DOI: https://doi.org/10.1109/MCSA.2000.895384

Furtado, E., Furtado, V., Silva, W.B., Rodrigues, D.W.T., da Silva Taddeo, L., Limbourg, Q., Vanderdonckt, J.: An ontology-based method for designing multiple user interfaces. In: Proceedings of International Workshop on Multiple User Interfaces, MUI' 01 (2001). https://www.researchgate.net/publication/2567741_An_Ontology-Based_Method_for_Universal_Design_of_User_Interfaces

Horvitz, E.: 1999. *Principles of Mixed-Initiative User Interfaces*. In: Proceedings of ACM Conference on Human Factors in Computing Systems CHI'99, Pittsburgh, May 15-20, 1999. ACM Press, New York, pp. 159–166. https://doi.org/10.1145/302979.303030

Hussain, J., Ul Hassan, A., Syed Muhammad Bilal, H., Ali, R., Afzal, M., Hussain, S., Bang, J., Banos, O., Lee, S. 2018. Model-based adaptive user interface based on context and user experience evaluation. Journal of Multimodal User Interfaces, 12, 2018, pp. 1–16. DOI: https://doi.org/10.1007/s12193-018-0258-2

Jameson, A.: 2003. Adaptive Interfaces and Agents. In: Jacko, J.A., Sears, A. (Eds.), *Human–Computer Interface Handbook*. Lawrence Erlbaum, Mahwah, pp. 305–330.

Kurtev, I, Bézivin, J., Aksit, M.: 2002. *Technological spaces: An initial appraisal.* In: Proc. of 4th International Symposium on Distributed Objects and Applications DOA 2002, University of California, Irvine, United States, 30 Oct.-1 Nov. 2002. URL: https://research.utwente.nl/en/publications/technological-spaces-an-initial-appraisal

Lavie, T., Meyer, J.: 2010. Benefits and costs of adaptive user interfaces. *International Journal of Human-Computer Studies* 68, pp. 508–524. DOI:

https://doi.org/10.1016/j.ijhcs.2010.01.004

Limbourg, Q., Vanderdonckt, J., Michotte, B., Bouillon, L., and Florins, M. 2004. *UsiXML: A User Interface Description Language Supporting Multiple Levels of Independence*. In: Proc. of Workshops in connection with the 4th Int. Conf. on Web Engineering (ICWE '04). Engineering Advanced Web Applications DIWE '04, Matera, M. and Comai, S. (Eds.), Rinton Press, pp. 325–338. URL:

https://www.researchgate.net/publication/220940216_UsiXML_A_User_Interface_Description_Language_Supporting_Multiple_Levels_of_Independence

López-Jaquero, V., Vanderdonckt, J., Montero, F., and González, P.: 2007. *Towards an Extended Model of User Interface Adaptation: the ISATINE framework*. In: Proc. of IFIP WG2.7/13.4 10th Conference on Engineering Human Computer Interaction jointly organized with IFIP WG 13.2 1st Conference on Human Centred Software Engineering and DSVIS - 14th Conference on Design Specification and Verification of Interactive Systems, EIS'2007, Salamanca, March 22-24, 2007, J. Gulliksen, M.B. Harning, Ph. Palanque (Eds.). Vol. 4940 of Lecture Notes in Computer Science. Springer, Berlin, pp.

374–392. DOI: https://doi.org/10.1007/978-3-540-92698-6_23

López-Jaquero, V., Montero, F. and González, P.: 2008. *Quality of Adaptation: User Cognitive Models in Adaptation Quality Assesment*. In: Proc. of 6th International Conference on Computer-Aided Design of User Interfaces, CADUI'2008, Albacete, June 11-13, 2008. López Jaquero, V., Montero, F., Molina, J.P., Vanderdonckt, J. (Eds.)., Springer, Berlin, pp. 265–276. DOI: https://doi.org/10.1007/978-1-84882-206-1_24

López-Jaquero, V., Montero, F. and Gonzalez, P.: 2009. AB-HCI: an interface multi-agent system to support human-centred computing. *IET Software* 3, 14. pp. 14–25. DOI: https://doi.org/10.1049/iet-sen:20070108

McKinley, P. K., Sadjadi, S. M., Kasten, E. P., and Cheng, B.H.: 2004. Composing Adaptive Software. *IEEE Computer* 37, 7. pp. 56–64. DOI: https://doi.org/10.1109/MC.2004.48

Motti, V.G., Vanderdonckt, J.: 2013. *A computational framework for context-aware adaptation of user interfaces*. In: Proceedings of the 7th IEEE International Conference on Research Challenges in Information Science RCIS '13. IEEE Computer Society Press, Los Alamitos, USA, pp. 1–12. DOI: https://doi.org/10.1109/RCIS.2013.6577709

Rosaci, R., Sarnè, G.M.L.: 2006. MASHA: A multi-agent system handling user and device adaptivity of Web sites. *User Modeling and User-Adapted Interaction* 16, 5. pp. 435–462. DOI: https://doi.org/10.1007/s11257-006-9015-4

Schlee, M. and Vanderdonckt, J.: 2004. *Generative Programming of Graphical User Interfaces*. In: Proceedings of 7th Interna-tional Working Conference on Advanced Visual Interfaces, AVI'2004, Gallipoli, May 25-28, 2004. pp. 403-406. ACM Press, New York. DOI: https://doi.org/10.1145/989863.989936

Sendín, M., López-Jaquero, V. and Collazos, C.A.: 2008. Collaborative Explicit Plasticity Framework: a Conceptual Scheme for the Generation of Plastic and Group-Aware User Interfaces. *Journal of Universal Computer Science* 14, 9. pp. 1447–1462. DOI: http://dx.doi.org/10.3217/jucs-014-09-1447

Totterdell, P. and Rautenbach, P.: 1990. Adaptation as a Problem Design. In: Browne, D., Totterdell, P., Norman, M. (Eds.), *Adaptive User Interfaces*. Computer And People Series, Acadamic Press, pp. 59–84.

Yigitbas, E., Jovanovikj, I., Biermeier, K., Sauer, S., Engels, G.: 2020. Integrated Model-driven Development of Self-adaptive User Interfaces. *International Journal on Software and Systems Modeling*, 19, pp. 1057–1081. DOI: https://doi.org/10.1007/s10270-020-00777-7.