Paired Sketching of Distributed User Interfaces: Workflow, Protocol, Software Support, and Experiment

MEHDI OUSMER, Université catholique de Louvain, Belgium
JEAN VANDERDONCKT, Université catholique de Louvain, Belgium
LAURA-BIANCA BILIUS, MintViz Lab, MANSiD Center, Ştefan cel Mare University of Suceava, Romania
RADU-DANIEL VATAVU, MintViz Lab, MANSiD Center, Ştefan cel Mare University of Suceava, Romania
MIHAIL TERENTI, MintViz Lab, MANSiD Center, Ştefan cel Mare University of Suceava, Romania

Fig. 1. Sketching distributed user interfaces with UbiSketch on various platforms (*left*) through paired sketching (*right*).

The evolving landscape of distributed user interfaces requires the prototyping stage be also distributed between users, tasks, platforms, and environments. To create a cohesive distribution of the user interface elements in such ecosystems, paired

Authors' Contact Information: Mehdi Ousmer, Université catholique de Louvain, Louvain Research Institute in Management and Organizations, Belgium, mehdi.ousmer@uclouvain.be; Jean Vanderdonckt, Université catholique de Louvain, Louvain Research Institute in Management and Organizations, Belgium, jean.vanderdonckt@uclouvain.be; Laura-Bianca Bilius, MintViz Lab, MANSiD Center, Ştefan cel Mare University of Suceava, Suceava, Romania, laura.bilius@usm.ro; Radu-Daniel Vatavu, MintViz Lab, MANSiD Center, Ştefan cel Mare University of Suceava, Suceava, Romania, radu.vatavu@usm.ro; Mihail Terenti, MintViz Lab, MANSiD Center, Ştefan cel Mare University of Suceava, Suceava, Romania, mihail.terenti@usm.ro.

ACM 2573-0142/2025/6-ARTEICS018
https://doi.org/10.1145/3735499

sketching has emerged as a collaborative design method that leverages multiple stakeholders' strengths, including designers, developers, and end users, working in pairs. In the context of developer experience applied to paired sketching for distributed user interfaces, we decomposed a workflow into four disciplines according to the Software and Systems Process Engineering Meta-Model (SPEM) notation. First, we defined a protocol to deploy paired sketching of distributed user interfaces, supported by UbiSketch, a collaborative software environment tailored featuring sketch recognition and whiteboarding. Second, to evaluate paired sketching for engineering interactive systems, we conducted an experiment involving five pairs of stakeholders who sketched a distributed user interface for inside-the-vehicule interaction distributed on four platforms: smartphone, tablet, pen display, and tabletop. Empirical results, from questionnaires, reactivity, intention, perceived satisfaction, and free comments, suggest a preference order in which the tabletop is ranked first, followed by the tablet, smartphone, and pen display. Based on these results, we discuss the potential of paired sketching for distributed user interfaces.

CCS Concepts: • **Hardware** → *Tactile and hand-based interfaces*; • **Human-centered computing** → **HCI design and evaluation methods**; *User studies*; **Interface design prototyping**.

Additional Key Words and Phrases: Distributed user interfaces, Multi-platform user interfaces, Paired sketching.

ACM Reference Format:
Mehdi Ousmer, Jean Vanderdonckt, Laura-Bianca Bilius, Radu-Daniel Vatavu, and Mihail Terenti. 2025. Paired Sketching of Distributed User Interfaces: Workflow, Protocol, Software Support, and Experiment. *Proc. ACM Hum.-Comput. Interact.* 9, 4, Article EICS018 (June 2025), 31 pages. https://doi.org/10.1145/3735499

1 Introduction

Sketching represents a key stage in the development lifecycle of user interfaces (UIs) [11, 42] due to the many intrinsic qualities of *sketches* [19], considered as fast depictions of the UI [28]. Furthermore, UI sketching favors participatory design [33] by involving any stakeholder (*e.g.,* designers, developers, end users, or user representatives) in sketching [48], especially when software tools are available to recognize and interpret the resulted sketches [9, 73]. UI sketching is highly effective in terms of input (one can start from scratch [11]) and output (the sketches directly inform the rest of the life cycle [45, 82]), execution (few resources are needed to implement sketching [28]). Moreover, UI representations conveyed by sketching are meaningful [38].

However, sketching often adheres to the paradigm in which a single person sketches one UI for one platform at one time (*e.g.,* one form for a website and another for a desktop UI). When a sketch is produced on paper, the sketcher physically communicates with the other participants to elicit comments and improve the design. When a sketch is produced through software [68], the communication is supported by a digital layer. When a sketch is managed using collaborative software [62] or simply collaboratively [19], two-way communication is needed [64]. However, this paradigm is no longer feasible for sketching Distributed User Interface (DUIs) [23, 78], where various users perform tasks on different platforms distributed in both space and time. For example, inside-the-vehicle interaction, as enabled by Vehicle-to-DUIs (V2DUIs) [6], involves both fixed and mobile devices distributed within the vehicle environment across the driver, the passenger next to them, and the other passengers [4, 5]. In this context, we address the following research question within the general framework of developer experience [24, 60]: *RQ=How to best support paired sketching of DUIs by carefully considering the impact of platforms on this activity?*.

Section 2 overviews the state-of-the-art for UI sketching, DUIs, and V2DUIs. Section 3 specifies a workflow where paired sketching is decomposed into four disciplines modeled using the Software and System Process Engineering Meta-Model. Based on these models, Section 4 describes UbiSketch, our collaborative environment for sketching DUIs across pairs of stakeholders and platforms, featuring sketch recognition and whiteboarding. Although UbiSketch can be configured to involve a predefined or variable number of stakeholders in V2DUIs, we propose a protocol for conducting paired DUI sketching, in which two stakeholders sketch one element after another before bringing them together. To evaluate the method for engineering interactive systems, Section 5 reports an experiment involving five pairs of stakeholders who sketched a V2DUI on four platforms. Section 6 discusses implications and limitations of paired sketching for DUIs and Section 7 concludes the paper.

2 Related Work

2.1 Sketching User Interfaces

Sketching, from a general activity [58] to UI design [42], is a well-established method. Initially developed using paper as a medium [70], sketching has evolved significantly due to emerging technologies and sketching software. Next, we discuss contributions in this space in the scientific community (Subsection 2.1.1) and commercially available sketching software applications (Subsection 2.1.2).

2.1.1 Sketching Software in the Scientific Community. The scientific community has largely investigated the support of UI sketching through several software applications, such as JavaSketchIt [12], SketchiXML [18], Calico [48], DigiSketch [35], Gambit [63], Eve [73], Akin [25], SketchInteraction [82], and Sketch2Wireframe [45], to name a few. All of them enable end users to produce sketches with typical pointing devices, such as a pen, a stylus, or a mouse. From the earliest sketching software, such as Calico [48] and SketchiXML [18], which relied primarily on pattern matching [12] and gesture recognition [9], recent advancements have led to applications leveraging deep learning techniques [36] to recognize UI fragments in sketches (low-fidelity) and transform them into wireframes (medium-fidelity) or final UIs (high-fidelity) [45, 82] in code [13]. Transitioning among the three levels of fidelity [49] is key to support progressive and iterative design [18]. Similarly, the navigation among sketches should also be subject to sketching [40], enabling an explicit linking to business processes, such as in AugIR+GC [34]. Similarly, taskSketch establishes a basis for sketching a task model that later initiates a UI sketching activity [55]. Several sketches of the same UI widget, including the most preferred ones [38], are used to train a sketch recognizer, allowing end users to add new representations with just a few samples [18]. Meanwhile, wireframes can be used in forward design processes facilitated by generative AI [10]. Interactive wireframes allow for an exploratory approach to UI sketching compared to static ones [39, 64].

Collaborative features, such as multi-user input management, whiteboarding, and videoconferencing, have been integrated into sketching software [48, 64] yet the predominant paradigm remains individual stakeholders sketching one UI or UI fragment at one time before sharing them [68, 73]; see Table 1. Most software allows some form of collaboration where several stakeholders can participate in the process. For example, AugIR+GC [34] distributes stakeholders interested in business processes on a wall screen while the stakeholders responsible for the UI sketching are gathered on a second wall screen. Most software does not explicitly specify the intended UI target, as a drawing can represent anything, including a user interface for any platform. General pair design [1] is a participatory design method where two stakeholders work on a common problem together. While general and flexible enough to accommodate any combination of stakeholders, the method does not include any precise protocol to carry out DUI sketching. For example, collaborative software such as Calico [48] and Gambit [63] can accommodate two or more stakeholders without any particular protocol. Consequently, UIs intended to run across multiple devices or platforms must be sketched separately for each required variation. For example, SketchiXML [18] and Eve [73] require users to select a predefined screen resolution for each platform, but lack support for creating compatible sketches across multiple platforms in a ubiquitous environment.

Interactions between users and various platforms through sketching require thoughtful design. To extend sketches across multiple devices, input and output modalities, and users, UI design must address specific properties and considerations of the design space [57, 71] to ensure integration and synchronous interactions. For example, Negulescu and Inamura [52] presented a system that allows users to engage with a robot by sketching the environment and specifying the affordances of objects and areas on a map. Hagbi et al. [32] introduced a design framework for developing in-place sketching, particularly focusing on augmented reality games. Chen et al. [15] developed an interactive tabletop sketching system tailored for educational scenarios. In general, the extension of sketching to ubiquitous computing environments, including those inside the vehicle, has not been explored in the scientific literature.

Table 1. Sketching software comparison with UbiSketch (bottom line). The columns represent: #Pl. (number of platforms supported for sketch input), Target (languages for UI targets), Coll.? (whether the software is collaborative), Fidelity (support of the three levels of fidelity—low, medium, and high—based on the following coding scheme: ○=no support, ◔=minimal support, ◑=basic support, ◕=advanced support, and ●=extended support), Reco. (number of widgets supported by sketch recognition), White.? (whether the software application features whiteboarding capabilities of the sketches), Video.? (whether the software application features videoconferencing capabilities).

Software	#Pl.	Target	Coll.?	Fidelity			Reco.	White.?	Video.?
				Low	Medium	High			
JavaSketchit [12]	1	Java	no	◑	○	◔	10	no	no
SketchiXML [18]	3	Java, HTML, XUL	no	◕	◕	◑	32	no	no
Calico [48]	1	-	yes	●	○	○	-	yes	yes
Digisketch [35]	1	-	no	◕	○	○	-	no	no
Gambit [63]	Any	HTML, Java	yes	◕	○	◔	-	no	no
IdeaVis [26]	1	Paper, drawing	yes	◑	○	○	-	no	no
UISkei [66]	1	Java	no	◑	◔	◔	10	no	no
UISKEI++ [65]	Any	Java	no	◑	○	○	10	no	no
PDotCaptured [33]	1	Paper, drawing	yes	◑	○	○	-	yes	no
SciSketch [15]	1	-	yes	●	○	○	-	yes	no
AugIR [40]	1	Any	yes	●	○	○	-	yes	no
AugIR+GC [34]	1	Any	yes	●	○	○	-	yes	no
Swire [36]	1	Any	no	●	○	●	*n*	no	no
Eve [73]	1	Mobile UI	no	◑	◑	◑	≥ 10	no	no
taskSketch [55]	1	Java	no	◑	○	○	-	no	no
SketchingInterfaces [82]	1	Any	no	◕	○	◕	*n*	no	no
Akin [25]	1	Any	no	◕	◑	○	*n*	no	no
UISketch [68]	1	Mobile UI	no	◕	◕	◑	≥ 20	no	no
Sketch-to-Code [13]	1	HTML	no	◕	○	◕	*n*	no	no
MetaMorph [67]	1	Mobile UI	no	◕	◕	◑	≥ 30	no	no
Sketch2Wireframe [45]	1	Any	no	◕	◕	○	*n*	no	no
Adobe XD	1	Any	no	○	◑	●	-	yes	no
FigJam	Any	Drawing	yes	◑	○	○	-	yes	yes
Sketch	1	Any	no	○	◑	●	-	yes	no
UbiSketch	Any	HTML, JavaScript	yes	◕	◑	◔	28	yes	yes

2.1.2 *Commercial Sketching Software.*

While the software applications discussed in Subsection 2.1.1 primarily focus on sketch recognition and transition from low-fidelity to high-fidelity with varying levels of support (Table 1), commercial sketching software tends to focus on one fidelity level with limited transition: low for FigJam and high for Adobe XD and Sketch, for example. FigJam is Figma's web-based collaborative sketching tool where any sketch is shared among all stakeholders through whiteboarding, but no sketch recognition is performed. Most of these applications allow the end user to sketch a single user interface for one environment or context of use at a time.

2.2 Vehicle to Distributed User Interfaces

DUIs are often referred to as UIs that are spatially and temporally distributed in a ubiquitous computing environment, typically running on different devices with different interaction capabilities and screen resolutions [23]. In general, DUI design is centered on distributing one or more interface elements across multiple devices or platforms [78]. This approach considers multi-user support in performing a wide range of tasks in different domains and contexts of use. As a consequence of these definitions, DUIs facilitate a highly adaptable and versatile interaction experience [54]. Previous work has explored use cases and scenarios that showcase the efficiency and flexibility of DUIs in various application areas, such as ubiquitous computing, gaming, or interactions inside smart vehicles [6, 21, 51, 80].

DUIs and V2X (Vehicle-to-everything) combine to form V2DUIs, an integral part of the smart vehicle family of concepts [6], which distribute UI elements across various devices, I/O modalities, users, and environments. For example, smart vehicle proxemics is pivotal for V2DUI [4] and encompasses input and output modalities across various distances to the vehicle. The framework structures interactive zones around the smart vehicle where controlling and receiving data from the vehicle is performed according to proximity. For example, in the personal zone, the driver touches the vehicle to interact with it; in the overt proximal and/or distant zone, the driver can control certain functions of the vehicle using their smartphone or smartwatch; lastly, in the covert zone, the driver uses a laptop to connect to the vehicle and access its data [4].

Designing V2DUIs encompasses the development of effective and safe interaction techniques [22], ensuring accessibility for both drivers and passengers through a variety of input modalities (*e.g.,* touch, voice, mid-air gestures [3]) and output modalities (*e.g.,* visual [16] or audio [74]). For example, Bilius and Vatavu [4] conducted a user study to explore user preferences related to gesture and voice commands for inside-the-vehicle interaction, examining both drivers' and passengers' perspectives. Kern and Schmidt [37] introduced a design space for in-vehicle UIs related to the placement of fixed and mobile devices in the vehicle environment. Diversifying interaction techniques and devices [74] for both drivers and passengers requires distributed interaction design [6].

2.3 Summary

Many software applications exist with general-purpose sketching capabilities, but few enable stakeholders to simultaneously sketch multiple parts of a DUI, especially with sketch recognition. This paper aims to go beyond the current paradigm involving one stakeholder sketching one UI fragment at a time for one platform by enabling pairs of stakeholders, *e.g.,* a designer playing the role of the guide who drives the sketching and a user representation who sketch together. However, when the software supports collaboration, no particularly precise method is available to structure sketching. Instead, we suggest paired sketching, a collaborative design approach in which two users work together to design the DUI. In this context, this paper contributes the following:

- A definition of a protocol for the paired sketching of DUIs, along with a supporting workflow modeled into four disciplines using the Software and System Process Engineering Meta-Model.
- An implementation of UbiSketch, a collaborative software environment tailored for sketching DUIs with pairs of stakeholders and platforms, featuring sketch recognition and whiteboarding.
- An experiment involving five pairs of stakeholders sketching a DUI for in-vehicle interaction on four platforms, designed to evaluate the developer experience (Dev-X) [60], a means of capturing how developers think and feel about their activities within their work environments.
- A comparison of paired sketching against pair design/drawing.

3 Engineering Methodological Support for Paired Sketching

By definition, a *method* is a particular form of procedure for accomplishing or approaching something, especially a systematic or established one. In the case of this work, the method is a systematic form for prototyping graphical

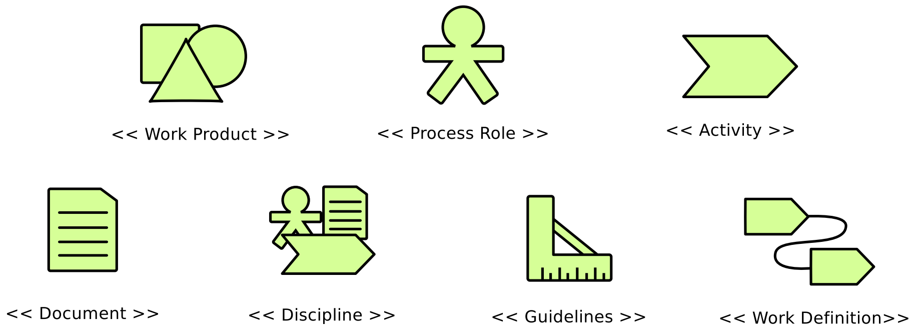

Fig. 2. A subset of the SPEM V2.0 notation relevant to paired sketching.

UIs, according to the procedure of paired sketching. This section defines our methodological support for paired sketching as envisioned in our UbiSketch software application. The Software and Systems Process Engineering Meta-Model (SPEM V2.0) is an Object Management Group (OMG) specification of any UML metamodel or profile used to represent a family of software development processes and their components. It forms a reference by providing the minimum set of process modeling elements to describe any software development process without adding specific models or constraints. To this end, we rely on seven concepts from the SPEM V2.0 notation [30] to formally describe the method of paired sketching (Fig. 2), as follows:

- *Work Definition*: constitutes a kind of operation that describes the work performed in the process.
- *Work Product*: any tangible piece of information that is produced, consumed, or modified by a process.
- *Process Role*: defines responsibilities over specific Work Products.
- *Activity*: constitutes a piece of work performed by a single Process Role.
- *Document*: a special kind (a stereotype) of Work Product.
- *Discipline*: partitions the Activities within a process according to a common theme.
- *Guidelines*: constitutes an element aimed at providing more detailed information about a resource.

The roles if the method are usually abstracted into designers, who must tell users what they mean by the artifact they have created, and users are expected to understand and respond to the message from the designer. Therefore, the Process Roles considered for our method are as follows:

- *Designer*: this role is responsible for producing DUI sketches, conducting the paired sketch session, and performing user tests with sketches, if relevant. We group all other relevant stakeholders under the umbrella "Designer" to simplify the definitions. Such stakeholders are practitioners, developers, graphic artists, etc.
- *User*: this role is responsible for providing feedback throughout the sketching session, both by assisting collaboratively sketching the UIs and prototypes and by validating the sketches of the prototypes.

Our method for paired sketching is described through four *disciplines* related to the four activities (Fig. 3). The purpose of constructing such a method is to support the activities observed, keeping in mind the basis of *sketching as a tool* to help designers and users to communicate effectively. To this end, we first abstract the procedure observed during sketching sessions [82] to make the method technology-independent. From the activity of sketching, we abstract the structure definition; from *prototyping*, we derive the behavior definition; and from *sharing/testing* and *discussing/reflecting*, we abstract testing and reflection, respectively. Each discipline is detailed in the next subsections via SPEM packages and workflows.

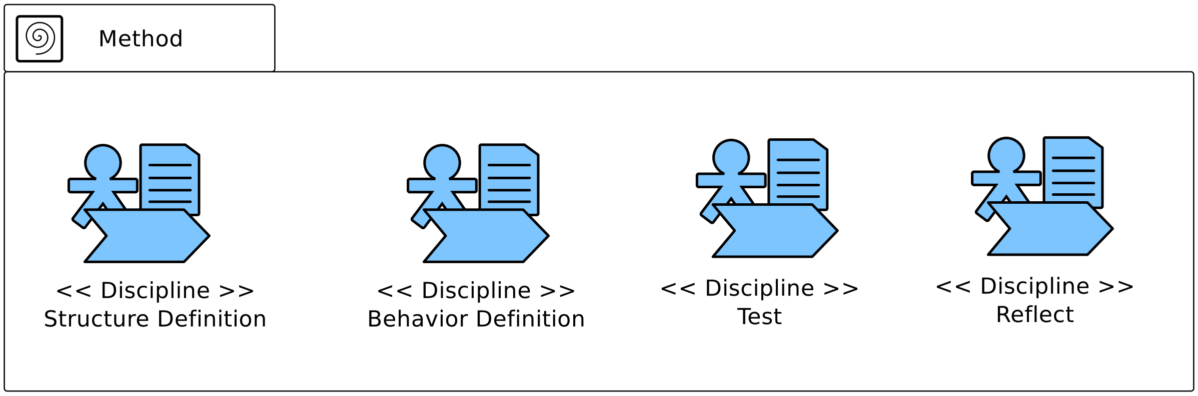

Fig. 3. We define the pair sketching method as a SPEM package with four disciplines: structure definition, behavior definition, test, and reflect; see the text for details.

3.1 Discipline 1: Structure Definition

The first thing conveyed with sketching across domains is structure [77]: designers define the UI structure by sketching the UI elements and composing them in containers. While interviewing users, the designer is concerned about the UI content and what is to be shown. The diagram in Fig. 4 shows:

Process Roles. The designer is primarily responsible for sketching the UIs and organizing the content as the specialist who will deliver a solution to the user, who will validate the work produced. The user can also produce sketches, if desired.

Documents. The designer uses and also contributes to *Task Analysis*, *User Stories*, *Scenarios*, *Personas*, and any type of knowledge regarding the user needs. However, the documents are not prerequisites for conducting this stage, but are rather receiving contributions throughout the process.

Guidelines. The designer can use guidelines for sketching UIs, such as *Heuristics* and *Design Guidelines* [2].

Work Products. These are represented by *User Interface drawings* that can be made in low-fidelity in the case of sketches and high-fidelity in the case of pixel-perfect interface tools. Mixed fidelity is also possible [17].

The workflow begins with the structure definition by designers and users and iterates to the next stage when both parties concur (*[structure iteration] / [structure ready]*); see Fig. 5. The designer can start the process by either *Analyzing Previously Produced Interfaces* and *Drawing User Interfaces* or simply by sketching them. For these activities, the designer uses documents such as *User Stories*, *Task Analysis*, *Scenarios*, along with guidelines such as *Design Guidelines* and *Heuristic Rules*. The user should provide feedback on the produced DUIs until the structure is agreed upon and validated. The designer organizes the content to proceed to the next stage.

3.2 Discipline 2: Behavior Definition

This stage adds a behavior definition to the sketches: the designer focuses on the practical usage of the system, or how the information will be presented, while the different sketches are assembled sequentially. The diagram in Fig. 6 shows the following SPEM constructs:

Process Roles. The designer is responsible for defining the behavior by organizing the structure produced in the former stage. The user provides feedback and validates the behavior.

Guidelines. The designer uses and contributes to *Task Analysis*, *User Stories*, *Scenarios*, and any type of **knowledge** regarding the user's needs.

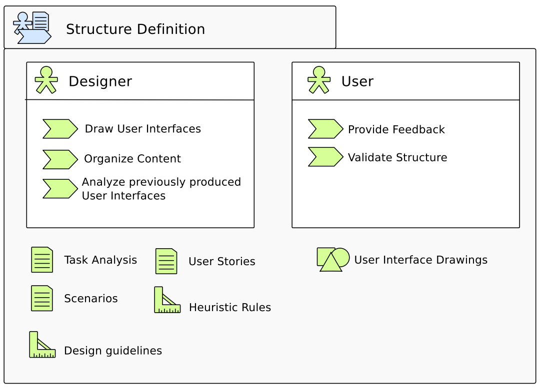

Fig. 4. Discipline for structure definition.

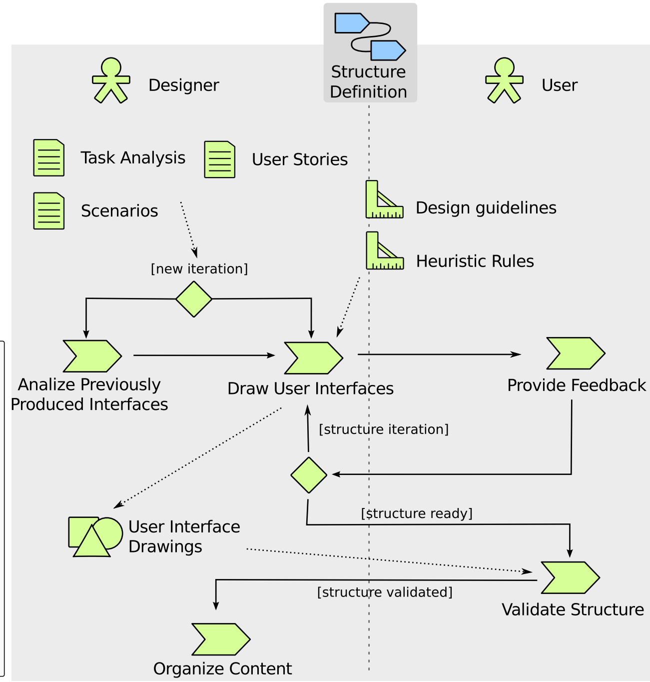

Fig. 5. Activity Diagram for structure definition.

Work Products. The *Work Products* can be low-fidelity UI prototypes, high-fidelity prototypes, or both.

In this stage, the user and the designer iterate until the behavior definition is agreed upon. Fig. 7 presents details about this stage, which starts with the designer assembling the different sketches produced previously (*Organizes the User Interfaces*) and then proceeds to *Make the Storyboard* which *connects* the sketches in a logical manner. This activity considers the *User Stories*, *Task Analysis*, *Scenarios* along with guidelines such as *Design Guidelines* and *Heuristics*. The designer and the user may iterate until an agreement about behavior is reached. The *User Interface Prototypes* are created in this activity for further validation by the user. They may also iterate backwards to the structure definition, since the definition of behavior (how) often leads to insights about the content to be presented (what) [7].

3.3 Discipline 3: Test

Each pair transfers their sketches on a wall, arranged to form a storyboard. An overall validation takes place where the navigation flows are played or tested. Participants typically point to the sketches following the previously designed navigation in a sort of storytelling. The diagram in Fig. 8 shows the following SPEM constructs:

Process Roles. The user is the most active in this discipline, being the stakeholder who is going to use and validate the produced design. The designer is responsible for specifying tasks when a user test is planned.

Guidelines. If a user test is planned, *Evaluation Protocols* can be used, such as *Thinking Aloud* where the user would explicitly describe what they are doing while navigating through the sketches.

Work Products. This discipline generates information about the usage of the designed solution when supported by a tool that registers the path taken by the user navigating with the prototype. In usability studies, designers often record videos of users interacting with the prototype, for documentation purposes.

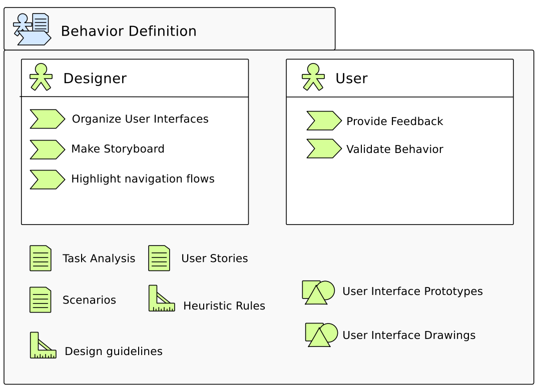

Fig. 6. Discipline for behavior definition.

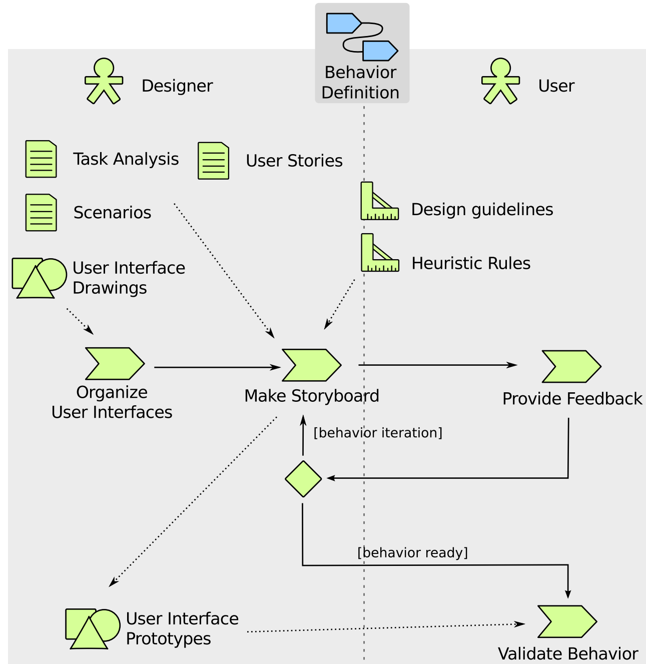

Fig. 7. Activity Diagram for behavior definition.

This stage implies active user participation since this role validates the prototype (Fig. 9). This activity starts with the designer specifying tasks that match the *Scenarios*, *User Stories*, *Task Analysis*. This should be evaluated by planning a user study with *Evaluation Protocols* where the user is testing the prototype and observed, or with techniques such as cognitive walkthrough [43]. The user should *Perform the Tasks* while *Using the prototype*, and the designer has the role of *Observing the usage*, from which *Usage Data* might be collected.

3.4 Discipline 4: Reflect

The modifications are agreed upon once the design is discussed and the potential problems highlighted. The group, the pair of stakeholders, proceed to another iteration of drawing/prototyping/sharing until both parties are satisfied with the results. The diagram in Fig. 10 shows the following SPEM constructs:

Process Roles. The designer and the user discuss potential problems and modifications based on the reflection about the *Usage Data* product in the previous step. Ultimately, the designer is responsible for producing a report documenting the decisions and eventually a *List of Requirements*.

Guidelines. Discussions, for example through *Brainstorming*, can take place at this stage.

Work Products. The designer produces a *List of Requirements* after each iteration. When the iterations are completed, a *Design Session Report* is written and kept for reference to future sessions.

In this stage, the designer and the user rely on brainstorming techniques to discuss potential *Problems* and *Modifications* (Fig. 11). This activity starts by splitting the work between the two, both performing essentially the same activity since they discuss problems and modifications. The designer is responsible for producing a report at the end of the activity, which typically includes a *Requirements List*. *Observations* can be seen as a series of

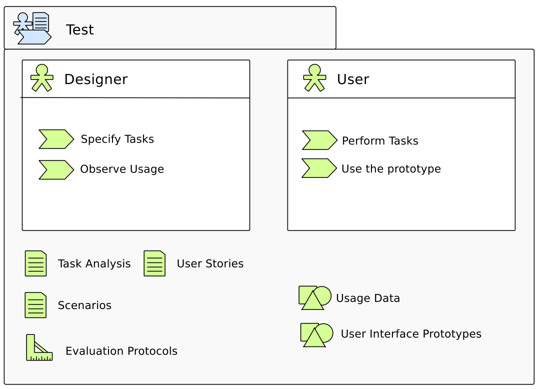

Fig. 8. Discipline for test.

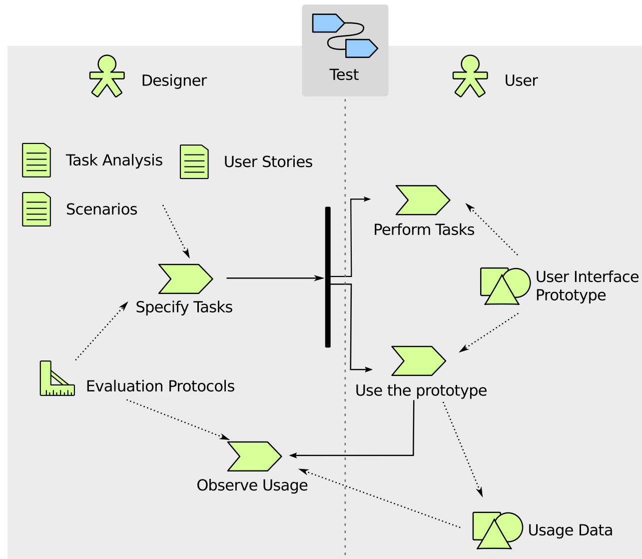

Fig. 9. Activity Diagram for test.

notes posted to the wall. In case the iteration is over (*[design phase finished]*) a report is produced; otherwise a new iteration is initiated.

Figure 12 presents the workflow that links all disciplines together involving all the *Work Products*, *Documents*, and *Guidelines*, represented as resources if consumed in the discipline or as outcomes, if produced.

4 Development of UbiSketch

Based on the models of Section 3, we developed UbiSketch, an interactive HTML5 application that runs on any platform equipped with a HTML5-compliant browser. The UbiSketch interface consists of a drawing area that uses the HTML5 <canvas> element and Javascript routines to capture mouse, pen, and touch events. On its left side, a toolbar enables switching from sketching to control functionalities. A wall screen synchronizes the sketches and arranges them like sheets of paper, enabling dragging, dropping, and (un)grouping. UbiSketch was designed to support many clients simultaneously running on smartphones, tablets, desktops, and tabletops managed by a central web server. Clients run the same frontend through a browser or a wrapper that displays UbiSketch without browser controls. Communication with the server is achieved through asynchronous requests via Asynchronous JavaScript and XML (Ajax). A stakeholder can select any platform: a lightweight device for a face-to-face meeting to sketch with a client and a large interactive surface for online meetings. A pair of stakeholders can choose two medium devices, such as a tablet and a laptop. A stakeholder shares any sketch by dragging it from the original device and dropping it on the whiteboard on the wall (Fig. 13).

The server manages the users, collaborative activities, and sketches through the Eclipse Sketch recognition API. The sketches can also be used in any native desktop application constructed with Eclipse, such as its graphical editors. This component processes the sketches independently of UbiSketch according to the *"lazy or postponed recognition"*, "where sketches are not immediately recognized and replaced with high-fidelity versions, in order to preserve the original look and maintain the designer's creative flow and ongoing conversation. Sketch recognition [9, 18] is deferred to completion or initiated on demand by the end user. The sketches are stored in InkML format.

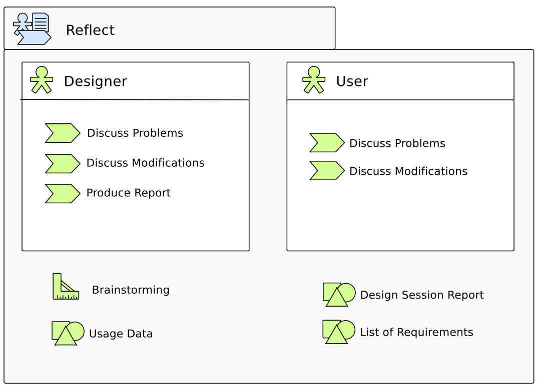

Fig. 10. Discipline for reflect.

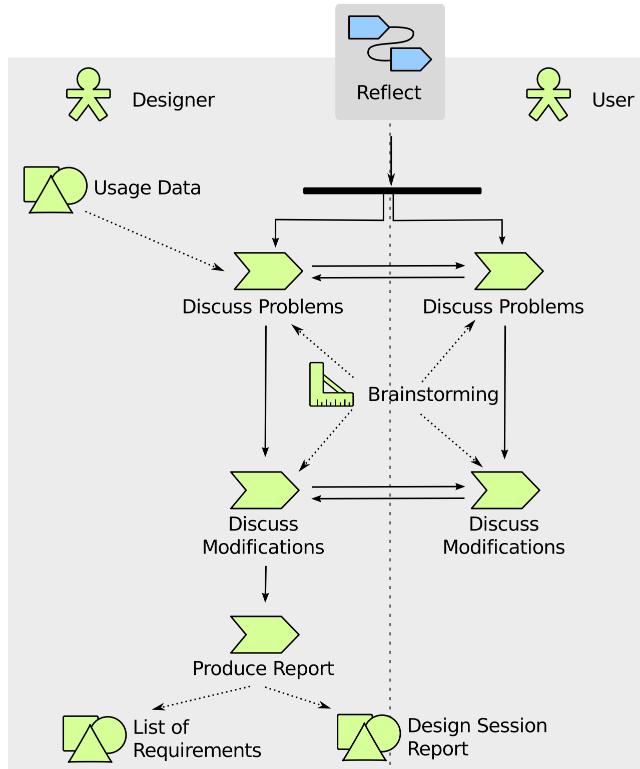

Fig. 11. Activity Diagram for reflect.

The implementation of UbiSketch is divided into two main modules: the *backend*, the server that responds to requests, manages the communication between devices, and maintains the database, and the *frontend*, the interface seen by designers and users across multiple devices. Fig. 16 depicts the relation between the backend and frontend: a single interaction of a user on a mobile device is shown on the left, the user is touching the screen to produce a stroke. This is captured by the onTouchEvent() method and is propagated to the backend that maintains the session and scenes. The backend updates the scene and transmits the changes to the other users.

4.1 Backend of UbiSketch

The backend is implemented as a cloud-based service to provide a lightweight application that can withstand a load increase. Since UbiSketch is a web application, there is no need to install anything. The software is deployed in Google App Engine to allow a group of servers to dynamically adjust the size of the service. The database is also distributed and replicated across the servers. The backend is responsible for the server that responds to requests from the frontend, and manages the communication between devices. This module also maintains the database and accesses it through the Objectify library. Fig. 14 shows that the backend is visible through a web service called GamService and exposes several methods to manipulate the objects of the core model. This service is the interface that can be accessed through the frontend and also external web services, such as websites and applications.

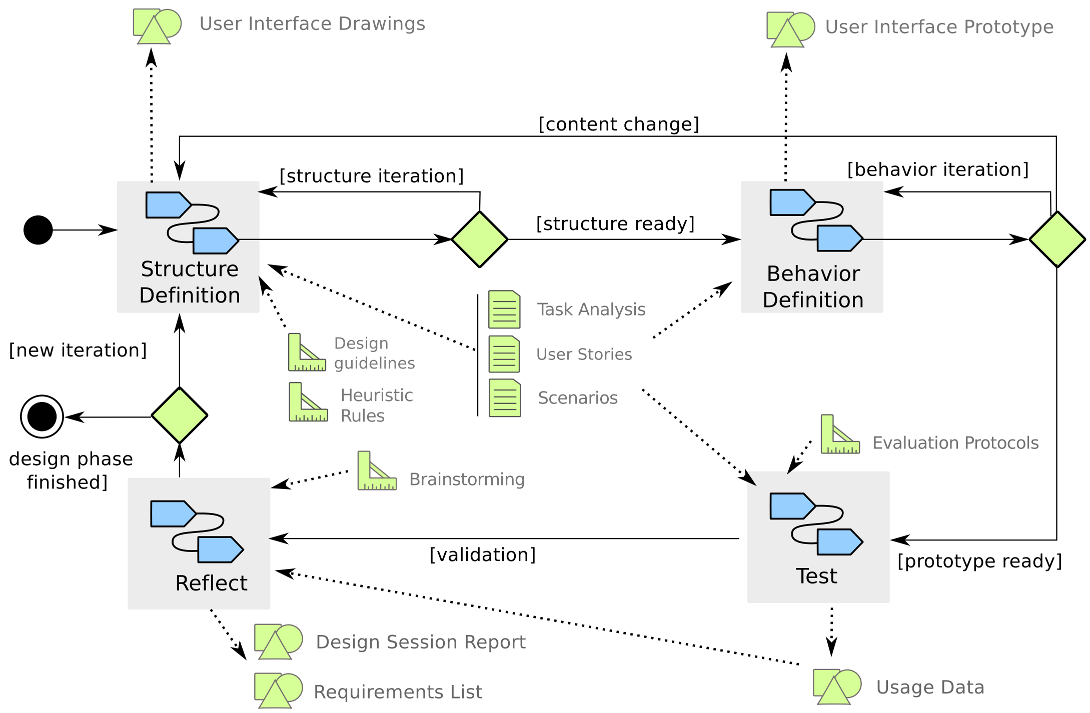

Fig. 12. Method workflow.

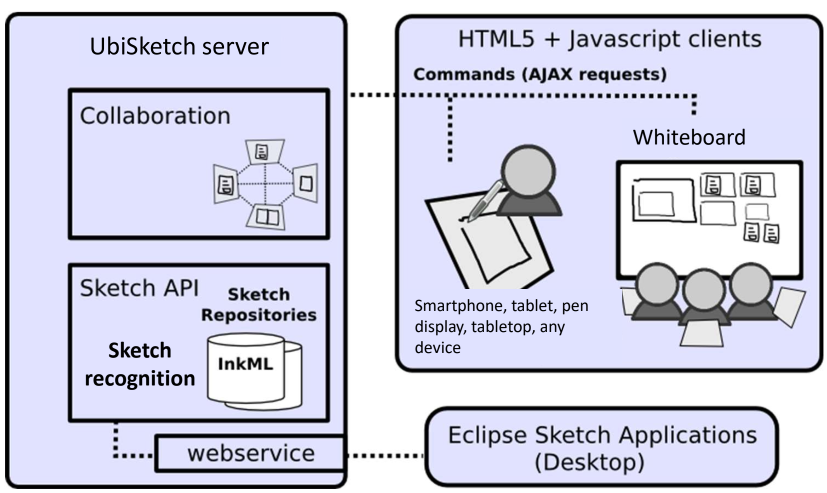

Fig. 13. Software architecture of UbiSketch.

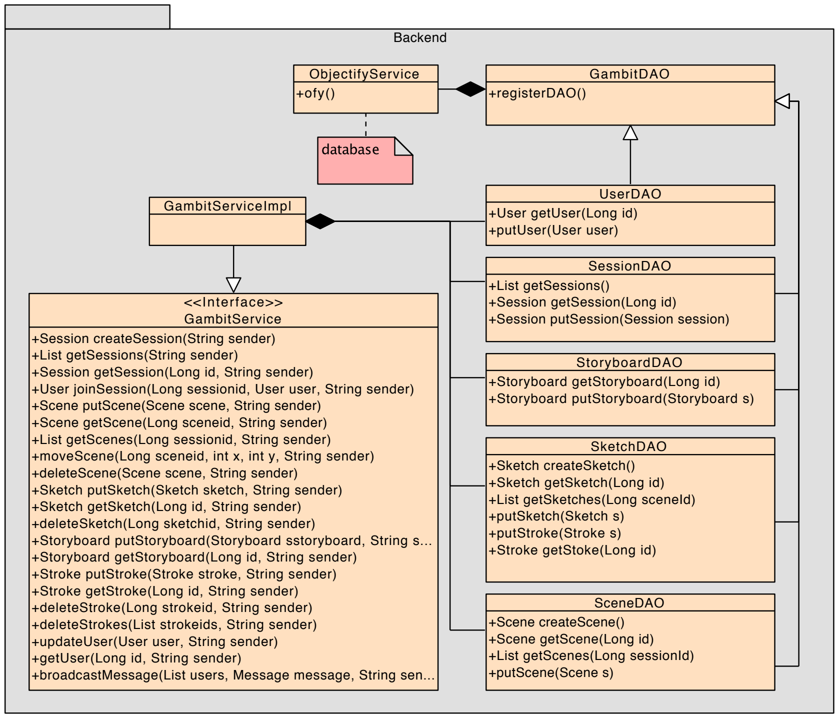

Fig. 14. Backend of UbiSketch.

4.2 Frontend of UbiSketch

The frontend is implemented in HTML5 and Javascript to be used on any Chromium-compatible device. The frontend is responsible for presenting the same interface on all devices, capturing mouse or touch events, and communicating with the backend to update *Scenes*, *Sketches*, and *Strokes*. Fig. 15 shows that the main object of the frontend is *Canvas*, which controls and updates the HTML canvas element through its Context2D object.

5 Evaluation

In general, paired sketching of UIs is referred to as a collaborative design technique in which two stakeholders work in pairs (e.g., a developer and an end user, a designer and a developer, or two developers) to prototype a UI through sketches. Each pair can work in parallel with other pairs to sketch any part of a DUI, either focusing on the same part to be distributed across the different platforms (the task remains constant, the platform varies) or concentrating on one platform to ensure consistency (the platform remains constant, the task varies). The distribution of stakeholders in pairs intrinsically reflects the distributive nature of DUI. Each pair session can address one variable at a time, such as the members of the pair, the task, the platform, or the environment. UbiSketch has been specifically designed to support DUI prototyping using paired sketching: the sketches are recorded by one member, by pairs of members according to tasks, platforms, and environments. *Developer experience* (Dev-X [24, 60]) is often referred to as a means to capture how developers think and feel about activities within their work environments [29]. In this context, we wanted to evaluate how UbiSketch can support paired sketching of DUIs and understand the impact of different platforms on this activity.

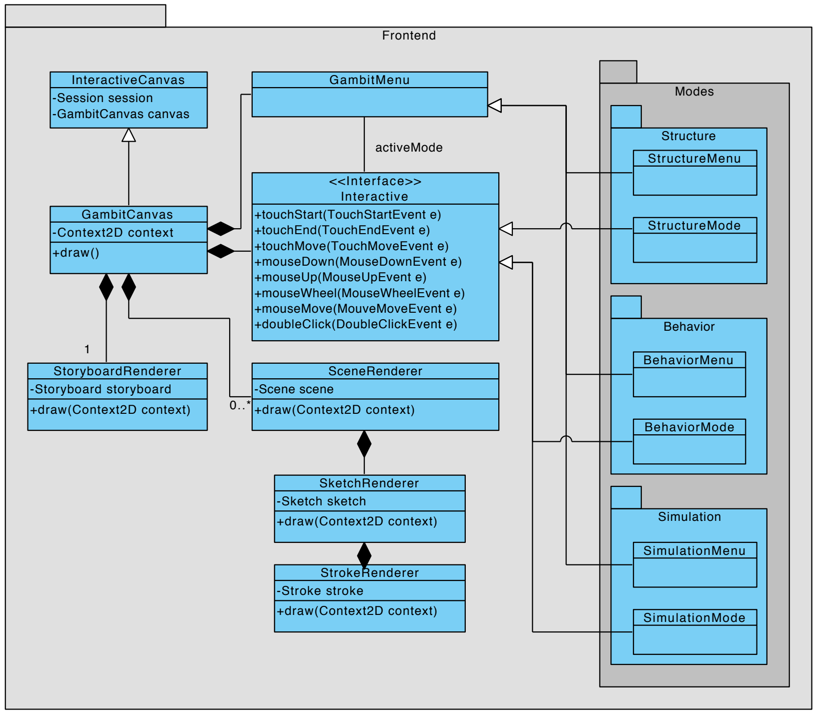

Fig. 15. Frontend of UbiSketch.

5.1 Method

5.1.1 Participants. To be compliant with the Dev-X context, our sample included professional designers and developers of UI/UX (10 men, M=26.7 and SD=4.9 years) with prior experience in DUIs. One participant reported using UI sketching frequently in his professional work, two participants reported moderate use, and the rest were familiar with and had been sporadically involved in sketching for UI design. Five pairs were formed based on the participants' self-reported expertise.

5.1.2 Apparatus and Setup. We conducted the experiment with UbiSketch running on four devices: *smartphone* (Apple iPhone 7 with a 4.7-inch 750×1334 pixels display; see Fig. 17 ⑥), *tablet* (Samsung P7100 Galaxy Tab 10.1v with a 10.1-inch 800×1280 pixels display; see Fig. 17 ④), *pen display* (Wacom Cintiq Pro 16 with a resolution of 3840×2160 pixels; Fig. 17 ⑤), and *tabletop* (Promethean ABAS595PEST ActivBoard 595 Pro with a HD resolution of 1280×800 pixels; see Fig. 1). All screens feature 16,777,216 colors and are touch-enabled via a pen. These resolutions were inspired by the most common screen resolutions for mobile, tablet, and desktop by BowserStack and by the display resolution for the tabletop. We randomly assigned devices to each of our five groups of participants, as follows (see Fig. 17 ①, ②, and ③):

- Group #1: *tabletop*, *tablet*, and *smartphone*.
- Group #2: *tablet, tabletop*, and *smartphone*.
- Group #3: *smartphone*, *tablet*, and *tabletop*.
- Group #4: *pen display, tablet,* and *smartphone*.
- Group #5: *tablet, tabletop*, and *pen display*.

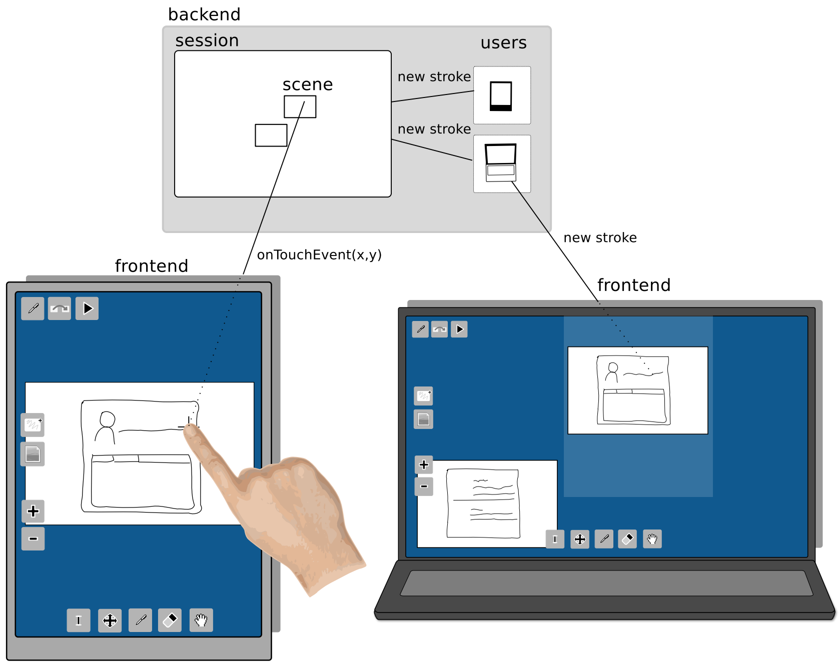

Fig. 16. Relation between the frontend and the backend of UbiSketch.

5.1.3 Protocol and task. Each group received instructions to sketch a V2DUI for multimedia infotainment using UbiSketch and their assigned platforms. We suggested the following protocol for paired sketching of DUIs:

> The first stakeholder sketches just a few UI features. The following rule of thumb is agreed upon between stakeholders: when one of them lifts the pen off the device for some time or hesitates about what to sketch next, it means that their turn is finished. The other stakeholder takes the pen and sketches some more features and/or modifies previously sketched features. Rounds follow each other until stakeholders feel that there is no longer a need to add, delete, or modify any feature of the interface. In the beginning, stakeholders can sketch silently without necessarily explaining or justifying what has been sketched unless they wish to do so. They are allowed to discuss design issues about any interface element at any time.

The main principles to follow during this protocol are:

(1) *Balance co-creation with co-modification*: Stakeholders are encouraged to create new features that can be potentially improved later in a way that is balanced between stakeholders.
(2) *Sketch incrementally*: Stakeholders are encouraged to sketch on each other's input, not just on their own.
(3) *Agree on a stopping criteria*: Stakeholders are invited to agree on stopping criteria.
(4) *Provide continuous feedback*: Stakeholders are encouraged to provide feedback to each other at any time on any sketched element, such as in the form of explanation, justification, motivation, interrogation, etc.

The experiment started with the participants signing a consent form and completing a demographic questionnaire. Participants were then presented UbiSketch, which they tested using the first platform for about ten minutes (Fig. 17, bottom). They were then asked to fill out the IBM Computer Satisfaction Usability Questionnaire

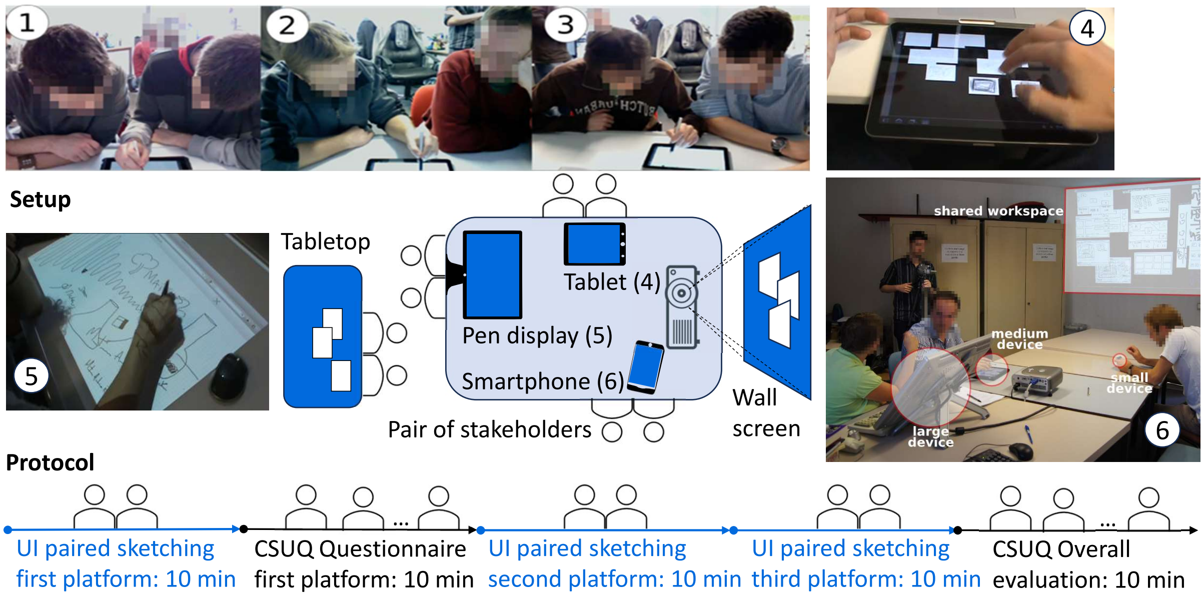

Fig. 17. Setup of the experiment (top): platforms (left and right) table setup (middle), and protocol experiment (bottom).

(CSUQ) [44] to provide feedback on UbiSketch and the devices they had just used.[1]. At this stage, the participants did not know about the other platforms assigned to their group. Subsequently, they used UbiSketch with the other two platforms for ten minutes per platform. Finally, they provided feedback about their experience with UbiSketch and all platforms, including comparisons between platforms; see the next section for the specific measures we evaluated.

5.1.4 Measures. The independent variable is:

- Device Type: a nominal variable specifying the device used for the V2DUI sketch with UbiSketch: smartphone, tablet, pen display, and tabletop.

We measured the following dependent variables:

- Dev-X Statement: an integer variable denoting the participant's rating for each IBM CSUQ [44] positive statement according to a 7-point Likert scale [47] (1=strongly disagree, 7=strongly agree).
- IBM CSUQ metric: an integer variable [1..7] denoting the aggregated rating of participants for four variables, *i.e.,* system usefulness (SysUse, Q1-Q8), information quality (InfoQual, Q9-Q14), interface quality (InterQual, Q15-Q18), and overall usability (Overall, Q1-Q19) [44].
- Reactivity: an integer variable [1..7] denoting the participants' ratings about the perception of UbiSketch time responsiveness [46] running on a specific platform [53].
- Intention to Use: an integer variable denoting the extent to which one intends to use [20] paired sketching with UbiSketch. This measure represents a perceptual judgment, rated from 1 to 7, of the method's efficacy and cost-effectiveness [14], related to the likelihood of the method being accepted in

[1]We used this questionnaire because it was empirically validated on a large number of interactive systems with very high reliability (Cronbach's α=.89) [44].

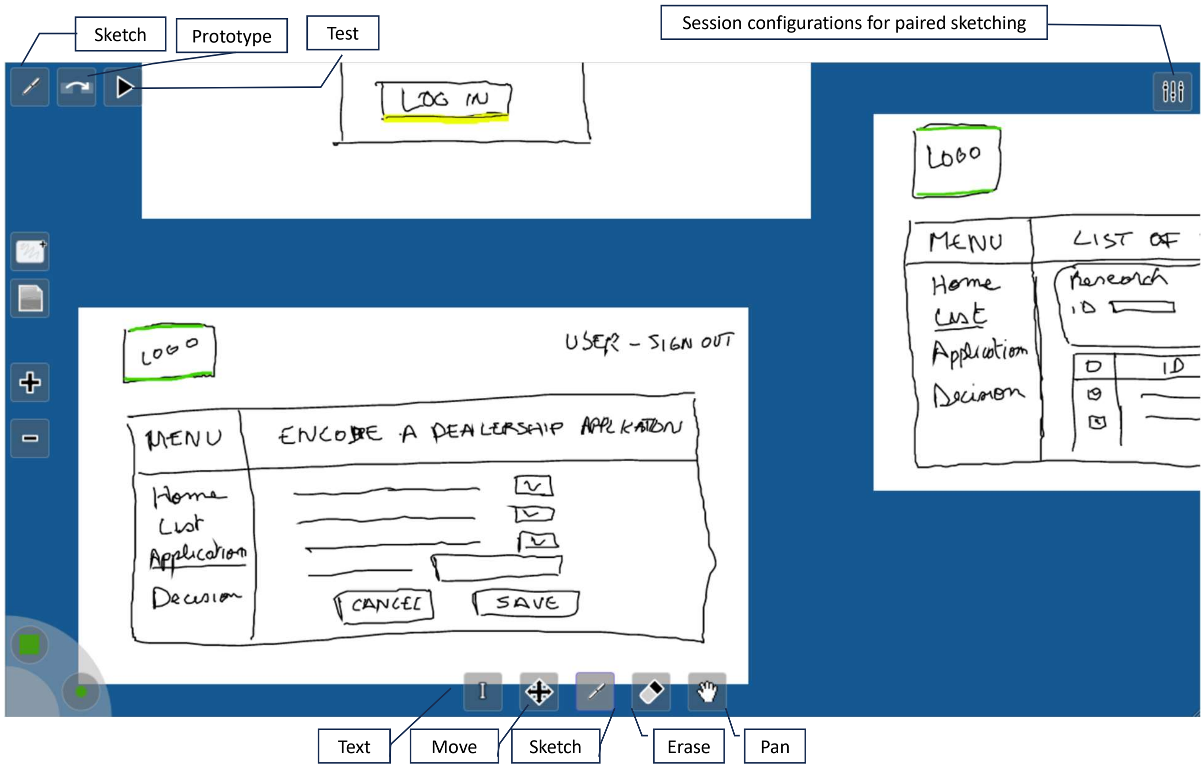

Fig. 18. Layout structure of the UbiSketch user interface.

practice, two important metrics in Dev-X [60]. The hypothesized causal relationships suggest that perceived ease of use and perceived usefulness affect intentions to use a method.

- Perceived Satisfaction: a rating on a 7-point Likert scale [47] denoting the overall satisfaction and importance of the six aforementioned measures (1=strongly dissatisfied, 7=strongly satisfied) [81].
- Comments: an optional set of three freeform feedback comments, either negative (*e.g.*, a critique), positive (*e.g.*, an appreciation), or constructive (*e.g.*, a suggestion for improvement). We also audio-recorded all conversations and converted them into transcripts.

5.2 Results and Discussion

We collected data for 5 pairs × 2 participants × 3 Device Types = 30 interactive sessions.

5.2.1 Dev-X Statements. Prior to analyzing the results of the Dev-X Statements, we computed various coefficients to assess the reliability of the IBM CSUQ (Table 2): the internal consistency reliability was excellent (Cronbach's α>.9), very strong (Spearman's ρ>.8), and very practical (Guttman's λ>.81 [31]). Although only n=10 participants were involved in the experiment, the global reliability was assessed as excellent as well as the individual Dev-X Statement via the Cronbach's α coefficients, all above .9; see Table 3. The individual difficulty ranged from medium to high (min=2.6, max=6.1). Some Dev-X Statement were assessed as not very discriminative (*e.g.*, Q_7, Q_{11}, Q_{12}) while others as very discriminative (*e.g.*, Q_2, Q_4, Q_6). In particular, Q_{15} was assessed as having a negative correlation ($\rho = -0.31$) with an inverse low discrimination ($D = -0.33$). A t-test

Table 2. Coefficients for the overall questionnaire reliability: type, value, and interpretation.

Coefficient	Value	Interpretation
Cronbach's α	0.947	Excellent [75]
Spearman's ρ (Halves)	0.942	Very strong [72]
Spearman's ρ (Odd-Even)	0.986	Very strong [72]
Guttman's λ (Halves)	0.851	Very practical [31]
Guttman's λ (Odd-Even)	0.969	Very practical [31]

Table 3. Coefficients for analyzing individual Dev-X Statement.

Coefficient	Q_1	Q_2	Q_3	Q_4	Q_5	Q_6	Q_7	Q_8	Q_9	Q_{10}	Q_{11}	Q_{12}	Q_{13}	Q_{14}	Q_{15}	Q_{16}	Q_{17}	Q_{18}
Cronbach's α	.942	.940	.941	.940	.940	.944	.941	.947	.946	.946	.950	.945	.946	.955	.946	.943	.946	.939
Difficulty	3.20	3.90	2.90	3.00	2.80	4.20	6.10	3.00	2.30	2.60	2.70	3.80	3.90	3.60	5.30	4.70	3.90	2.60
Discrimination	3.00	4.67	4.33	5.33	4.00	4.67	1.67	4.00	1.00	1.33	1.00	1.00	1.67	1.33	-0.33	2.00	4.00	2.33
Correlation	0.77	0.94	0.91	0.96	0.89	0.93	0.81	0.90	0.45	0.53	0.40	0.22	0.50	0.52	-0.31	0.61	0.87	0.57

revealed that two of the nineteen Dev-X Statements were significantly above the median Mdn=4, as follows: Q_7 ("It was easy to learn to use this system", M=6.1, SD=0.8, $p\leq.001$, r=.93) and Q_{15} ("The organization of information on the system screens is clear", M=5.30, SD=0.9, p=.00098), which suggest that UbiSketch was easy to learn and its structure was straightforward, whatever the platform used. Four Dev-X Statements were significantly below the median: Q_9 ("The system gives error messages that clearly tell me how to fix problems", M=2.30, SD=1.49, $p\leq.0028$), Q_{10} ("Whenever I make a mistake using the system, I recover easily and quickly", M=2.60, SD=1.17, p=.0022), Q_{11} ("The information (such as online help, on-screen messages, and other documentation) provided with this system is clear", M=2.70, SD=1.49, p=.011), and Q_{18} ("This system has all the functions and capabilities I expect it to have", M=2.60, SD=1.43, p=.01). UbiSketch did not produce any error message to not disrupt the sketching process. When a UI element was not recognized, it was simply left as is, and the stakeholder could delete any DUI element not recognized. Furthermore, not all DUI elements could be recognized, which did not create any differentiation on the UbiSketch interface.

Fig. 19 shows a divergent stacked bar graph with the Dev-X Statements ratings for each Device Type. This graph is adequate to show the spread of negative (1=strongly disagree) and positive (7=strongly agree) values because they align with each other around the neutral point of the scale (Mdn=4), therefore facilitating comparison [56]. A one-way ANOVA showed a statistically significant difference in Dev-X Statements between Device Types ($F_{3,186}$=35.25, $p\leq0.001$) with a large effect size (η^2=0.36), suggesting that the devices were appreciated differently. For example, a Kruskal-Wallis test (H=66.67, df=3, $p\leq0.001$) with Nemenyi pairwise comparisons showed that the tabletop received higher ratings than the smartphone (R=68.79, $p\leq0.001$), the pen display (R=58.04, $p\leq0.001$), and the tablet (R=39.11, p=0.0019). Looking more closely at the ratings for each Device Type, the results fall into three main categories:

(1) *Positive ratings that do not vary according to Device type*: statements Q_6, Q_7, Q_{15}, Q_{16}, and Q_{17} belong to this category. For example, Q_6 ("I feel comfortable using this system") received the following average ratings: M=1.5 for the pen display (SD=0.5), M=3 for the smartphone (SD=1.0), M=4.75 for the tablet (SD=0.43), and M=7 for the tabletop, with an average of M=4.2 (SD=1.94). These positive statements were mostly related to ease of use and learning. For example, Q_7 ("It was easy to learn to use this system") received the highest average value (M=6.10, SD=0.8), reinforcing that paired sketching supported by UbiSketch was

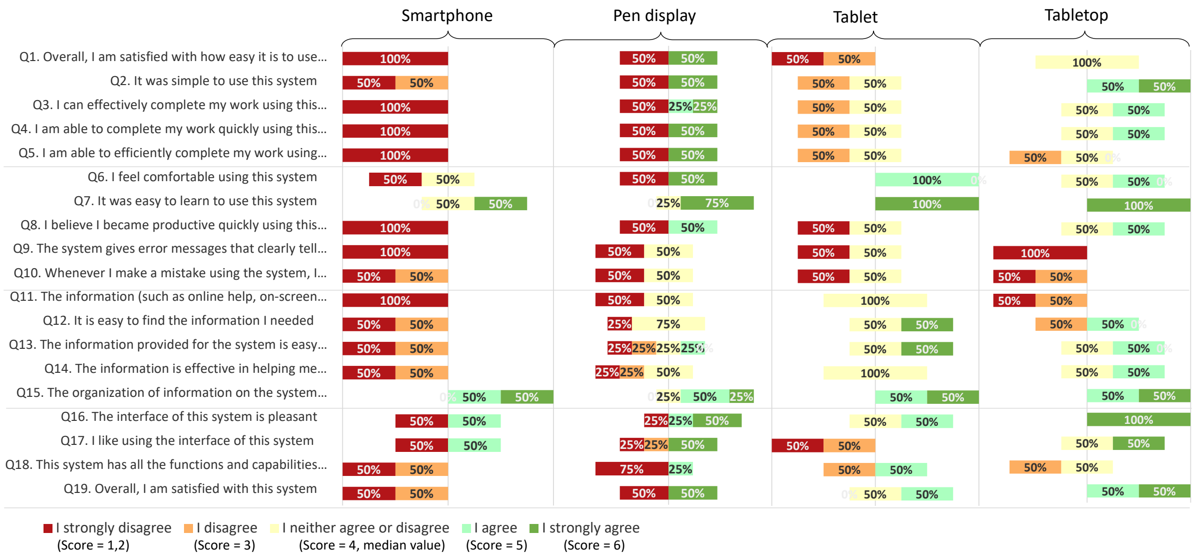

Fig. 19. Ratings of the Dev-X Statements of UbiSketch per Device Type.

easy to use and learn, one of the most important benefits. No tutorials, demonstrations, or explanations were provided during the experiment to allow the participants to explore and use UbiSketch as they wished. Statement Q_{15} ("The organization of information on the system screen is clear") received the second highest average score (M=5.3, SD=0.9), showing that the UbiSketch layout, decomposed into three zones (Fig. 18), was well perceived. Statement Q_{16} ("The interface of this system is pleasant") came in third position (M=4.7, SD=1.4), suggesting that the participants enjoyed paired sketching in UbiSketch.

(2) *Initially negative, progressively positive ratings that vary largely according to Device Type*: statements belong to this category evolved favorably from the smallest device to the largest. Most participants agreed that screen size and resolution impact performance. The smartphone, the smallest device in our experiment, received the worst results, contrasted by the tabletop, the largest device. For example, Q_1 ("Overall, I am satisfied with how easy it is to use the system") was rated M=1.5 for the pen display (SD=0.5), M=2.5 for the smartphone and the tablet (SD=0.5), and M=6.5 for the tabletop (SD=0.5), with an overall score of M=3.2, SD=1.83. This statement starts with a negative result (all participants agreed the device was too small, even for sketching a smartphone UI, and tended to prefer larger devices even for sketching small-size UIs) to end up with a moderate result (all participants reached the median value). Participants also reported that the CPU performance affected sketching recognition, with the smallest device also having the lowest computational capabilities compared to the largest. Device speed matched screen size preferences. This finding could explain why items related to efficiency were rated lower on smaller devices than on larger ones. For example, statements Q_3 ("I can effectively complete my work using this system") and Q_4 ("I am able to complete my work quickly using this system") received low ratings (M=2.9, SD=2.0 and M=3.0, SD=2.3, respectively), which increased in the end. Participants perceived not completing their tasks quickly enough (*e.g.*, Q_3, Q_4, Q_5, and Q_8), although they felt comfortable with UbiSketch (Q_6).

(3) *Irrelevant items received inconsistent ratings*: some statements were irrelevant for UbiSketch although we kept them in the evaluation to not endanger the empirical validation of the questionnaire. For example, statement Q_9 ("The system gives error messages that tell me how to fix problems") received the lowest

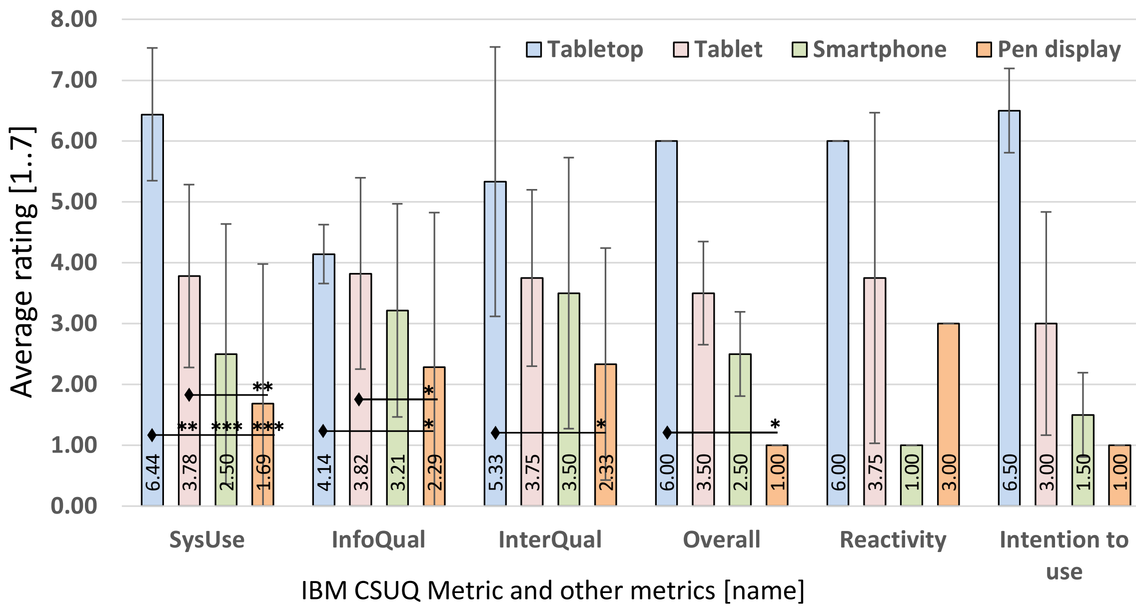

Fig. 20. Pairwise comparison of the various device types used in the experiment. Error bars show 95% confidence intervals; star notations denote $p \leq .05^{*}$, $p \leq .01^{**}$, $p \leq .001^{***}$.

score on average (M=2.3, SD=1.4) because no message of any type was produced by UbiSketch while sketching, and statement Q_{10} ("Whenever I make a mistake using the system, I recover easily and quickly") received the second lowest score (M=2.6, SD=1.1). UbiSketch did not inform the stakeholder which parts of the sketch were recognized, which was deferred to the end of the task or on demand. Several tools automatically recognize the UI fragment being sketched [18, 25, 45], but this option was not automatically activated in our experiment. The participants were free to assess any Dev-X statement or to consider them as not appropriate.

5.2.2 CSUQ Metrics. The IBM CSUQ metrics received moderate ratings when averaged across all Device types: system usefulness (M=3.64, SD=2.12), information quality (M=3.46, SD=1.57), interface quality (M=3.73, SD=1.79), and overall (M=3.47, SD=1.92). Given the rating variations according to Device Type observed in our experiment, we present individual ratings for each CSUQ metric; see Fig. 20. A one-way ANOVA revealed a main effect of Device Type on SysUse ($F_{3,60}$=37.83, $p \leq 0.001$) with a large effect size (η^2=0.65) and on InfoQual ($F_{3,36}$=8.43, $p \leq 0.00022$, η^2=0.41), but not on InterQual ($F_{3,24}$=2.07, p=0.13, *n.s.*).

Kruskal-Wallis non-parametric tests [41] and Nemenyi pairwise comparisons showed a statistically significant difference between SysUse (H=40.06, df=3, $p \leq .001^{***}$) of the various conditions of Device Type, highlighting the *tabletop* as the best device; see Fig. 20 for details. For example, the *tabletop* was ranked significantly better than the *pen display* (R=39.03, q=8.38, $p \leq .001$), the *tablet* (R=16.59, q=3.56, p=.0018), and the *smartphone* (R=30.12, q=6.47, $p \leq .001$). Also, the *tablet* was perceived as significantly more useful than the *pen display*. This result is probably due to the oblique position of the *pen display* that induces hand fatigue, unlike the *tablet* that is flat on the table. We also found that the *pen display* was outperformed by the *tabletop* (R=23.92, q=4.39, p=.011) and the *tablet* (R=19.21, q=4.08, p=.020) on the InfoQual measure. With regard to InterQual, the *tabletop* significantly outperformed the *pen display* (R=14.25, q=3.96, p=.026). The same results were also found for the Overall metric.

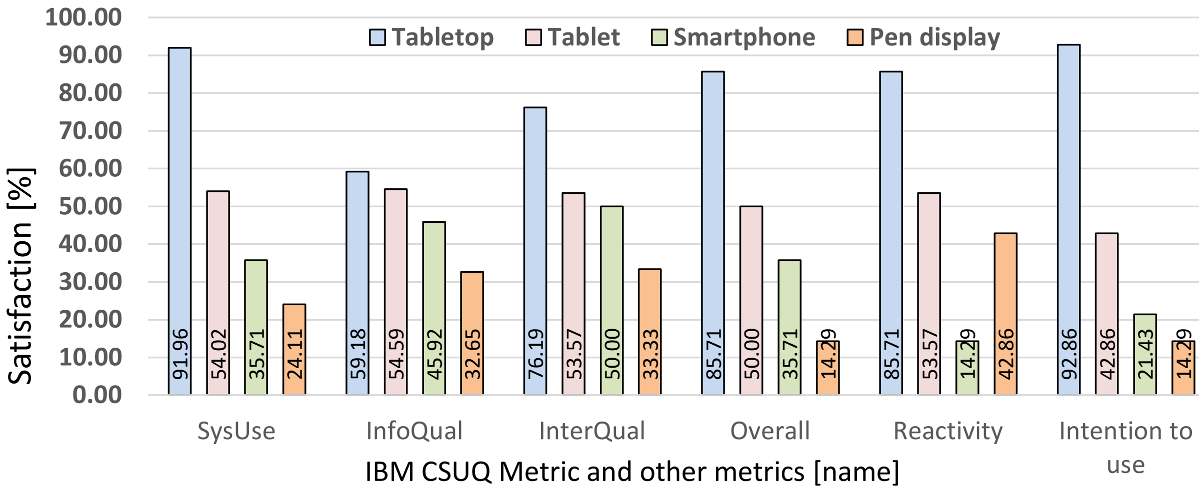

Fig. 21. Perceived satisfaction of using UbiSketch on various platforms during our experiment.

Fig. 21 also shows the Perceived Satisfaction for the CSUQ metrics per Device Type. Participants expressed their satisfaction in the same order for all CSUQ metrics: tabletop, tablet, smartphone, and pen display. For example, the SysUse metric received a percentage of perceived satisfaction of 91.96% for the tabletop, 54.02% for the tablet, 35.71% for the smartphone, and 24.11% for the pen display. Although the order of the first three platforms was predictable, the pen display was less satisfying mainly due to its oblique position forcing the stakeholder to sketch with the hand raised, tiring for a long period. The percentage difference between the most preferred platform, *i.e.,* the tabletop, and the least preferred platform, *i.e.,* the pen display, was minimal for the InfoQual metric (Δ=44.83%) and maximal for the Overall metric (Δ=83.33%), suggesting that the variation in satisfaction was perceived as less different between the conditions of Device Type for InfoQual.

5.2.3 Reactivity. We present average ratings of Perceived satisfaction per Device Type for each CSUQ metric (Fig. 20) and for perceived Satisfaction (Fig. 21), respectively. We found that the *tabletop* was perceived by our participants as the most reactive device, followed by the *tablet*, the *pen display*, and the *smartphone*, without statistically significant differences (H=3.35, df=3, p=.29, *n.s.*). For the first time, the tablet was ranked second after the tabletop as it was equipped with a faster CPU.

5.2.4 Intention to Use. An interesting result was obtained with the Intention to Use measure, showing that the *tabletop* was the most preferred platform that the participants expressed intentions to use again in the future for UI sketching. In general, the scores collected for the smartphone and the pen display suggest that these device types should be discarded. These results are complemented by the perceived Satisfaction to use UbiSketch for paired sketching. We found that *tabletop* (Fig. 1) was the most preferred device in terms of system usefulness (92% of the participants agreed on this measure), information quality (59%), interface quality (76%), overall usability (85%), reactivity (85%), and intention to use in the future (93%). In addition, the participants felt the most comfortable with the *tabletop* due to its large, horizontal, and stable surface, mimicking a conventional desktop working scenario. Even if the participants did not use the entire surface of the *tabletop*, its large display provided the freedom to do so. The *pen display* scored the lowest for most measures except responsiveness, mainly due to CPU performance. The *tablet* represents the second most preferred device for paired DUI sketching with a

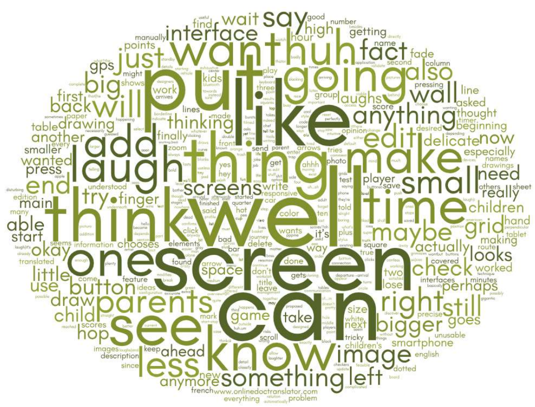

Fig. 22. Smartphone.

Fig. 23. Tablet.

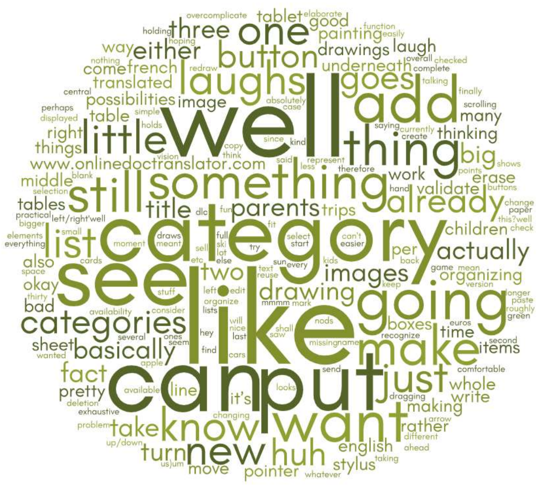

Fig. 24. Pen display.

Fig. 25. Tabletop.

Fig. 26. Word clouds per Device Type.

similar rating across the evaluated measures (SysUse=54%, InfoQual=55%, InterQual=54%, Overall=50%, Reactivity=54%, and Intention to Use=43%).

5.2.5 Conversations. We acknowledge that good communication during the paired sketching process is a learned activity that involves a variety of skills including listening, speaking, monitoring, understanding, and clarifying meaning. Paired sketching highlights what it is like to co-create by communicating [26, 59]: both stakeholders influence the sketching, but cannot control it entirely. Therefore, we also analyzed participants' conversations from their transcripts by using *Word Clouds*, a special visualization of text in which the more frequently used words are highlighted more prominently. [50] describe an experience using word clouds to inform qualitative research, where this kind of visualization allows researchers to grasp the common themes in the text. Based on the audio records of the participants' conversations, we created a transcript for each Device Type: the tabletop

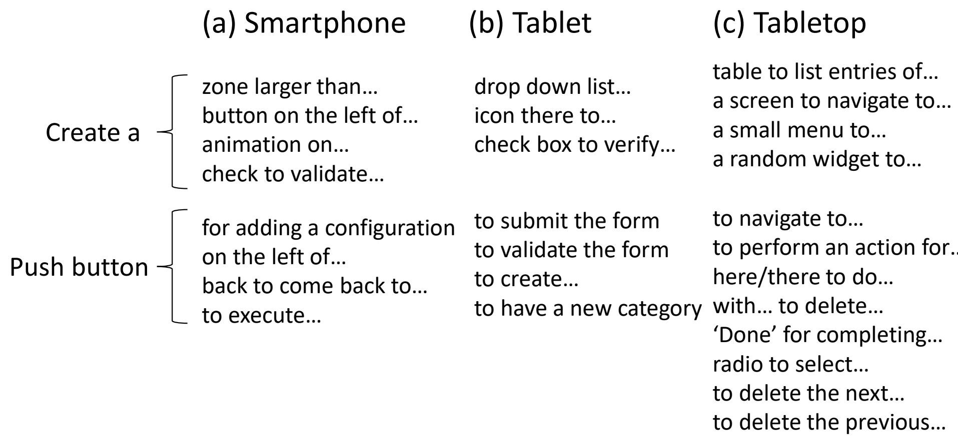

Fig. 27. Examples of topic modeling per Device Type.

(24,985 utterances) and the tablet (25,328 utterances) received the most extensive conversations, followed by the pen display (16,808 utterances) and the smartphone (13,255 utterances). Once again, the larger the device, the more extensive the conversation was. Based on the transcripts, we created four word clouds containing the 50 most frequently used terms. Even though a more profound analysis is needed to grasp the context of words used in different utterances, we can compare the overall differences in participants' conversations (Fig. 26): 557 unique words were produced for the tabletop, 492 for the smartphone, 418 for the tablet, and 210 for the pen display.

The most frequent occurrences were represented by the words "Yes", "Yeah" (*e.g.,* 93 occurrences of "yes" and 51 occurrences of "yeah" for the smartphone *vs.* 30 occurrences of "No"), "Well", "Okay", "Ok", all expressing some form of consensus between the stakeholders, a key aspect in their conversations. Other words expressed hesitations, such as "Uh", "Ah", "Oh". After removing these terms from the word clouds (Fig. 26), "do/make" were the most frequent since participants discussed how the users (represented by imaginary terms like "user", "parent", "kids", "driver") would engage with the interface. The participants used the tablet to sketch the mobile UIs, perhaps explaining why the words "here" and "there" have a high occurrence rate. Subjects also complained about the smartphone's small screen size; this is why the word "screen" received a higher occurrence in the smartphone word cloud. Action terms, such as "think", "place", "put", "press", "say", were also used frequently. The tablet and the tabletop received terms referring to user action, such as "select", "erase", "write", and "display". These results suggest that term occurrence rates match the screen size of the involved devices: participants were more expressive when using larger devices than small ones. Topic modeling [8] revealed the most common phrase structures; see Fig. 27.

6 Implications of Using Paired Sketching

6.1 Takeaway Findings

Our experiment yielded several key findings that inform future research and development for paired sketching:

Supported naturalness: The participants were looking for a large enough workspace for comfort purposes and supporting creativity. Considering our research question, we conclude that the *tabletop* was favored by our participants, particularly in terms of perceived usefulness and intention for future use. Translating to an in-vehicle context, larger input surfaces (*e.g.,* tabletops, dashboard touchscreens, or fold-out tablets) facilitate natural and intuitive sketching, providing a clear overview of the final result. When inside the vehicle, using

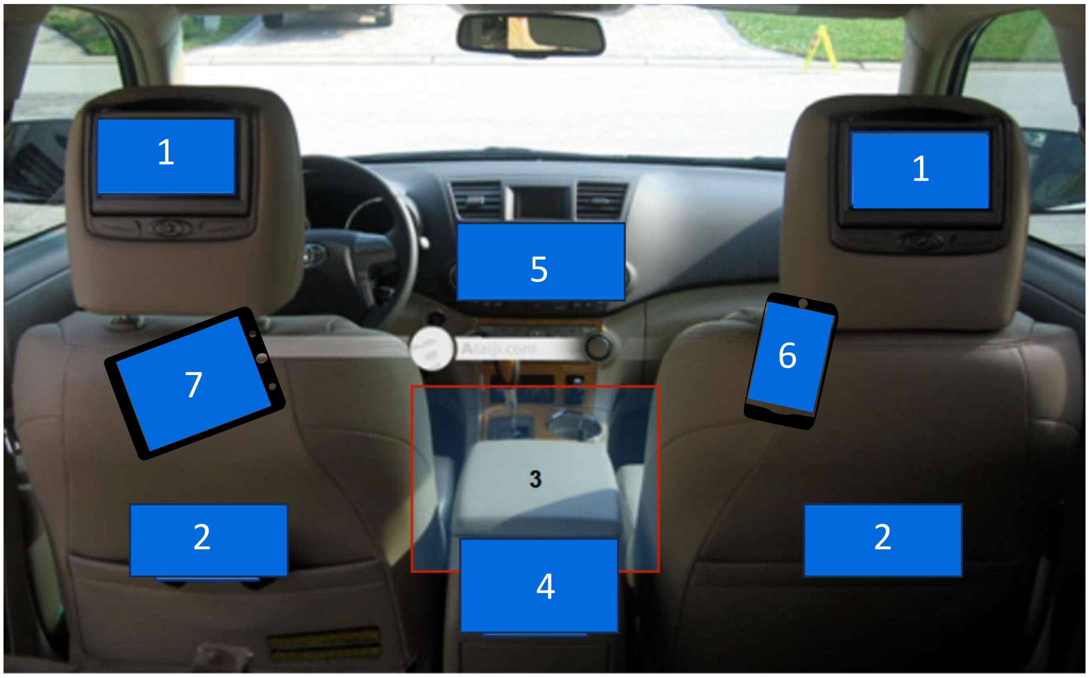

Fig. 28. Interactive surfaces and potential installation for an inside-the-vehicle V2DUI.

DUIs enables drivers and passengers to collaborate on the sketched variations of dashboard layouts or control interfaces that meet their needs and preferences; see Fig. 28 for an overview of the interactive surface distribution: driver's main display (5), the rear passengers' head-up displays (1), the rear passengers' personal displays (2) and their shared display (4) or transparent area (1), their mobile devices such as smartphones (6) or tablets (7), without counting the copilot's display(s) for infotainment.

Increased efficiency: Although all sessions were limited to ten minutes, the collaborative nature of paired sketching accelerated the design process by reducing the time required to complete the task. We observed that the participants produced more UI fragments on larger platforms than on smaller ones. In an in-vehicle environment, using distributed interfaces enables efficiency because both the driver and passengers have different needs and responsibilities, therefore sketching can be designed to accommodate specific requirements on each interactive surface. The driver benefits from dashboard displays for safe access to functions without distraction, while passengers can use the infotainment system, rear-seat entertainment interactive surfaces, and smartphones to access media, adjust their environment properties, and assist the driver with various tasks.

Enhanced creativity: Paired sketching can led to more innovative designs by combining diverse perspectives and expertise, particularly when one participant plays the role of the designer and the other of the user representative. In the context of in-vehicle interaction, passengers and drivers can engage in creative collaboration to design interactive features. For example, by sketching features of the navigation system, the driver can gain insights into what features are important for maintaining the eyes on the road. Passengers can contribute additional features like detailed route previews and points of interest, sketched in a way that does not distract the driver.

Explicit consistency: The iterative refinement process ensured greater consistency between different DUI fragments. Sketches were shared in real time (Fig. 17), allowing each group of participants to monitor what the others were doing and draw inspiration from it. This process ensured that the final design was user-friendly and optimized, improving the overall user experience and prioritizing safety. Thus, passengers could easily navigate menus or select content without needing the assistance of the driver.

Improved focus: As pairs of participants talked more often and designated UI elements more explicitly, the focus on the sketched UI was maintained. Paired sketching is more likely to lead to better UI designs due to the common decision making it involves. Moreover, designers will display a higher confidence in their decisions as they are shared and taken in collaboration.

Provide extensive and explicit feedback: When a stakeholder sketches something, unless decided otherwise, the other stakeholder can provide immediate feedback to reinforce the current design or suggestions to improve it. Since stakeholders may have different levels of design expertise and experience, they can more effectively share their design knowledge and ensure knowledge transfer from the highest experienced stakeholder to the others.

6.2 Paired Sketching *vs.* Pair Design

Paired sketching, as described in this paper, may appear similar to pair design or pair drawing, but presents distinct advantages for UIs, as follows:

- *Focus on UI concepts over UI detail.* Paired sketching emphasizes UI design ideas, concepts, structure, and interaction flow, enabling rapid iteration and brainstorming. At the same time, it often focuses on UI aesthetics, precision and artistic UI fragments, which slows down conceptual design.
- *Faster iteration and ideation.* Paired sketching is quick and informal, enabling stakeholders to explore multiple UI concepts in a short amount of time. In contrast, pair drawing requires more precision, which can hinder fast ideation.
- *Encourages parallel collaboration among stakeholders.* Paired sketching prioritizes communication and reaching consensus as stakeholders discuss and refine ideas. This is particularly suitable for DUIs since UI fragments are sketched in parallel by multiple pairs of stakeholders. On the other hand, pair drawing often involves one person drawing while the other observes, which can reduce active collaboration.
- *Reduced perfectionism.* Paired sketching is meant to be rough yet evolving, encouraging stakeholders to focus on the bigger picture. In contrast, pair drawing emphasizes polished UI fragments, which may lead to over-focusing on details too early.
- *Facilitates user experience workflows.* Paired sketching aligns with wireframing, prototyping, and user flow mapping, all crucial for UI design, in particular for abstract UIs [76] and concrete UIs [79]. In contrast, pair drawing may be more useful for final assets, illustrations, or branding, such as for the final UI rather than layout ideation.
- *Supports agile approaches.* Paired sketching can be used in design sprints in the case of agile development, rapid prototyping, and lean user experience processes. However, pair drawing is more suitable for final-stage visuals, such as high fidelity UI, which may not fit agile workflows, an important metric for Dev-X.

6.3 Limitations

While the experiment was conducted in the application area of V2DUIs [6], we believe that it can be reproduced for other areas and contexts of use, such as DUIs for smart homes, *e.g.,* sketch a UI distributed in a smart home system by involving multiple devices that are spatially distributed involving novel devices such as rings [27] or radars [69] or a cross-device productivity suite for modeling [48]. For such applications, paired sketching offers two main advantages in the context of developer experience [24, 29, 60]:

- A shared perception and understanding of the sketching process, *i.e.,* participants working in pairs share the design goals and their solutions, which is not the case in individual sketching.

- A multidisciplinary perspective, *i.e.,* the collaborative nature of paired sketching incorporates diverse viewpoints, improving creativity and robustness of design.

However, despite its advantages, paired sketching also presents several challenges:

- *Coordination and communication overheads*: coordinating schedules and communication between stakeholders can be challenging when spatially and temporally distributed. For synchronous setups, UbiSketch incorporates a video conferencing system (Fig. 1).
- *Conflict resolution*: UbiSketch does not incorporate any mechanism to help conflict resolution, therefore requiring effective conflict resolution strategies outside its setup.

We did not test how UbiSketch would perform when each device is manipulated by more than two users, such as involving groups of three or four stakeholders. We expect the tabletop to still be preferred due to its larger size, but the coordination problem would become more acute. The evaluation of UbiSketch conducted in the context of V2DUIs (Section 5) is also prone to some threats to validity:

- *Threats to external validity*: The convenience sampling used in the evaluation possesses intrinsic biases, such as a potential lack of variety or representativeness of the actual population. However, we provide information on how the participants used and perceived paired sketching supported by UbiSketch. Future experiments should confirm these insights with more diverse populations of practitioners and stakeholders. Secondly, the findings of our evaluation, which focused on an in-vehicle infotainment system, may not generalize directly to other application domains, including other use cases in automotive UIs [61].
- *Threats to internal validity*: The participants were all experienced in UI design, but in different degrees. Moreover, their ability to express themselves through sketching, even if it did not require drawing skills, was atypical. We addressed this limitation by informing participants that they were not expected to produce a perfect prototype; instead, our focus was on understanding their impressions of the system.
- *Threats to construct validity*: Our experiments followed a within-subject design but could be replicated in the future using a between-subject approach, in which pairs of stakeholders interact with only one device or platform rather than being exposed to multiple devices.

7 Conclusion and Future Work

Paired sketching is suitable for prototyping DUIs because it fosters real-time collaboration and rapid ideation, both of which are critical in addressing the complexity and coordination challenges inherent in distributed systems. By working in pairs composed of developers, designers, end users, or their representatives, both the user experience and the developer experience can be simultaneously considered. This aspect should ensure that the DUI elements align with the distributed nature of the application, such as latency, consistency, and data synchronization concerns. For example, when an end user sketches a DUI element, the developer can immediately respond with feedback on its feasibility and development cost. Conversely, when a developer sketches a DUI element, the end user can immediately check whether it is aligned with the task requirements and their preferences. In case of convergence, the process can proceed to the next elements, while in case of divergence, a consensus procedure can be engaged. This collaborative approach encourages diverse viewpoints, immediate feedback, and quick iteration, helping to identify and resolve potential usability or technical issues early in the design process. Moreover, pair sketching promotes shared understanding and consensus among team members, crucial when designing DUIs that span multiple users, tasks, devices, platforms, and environments.

The key finding of our work is that most measures, such as the statements for developer experience, intention to use, reactivity, and expressive conversation, match the screen size and resolution of the involved devices: the larger the screen size, the better the results. Although our participants took into account the constraints imposed by each platform in their DUI sketches, they expressed a preference for larger screens, which they found to better support creativity. Future research could explore ways to optimize the design process, address the challenges we identified, and expand application opportunities to other areas. In addition, UbiSketch was developed to allow stakeholders to participate in paired sketching for UI design in general and DUI design in particular. The main interaction modality is still graphical, but as gesture-based interaction has started to replace conventional modalities, UI sketching could be extended to support gesture-based input as well.

Acknowledgments

Mehdi Ousmer and Jean Vanderdonckt are supported by the EU EIC Pathfinder-Awareness Inside challenge "Symbiotik" project (1 Oct. 2022-30 Sep. 2026) under Grant no. 101071147. Laura-Bianca Bilius and Radu-Daniel Vatavu acknowledge support from the NetZeRoCities Competence Center funded by the European Union-NextGenerationEU and the Romanian Government under the National Recovery and Resilience Plan for Romania, contract no. 760007/30.12.2022 with the Romanian Ministry of Research, Innovation, and Digitalization through the specific research projects P3, Smart and sustainable buildings and P4, Smart mobility and infrastructure. We are very grateful to Ugo Sangiorgi for his contributions to this work.

References

[1] Gretchen Anderson and Christopher Noessel. 2016. *Pair Design.* O'Reilly Media, Inc. https://www.oreilly.com/library/view/pair-design/9781492042907/

[2] Abdo Beirekdar, Marc Keita, Monique Noirhomme, Frédéric Randolet, Jean Vanderdonckt, and Céline Mariage. 2005. Flexible Reporting for Automated Usability and Accessibility Evaluation of Web Sites. In *Human-Computer Interaction - INTERACT 2005*, Maria Francesca Costabile and Fabio Paternò (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 281–294. https://doi.org/10.1007/11555261_25

[3] Laura-Bianca Bilius and Radu-Daniel Vatavu. 2020. A Synopsis of Input Modalities for In-Vehicle Infotainment and Consumption of Interactive Media. In *Proceedings of the ACM International Conference on Interactive Media Experiences (IMX '20).* ACM, New York, NY, USA, 195–199. https://doi.org/10.1145/3391614.3399400

[4] Laura-Bianca Bilius and Radu-Daniel Vatavu. 2021. A Multistudy Investigation of Drivers and Passengers' Gesture and Voice Input Preferences for In-Vehicle Interactions. *Journal of Intelligent Transportation Systems* 25 (2021), 197–220. https://doi.org/10.1080/15472450.2020.1846127

[5] Laura-Bianca Bilius, Radu-Daniel Vatavu, and Nicolai Marquardt. 2021. Smart Vehicle Proxemics: A Conceptual Framework Operationalizing Proxemics in the Context of Outside-the-Vehicle Interactions. In *Human-Computer Interaction (INTERACT '21)*, Carmelo Ardito, Rosa Lanzilotti, Alessio Malizia, Helen Petrie, Antonio Piccinno, Giuseppe Desolda, and Kori Inkpen (Eds.). Springer International Publishing, Cham, 150–171. http://dx.doi.org/10.1007/978-3-030-85616-8_11

[6] Laura-Bianca Bilius, Radu-Daniel Vatavu, and Jean Vanderdonckt. 2024. Expanding V2X with V2DUIs: Distributed User Interfaces for Media Consumption in the Vehicle-to-Everything Era. In *Proceedings of the ACM International Conference on Interactive Media Experiences* (Stockholm, Sweden) *(IMX '24).* Association for Computing Machinery, New York, NY, USA, 394–401. https://doi.org/10.1145/3639701.3663643

[7] Yashwant Singh Bisht, Rajesh Singh, Anita Gehlot, Shaik Vaseem Akram, Amit Kumar Thakur, Neeraj Priyadarshi, and Bhekisipho Twala. 2024. Recent Trends and Technologies in rapid prototyping and its Inclination towards Industry 4.0. *Sustainable Engineering and Innovation* 6, 1 (Jun. 2024), 141–154. https://doi.org/10.37868/sei.v6i1.id288

[8] David M. Blei. 2012. Probabilistic topic models. *Commun. ACM* 55, 4 (apr 2012), 77–84. https://doi.org/10.1145/2133806.2133826

[9] Jared N. Bott and Joseph J. Laviola Jr. 2015. The WOZ Recognizer: A Wizard of Oz Sketch Recognition System. *ACM Trans. Interact. Intell. Syst.* 5, 3, Article 15 (oct 2015), 38 pages. https://doi.org/10.1145/2743029

[10] Paul Brie, Nicolas Burny, Arthur Sluÿters, and Jean Vanderdonckt. 2023. Evaluating a Large Language Model on Searching for GUI Layouts. *Proc. ACM Hum.-Comput. Interact.* 7, EICS, Article 178 (jun 2023), 37 pages. https://doi.org/10.1145/3593230

[11] Bill Buxton. 2007. *Sketching User Experiences: Getting the Design Right and the Right Design.* Morgan Kaufmann. https://www.amazon.com/Sketching-User-Experiences-Interactive-Technologies/dp/0123740371

[12] Anabela Caetano, Neri Goulart, Manuel Fonseca, and Joaquim Jorge. 2002. Sketching User Interfaces with Visual Patterns. In *SIACG2002 - 1st lbero-American Symposium in Computer Graphics*, Xavier Pueyo, Manuel Próspero dos Santos, and Luiz Velho (Eds.). The Eurographics Association. https://doi.org/10.2312/pt.20021420

[13] Tommaso Calò and Luigi De Russis. 2022. Style-Aware Sketch-to-Code Conversion for the Web. In *Companion of the Proceedings of ACM SIGCHI Symposium on Engineering Interactive Computing Systems* (Sophia Antipolis, France) *(EICS '22 Companion).* Association for Computing Machinery, New York, NY, USA, 44–47. https://doi.org/10.1145/3531706.3536462

[14] Cheng-Min Chao. 2019. Factors Determining the Behavioral Intention to Use Mobile Learning: An Application and Extension of the UTAUT Model. *Frontiers in Psychology* 10 (2019). https://doi.org/10.3389/fpsyg.2019.01652

[15] Rui Chen, Po-Jui (Ray) Chen, Rui Feng, Yilin (Elaine) Liu, Andy Wu, and Ali Mazalek. 2014. SciSketch: a tabletop collaborative sketching system. In *Proceedings of the 8th International Conference on Tangible, Embedded and Embodied Interaction* (Munich, Germany) *(TEI '14)*. Association for Computing Machinery, New York, NY, USA, 247–250. https://doi.org/10.1145/2540930.2540973

[16] Ashley Colley, Jonna Häkkilä, Bastian Pfleging, and Florian Alt. 2017. A Design Space for External Displays on Cars. In *Proceedings of the 9th International Conference on Automotive User Interfaces and Interactive Vehicular Applications Adjunct (AutomotiveUI '17)*. ACM, New York, NY, USA, 146–151. https://doi.org/10.1145/3131726.3131760

[17] Adrien Coyette, Suzanne Kieffer, and Jean Vanderdonckt. 2007. Multi-fidelity Prototyping of User Interfaces. In *Proceedings of 11th IFIP TC 13 International Conference on Human-Computer Interaction, INTERACT '20'07, Rio de Janeiro, Brazil, September 10-14, 2007 (Lecture Notes in Computer Science, Vol. 4662)*, Maria Cecília Calani Baranauskas, Philippe A. Palanque, Julio Abascal, and Simone Diniz Junqueira Barbosa (Eds.). Springer, 150–164. https://doi.org/10.1007/978-3-540-74796-3_16

[18] Adrien Coyette, Sascha Schimke, Jean Vanderdonckt, and Claus Vielhauer. 2007. Trainable Sketch Recognizer for Graphical User Interface Design. In *Proceedings of 11th IFIP TC 13 International Conference on Human-Computer Interaction, INTERACT '07, Rio de Janeiro, Brazil, September 10-14, 2007 (Lecture Notes in Computer Science, Vol. 4662)*, Maria Cecília Calani Baranauskas, Philippe A. Palanque, Julio Abascal, and Simone Diniz Junqueira Barbosa (Eds.). Springer, 124–135. https://doi.org/10.1007/978-3-540-74796-3_14

[19] Brock Craft and Paul Cairns. 2009. Sketching sketching: outlines of a collaborative design method. In *Proceedings of the 23rd British HCI Group Annual Conference on People and Computers: Celebrating People and Technology* (Cambridge, United Kingdom) *(BCS-HCI '09)*. BCS Learning & Development Ltd., Swindon, GBR, 65–72.

[20] Fred D. Davis, Richard P. Bagozzi, and Paul R. Warshaw. 1989. User Acceptance of Computer Technology: A Comparison of Two Theoretical Models. *Management Science* 35, 8 (1989), 982–1003. https://doi.org/10.1287/mnsc.35.8.982 arXiv:https://doi.org/10.1287/mnsc.35.8.982

[21] Alexandre Demeure, Jean-Sébastien Sottet, Gaëlle Calvary, Joëlle Coutaz, Vincent Ganneau, and Jean Vanderdonckt. 2008. The 4C Reference Model for Distributed User Interfaces. In *Proceedings of the 4th Int. Conference on Autonomic and Autonomous Systems*. IEEE, USA, 61–69. http://dx.doi.org/10.1109/ICAS.2008.34

[22] Henrik Detjen, Sarah Faltaous, Bastian Pfleging, Stefan Geisler, and Stefan Schneegass. 2021. How to Increase Automated Vehicles' Acceptance through In-Vehicle Interaction Design: A Review. *International Journal of Human–Computer Interaction* 37, 4 (2021), 308–330. https://doi.org/10.1080/10447318.2020.1860517

[23] Niklas Elmqvist. 2011. Distributed User Interfaces: State of the Art. In *Distributed User Interfaces: Designing Interfaces for the Distributed Ecosystem*, José A. Gallud, Ricardo Tesoriero, and Victor M.R. Penichet (Eds.). Human-Computer Interaction Series. Springer, London, 1–12. https://doi.org/10.1007/978-1-4471-2271-5_1

[24] Fabian Fagerholm and Jürgen Münch. 2012. Developer experience: Concept and definition. In *Proceedings of the International Conference on Software and System Process (ICSSP '12')*. 73–77. https://doi.org/10.1109/ICSSP.2012.6225984

[25] Nishit Gajjar, Vinoth Pandian Sermuga Pandian, Sarah Suleri, and Matthias Jarke. 2021. Akin: Generating UI Wireframes From UI Design Patterns Using Deep Learning. In *Companion Proceedings of the 26th International Conference on Intelligent User Interfaces* (College Station, TX, USA) *(IUI '21 Companion)*. Association for Computing Machinery, New York, NY, USA, 40–42. https://doi.org/10.1145/3397482.3450727

[26] Florian Geyer, Jochen Budzinski, and Harald Reiterer. 2012. IdeaVis: a hybrid workspace and interactive visualization for paper-based collaborative sketching sessions. In *Proceedings of the 7th Nordic Conference on Human-Computer Interaction: Making Sense Through Design* (Copenhagen, Denmark) *(NordiCHI '12)*. Association for Computing Machinery, New York, NY, USA, 331–340. https://doi.org/10.1145/2399016.2399069

[27] Bogdan-Florin Gheran, Radu-Daniel Vatavu, and Jean Vanderdonckt. 2023. New Insights into User-Defined Smart Ring Gestures with Implications for Gesture Elicitation Studies. In *Extended Abstracts of the ACM Conference on Human Factors in Computing Systems* (Hamburg, Germany) *(CHI EA '23)*. Association for Computing Machinery, New York, NY, USA, Article 216, 8 pages. https://doi.org/10.1145/3544549.3585590

[28] Saul Greenberg, Sheelagh Carpendale, Nicolai Marquardt, and Bill Buxton. 2012. *Sketching User Experiences - The Workbook*. Academic Press. http://store.elsevier.com/product.jsp?isbn=9780123819598

[29] Michaela Greiler, Margaret-Anne Storey, and Abi Noda. 2023. An Actionable Framework for Understanding and Improving Developer Experience. *IEEE Trans. Softw. Eng.* 49, 4 (April 2023), 1411–1425. https://doi.org/10.1109/TSE.2022.3175660

[30] Object Management Group. 2018. Software & Systems Process Engineering Metamodel (SPEM V2.0). https://www.omg.org/spec/SPEM/2.0/About-SPEM

[31] Louis Guttman. 1944. A Basis for Scaling Qualitative Data. *American Sociological Review* 9, 2 (1944), 139–150. https://doi.org/10.2307/2086306

[32] Nate Hagbi, Raphael Grasset, Oriel Bergig, Mark Billinghurst, and Jihad El-Sana. 2015. In-Place Sketching for Augmented Reality Games. *Comput. Entertain.* 12, 3, Article 3 (feb 2015), 18 pages. https://doi.org/10.1145/2702109.2633419

[33] Matthias Heintz, Effie Law, and Pamela Andrade. 2018. Comparison of paper- and tool-based participatory design approaches: a case study with PDotCapturer. In *Proceedings of the 32nd International BCS Human Computer Interaction Conference* (Belfast, United Kingdom) *(HCI '18)*. BCS Learning & Development Ltd., Swindon, GBR, Article 29, 12 pages. https://doi.org/10.14236/ewic/HCI2018.29

[34] Marc Hesenius, Markus Kleffmann, and Volker Gruhn. 2021. AugIR Meets GestureCards: A Digital Sketching Environment for Gesture-Based Applications. *Interacting with Computers* 33, 2 (05 2021), 134–154. https://doi.org/10.1093/iwcomp/iwab017 arXiv:https://academic.oup.com/iwc/article-pdf/33/2/134/40358790/iwab017.pdf

[35] Ramon Linus Hofer and Andreas Kunz. 2010. Digisketch: taming anoto technology on LCDs. In *Proceedings of the 2nd ACM SIGCHI Symposium on Engineering Interactive Computing Systems* (Berlin, Germany) *(EICS '10)*. Association for Computing Machinery, New York, NY, USA, 103–108. https://doi.org/10.1145/1822018.1822034

[36] Forrest Huang, John F. Canny, and Jeffrey Nichols. 2019. Swire: Sketch-based User Interface Retrieval. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–10. https://doi.org/10.1145/3290605.3300334

[37] Dagmar Kern and Albrecht Schmidt. 2009. Design Space for Driver-Based Automotive User Interfaces. In *Proceedings of the 1st International Conference on Automotive User Interfaces and Interactive Vehicular Applications (AutomotiveUI '09)*. ACM, New York, NY, USA, 3–10. https://doi.org/10.1145/1620509.1620511

[38] Suzanne Kieffer, Adrien Coyette, and Jean Vanderdonckt. 2010. User interface design by sketching: a complexity analysis of widget representations. In *Proceedings of the 2nd ACM SIGCHI Symposium on Engineering Interactive Computing Systems* (Berlin, Germany) *(EICS '10)*. Association for Computing Machinery, New York, NY, USA, 57–66. https://doi.org/10.1145/1822018.1822029

[39] Suzanne Kieffer and Jean Vanderdonckt. 2023. A Comparison of Paper Sketch and Interactive Wireframe by Eye Movements Analysis, Survey, and Interview. In *Proceedings of 56th Hawaii International Conference on System Sciences* (Maui, Hawaii, USA) *(HICSS 2023)*, Tung X. Bui (Ed.). ScholarSpace, 4067–4076. https://hdl.handle.net/10125/103128

[40] Markus Kleffmann, Marc Hesenius, and Volker Gruhn. 2015. Connecting UI and business processes in a collaborative sketching environment. In *Proceedings of the 7th ACM SIGCHI Symposium on Engineering Interactive Computing Systems* (Duisburg, Germany) *(EICS '15)*. Association for Computing Machinery, New York, NY, USA, 200–205. https://doi.org/10.1145/2774225.2775076

[41] William H. Kruskal and W. Allen Wallis. 1952. Use of Ranks in One-Criterion Variance Analysis. *J. Amer. Statist. Assoc.* 47, 260 (1952), 583–621. https://doi.org/10.1080/01621459.1952.10483441

[42] J.A. Landay and B.A. Myers. 2001. Sketching interfaces: toward more human interface design. *Computer* 34, 3 (2001), 56–64. https://doi.org/10.1109/2.910894

[43] Clayton Lewis, Peter G. Polson, Cathleen Wharton, and John Rieman. 1990. Testing a walkthrough methodology for theory-based design of walk-up-and-use interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Seattle, Washington, USA) *(CHI '90)*. Association for Computing Machinery, New York, NY, USA, 235–242. https://doi.org/10.1145/97243.97279

[44] James R. Lewis. 2002. Psychometric Evaluation of the PSSUQ Using Data from Five Years of Usability Studies. *International Journal of Human–Computer Interaction* 14, 3-4 (2002), 463–488. https://doi.org/10.1080/10447318.2002.9669130

[45] Xudong Liang and Tao Lin. 2023. Sketch2Wireframe: an automatic framework for transforming hand-drawn sketches to digital wireframes in UI design. *The Visual Computer* 47, 260 (2023), 583–621. https://doi.org/10.1007/s00371-023-03188-z

[46] Lassi A. Liikkanen and Paula Gómez Gómez. 2013. Designing interactive systems for the experience of time. In *Proceedings of the 6th International Conference on Designing Pleasurable Products and Interfaces* (Newcastle upon Tyne, United Kingdom) *(DPPI '13)*. Association for Computing Machinery, New York, NY, USA, 146–155. https://doi.org/10.1145/2513506.2513522

[47] 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

[48] Nicolas Mangano, Alex Baker, and André van der Hoek. 2008. Calico: a prototype sketching tool for modeling in early design. In *Proceedings of the 2008 International Workshop on Models in Software Engineering* (Leipzig, Germany) *(MiSE '08)*. Association for Computing Machinery, New York, NY, USA, 63–68. https://doi.org/10.1145/1370731.1370747

[49] Michael McCurdy, Christopher Connors, Guy Pyrzak, Bob Kanefsky, and Alonso Vera. 2006. Breaking the fidelity barrier: an examination of our current characterization of prototypes and an example of a mixed-fidelity success. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Montréal, Québec, Canada) *(CHI '06)*. Association for Computing Machinery, New York, NY, USA, 1233–1242. https://doi.org/10.1145/1124772.1124959

[50] Carmel McNaught and Paul Lam. 2014. Using Wordle as a Supplementary Research Tool. *The Qualitative Report* 15, 3 (2014), 630–643. https://doi.org/10.46743/2160-3715/2010.1167

[51] 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 Systems (EICS '11)*. ACM, New York, NY, USA, 11–20. https://doi.org/10.1145/1996461.1996488

[52] Matei Negulescu and Tetsunari Inamura. 2011. Exploring sketching for robot collaboration. In *Proceedings of the 6th International Conference on Human-Robot Interaction* (Lausanne, Switzerland) *(HRI '11)*. Association for Computing Machinery, New York, NY, USA, 211–212. https://doi.org/10.1145/1957656.1957736

[53] Jakob Nielsen. 1994. *Response Times: The Three Important Limits*. Morgan Kaufmann, San Francisco, CA, USA, Chapter 5.

[54] Victor M. R. Penichet, Maria Dolores Lozano, Jose A. Gallud, and Ricardo Tesoriero. 2014. 12 + 1 Questions in the Design of Distributed User Interfaces. In *Proceedings of the 2014 Workshop on Distributed User Interfaces and Multimodal Interaction* (Toulouse, France) *(DUI*

'14). Association for Computing Machinery, New York, NY, USA, 37–41. https://doi.org/10.1145/2677356.2677664

[55] Jorge-Luis Pérez-Medina and Karina-Beatriz Jimenes-Vargas. 2020. Recognizing a sketched task model by multi-stroke gesture recognition. In *Companion Proceedings of the 12th ACM SIGCHI Symposium on Engineering Interactive Computing Systems* (Sophia Antipolis, France) *(EICS '20 Companion)*. Association for Computing Machinery, New York, NY, USA, Article 5, 6 pages. https://doi.org/10.1145/3393672.3398494

[56] Fabio Petrillo, André Suslik Spritzer, Carla Dal Sasso Freitas, and Marcelo Pimenta. 2011. Interactive analysis of Likert scale data using a multichart visualization tool. In *Proceedings of the 10th Brazilian Symposium on Human Factors in Computing Systems and the 5th Latin American Conference on Human-Computer Interaction* (Porto de Galinhas, Pernambuco, Brazil) *(IHC+CLIHC '11)*. Brazilian Computer Society, Porto Alegre, BRA, 358–365.

[57] A. Peñalver, J.J. López-Espín, J.A. Gallud, E. Lazcorreta, and F. Botella. 2011. Distributed User Interfaces: Specification of Essential Properties. In *Distributed User Interfaces*, J. Gallud, R. Tesoriero, and V. Penichet (Eds.). Springer, London, 13–21. https://doi.org/10.1007/978-1-4471-2271-5_2

[58] Albert W. Porter. 1977. *The Art of Sketching*. Davis Publications Inc., USA. https://www.wob.com/en-ie/books/albert-w-porter/art-of-sketching/9780871920928/GOR006970913

[59] Daniela Faas Qifang Bao and Maria Yang. 2018. Interplay of sketching & prototyping in early stage product design. *International Journal of Design Creativity and Innovation* 6, 3-4 (2018), 146–168. https://doi.org/10.1080/21650349.2018.1429318 arXiv:https://doi.org/10.1080/21650349.2018.1429318

[60] Abdul Razzaq, Jim Buckley, Qin Lai, Tingting Yu, and Goetz Botterweck. 2024. A Systematic Literature Review on the Influence of Enhanced Developer Experience on Developers' Productivity: Factors, Practices, and Recommendations. *ACM Comput. Surv.* 57, 1, Article 13 (Oct. 2024), 46 pages. https://doi.org/10.1145/3687299

[61] Andreas Riegler, Philipp Wintersberger, Andreas Riener, and Clemens Holzmann. 2018. Investigating User Preferences for Windshield Displays in Automated Vehicles. In *Proceedings of the 7th ACM International Symposium on Pervasive Displays* (Munich, Germany) *(PerDis '18)*. Association for Computing Machinery, New York, NY, USA, Article 8, 7 pages. https://doi.org/10.1145/3205873.3205885

[62] Ugo Sangiorgi and Jean Vanderdonckt. 2012. GAMBIT: Addressing multi-platform collaborative sketching with html5. In *Proceedings of the 4th ACM SIGCHI Symposium on Engineering Interactive Computing Systems* (Copenhagen, Denmark) *(EICS '12)*. Association for Computing Machinery, New York, NY, USA, 257–262. https://doi.org/10.1145/2305484.2305527

[63] Ugo Braga Sangiorgi. 2012. Addressing multi-platform collaborative sketching. In *Proceedings of the ACM SIGCHI Symposium on Engineering Interactive Computing Systems, EICS'12, Copenhagen, Denmark - June 25 - 28, 2012*, Simone Diniz Junqueira Barbosa, José Creissac Campos, Rick Kazman, Philippe A. Palanque, Michael D. Harrison, and Steve Reeves (Eds.). ACM, 309–312. https://doi.org/10.1145/2305484.2305538

[64] Ugo Braga Sangiorgi, François Beuvens, and Jean Vanderdonckt. 2012. User interface design by collaborative sketching. In *Proceedings of the Designing Interactive Systems Conference* (Newcastle Upon Tyne, United Kingdom) *(DIS '12)*. Association for Computing Machinery, New York, NY, USA, 378–387. https://doi.org/10.1145/2317956.2318013

[65] Vinícius C.V.B. Segura and Simone D.J. Barbosa. 2013. UISKEI++: multi-device wizard of oz prototyping. In *Proceedings of the 5th ACM SIGCHI Symposium on Engineering Interactive Computing Systems* (London, United Kingdom) *(EICS '13)*. Association for Computing Machinery, New York, NY, USA, 171–174. https://doi.org/10.1145/2494603.2480337

[66] Vinícius C. V. B. Segura, Simone D. J. Barbosa, and Fabiana Pedreira Simões. 2012. UISKEI: a sketch-based prototyping tool for defining and evaluating user interface behavior. In *Proceedings of the International Working Conference on Advanced Visual Interfaces* (Capri Island, Italy) *(AVI '12)*. Association for Computing Machinery, New York, NY, USA, 18–25. https://doi.org/10.1145/2254556.2254564

[67] Vinoth Pandian Sermuga Pandian, Sarah Suleri, Christian Beecks, and Matthias Jarke. 2021. MetaMorph: AI Assistance to Transform Lo-Fi Sketches to Higher Fidelities. In *Proceedings of the 32nd Australian Conference on Human-Computer Interaction* (Sydney, NSW, Australia) *(OzCHI '20)*. Association for Computing Machinery, New York, NY, USA, 403–412. https://doi.org/10.1145/3441000.3441030

[68] Vinoth Pandian Sermuga Pandian, Sarah Suleri, and Prof. Dr. Matthias Jarke. 2021. UISketch: A Large-Scale Dataset of UI Element Sketches. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) *(CHI '21)*. Association for Computing Machinery, New York, NY, USA, Article 319, 14 pages. https://doi.org/10.1145/3411764.3445784

[69] Alexandru-Ionut Siean, Cristian Pamparau, Arthur Sluÿters, Radu-Daniel Vatavu, and Jean Vanderdonckt. 2023. Flexible gesture input with radars: systematic literature review and taxonomy of radar sensing integration in ambient intelligence environments. *J. Ambient Intell. Humaniz. Comput.* 14, 6 (2023), 7967–7981. https://doi.org/10.1007/s12652-023-04606-9

[70] Carolyn Snyder. 2004. *Paper Prototyping: The Fast and Easy Way to Design and Refine User Interfaces*. Morgan Kaufmann. https://doi.org/10.1016/B978-1-55860-870-2.X5023-2

[71] David Socha and Josh Tenenberg. 2015. Sketching and conceptions of software design. In *Proceedings of the Eighth International Workshop on Cooperative and Human Aspects of Software Engineering* (Florence, Italy) *(CHASE '15)*. IEEE Press, 57–63.

[72] Charles Spearman. 1904. The Proof and Measurement of Association between Two Things. *The American Journal of Psychology* 15, 1 (jan 1904), 72–101. https://doi.org/10.2307/1412159

[73] Sarah Suleri, Vinoth Pandian Sermuga Pandian, Svetlana Shishkovets, and Matthias Jarke. 2019. Eve: A Sketch-based Software Prototyping Workbench. In *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI EA '19)*. Association for Computing Machinery, New York, NY, USA, 1–6. https://doi.org/10.1145/3290607.3312994

[74] Moustafa Tabbarah, Yusheng Cao, Ahmad Abu Shamat, Ziming Fang, Lingyu Li, and Myounghoon Jeon. 2023. Novel In-Vehicle Gesture Interactions: Design and Evaluation of Auditory Displays and Menu Generation Interfaces. In *Proceedings of the 15th International Conference on Automotive User Interfaces and Interactive Vehicular Applications (AutomotiveUI '23)*. ACM, New York, NY, USA, 224–233. https://doi.org/10.1145/3580585.3607164

[75] Mohsen Tavakol and Reg Dennick. 2011. Making sense of Cronbach's alpha. *International Journal of Medical Education* 2 (2011), 53–55. https://doi.org/10.5116/ijme.4dfb.8dfd arXiv:http://www.ijme.net/archive/2/cronbachs-alpha.pdf

[76] Vi Tran, Jean Vanderdonckt, Ricardo Tesoriero, and François Beuvens. 2012. Systematic generation of abstract user interfaces. In *ACM SIGCHI Symposium on Engineering Interactive Computing Systems, EICS'12, Copenhagen, Denmark - June 25 - 28, 2012*, Simone Diniz Junqueira Barbosa, José Creissac Campos, Rick Kazman, Philippe A. Palanque, Michael D. Harrison, and Steve Reeves (Eds.). ACM, 101–110. https://doi.org/10.1145/2305484.2305502

[77] Barbara Tversky. 2002. What do Sketches say about Thinking. In *Proc. of 2002 AAAI Spring Symposium, Sketch Understanding Workshop*. AAAI, 1–4. https://aaai.org/papers/0022-ss02-08-022-what-do-sketches-say-about-thinking/

[78] Jean Vanderdonckt. 2010. Distributed User Interfaces: How to Distribute User Interface Elements across Users, Platforms, and Environments. In *Proc. of the 11th Congreso Internacional de Interacción Persona-Ordenador Interacción*. AIPO, Valencia, 20–32. http://www.usixml.org/en/vanderdonckt-j-distributed-user-interfaces-how-to-distribute-user-interface-elements-across-users-platforms-and-environm.html?IDC=465&IDD=1433

[79] Jean Vanderdonckt and Francisco Montero Simarro. 2010. Generative pattern-based design of user interfaces. In *Proceedings of the 1st International Workshop on Pattern-Driven Engineering of Interactive Computing Systems, PEICS '10, Berlin, Germany, July 20, 2010*, Kai Breiner and Gerrit Meixner (Eds.). ACM, 12–19. https://doi.org/10.1145/1824749.1824753

[80] Chris Vandervelpen and Karin Coninx. 2004. Towards model-based design support for distributed user interfaces. In *Proceedings of the Third Nordic Conference on Human-Computer Interaction* (Tampere, Finland) *(NordiCHI '04)*. Association for Computing Machinery, New York, NY, USA, 61–70. https://doi.org/10.1145/1028014.1028023

[81] Viswanath Venkatesh and Fred D. Davis. 2000. A Theoretical Extension of the Technology Acceptance Model: Four Longitudinal Field Studies. *Management Science* 46, 2 (2000), 186–204. https://doi.org/10.1287/mnsc.46.2.186.11926 arXiv:https://doi.org/10.1287/mnsc.46.2.186.11926

[82] Christoph Wimmer, Alex Untertrifaller, and Thomas Grechenig. 2021. SketchingInterfaces: A Tool for Automatically Generating High-Fidelity User Interface Mockups from Hand-Drawn Sketches. In *Proceedings of the 32nd Australian Conference on Human-Computer Interaction* (Sydney, NSW, Australia) *(OzCHI '20)*. Association for Computing Machinery, New York, NY, USA, 538–545. https://doi.org/10.1145/3441000.3441015

Received 14 February 2025; revised 12 March 2025; revised 27 April 2025; accepted 6 May 2025