Engineering Slidable Graphical User Interfaces with SLIME

ARTHUR SLUŸTERS, Université catholique de Louvain, Belgium
JEAN VANDERDONCKT, Université catholique de Louvain, Belgium
RADU-DANIEL VATAVU, Ştefan cel Mare University of Suceava, Romania

Intra-platform plasticity regularly assumes that the display of a computing platform remains fixed and rigid during interactions with the platform in contrast to *reconfigurable displays*, which can change form depending on the context of use. In this paper, we present a model-based approach for designing and deploying graphical user interfaces that support *intra-platform plasticity for reconfigurable displays*. We instantiate the model for E3SCREEN, a new device that expands a conventional laptop with two slidable, rotatable, and foldable lateral displays, enabling *slidable user interfaces*. Based on a UML class diagram as a domain model and a SCRUD list as a task model, we define an abstract user interface as interaction units with a corresponding master-detail design pattern. We then map the abstract user interface to a concrete user interface by applying rules for the reconfiguration, concrete interaction, unit allocation, and widget selection and implement it in JavaScript. In a first experiment, we determine display configurations most preferred by users, which we organize in the form of a state-transition diagram. In a second experiment, we address reconfiguration rules and widget selection rules. A third experiment provides insights into the impact of the lateral displays on a visual search task.

CCS Concepts: • **Human-centered computing → Graphical user interfaces**; *Interactive systems and tools*; • **Software and its engineering → Graphical user interface languages**; **System modeling languages**; *Model-driven software engineering*; Runtime environments;

Additional Key Words and Phrases: Adaptation; Inter-platform plasticity; Intra-platform plasticity; Plasticity of user interfaces; Reconfigurable display; Rotatable display; Slidable screen; Slidable User Interface.

ACM Reference Format:
Arthur Sluÿters, Jean Vanderdonckt, and Radu-Daniel Vatavu. 2021. Engineering Slidable Graphical User Interfaces with SLIME. *Proc. ACM Hum.-Comput. Interact.* 5, EICS, Article 200 (June 2021), 29 pages. https://doi.org/10.1145/3457147

1 INTRODUCTION

The term "plasticity" expresses the capacity of biological tissues to undergo continuous deformations to adapt to external constraints in order to preserve function without rupture. In applied materials, plasticity refers to the capacity of a material to be subjected to deformations, such as pressure and extension, and stay in the deformed state when the external force stops. Applied to interactive systems, the plasticity is defined as "the ability of user interfaces to withstand variations of the context of use while preserving usability" [6, 7] and to preserve human values [8, 12].

The plasticity of UIs has been examined in a large body of works [7–9, 15, 16, 19, 30, 39, 43, 48, 55, 59] that addressed the problem of adapting user interfaces to contextual variations. The *context*

Authors' addresses: Arthur Sluÿters, Université catholique de Louvain, LouRIM Institute, Place des Doyens, 1, Louvain-la-Neuve, 1348, Belgium, arthur.sluyters@uclouvain.be; Jean Vanderdonckt, Université catholique de Louvain, LouRIM Institute, Place des Doyens, 1, 1348, Louvain-la-Neuve, Belgium, jean.vanderdonckt@uclouvain.be; Radu-Daniel Vatavu, Ştefan cel Mare University of Suceava, MintViz Lab, MANSiD, 13 Universitatii, Suceava, 720229, Romania, radu.vatavu@usm.ro.

2573-0142/2021/6-ART200 $15.00
https://doi.org/10.1145/3457147

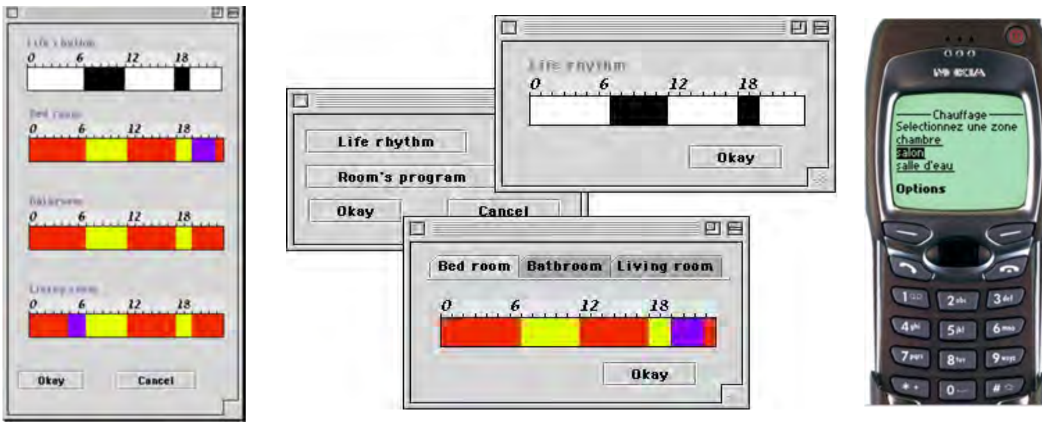


Fig. 1. An example of a mixed-platform plastic user interface [7].

of use [8, 14] is defined as the triple $C = (U, P, E)$, where U denotes a model of the user and their tasks, P denotes a platform model, and E a model of the environment.

Among these three dimensions, the platform has received considerable attention [7, 32, 68], probably because its parameters, such as display size, display resolution, and display density largely affect the usability of an interactive system. *Intra-platform plasticity* addresses a specific platform, *e.g.*, one display. *Inter-platform plasticity* covers interactive scenarios involving multiple platforms, *e.g.*, from a desktop computer to a smartphone. *Mixed-platform plasticity* combines the former, *e.g.*, MMS [7] has employed a constraint solver to produce mixed-platform plastic UIs (Fig. 1): when the display resolution is large enough, multiple features of a house energy monitoring system are displayed at once (left); when the resolution is reduced, separated windows are generated (center); and when mobile, system features are displayed sequentially (right).

Intra-platform plasticity has assumed that the platform, such as the screen, the display, or the monitor [24], remains fixed as opposed to *reconfigurable displays* [47] that can change forms. Consequently, there are no model-based approaches available to support intra-platform plasticity for reconfigurable displays [20, 56]. In this context, the contributions of this work are as follows:

- We revisit the state-of-the-art for platform user interface plasticity by addressing the intra-, inter-, and mixed-plasticity concepts (Section 2).
- We refine intra-platform plasticity for the case of reconfigurable displays by considering four properties of such displays: extensibility, extendability, expandability, and extractability. We illustrate these properties with E3Screen, a new prototype designed to enhance any flat screen, *e.g.*, of a laptop, with two slidable, rotatable, and foldable lateral displays (Section 3).
- We introduce Slime, a model-based approach for designing and deploying Graphical User Interfaces (GUIs) that supports the four properties, and we illustrate two interactive applications: (i) an on-line newspaper and (ii) a car rental application (Section 4).
- We conduct three experiments. In a first experiment, we determine user preferred configurations for E3Screen, which we organize in the form of a state-transition diagram for a configuration model (Section 5). In a second experiment, we determine reconfiguration and widget selection rules (Section 6). Finally, we conduct a third experiment to investigate the impact of E3Screen on a visual search task (Section 7).

We conclude with a discussion in Section 8 regarding the advantages and limitations of the process induced by Slime and E3Screen, and propose directions for future work.

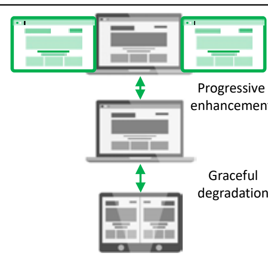
Plasticity type		Illustration	References
Intra-platform			Elastic Windows [31], Nokia applet [32], PlastiXML [13], FWL [68], Layout Hints [54], AAUI [27], OR-Constraints [29]
Mixed-platform		both	MMS [7], COMETS [17], FlexClock [21]
Inter-platform	Fixed display		Abstract Interface [57], Attach Me [23], DistriXML [41]
	Reconfigurable display		this paper

Table 1. State-of-the-art for platform plasticity.

2 STATE OF THE ART

Table 1 provides an overview of selected work from platform-plastic GUIs: intra-platform when plasticity is implemented within the same platform (*e.g.*, on a desktop computer), inter-platform when plasticity is implemented across two or more platforms (*e.g.*, from a desktop computer to a smartphone), and mixed-platform when the previous cases are combined.

2.1 Intra-platform Plasticity

"Elastic Windows" [31] fall in this category as windows are automatically resized depending on constraints imposed by the tasks carried by the user. According to the user role, *e.g.*, organizer of meetings, windows resize to minimize overlapping and maximize visibility. The "Nokia Applet" [32], used to manage a set-top box, a media player, and a smart TV, resizes its window and expands and collapses a hierarchy of widgets according to resolution of the display and user's task (Fig. 2). When the user employs the media player, the hierarchy of controls for this device is expanded, while other hierarchies are collapsed. Animated transitions [18] ensure a smooth continuation between the two states of the user interface. "PlastiXML" [13] defines the plasticity of a point, a set of (x, y) coordinates where the GUI can change according to display size and resolution constraints. The designer specifies all the plasticity points in the editor and produces a Mealy machine as a model. The "Flexible Widget Layout" (FWL) [68] solves a local Constraint Satisfaction Problem (CSP) specified by constraints imposed on widgets, their types, location, and size.[1] A Java engine restructures the contents of a Java window automatically according to these constraints.

[1]See http://takty.stxst.com/res/fwl/index.en.html

Fig. 2. A Java applet for intra-platform plasticity with automatic restructuring [32].

Instead of solving the layout problem analytically, Raneburger *et al.* [54] adopted a two-step heuristic approach in which GUI layout parameters are specified in a platform-independent manner via reusable transformation rules. Missing or platform-dependent parameters are determined automatically based on "Layout Hints" and scrolling preferences. This method has been used to (semi-)automatically generate a UI layout for a particular platform, but not for restructuring it at run-time. "Adaptive layout for Automotive UIs" (AAUI) [27] generates GUIs for in-vehicle interaction, for which screen constraints are known at design time and, thus, do not require run-time restructuring. "OR-Constraints" [29] are introduced to solve the constraint-based GUI layout by accommodating several variations of the aspect ratio with the ORCSolver adaptive GUI renderer.

2.2 Inter-platform Plasticity

The "Abstract Interface" [57] exploits the logical definition of the user data and the task to automatically select which UI components should be employed at run-time and to structure the GUI layout accordingly. "Attach Me/Detach Me" [23] enables users to detach parts of the GUI, *e.g.*, a toolbar, from one platform and to move it to another. This scenario instantiates a Distributed User Interface (DUI) [42], which is at the heart of "DistriXML" [41], where moving a GUI from one platform to another renders it again. Although such a type of plasticity is achieved during run-time, the characteristics of the platform remain constant since their displays are fixed. Graceful degradation [19] and its counterpart, progressive enhancement [44], occur when the plastic GUI is moved from one platform, *e.g.*, a desktop computer, to a smaller one such as a smartphone and, respectively, to a larger one, such as a wall display.

2.3 Mixed-platform Plasticity

MMS [62] (Fig. 1) supports mixed-platform plasticity by changing within a given platform and across platforms alike. The "Comets" (COntext sensitive Multi-target widgETS) [17] exploit a decision graph to select automatically a particular widget or component based on the data to display, the task, and the constraints of the target platform. Mixed-platform plasticity is expressed directly at the level of a "Comet" instead of the level of the interactive application.

Restructuring the UI layout can occur when different platforms are coupled to form a new ecosystem [3], which can be decoupled afterwards. The new interaction surface resulting from this process is created and managed by the system, but the windows contained in this space are not necessarily managed automatically. "FlexClock" [21] displays the date and time of the day depending on the length and height of the container window according to twenty-six different layouts computed at run-time. For example, the date can be displayed as a label or as part of the calendar.[2] "FlexClock" supports intra-platform plasticity as it resizes its GUI depending on the window dimensions, and inter-platform plasticity since this the application runs on top of Qtk [22].

[2]See https://www.youtube.com/watch?v=bCW9CjaDQAA

Sendin *et al.* [59] differentiate plasticity frameworks by whether they work at design-time (with respect to a predictive context of use) or run-time (with respect to an effective context of use), since not all the contexts of use can be predicted at design time. To support group awareness, Roda *et al.* [55] defined a dichotomic view of plasticity, where the plasticity framework is duplicated, once for predictive contexts of use and again for effective contexts of use.

All platforms considered in the prior work have a fixed form factor. None of the prior work, to the best of our knowledge, can support plasticity when the platforms have variable form factors, such as in the case of reconfigurable displays [20, 33, 47]. In their survey for characterizing the dimensions of the context of use, Roda *et al.* [55] observed that a number of 53 references (representing 11.45% of the surveyed pool) had considered the display or platform in plasticity (platform-plasticity), among which a number of 46 (9.44%) considered screen size, 8 references (1.73%) considered screen resolution, and 3 references (0.65%) other aspects of the display.

3 PROPERTIES OF RECONFIGURABLE DISPLAYS

To specify plasticity for reconfigurable displays, we rely on the following four specific properties [66] for displays that can change shape:

(1) *Extensibility* refers to the quality property of a platform to increase its physical form factor by deformations of the material from which it was made. In our case, it refers to the GUI quality property to extend itself based only on intrinsic, embedded, or embarked features. For example, Fig. 2 demonstrates such a capability since all the variations are foreseen at design-time, *i.e.*, predictive plasticity according to [59].
(2) *Extendability* refers to as the quality property of a platform to increase its physical form factor through the addition of new components. For example, a new platform in a new environment can be connected to existing platforms from that environment [30]. In our case, the property refers to the GUI quality property to extend itself by adding new components not expected at design time, but are possible at run-time, *i.e.*, effective plasticity according to [59]. For example, an attachable user interface [23] receives a new component at run-time that is "plastified" in a free placeholder.
(3) *Expandability* refers to the quality property of a platform to increase its physical form factor through the reuse, restructuring, and reconfiguration of its internal modules and components. In our case, it refers to the GUI quality property to restructure itself based on the components included in it. For example, re-layouting [45] connects to this property.
(4) *Extractability* refers to the quality property of a platform to allow removal of its components that can be used independently of the platform from which they originated. In our case, it refers to the GUI quality to remove some of its components. For instance, a detachable user interface [23] is able to cut some of its parts and remove them from the display.

To explore these properties for reconfigurable displays, we consider E3SCREEN [66], a prototype in the form of an overlay containing two identical display panels, acting as secondary and tertiary displays, respectively, that can be attached to the backside of any tablet, laptop, or PC screen representing the main or primary display. Each panel slides laterally and can be rotated up to 180°, enabling a complete rotation in either direction; see [66] for details and Fig. 3 for a photograph. Hence, contrarily to double-monitor [24] or triple-monitor [39] systems, E3SCREEN is reconfigurable and implements *extendability* (*i.e.*, the overlay is added to the primary display), *expandability* (parts of the overlay can be reconfigured, *e.g.*, slided, rotated, folded, to change the form factor of the multi-display system), and *extractability* (the display panels can be removed from the overlay and used independently) (Fig. 4). E3SCREEN enlarges the user's visual field [5] and doubles and, respectively, triples the interaction surface to form an adjustable [34] and rotatable panoramic view.

Fig. 3. E3Screen employed as a reconfigurable display. Image source: Slidenjoy with permission.

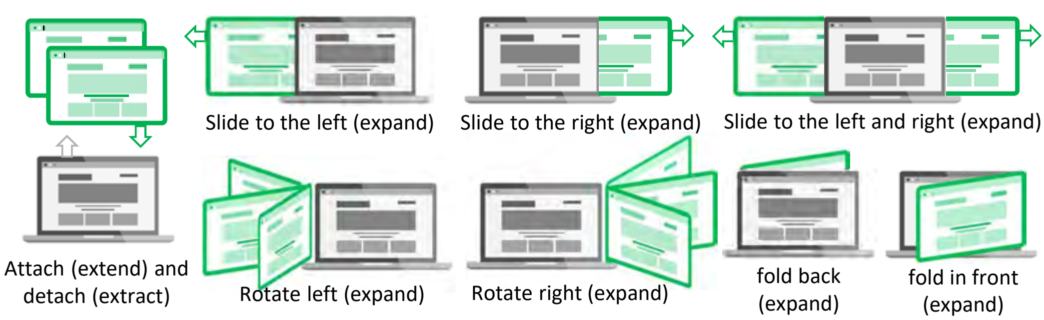


Fig. 4. Extendability, extractability, and expandability properties of E3Screen [66].

Unlike a setup with three fixed screens and unlike a monitor stand with a table mount for three monitors, E3Screen combines a wide and flexible interaction surface with minimal cluttering.

4 THE SLIME MODEL-BASED APPROACH FOR SLIDABLE USER INTERFACES

4.1 Model-based Approach

Fig. 5 shows the Slime model-based approach for intra-platform plasticity according to the following four levels of abstraction of the Cameleon Reference Framework (CRF) [8]:

- *Task & Concepts*: located at the Computing Independent Model (CIM) level [51], the domain model is represented as a UML V2.5 Class Diagram and stored as a XML file to facilitate parsing. The task model consists of a list of potential SCRUD operations (Search-Create-Read-Update-Delete [52]) that can be performed on the classes of the domain model (Fig. 6) [1]: create a new object, retrieve an existing object, update an existing object, delete an existing object, and search for a given object.
- *Abstract User Interface*: located at the Platform Independent Model (PIM) level, a configurator for the Abstract User Interface (AUI) applies recursively a master-detail design pattern [51] for the classes of the domain model connected via aggregation. This process results in a chain of relationships from the master class.

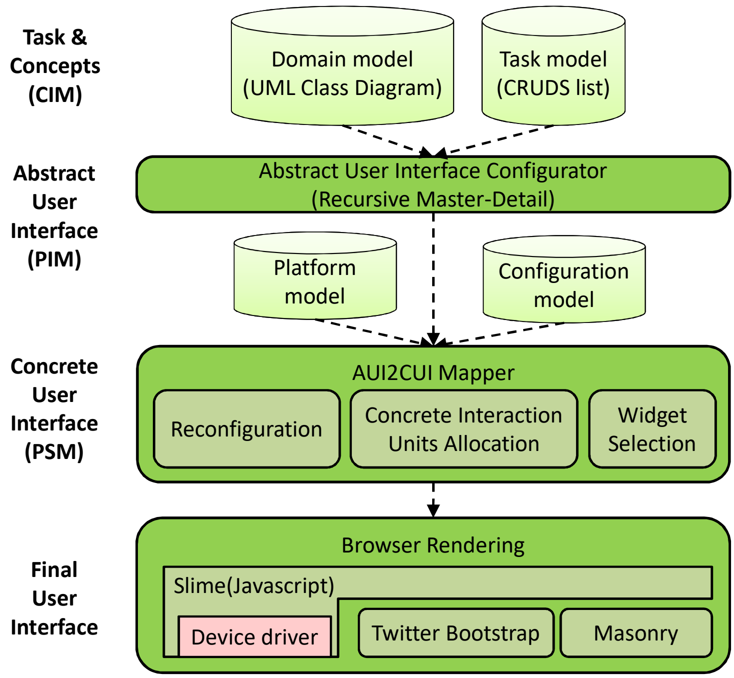


Fig. 5. The SLIME model-based approach following to the Cameleon Reference Framework [8].

- *Concrete User Interface*: located at the Platform Specific Model (PSM) level, the abstract user interface is mapped onto a concrete user interface by exploiting three sets of rules —reconfiguration, allocation, and widgets selection—on two models—a platform model and a novel configuration model—detailed next.
- *Final User Interface*: SLIME, a JavaScript environment that renders the concrete user interface at run-time via a device driver, Twitter Bootstrap, and Masonry.

4.1.1 Initiating the Task & Concepts. Fig. 6, right, illustrates the simplified domain model with a sequence of classes, classes attributes and properties, and semantic relations between classes. The master-detail design pattern identifies the starting class and the detail class by following any relationship between the two classes. The design pattern is then applied recursively: relations are followed from a source class to a target class to create a sequence until relations are exhausted. Any relationship, other than "Is-Part-of" (aggregation or composition), is transformed into a directed composition to connect the classes in the sequence. Other relations are not supported in our case to keep the process simple. Each class comes with a predefined set of SCRUD methods with permissions stored in roles. For example, a class could be specified as viewable only (in which case only the Read method is permitted), editable (only the Update method is permitted), or inputtable (only the Create method is permitted).

4.1.2 Transformation to AUI. Fig. 6 depicts how the domain model is transformed into an abstract user interface. The sequence of classes determined in the previous step is mirrored into a sequence of abstract interaction units with the same names and attributes. The resulting AUI consists of a hierarchy of Abstract Interaction Units (AUIs) following the W3C definition.[3]

[3]See https://www.w3.org/TR/abstract-ui/.

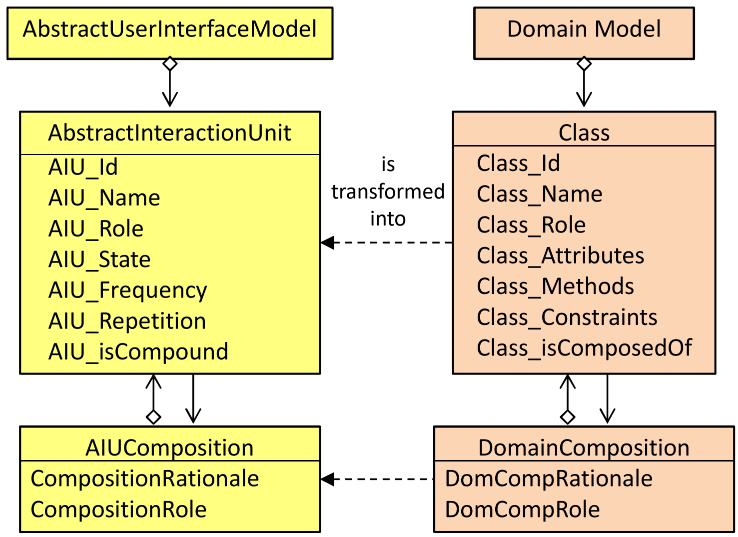


Fig. 6. Domain to AUI model-2-model transformation.

4.1.3 Transformation to CUI. To transform the platform-independent AUIs into one or many CUIs, two models are exploited. A *platform model* consists of a static model (predictive [59]) specified at design time that captures the most important attributes of the platform affecting its rendering, such as the number of displays, their size, resolution, DPI, number of colors, etc. For example, the platform model of E3Screen specifies that the main display has a 15-inch screen diagonal and each panel has full-HD 1920×1080 pixel resolution. Second, a *configuration model* consists of a dynamic model (effective [59]) that is specified once at design-time and updated at run-time depending on the configuration of the displays. As opposed to the platform model, the configuration model evolves by exploiting the E3Screen configurations as follows:

(1) The number of displays (one, two, or three) that are used simultaneously, following Truemper *et al.*'s [64] treatment of usability aspects for multiple monitor displays.
(2) The number of available displays connects to the number of panels that are employed by a particular use case scenario (none, left, right, or both) [34, 49].
(3) The orientation of the rotatable displays with respect to the primary display, from $-180°$ to $+180°$, according to the discussions from [58, 63].

The AUI of the previous step is mapped onto a Concrete User Interface, made up of Concrete Interaction Units (CIUs) by using sets of rules contained in three modules:

(1) *Reconfiguration*: CIUs are allocated to displays as specified in the platform model with the hierarchy root on the main display and sub-CIUs on additional displays. For instance, in case of an Update method, the list of objects of the class of interest is assigned to the main, central panel, while the editable CIU is assigned to a lateral display.
(2) *CIU Allocation*: a graphical container is allocated to each abstract interaction unit as previously defined. The sequence of AIUs is mapped onto a hierarchy of graphical containers with the application window as the first source one, then the group boxes, etc.
(3) *Widget selection*: a widget is selected for each class attribute depending on the role played by the SCRUD methods according to simple selection rules, *e.g.*, [2, 53, 65]. For example, a Last name attribute will select an editable field for an Update method or a text for a Read method.

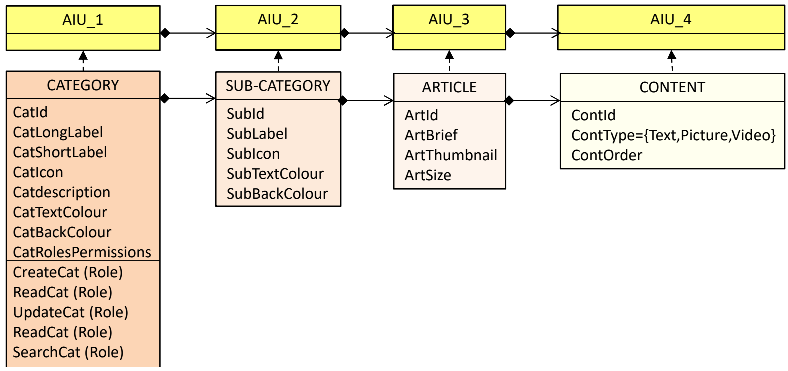


Fig. 7. The domain model of the on-line newspaper application and its corresponding AUI.

4.1.4 Rendering of the FUI. The Final User Interface (FUI) is rendered in SLIME as a Javascript code based on three components (Fig. 5): a device driver, Twitter Bootstrap, and Masonry.

A custom device driver developed for E3SCREEN extends the resolution of the primary display on the two lateral displays and detects the parameters for the configuration model: the number of displays and their size and orientation.

Twitter Bootstrap[4] is a CSS and JavaScript toolkit that includes a catalog of pre-defined layouts, structures, and shapes. GUIs are structured as hierarchies of UI elements, which are recursively decomposed into sub-components until leaf nodes. This can be achieved via several User Interface Description Languages [38, 67]. The various levels of the hierarchy can be mapped onto such pre-defined layouts to be embedded in the code. Thus, Twitter Bootstrap is an efficient tool to easily design the layout from scratch for various devices and resolutions, including the resolution of an expanded display. This toolkit allows us to build quickly "responsive design" [40] that adapts automatically the design of the website according to the device or the resolution of the display. We can describe each layout based on media queries. A *media query* is a part of code dedicated to a specific media in the CSS code. Each media query runs in pair with a device (or resolution) so that the browser can use the appropriate part of the CSS. Each time a display is expanded, the CSS is used to adapt the layout via the correct media queries according to the current resolution.

JQuery Masonry[5] is a jQuery plugin that arranges the elements of a website to fit the browser viewport as best as possible. To achieve this desiderate, jQuery Masonry assigns each element to the viewport grid to fill as most gaps as possible. The result is a more compact display of the layout. The jQuery Masonry mechanism enables rearranging the main structure of the layout depending on the display configuration. The developer needs to specify how the UI divides into several logical elements, after which Masonry can perform the reorganization automatically while users modify the resolution of the display or the browser viewport.

4.2 Application #1: The On-line Newspaper

The first application that we implemented with SLIME was an on-line newspaper. Its domain model consists of a sequence of four classes (Fig. 7): a category of news is decomposed into subcategories, each subcategory holds a certain number of articles, and each article contains an ordered table of contents of three types: text, picture, and video. We used a catalog of 139 articles falling into several categories ("Top news", "National news", "International news", "Unusual news"), which

[4]See https://getbootstrap.com/2.0.2/
[5]See https://masonry.desandro.com/

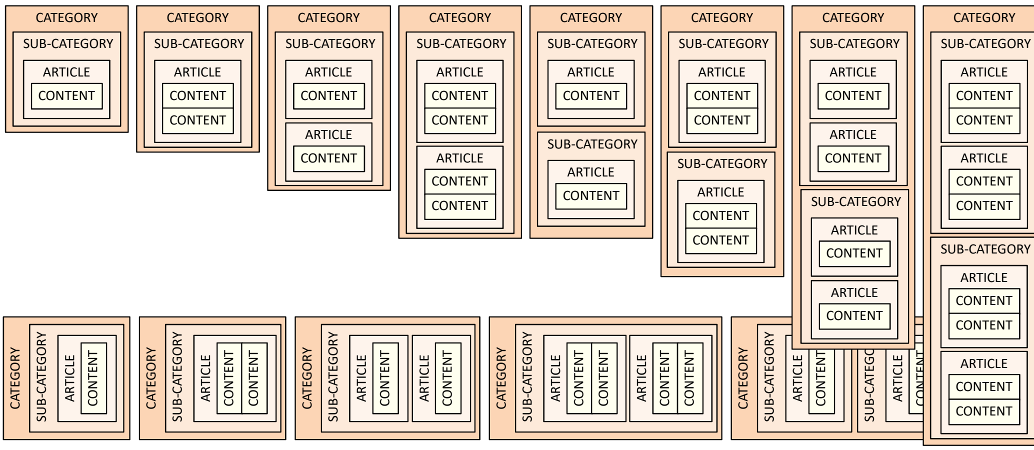


Fig. 8. Allocations of the interaction units for the on-line newspaper application: vertical *vs.* horizontal.

Fig. 9. The on-line newspaper application in Bootstrap (left) and Masonry (right).

were decomposed in subcategories. The original on-line newspaper displayed a fixed menu of categories on the left side of the screen and the current article on the right side in one large column. The subcategories could be expanded or collapsed according to display configurations (Fig. 8). This original design did not provide a genuine "Overview+Contents" approach. Therefore, we restructured the UI in terms of categories and subcategories, article sections, and level of detail for each sub-section. All these levels could thus be manipulated into rectangular regions (Fig. 9).

The steps of an expansion to the right (Fig. 10) are as follows:

(1) The primary screen is in its normal position.
(2) The user expands a lateral screen to the right.
(3) The UI detects the expansion event and triggers a restructuring mechanism by passing as parameters the coordinates of the window following the expanding surface.
(4) The initial coordinates of the window $\{(x_1, y_1), (x_2, y_2)\}$ are expanded to the target window $\{((x_1, y_1), (x3, y_3)\}$, where $y_3{=}y_2$ in the case of a full screen window that is expanded.
(5) The restructuring mechanism renders the new configuration of the slidable UI.
(6) When the user stops the expansion, the mechanism stops.

Fig. 10. Expandability of the slidable user interface to the right.

Fig. 11. The car rental application in Bootstrap (left), Masonry (center), and combined variants (right).

4.3 Application #2: The Car Rental Web Site

Our second application was a car rental on-line application that was previously implemented in one small page with tab navigation: the first tab contains information about the the user, the second about the user address, the third about the user's preferences about cars, and the last tab enables posting comments and submitting the rental form. This wizard-like UI design does not permit adaption of the content according to the characteristics of the display. We thus decided to restructure the content into seven sections: personal information, address, car information, booking duration, additional equipment selection and options, comments and form submission, and summary of the booking; see Fig. 11.

5 EXPERIMENT #1: PREFERRED CONFIGURATIONS FOR LATERAL DISPLAYS

To understand users' preferences for display configurations and slidable UIs, we conducted an experiment with the following null hypothesis:

> H_0: *There is no difference in user preference between the normal configuration (i.e., one primary screen only) and other configurations when E3Screen is deployed.*

5.1 Method

5.1.1 Participants. A total number of N=103 participants (34 female) aged between 15 and 65 years (M=28.9, SD=12.3, Mdn=24.0 years) were recruited via contact lists. Participants' professional occupations included secretary, teacher, director, psychologist, self-employed, unemployed, retired, and students in Engineering, Law, Economics, Physiotherapy, Management, and Criminology. All the participants reported frequent use of computers and smartphones for various purposes, such as email (83.5%), social networks (68.9%), watching videos on YouTube (79.6%), web browsing (88.3%), document writing (78.6%), playing video games (31.1%), and reading online press (48.5%). About 75% of the participants were already using at least two displays (*e.g.*, a laptop and a smartphone) on a regular basis. A percent of 62% of the participants was aware of the fact that using at least one additional screen could help increase their productivity in the workplace [24] and reduce workload [60]. A number of 94 participants (91%) were right-handed and 9 (9%) were left-handed. We assigned participants to four groups: students (29.1%), lower executive and administrative people (30.1%), senior executive up to top managers (17.5%), and the rest of the respondents that

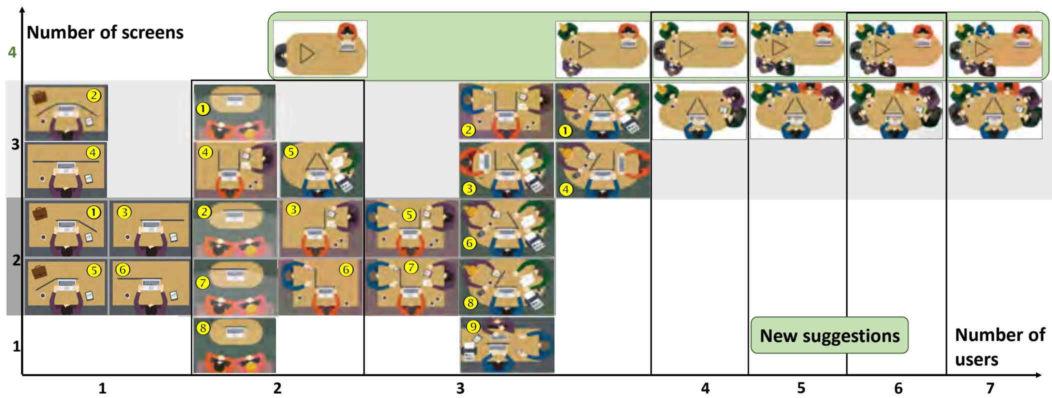


Fig. 12. Display configurations from our first experiment in decreasing order of their frequency.

declared themselves independent, retired, or unemployed. None of the participants was not familiar with the E3Screen prototype.

5.1.2 Apparatus. The prototype running on a Sony VAIO FIT E laptop was placed in the middle of a table in a meeting room so that the physical distance to the participant was between 18 and 24 inches (46 and 61 cm, respectively).[6] The prototype was switched on, showing a typical Windows desktop with folders, shortcuts, and applications that could be launched.

5.1.3 Procedure and Task. After signing a consent form, filling in a socio-demographic questionnaire, and watching a demonstration video, participants were asked to propose seven display configurations corresponding to use cases for one user (*i.e.*, the participants themselves) to seven users. Once ready, they demonstrated the configurations.

5.1.4 Experiment Design. Our study was a within-subjects design [10] with one independent variable, the number of users (from 1 to 7), for which display configurations were elicited. This design resulted in 103 participants × 7 configurations = 721 proposals.

5.2 Results

Fig. 12 shows the display configurations proposed by our participants in decreasing order of their frequency; see [66] for more details. The most frequent configuration for one user (45/103=44%) was the right panel deployed at an angle of 45°, since most users working alone deploy first the panel corresponding to their dominant hand; see the illustration ① in the leftmost column of Fig. 12. When users need more screen space, the left panel is expanded at a similar angle (②=39/103=38%). Flat configurations come afterwards: the right panel is first deployed (③=7/103=5%) with the left one after (⑥=3/103=3%) or both (④).

The most frequent configuration for two users (①=71/103= 69% in the second column of Fig. 12) was the full flat followed by the lateral flat (right: ②=11/103=11%; left: ⑦= 2/103= 2%). Next came the "L-shaped" configuration (right: ③=8/103=8%; left: ⑥ = 3/103=3%), the "U-shaped" (④=4/103=4%), and the "triangle" configuration (⑤=3/103=3%), respectively. At this stage of the experiment, a new configuration was suggested: the "remote triangle", which arranges the three panels in a triangle,

[6] The distance between the user and the screen should be of at least 3 inches (7.6 cm) and 25 inches (63.5 cm) according to the device control perspective [46]. Moreover, displays should not be positioned above 75°from the horizontal line of sight and at a maximum angle of 45°, according to [25, 46].

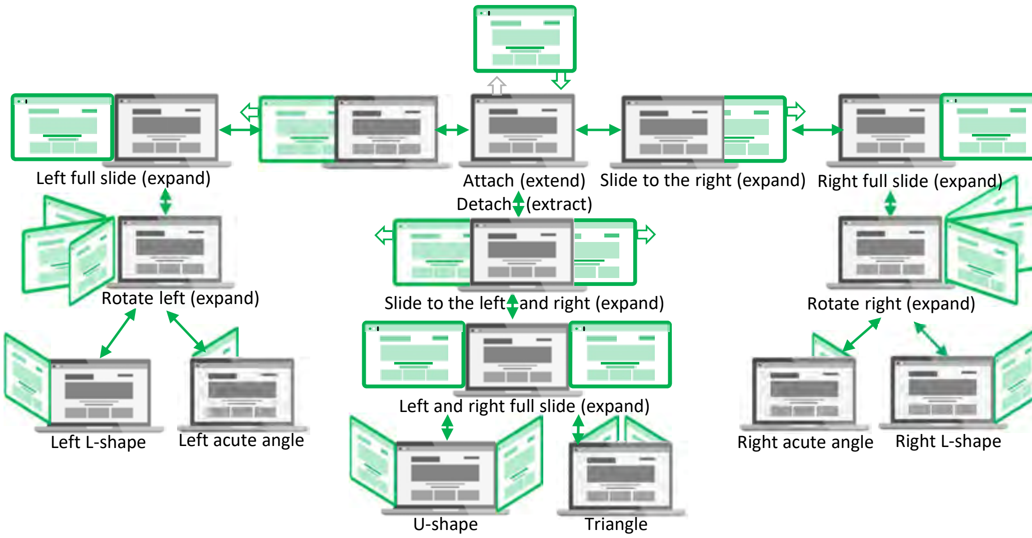


Fig. 13. The state-transition diagram for the most preferred display configurations.

Tasks given	**Order of preference**
Reading newspapers + one secondary task	1;3;2;4
Writing a document + one secondary task	1;3;2;4
Watching a video + one secondary task	1;2;3;4
Playing videos games + one secondary task	1;2;3;4
Reading newspapers + two secondary tasks	3;4;1;2
Writing a document + two secondary tasks	3;4;1;2
Watching a video + two secondary tasks	3;4;1;2
Playing videos games + two secondary tasks	3;4;1;2

Table 2. Participants' preferences for display configurations according to the task to perform.

but positions the device towards the opposite side of the table with a connecting cable to the host computer. This configuration is added in green in Fig. 12.

The most frequent configuration for three users was the "triangle", followed by "U-shaped." Another possible configuration combines a right angle on one side and an obtuse angle on the opposite panel (3=10/103=10% and 4=6/103=6%). For four to seven users, the "triangle" was always the one configuration that was proposed by the participants.

Participants were also asked which configuration from the catalogue reproduced in Fig. 14 was preferable under specific situations that were presented to them as statements. To provide participants with a representative context, the slidable UI before and after expansion (Fig. 15 and 16) were demonstrated. Table 2 shows the most frequently selected configurations.

5.3 Design Considerations

Based on our results, we present design considerations for reconfigurable displays:

- *Deploy the panels to match the number of users or groups.* Participants opened displays depending on the amount of avatars, without any apparent need for deploying an extra display if the number of users did not require it. Exceptions occurred for less than two users.

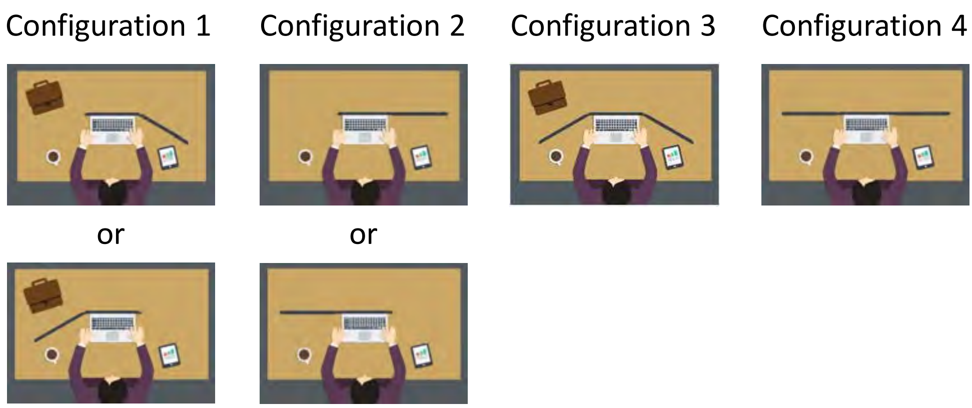


Fig. 14. The four display configurations tested during the experiment.

- *Orient displays to match the partitioning of people around the table.* Participants oriented the displays towards the avatars they imagined sitting around the table with one display per user. The right angle and the triangle were the most preferred configuration for two and more than three users, respectively.
- *Prefer configurations with maximal exposure.* When the number of avatars was small, no particular constraint was imposed to the participants. However, when the number increased, thus posing constraints in terms of the physical space available to the users, participants switched to configurations maximizing the visual field of view for all the users, while also optimizing physical space. This aspect becomes significant for more than two users.
- *Prefer symmetrical configurations of displays.* The most frequently proposed configurations were symmetrical: the full flat screen for two users, followed by the "U-shaped" and the "triangle" for three users. Inverted configurations, such as the "L-shaped" on the right panel for right-handed users or on the left for left-handed users, also relate to symmetry.
- *Adjust the display orientation according to the degree of formality.* Two users that have a close relationship, *e.g.* collaborators, friends, relatives, can share a full flat configuration. When the relationship is more formal, the angle of the corresponding panel increases. Similarly, the "U-shaped" was considered less formal for three users in collaboration than the "V-shaped" with an obtuse angle. A wider angle indicates more formality between stakeholders, a more distant relationship, and/or a smaller degree of involvement in the collaboration.
- *Exploit the "remote triangle" for more users.* The "remote triangle" can support more formal configurations when many users are involved. The distance between the owner and the rest of the group reflects the potential degree of formality or prevents the rest of the group from interfering with the owner, *e.g.*, during a collaboration scenario when a vendor delivers a presentation to multiple stakeholders simultaneously. When formality is not important, we found that some of the participants switched from grouping several "remote triangles" to a "star-shaped" one.
- *Support for deterritorialization and reterritorialization.* Reterritorialization is the restructuring of a space that has experienced deterritorialization before. While deterritorialization and reterritorialization co-exist, relative deterritorialization, which is always accompanied by reterritorialization, is distinguished from absolute deterritorialization, which gives rise to a permanent displacement. Configurable displays should support this phenomenon, *e.g.*, by capturing operations performed to undo/redo them.

Fig. 15. A selection of slidable user interfaces before expansion.

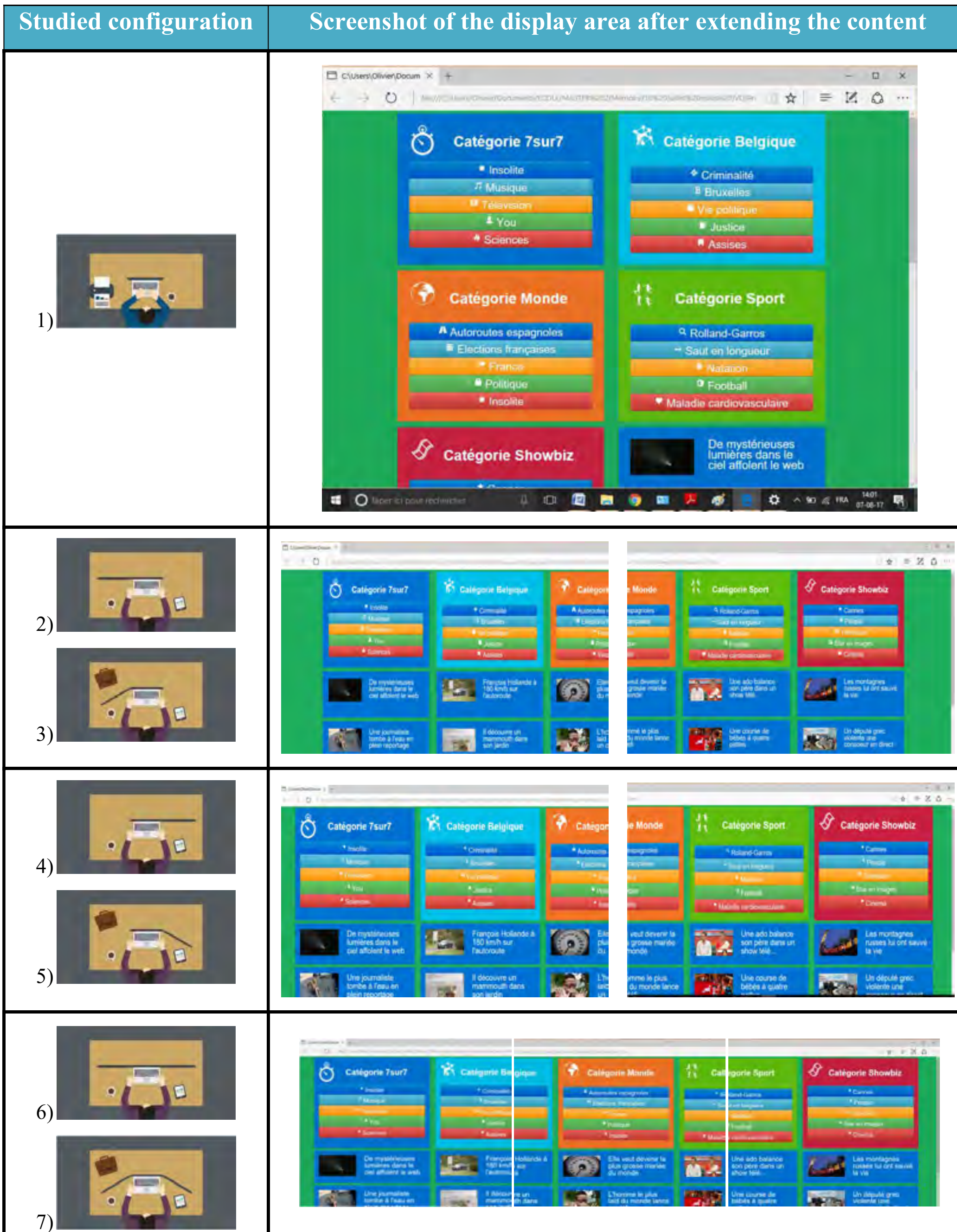


Fig. 16. A selection of slidable user interfaces after expansion.

6 EXPERIMENT #2: COMPARISON OF DESIGN ALTERNATIVES

To investigate the impact of the three design alternatives on perceived usability, we conducted a second experiment to evaluate the perceived ease of use of the layout automatically generated by Slime in all the three major design alternatives. We are interested to the following null hypothesis:

> H_0: *There is no difference in usability between the three design alternatives for slidable configurations when E3Screen is deployed.*

6.1 Method

6.1.1 Participants. Our sample consisted of 15 participants (6 female and 9 male) aged between 30 and 82 years (M=39.13, SD=18.90). None of the participants was engaged with the first experiment. Also, none of the participants was familiar with E3Screen. No compensation was offered.

6.1.2 Task and Stimuli. We used the two applications developed with Slime for E3Screen: the on-line newspaper (see Section 4.2) and the car rental application (see Section 4.3). For each application, we considered the following three design alternatives:

1. *Bootstrap only.* A Bootstrap-based layout with three underlying techniques: automatic font scaling to maintain the physical size of the text under different screen configurations, automatic form elements scaling for labels, inputs, text areas, and screen divisions. Automatic media scaling was used to adjust the size and detail of pictures and videos.
2. *Masonry only.* We designed a Masonry-based layout that restructures its first-level regions and UI elements first vertically and then horizontally. Since all the individual UI elements are subject to this restructuring, this design is very versatile, but also changing a lot.
3. *Bootstrap and Masonry combined.* We designed a layout combining both approaches based on a unifying principle: when the viewport or the screen resolution increases after a screen has been expanded, all the UI elements are reordered with Masonry and their sizes increased proportionally with Bootstrap.

Participants were randomly assigned to one design alternative and application (*e.g.*, Bootstrap and the on-line newspaper) and to its counterpart (*e.g.*, Masonry for the car rental application), then randomly to the combination of techniques for both applications. Participants were encouraged to explore each design alternative and to expand the primary screen to the left and right with the corresponding panels of E3Screen. Overall, we had 15 participants × 4 configurations = 45 trials.

6.1.3 Procedure. Participants were given five minutes to discover Slime and the E3Screen but without any interaction. The four trials were administrated per participant with a time limit of seven minutes, but each participant was free to stop the exploration before the time elapsed. After each trial, the IBM Computer System Usability Questionnaire (CSUQ) [36] was filled. This questionnaire enables participants to report their perceived usability of the configuration through sixteen closed questions in the form of 7-point Likert scales [37] (1=strongly disagree, 2=largely disagree, 3=disagree, 4=neutral, 5=agree, 6=largely agree, and 7=strongly agree). We appended the questionnaire with four statements related to the presentation guidelines [4] of the design alternative, as follows: "I always know where I am and how to go wherever I want" (item 20), "Returns and exits are clear and visible" (item 21), "The address of the current page allows me to easily return" (item 22), and "The colors are chosen to allow readable information" (item 23).

6.1.4 Experiment Design. Our study was a between-subjects design counterbalanced for the first part and equivalent for the second [10] with two independent variables:

- Application, nominal variable with two categories representing the Slime-based applications: "on-line newspaper" and "car rental."

- Design alternative, nominal variable with three categories representing the design alternatives "Bootstrap," "Masonry," and "Both."

The dependent variables were:

- System usefulness (SysUse): average of items 1 to 8.
- Quality of the information (InfoQual): average of items 9 to 15.
- Quality of the interaction (InterQual): average of items 16 to 18.
- System quality (Overall): average of items 1 to 19.
- Presentation: average of items 20 to 23.
- Mean: average of all the items 1 to 23.

6.2 Results

Fig. 17 shows the dependent variables for the three design alternatives, per application and overall.

Regarding the on-line newspaper, since the distribution was not normal (Shapiro-Wilk tests $W_{Bootstrap}$=.86, $W_{Masonry}$=.89, and W_{Both}=.88, p<.05), we employed Wilcoxon signed-rank tests for paired samples. We found a statistically significant difference in the ratings of the Bootstrap condition (M=5.93) with respect to Masonry (M=5.40, T=2933, $Z-score$=8.76, p<.001, r=.34) and between Both (M=5.54, T=2371, Z=7.65, p<.001, r=.30). The perceived order of usability—Bootstrap, Both, and Masonry—was confirmed for the car rental system as well. Participants perceived higher usability when interacting with the Bootstrap (M=5.77) than Both (M=5.52, T=244, Z=8.45, p<.001, r=.32) and Masonry (M=5.34, T=196, Z=6.67, p<.001, r=.25). While the Bootstrap condition was always superior, we observed a statistically significant difference between the on-line newspaper and the car rental (U=50955, Z=2.49, p=.006, r=.09). Overall, the on-line newspaper was preferred over the car rental application. The main reason was that the on-line newspaper was composed of related regions and subregions in a hierarchical decomposition, a design that was appreciated by participants, as opposed to the series of different and independent regions for the car rental application. For each application, the Bootstrap alternative received the best scores for both the IBM CSUQ and our metric for presentation. The green background color used for some news sections was perceived more distinctive than for the two other alternatives. The news ordering changes at every news opening, which can be perceived as largely disturbing [28], in particular when more space becomes available. Therefore, the stability of the Bootstrap version is assessed as more predictable and acceptable than in the ever-changing layout of the Masonry condition, even if this technique optimizes the screen real estate. In the Masonry condition, system usability (SysUse) decreased. The overall satisfaction (Overall) was the worst for Both and Masonry. The InfoQual and Presentation criteria of each version were not better than the rest.

7 EXPERIMENT #3: EVALUATION OF DISPLAY CONFIGURATIONS

To investigate the impact of the display configurations on task performance, we conducted a third experiment to test the impact of the layout generated by Slime on a visual search task. Our null hypothesis was:

> H_0: *There is no difference between the search time and the workload of a normal configuration (i.e., one screen only) and other configurations when E3Screen is deployed.*

7.1 Method

7.1.1 Participants. Our sample consists of 10 participants (4 female and 6 male) aged between 21 and 54 years (M=29.9, SD=10.75). All participants were using computers daily. None of them was familiar with E3Screen. No compensation was offered.

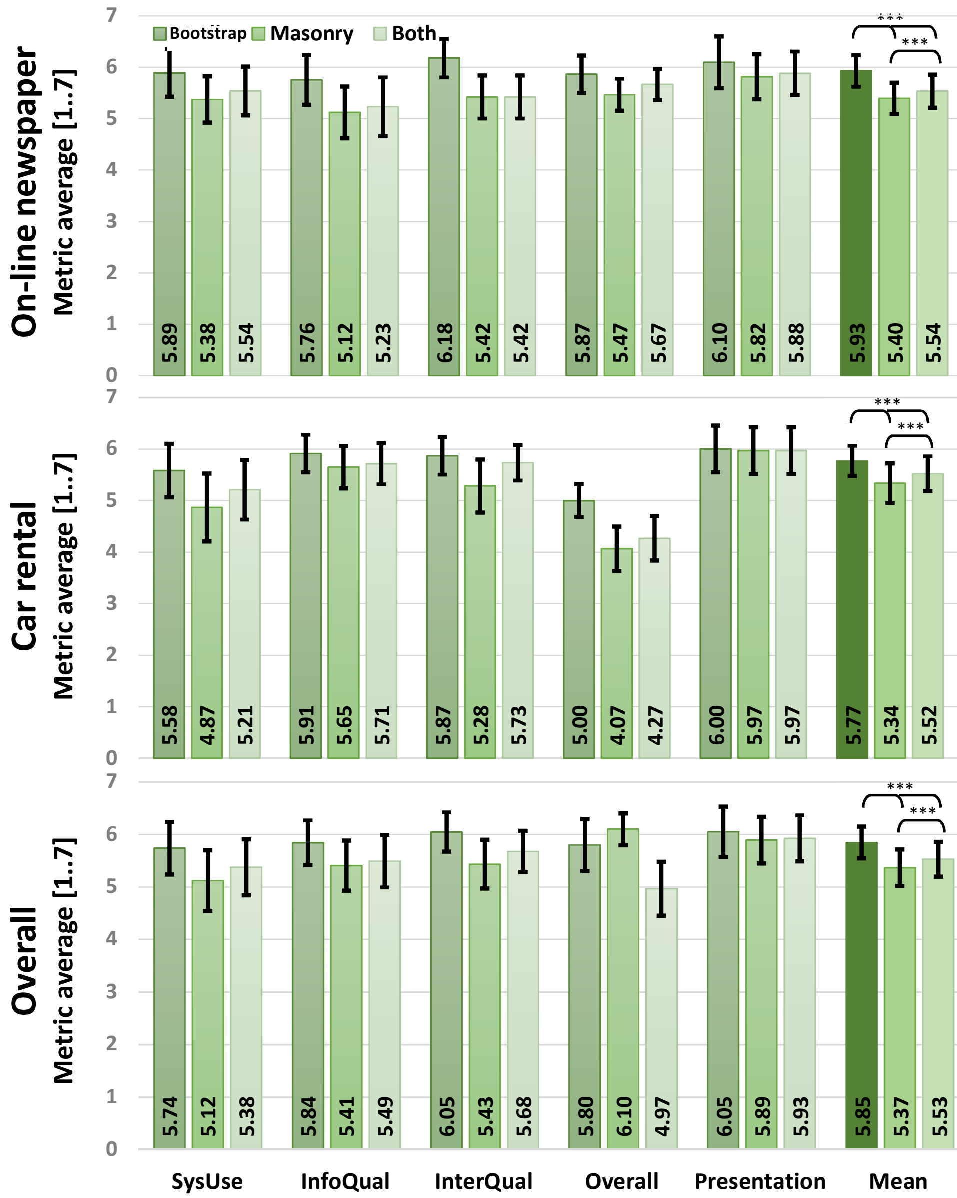


Fig. 17. Results for the three design alternatives used in the second experiment: Bootstrap, Masonry, and combined. Error bars show 95% confidence intervals.

7.1.2 Task. We considered the on-line newspaper application (see Section 4.2), for which we randomly selected fourteen articles from five typical news categories ("Seven days a week," "Nationwide," "Worldwide," "Sport," and "Showbiz;" see the Appendix B for screenshots), and created a base of 70 articles. Participants were instructed to perform a visual search task consisting of browsing the categories and their articles to find a randomly selected item. The process was repeated for the seven configurations of E3Screen from Fig. 15 for one user. Each configuration was randomly selected as well. Overall, we had 10 participants × 7 configurations = 70 trials.

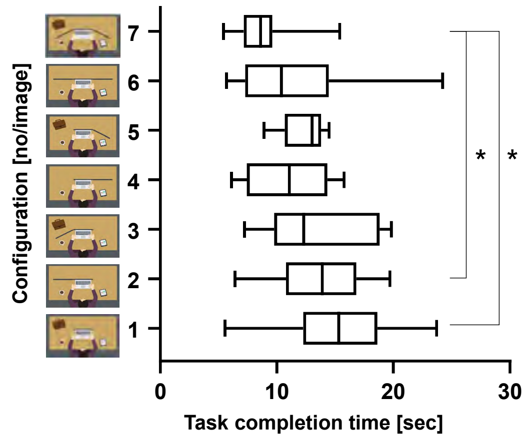


Fig. 18. Task completion times for each the display configuration.

7.1.3 Apparatus and setup. E3Screen was placed on a table in a quiet room with a whiteboard. Participants were provided with a comfortable chair in front of E3Screen. The two lateral panels of E3Screen were deployed to the initial UI before expansion is ensured by Slime. When a randomly selected item was displayed on the whiteboard, Slime was activated to expand the initial slidable UI into its final state depending on the configuration selected, thus triggering a stopwatch that recorded the time elapsed until the participant selected the item.

7.1.4 Procedure. Participants were given five minutes to discover Slime and E3Screen. The seven trials were then administrated and timed. After completing the trials, the IBM Post-Study System Usability Questionnaire (PSSUQ) [35] and the NASA-TLX questionnaire [26] were filled in according to Casner and Brian's [11] procedure and checked by the experimenter. The PSSUQ questionnaire is appended with an optional list of three positive and three negative aspects that a participant may want to report.

7.1.5 Experiment design. Since each participant was exposed to all seven configurations, our experiment was a within-subjects design [10] with the following independent variables:

- Configuration, nominal variable with seven conditions (C1 to C7), representing the seven configurations reproduced in Fig. 15 and C1 the control condition.
- Item, numerical variable with seventy conditions, 1 to 70, identifying the stimuli.

The dependent variables were:

- Task completion time, representing the time elapsed between the presentation of the item and its selection on E3Screen.
- Task completion rate, representing the percentage of correctly selected items on the first trial.
- IBM PSSUQ scores for each statement.
- NASA-TLX scores: the scores entered by each participant in the NASA-TLX application [50] regarding the various measures, *i.e.* physical demand, mental demand, temporal demand, performance, effort, and frustration.

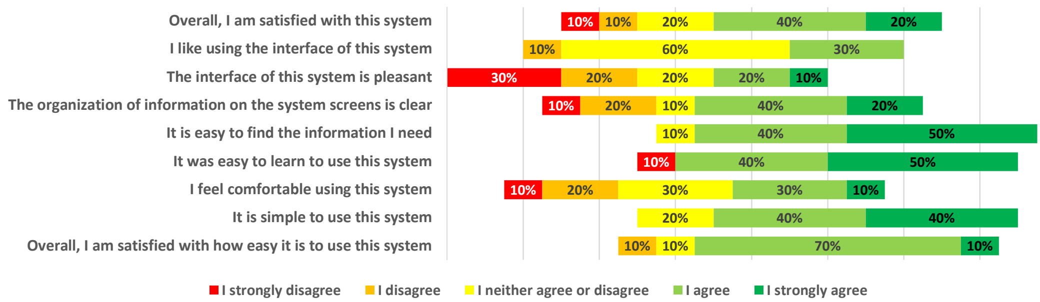


Fig. 19. Distribution of participants' answers to the IBM PSSUQ questionnaire used in the third experiment.

7.2 Results

The task completion rate averaged for all the trials was 95.7% (67/70). Figure 18 depicts the distribution of task completion times for the seven configurations, with the first configuration serving as a control condition (no panel was deployed). This configuration had the largest task times (M=15.22s), the second largest standard deviation (SD=5.47s), and the second largest coefficient of variation (CV=35.97%). The fastest configurations were as follows: configuration 7 (M=8.95s, SD=2.69, CV=30.15%), configuration 4 (M=10.91s, SD=3.52, CV=32.29%), configuration 6 (M=11.47s, SD=5.64, CV=49.20%), configuration 5 (M=12.46s, SD=1.83, CV=14.70%), configuration 3 (M=13.46s, SD=4.61, CV=34.24%), and configuration 2 (M=13.66s, SD=4.13, CV=30.24%), respectively. Overall, the averaged task completion time with two panels (conditions 6 and 7, M=10.21s) was faster than the only one panel deployed (conditions 2 to 5, M=12.62s), which was faster than the control condition (condition 1, M=15.22s). This result suggests that the two laterally deployed panels reduce the task completion time with respect to one or the no panel conditions.

The average of all the conditions was M=12.30s. The highest-performing participants completed the task in about 7 seconds. We also noticed that participants preferred to extend the screen to their right while working on a configuration with just one added screen. Moreover, the angle difference (0°–45°) did not seem to affect performance. When comparing the average task time of the four configurations (featuring one extra display), we observe that there was not much difference between the second screen positioned to the left on the same plane and at 45° (13.7s and 13.5s, respectively). Nevertheless, there was a difference of 1.6 seconds between the two configurations with one screen on the right. In that case, the angle has an effect and the 0°had better performance than 45°. This result suggests that, among all possible configurations with one additional screen, users prefer to extend the screen to the right at the same angle as the primary screen. In conclusion, task completion times are not equivalent for all configurations.

Figure 19 shows the distribution of the answers provided by the participants to the IBM PSSUQ questionnaire. Only question 7 received a mean under the midpoint value. However, only four of the results were significantly higher than the midpoint value, suggesting that the answers for these conditions reflect a positive appreciation of usability, as follows: question 1 (t=3.207, df=9, p=.0107*), question 2 (t=4.811, df=9, p=.0010**), question 4 (t=3.087, df=9, p=.0130*), and question 5 (t=6.332, df=9, p=.0001***), respectively.

Figure 20 shows the averaged scores of the various dimensions of the NASA-TLX test. Based on [50], we computed an average score of 18.1/21 for the performance aspect, which represents a high value. Moreover, the smallest score was 15. This result means that all respondents considered that they were successful at completing the tasks. Furthermore, there was an average of 2.9 for the

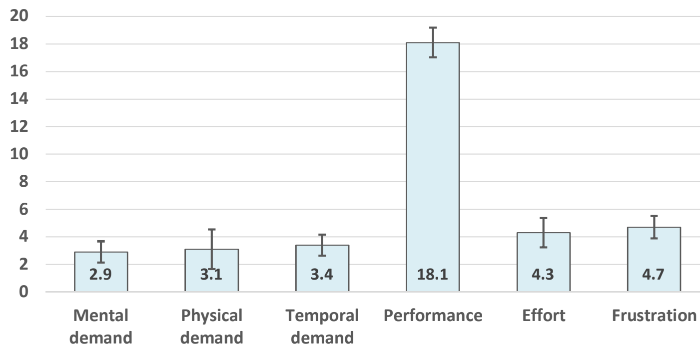


Fig. 20. Distribution of scores from the NASA-TLX test used in the third experiment. Error bars show 95% confidence intervals.

mental demand. The other aspects, physical demand, temporal demand, effort, and frustration, also registered low average scores. Thus, we can conclude that the participants subjectively assessed the various tasks as having a moderate workload with small variations.

8 CONCLUSION

We revisited the notion of plasticity of user interfaces by refining it into inter-platform plasticity and intra-platform plasticity and considering the effect of reconfigurable displays. To this end, we introduced a model-based approach for addressing specific cases of plasticity, which we illustrated with the E3Screen device and two applications. Future work will consider more experiments to understand the cognitive load induced by these various display configurations as well as their effect on the visual field of view and, correspondingly, user performance, extending Tan *et al.*'s [61] results on dual-monitor systems towards flexible configurations involving three displays.

ACKNOWLEDGMENTS

The authors would like to thank Thomas Castro, Charlie Jeunehomme, and Laurent Wéry from the Slide'n'joy company for the E3Screen device. This work is supported by the Nomos project of the Hubert Curien Programme funded by Wallonie-Bruxelles-International under Grant No. 454410. Radu-Daniel Vatavu acknowledges support from a grant of the Romanian National Authority for Scientific Research and Innovation, CNCS/CCCDI-UEFISCDI, project number PN-III-P3-3.6-H2020-2020-0034 (12/2021).

REFERENCES

[1] Pierre A. Akiki, Arosha K. Bandara, and Yijun Yu. 2013. Cedar studio: an IDE supporting adaptive model-driven user interfaces for enterprise applications. In *Proc. of ACM SIGCHI Symposium on Engineering Interactive Computing Systems (EICS'13)*, Peter Forbrig, Prasun Dewan, Michael Harrison, and Kris Luyten (Eds.). ACM Press, NewYork, USA, 139–144.

[2] Helmut Balzert. 1996. From OOA to GUIs: The JANUS System. *J. Object Oriented Program.* 8, 9 (1996), 43–47.

[3] Nicolas Barralon, Joëlle Coutaz, and Christophe Lachenal. 2007. Coupling Interaction Resources and Technical Support. In *Universal Access in Human-Computer Interaction. Ambient Interaction*, Constantine Stephanidis (Ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, 13–22. https://doi.org/10.1007/978-3-540-73281-5_2

[4] Abdo Beirekdar, Jean Vanderdonckt, and Monique Noirhomme-Fraiture. 2002. A Framework and a Language for Usability Automatic Evaluation of Web Sites by Static Analysis of HTML Source Code. In *Proceedings of the Fourth International Conference on Computer-Aided Design of User Interfaces (CADUI '02)*, Christophe Kolski and Jean Vanderdonckt (Eds.). Kluwer, Dordrecht, The Netherlands, 337–348. https://doi.org/10.1007/978-94-010-0421-3_29

[5] Xiaojun Bi and Ravin Balakrishnan. 2009. Comparing Usage of a Large High-resolution Display to Single or Dual Desktop Displays for Daily Work. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '09)*. ACM, New York, NY, USA, 1005–1014. https://doi.org/10.1145/1518701.1518855
[6] Gaëlle Calvary, Joëlle Coutaz, Olfa Dâassi, Lionel Balme, and Alexandre Demeure. [n. d.]. Towards a New Generation of Widgets for Supporting Software Plasticity: The "Comet". In *Proc. of Joint Working Conferences on Engineering Human Computer Interaction and Interactive SystemsEHCI-DSVIS 2004*, Rémi Bastide, Philippe A. Palanque, and Jörg Roth (Eds.).
[7] Gaëlle Calvary, Joëlle Coutaz, and David Thevenin. 2001. Supporting Context Changes for Plastic User Interfaces: A Process and a Mechanism. In *People and Computers XV—Interaction without Frontiers*, Ann Blandford, Jean Vanderdonckt, and Phil Gray (Eds.). Springer, London, 349–363.
[8] Gaëlle Calvary, Joëlle Coutaz, David Thevenin, Quentin Limbourg, Laurent Bouillon, and Jean Vanderdonckt. 2003. A Unifying Reference Framework for multi-target user interfaces. *Interacting with Computers* 15, 3 (2003), 289–308. https://doi.org/10.1016/S0953-5438(03)00010-9
[9] Gaelle Calvary, Joëlle Coutaz, David Thevenin, Quentin Limbourg, Nathalie Souchon, Laurent Bouillon, Murielle Florins, and Jean Vanderdonckt. 2002. Plasticity of User Interfaces: A Revised Reference Framework. In *Proceedings of the First International Workshop on Task Models and Diagrams for User Interface Design* (July 2002) *(TAMODIA '02)*, Costin Pribeanu and Jean Vanderdonckt (Eds.). INFOREC Publishing House, Bucharest, 127–134. https://pdfs.semanticscholar.org/5eaf/977231fb4d48a0ee9ad77941078ee6951b31.pdf
[10] Donald T. Campbell and Julian C. Stanley. 1963. *Experimental and quasi-experimental designs for research.* Rand McNally & Company, Chicago, USA, Chapter 5-Experimental and quasi-experimental designs for research, 1–76. https://www.sfu.ca/~palys/Campbell&Stanley-1959-Exptl&QuasiExptlDesignsForResearch.pdf
[11] Stephen M. Casner and Brian F. Gore. 2010. *Measuring and Evaluating Workload: A Primer.* Technical Report NASA/TM—2010-216395. National Aeronautics and Space Administration. https://matb-files.larc.nasa.gov/Workload_Primer_TM_Final.pdf
[12] Gilbert Cockton. 2004. Value-Centred HCI. In *Proceedings of the Third Nordic Conference on Human-Computer Interaction (NordiCHI '04)*. Association for Computing Machinery, New York, NY, USA, 149–160. https://doi.org/10.1145/1028014.1028038
[13] Benoǐt Collignon, Jean Vanderdonckt, and Gaëlle Calvary. 2008. An Intelligent Editor for Multi-presentation User Interfaces. In *Proceedings of the ACM Symposium on Applied Computing* (March 16-20, 2008) *(SAC '08)*. ACM, New York, NY, USA, 1634–1641. https://doi.org/10.1145/1363686.1364072
[14] Joëlle Coutaz. 2010. User Interface Plasticity: Model Driven Engineering to the Limit!. In *Proceedings of the 2nd ACM SIGCHI Symposium on Engineering Interactive Computing Systems* (June 2010) *(EICS '10)*. Association for Computing Machinery, New York, NY, USA, 1–8. https://doi.org/10.1145/1822018.1822019
[15] Joëlle Coutaz and Gaëlle Calvary. 2012. *HCI and Software Engineering : Designing for User Interface Plasticity, Chapter 52.* CRC Press, Boca Raton, USA, Chapter The Human-Computer Interaction Handbook : Fundamentals, Evolving Technologies and Emerging Applications, 1–26. https://doi.org/10.1201/b11963
[16] Alexandre Demeure and Gaëlle Calvary. 2003. Plasticity of user interfaces: towards an evolution model based on conceptual graphs. In *Proceedings of the 15th French-speaking conference on human-computer interaction on 15eme Conference Francophone sur l'Interaction Homme-Machine (IHM '03)*, Khaldoun Zreik, Thomas Baudel, Christophe Kolski, and Eric Brangier (Eds.), Vol. 51. ACM Press, New York, NY, USA, 80–87. https://doi.org/10.1145/1063669.1063681
[17] Alexandre Demeure, Gaëlle Calvary, Joëlle Coutaz, and Jean Vanderdonckt. 2007. The Comets Inspector: Towards Run Time Plasticity Control Based on a Semantic Network. In *Task Models and Diagrams for Users Interface Design*, Karin Coninx, Kris Luyten, and Kevin A. Schneider (Eds.). Springer, Berlin, Heidelberg, 324–338. https://doi.org/10.1007/978-3-540-70816-2_23
[18] Charles-Eric Dessart, Vivian Genaro Motti, and Jean Vanderdonckt. 2012. Animated Transitions between User Interface Views. In *Proceedings of the International Working Conference on Advanced Visual Interfaces (AVI '12)*. Association for Computing Machinery, New York, NY, USA, 341–348. https://doi.org/10.1145/2254556.2254623
[19] Murielle Florins, Francisco Montero Simarro, Jean Vanderdonckt, and Benjamin Michotte. 2006. Splitting Rules for Graceful Degradation of User Interfaces. In *Proceedings of the ACM International Working Conference on Advanced Visual Interfaces (AVI '06)*. Association for Computing Machinery, New York, NY, USA, 59–66. https://doi.org/10.1145/1133265.1133276
[20] Antonio Gomes and Roel Vertegaal. 2014. PaperFold: A Shape Changing Mobile Device with Multiple Reconfigurable Electrophoretic Magnetic Display Tiles. In *Proceedings of ACM International Conference on Human Factors in Computing Systems, Extended Abstracts (CHI EA '14)*. ACM, New York, NY, USA, 535–538. https://doi.org/10.1145/2559206.2574770 See video at https://www.youtube.com/watch?v=IBdFQ6ICD-4.
[21] Donatien Grolaux, Peter Van Roy, and Jean Vanderdonckt. 2002. FlexClock, a Plastic Clock Written in Oz with the QTk toolkit. In *Proceedings of the First International Workshop on Task Models and Diagrams for User Interface*

Design (July 18-19, 2002) *(TAMODIA '02)*, Costin Pribeanu and Jean Vanderdonckt (Eds.). INFOREC Publishing House, Bucharest, Romania, 135–142. https://www.info.ucl.ac.be/~pvr/FlexClockTamodia.pdf See video at https://www.youtube.com/watch?v=bCW9CjaDQAA.

[22] Donatien Grolaux, Peter Van Roy, and Jean Vanderdonckt. 2001. QTk - A Mixed Declarative/Procedural Approach for Designing Executable User Interfaces. In *Engineering for Human-Computer Interaction (EHCI '01)*, Murray Reed Little and Laurence Nigay (Eds.). Springer, Berlin, Heidelberg, 109–110. https://doi.org/10.1007/3-540-45348-2_12

[23] Donatien Grolaux, Jean Vanderdonckt, and Peter Van Roy. 2005. Attach Me, Detach Me, Assemble Me Like You Work. In *Proceedings of IFIP TC 13 International Conference on Human-Computer Interaction* (September 12-16, 2005) *(INTERACT '05)*, Maria Francesca Costabile and Fabio Paternò (Eds.). Springer, Berlin, Heidelberg, 198–212.

[24] Jonathan Grudin. 2001. Partitioning Digital Worlds: Focal and Peripheral Awareness in Multiple Monitor Use. In *Proceedings of the ACM Conference on Human Factors in Computing Systems* (March 2001) *(CHI '01)*. ACM, New York, NY, USA, 458–465. https://doi.org/10.1145/365024.365312

[25] Chris Harrison and Scott E. Hudson. 2011. A New Angle on Cheap LCDs: Making Positive Use of Optical Distortion. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology (UIST '11)*. ACM, New York, NY, USA, 537–540. https://doi.org/10.1145/2047196.2047266

[26] Sandra G. Hart and Lowell E. Staveland. 1988. Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research. In *Human Mental Workload*, Peter A. Hancock and Najmedin Meshkati (Eds.). Advances in Psychology, Vol. 52. North-Holland, 139–183. https://doi.org/10.1016/S0166-4115(08)62386-9

[27] Renate Häuslschmid, Klaus Bengler, and Cristina Olaverri-Monreal. 2013. Graphic Toolkit for Adaptive Layouts in In-Vehicle User Interfaces. In *Proceedings of the 5th International Conference on Automotive User Interfaces and Interactive Vehicular Applications (AutomotiveUI '13)*. Association for Computing Machinery, New York, NY, USA, 292–298. https://doi.org/10.1145/2516540.2516580

[28] Bowen Hui, Grant A. Partridge, and Craig Boutilier. 2009. A probabilistic mental model for estimating disruption. In *Proceedings of the 14th ACM International Conference on Intelligent User Interfaces (IUI '08)*. ACM, New York, NY, USA, 287–296. https://doi.org/10.1145/1502650.1502691

[29] Yue Jiang, Ruofei Du, Christof Lutteroth, and Wolfgang Stuerzlinger. 2019. ORC Layout: Adaptive GUI Layout with OR-Constraints. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19)*. Association for Computing Machinery, New York, NY, USA, Article Paper 413, 12 pages. https://doi.org/10.1145/3290605.3300643

[30] Tero Jokela, Ming Ki Chong, Andrés Lucero, and Hans Gellersen. 2015. Connecting Devices for Collaborative Interactions. *interactions* 22, 4 (June 2015), 39–43. https://doi.org/10.1145/2776887

[31] Eser Kandogan and Ben Shneiderman. 1996. Elastic Windows: Improved Spatial Layout and Rapid Multiple Window Operations. In *Proceedings of the Workshop on Advanced Visual Interfaces (AVI '96)*. Association for Computing Machinery, New York, NY, USA, 29–38. https://doi.org/10.1145/948449.948454

[32] Heikki Keränen and Johan Plomp. 2002. Adaptive Runtime Layout of Hierarchical UI Components. In *Proceedings of the Second Nordic Conference on Human-Computer Interaction (NordiCHI '02)*. Association for Computing Machinery, New York, NY, USA, 251–254. https://doi.org/10.1145/572020.572058

[33] Mohammadreza Khalilbeigi, Roman Lissermann, Wolfgang Kleine, and Jürgen Steimle. 2012. FoldMe: Interacting with Double-sided Foldable Displays. In *Proceedings of the Sixth International Conference on Tangible, Embedded and Embodied Interaction (TEI '12)*. ACM, New York, NY, USA, 33–40. https://doi.org/10.1145/2148131.2148142

[34] Sangsu Lee, Hyunjeong Kim, Yong-ki Lee, Minseok Sim, and Kun-pyo Lee. 2011. Designing of an Effective Monitor Partitioning System with Adjustable Virtual Bezel. In *Human Centered Design*, Masaaki Kurosu (Ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, 537–546. https://doi.org/10.1007/978-3-642-21753-1_60

[35] James R. Lewis. 1992. Psychometric Evaluation of the Post-Study System Usability Questionnaire: The PSSUQ. *Proceedings of the Human Factors Society Annual Meeting* 36, 16 (1992), 1259–1260. https://doi.org/10.1177/154193129203601617 arXiv:https://doi.org/10.1177/154193129203601617

[36] James R. Lewis. 1995. IBM computer usability satisfaction questionnaires: Psychometric evaluation and instructions for use. *International Journal of Human-Computer Interaction* 7, 1 (1995), 57–78. https://doi.org/10.1080/10447319509526110 arXiv:https://doi.org/10.1080/10447319509526110

[37] Rensis Likert. 1932. A technique for the measurement of attitudes. *Archives of Psychology* 22, 140 (1932), 55–. http://psycnet.apa.org/record/1933-01885-001

[38] Quentin Limbourg, Jean Vanderdonckt, Benjamin Michotte, Laurent Bouillon, and Murielle Florins. 2004. UsiXML: A User Interface Description Language Supporting Multiple Levels of Independence. In *Proceedings of Workshops in connection with the 4th International Conference on Web Engineering (ICWE '04). Engineering Advanced Web Applications* (28-30 July, 2004) *(DIWE '04)*, Maristella Matera and Sara Comai (Eds.). Rinton Press, 325–338. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.69.3572&rep=rep1&type=pdf

[39] Anderson Maciel, Luciana P. Nedel, Eduardo M. Mesquita, Marcelo H. Mattos, Gustavo M. Machado, and Carla M. D. S. Freitas. 2010. Collaborative Interaction through Spatially Aware Moving Displays. In *Proceedings of the 2010 ACM*

Symposium on Applied Computing (March 2010) *(SAC '10)*. Association for Computing Machinery, New York, NY, USA, 1229–1233. https://doi.org/10.1145/1774088.1774346

[40] Ethan Marcotte. 2015. *Responsive Design.* A Book Apart.

[41] José Pascual Molina Massó, Jean Vanderdonckt, and Pascual González López. 2006. Direct manipulation of user interfaces for migration. In *Proceedings of the 11th International Conference on Intelligent User Interfaces, IUI 2006, Sydney, Australia, January 29 - February 1, 2006*, Cécile Paris and Candace L. Sidner (Eds.). ACM, 140–147. https://doi.org/10.1145/1111449.1111483

[42] Jérémie Melchior, Jean Vanderdonckt, and Peter Van Roy. 2011. A model-based approach for distributed user interfaces. In *Proceedings of the 3rd ACM SIGCHI Symposium on Engineering Interactive Computing System (EICS '11)*, Fabio Paternò, Kris Luyten, and Frank Maurer (Eds.). ACM Press, New York, NY, USA, 11–20. https://doi.org/10.1145/1996461.1996488

[43] Francisco Montero, Víctor López-Jaquero, Jean Vanderdonckt, Pascual González, María Lozano, and Quentin Limbourg. 2006. Solving the Mapping Problem in User Interface Design by Seamless Integration in IdealXML. In *Proceedings of the 12th International Workshop on Design, Specification, and Verification of Interactive Systems* (July 13-15, 2006) *(DSV-IS '06)*, Stephen W. Gilroy and Michael D. Harrison (Eds.). Springer, Berlin, Heidelberg, 161–172. https://doi.org/10.1007/11752707_14

[44] Michael Nebeling, Fabrice Matulic, Lucas Streit, and Moira C. Norrie. 2011. Adaptive Layout Template for Effective Web Content Presentation in Large-Screen Contexts. In *Proceedings of the 11th ACM Symposium on Document Engineering (DocEng '11)*. Association for Computing Machinery, New York, NY, USA, 219–228. https://doi.org/10.1145/2034691.2034737

[45] Michael Nebeling and Moira C. Norrie. 2013. Responsive Design and Development: Methods, Technologies and Current Issues. In *Proceedings of International Conference on Web Engineering (ICWE '13)*, Florian Daniel, Peter Dolog, and Qing Li (Eds.). Springer, Berlin, Heidelberg, 510–513.

[46] J.M. O'Hara, W.S. Brown, P.M. Lewis, and P. Persensky. 2002. Human-System Interface Design Review Guidelines, NUREG-0700, Rev. 2.

[47] Takashi Ohta. 2019. *Interface Support for Creativity, Productivity, and Expression in Computer Graphics.* Open University Press, Hershey, PA, USA, Chapter An Intuitive Interface for Interactively Pairing Multiple Mobile Devices: Dynamic Reconfiguration of Multiple Screens and a Variety of Content Designs, 1–26. https://doi.org/10.4018/978-1-5225-7371-5.ch001

[48] Raquel Oliveira, Sophie Dupuy-Chessa, and Gaëlle Calvary. 2015. Plasticity of User Interfaces: Formal Verification of Consistency. In *Proceedings of the 7th ACM SIGCHI Symposium on Engineering Interactive Computing Systems (EICS '15)*. Association for Computing Machinery, New York, NY, USA, 260–265. https://doi.org/10.1145/2774225.2775078

[49] Antti Oulasvirta and Lauri Sumari. 2007. Mobile Kits and Laptop Trays: Managing Multiple Devices in Mobile Information Work. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '07)*. ACM, New York, NY, USA, 1127–1136. https://doi.org/10.1145/1240624.1240795

[50] Vinoth Pandian Sermuga Pandian and Sarah Suleri. 2020. NASA-TLX Web App: An Online Tool to Analyse Subjective Workload. arXiv:cs.HC/2001.09963 https://arxiv.org/abs/2001.09963

[51] Oscar Pastor and Juan Carlos Molina. 2007. *Model-driven architecture in practice - a software production environment based on conceptual modeling.* Springer, Berlin.

[52] Richard Paštěka, Mathias Forjan, and Veronika David. 2018. A Single Point of Contact Data Platform for Rehabilitative Exercises and Equipment: Development of the REHABitation Database. In *Proceedings of the 8th International Conference on Software Development and Technologies for Enhancing Accessibility and Fighting Info-Exclusion (DSAI 2018)*. Association for Computing Machinery, New York, NY, USA, 317–322. https://doi.org/10.1145/3218585.3218676

[53] Angel Puerta and Martin Hu. 2009. UI Fin: A Process-Oriented Interface Design Tool. In *Proceedings of the 14th International Conference on Intelligent User Interfaces (IUI '09)*. Association for Computing Machinery, New York, NY, USA, 345–354. https://doi.org/10.1145/1502650.1502698

[54] David Raneburger, Roman Popp, and Jean Vanderdonckt. 2012. An Automated Layout Approach for Model-driven WIMP-UI Generation. In *Proceedings of the 4th ACM SIGCHI Symposium on Engineering Interactive Computing Systems* (June 25-26, 2012) *(EICS '12)*. ACM, New York, NY, USA, 91–100. https://doi.org/10.1145/2305484.2305501

[55] Cristina Roda, Elena Navarro, Uwe Zdun, Víctor López-Jaquero, and Georg Simhandl. 2018. Past and future of software architectures for context-aware systems: A systematic mapping study. *Journal of Systems and Software* 146 (2018), 310 – 355. https://doi.org/10.1016/j.jss.2018.09.074

[56] R. Schmidt, E. Penner, and S. Carpendale. 2004. Reconfigurable Displays.. In *Proceedings of the UbiComp'04 Workshop on Ubiquitous Display Environments (UbiDisplay '04)*, Chris Johnson (Ed.). ACM, New York, USA. https://innovis.cpsc.ucalgary.ca/innovis/uploads/Publications/Publications/ubidisplay04-schmidt-carpendale.pdf

[57] Kevin A. Schneider and James R. Cordy. 2001. Abstract User Interfaces: A Model and Notation to Support Plasticity in Interactive Systems. In *Proceedings of the International Workshop on Design, Specification, and Verification of Interactive Systems (DSV-IS '01)*, Chris Johnson (Ed.). Springer, Berlin, Heidelberg, 28–48. https://doi.org/10.1007/3-540-45522-1_3

[58] Stacey D. Scott, M. Sheelagh T. Carpendale, and Kori M. Inkpen. 2004. Territoriality in Collaborative Tabletop Workspaces. In *Proceedings of the 2004 ACM Conference on Computer Supported Cooperative Work (CSCW '04)*. ACM, New York, NY, USA, 294–303. https://doi.org/10.1145/1031607.1031655

[59] Montserrat Sendín, Víctor López-Jaquero, and César A. Collazos. 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 (may 2008), 1447–1462. http://www.jucs.org/jucs_14_9/collaborative_explicit_plasticity_framework.

[60] Ramona E. Su and Brian P. Bailey. 2005. Put Them Where? Towards Guidelines for Positioning Large Displays in Interactive Workspaces. In *Proceedings of IFIP TC 13 International Conference on Human-Computer Interaction (INTERACT '05)*, Maria Francesca Costabile and Fabio Paternò (Eds.). Springer, Berlin, Heidelberg, 337–349. https://doi.org/10.1007/11555261_29

[61] Desney S. Tan and Mary Czerwinski. 2003. Effects of Visual Separation and Physical Discontinuities when Distributing Information across Multiple Displays. In *Proceedings of IFIP TC13 International Conference on Human-Computer Interaction* (September 1-5, 2003) *(INTERACT '03)*, Matthias Rauterberg, Marino Menozzi, and Janet Wesson (Eds.). IOS Press. https://www.microsoft.com/en-us/research/wp-content/uploads/2003/11/ozchiphyssep03.pdf

[62] David Thevenin and Joëlle Coutaz. 1999. Plasticity of User Interfaces: Framework and Research Agenda. In *Proceedings of the IFIP TC13 International Conference on Human-Computer Interaction* (August 30-September 3, 1999) *(INTERACT '99)*, M. Angela Sasse and Chris W. Johnson (Eds.). IOS Press, Amsterdam, 110–117. http://iihm.imag.fr/thevenin/papiers/Interact99/Plasticity.Interact99-WWW.pdf

[63] Jennifer Thom-Santelli. 2009. Expressing Territoriality in Collaborative Activity. In *Proceedings of the ACM 2009 International Conference on Supporting Group Work (GROUP '09)*. ACM, New York, NY, USA, 389–390. https://doi.org/10.1145/1531674.1531738

[64] Jacob M. Truemper, Hong Sheng, Michael G. Hilgers, Richard H. Hall, Morris Kalliny, and Basanta Tandon. 2008. Usability in Multiple Monitor Displays. *SIGMIS Database* 39, 4 (Oct. 2008), 74–89. https://doi.org/10.1145/1453794.1453802

[65] Jean Vanderdonckt and Thanh-Diane Nguyen. 2019. MoCaDiX: Designing Cross-Device User Interfaces of an Information System based on its Class Diagram. *Proc. ACM Hum. Comput. Interact.* 3, EICS (2019), 17:1–17:40. https://doi.org/10.1145/3331159

[66] Jean Vanderdonckt and Radu-Daniel Vatavu. 2021. Extensible, Extendable, Expandable, Extractable: The 4E Design Approach for Reconfigurable Displays. *International Journal of Human–Computer Interaction* (2021). https://doi.org/10.1080/10447318.2021.1908666

[67] Marco Winckler, Jean Vanderdonckt, Adrian Stanciulescu, and Francisco Trindade. 2008. Cascading Dialog Modeling with UsiXML. In *Interactive Systems. Design, Specification, and Verification*, T. C. Nicholas Graham and Philippe Palanque (Eds.). Springer, Berlin, Heidelberg, 121–135. https://doi.org/10.1007/978-3-540-70569-7_12

[68] Takuto Yanagida, Hidetoshi Nonaka, and Masahito Kurihara. 2009. Personalizing Graphical User Interfaces on Flexible Widget Layout. In *Proceedings of the 1st ACM SIGCHI Symposium on Engineering Interactive Computing Systems (EICS '09)*. Association for Computing Machinery, New York, NY, USA, 255–264. https://doi.org/10.1145/1570433.1570481 See demonstration at http://takty.stxst.com/res/fwl/index.en.html.

A EXAMPLES OF FREQUENT DISPLAY CONFIGURATIONS FOR E3SCREEN

Fig. 21 shows some examples of frequent configurations.

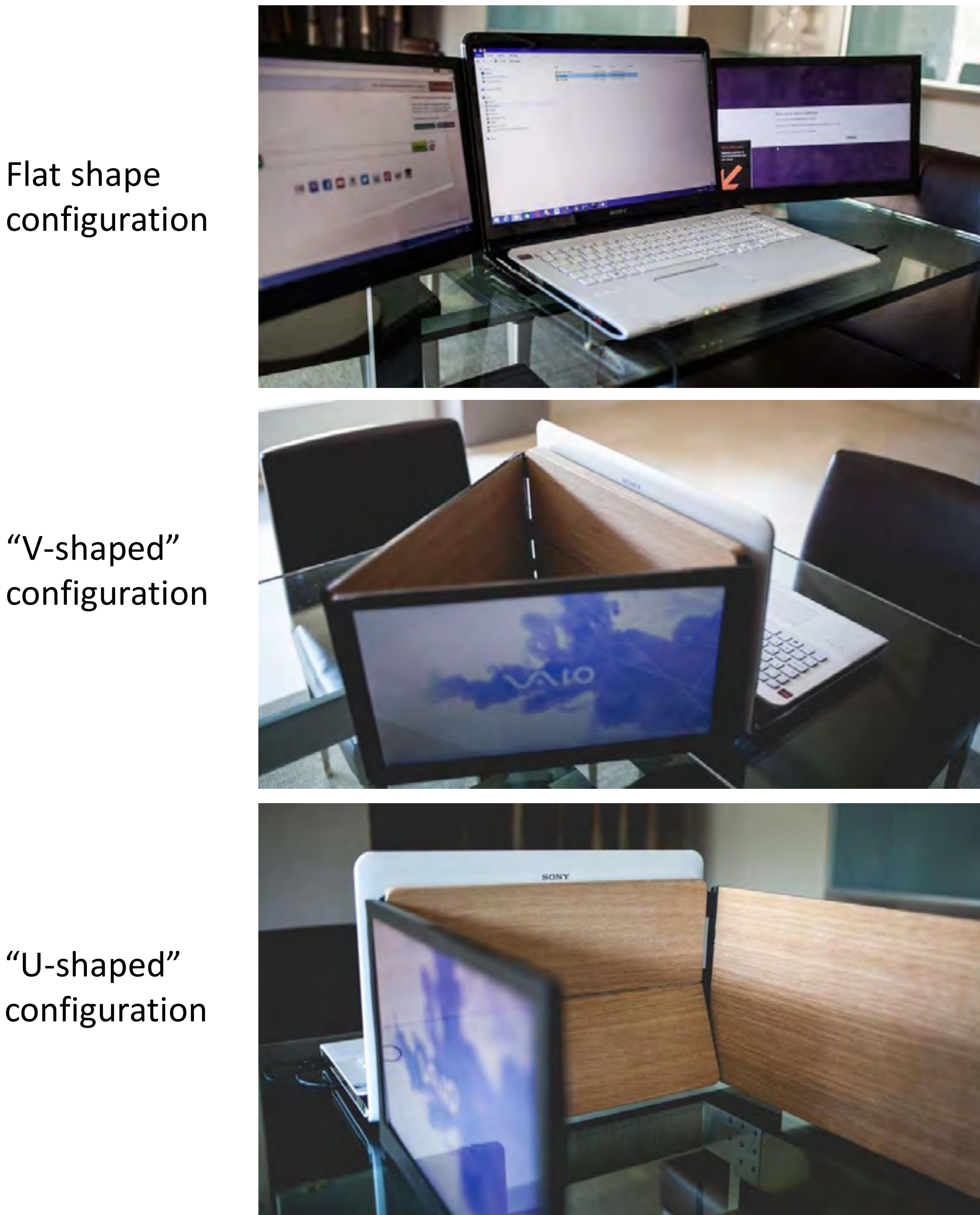


Fig. 21. Examples of frequent display configurations.

B STIMULI USED DURING THE THIRD EXPERIMENT

Fig. 22 to 27 show the five categories of articles used in the third experiment.

Fig. 22. Articles from the "Seven days a week" category.

Fig. 23. One articles expanded in the "Seven days a week" category.

Received February 2020; revised October 2020; accepted February 2021

Fig. 24. Articles from the “Nationwide” category.

Fig. 25. Articles from the “Worldwide” category.

Fig. 26. Articles from the “Sport” category.

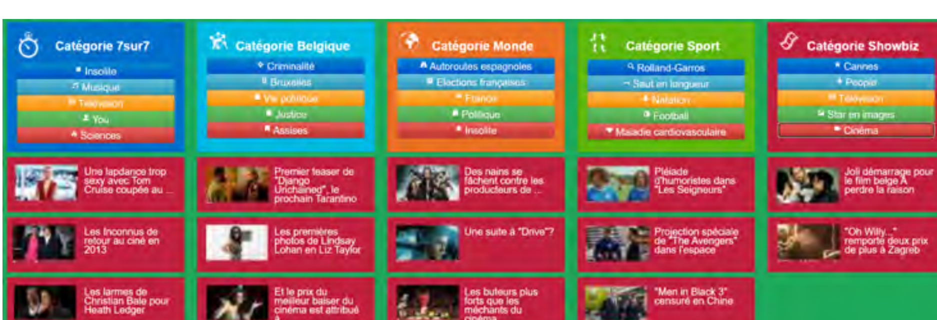


Fig. 27. Articles from the “Showbiz” category.