CROSSIDE: A Cross-Surface Collaboration by Sketching Design Space

Jorge-Luis Pérez-Medina[1], Jean Vanderdonckt[2] and Santiago Villarreal-Narvaez[2]
[1] Intelligent & Interactive Systems Lab, Universidad de las Américas. Quito, Ecuador
[2] Université catholique de Louvain (UCL). Louvain-la-Neuve, Belgium

Abstract— **This paper introduces, motivates, defines, and exemplifies CROSSIDE, a design space for representing capabilities of a software for collaborative sketching in a cross-surface setting, i.e., when stakeholders are interacting with and across multiple interaction surfaces, ranging from low-end devices such as smartwatches, mobile phones to high-end devices like wall displays. By determining the greatest common denominator in terms of system properties between forty-one references, the design space is structured according to seven dimensions: user configurations, surface configurations, input interaction techniques, work methods, tangibility, and device configurations. This design space is aimed at satisfying three virtues: *descriptive* (i.e., the ability to systematically describe any particular work in cross-surface interaction by sketching), *comparative* (i.e., the ability to consistently compare two or more works belonging to this area), and *generative* (i.e., the ability to generate new ideas by identifying potentially interesting, undercovered areas). A radar diagram graphically depicts the design space for these three virtues.**

I. INTRODUCTION

In many domains of human activity, across them [31] and within [17], *sketching* is largely used as a quick, efficient, and cost-effective tool for expressing ideas, illustrating design concepts and solutions as well as for sharing them with diverse people [49]. During the early stages [25] of product development until its final completion, sketching immediately plays some role when co-design is a must [52]. Sketching facilitates the sharing of abstract ideas and the insights inside a team while offering a common ground for the discussion, especially when the participants come from different cultural and social backgrounds. Sketching usually uses basic drawings to foster everyone's participation [10].

Sketching, as one particular form of drawing, belongs to the first human intellectual skills and abilities that are acquired even before speaking, writing, and precise drawing. By one year, infants understand that a sequence of sounds form a word that symbolically represents an action, a relation, or an object they can point to rather than reaching it. Our human ability to perceive and to recognize elements in a graphical representation is in itself a statement of the expression power yielded by such form of expression.

A *sketch*, coming from the Greek word $\sigma\chi\epsilon\delta\iota o\zeta$ (schedios - *done ex tempore*), consists in a quick provisional freehand drawing executed without any constraint in any medium. A sketch may convey a message, record or develop an abstraction or may be used as a mean for explaining something, for example an image. A sketch is considered as a very quick drawing aimed at communicating a message, which can be understood, misinterpreted, or even ignored. But there is always a message that is different from the one of a *drawing*. For these reasons, within many professions such as industrial and architecture design [52], representing actions, objects, and their relations using different forms of sketches such as drafts, blueprints, and prototypes, has always been very important. They make visible the stakeholder's contribution, which could be different than expected.

The audience of stakeholders involved in sketching activities is now recognized to be wider. Consequently, the pool of software tools to support collaborative sketching, both commercial and from the academia, also becomes more filled. *How much sketching activities could be supported by collaborative tools* is therefore a key question addressed in this paper. Researchers should be informed about capabilities of existing tools to update their research agenda and to better understand similarities and differences. Practitioners should also be informed about which tool would match their requirements for conducting collaborative sketching.

For this purpose, the remainder of this paper is structured as follows: Section 2 reviews the concept of sketching across some disciplines, mainly those involving some form of co-design, and discusses selected works in these disciplines; based on these observations, Section 3 introduces, motivates, and defines CROSSIDE, a design space for representing the key capabilities of software for collaborative sketching, with a special attention paid to cross-surface collaboration; Section 4 exemplifies some case studies with this design space; Section 5 concludes this paper by explaining how this design space could be systematically used to describe, compare, and invent tools for collaborative sketching on multiple surfaces of interaction.

II. RELATED WORK

In Information and Communications Technologies (ICT) as well as in computer and software systems engineering, a significant amount of resources is devoted to designing a concept, a service, a solution which could be later on revealed as inadequate. This could happen when the functional requirements are not satisfied, when the user experience is not met, or when the concept is simply not technologically feasible or too expensive. When such as mismatch is discovered late in the development life cycle or after the deployment, adapting what is required to be changed represents a high cost.

Early sketching in the development life cycle helps creating alternative solutions and comparing them. It helps to ensure that the right approach to design the right solution is put in place. It keeps the design and development right. With a sketch, designers and other stakeholders are able to identify figures, arrows, symbols and other elements that were deliberately chosen by a computer actor to communicate with stakeholders, to illustrate the requirements and share design ideas. It is more efficient and effective than any textual, graphical or formal specifications.

By using sketches, designers become more motivated, more creative, and perhaps more able to address the challenges of creating a successful design, and thus to produce a better design outcome [56]. By definition, prototypes are scaled-down versions of what will be built. Designers use them because they are faster and cheaper to create than the final blueprints. Sketching is a quick provisional drawing, it therefore matches the requirements of prototyping [49].

In Human-Computer Interaction (HCI) design, Collaborative User-Centered Design (CUCD) process suggests using sketch to understanding and designing all the aspects of a user interface (UI) design and for getting involved a wide community of stakeholders, such as, but not limited to: user researcher, information architect, user interface interaction designers, user testing specialists, software developers, marketing personnel and software products leaders. Early sketches inform the development of User interface conceptual, interaction style, and even other related material such training resources, support services, online help, etc. UI usability can be seen as a design problem of “wicked nature”, as a problem to be solved – the more you try to solve the problem, the more you discover the complexity of the UI usability and more you are able to suggest solutions in the early design phase of the CUCD life cycle. This is because usually original usability problem has implications/consequences that cannot be known in advance. Sketches have been shown very powerful to build a consensus and a trade-off when usability problems are conflicting with other major quality factors, such as security. A UI sketch reveals how much a system could be usable, secure.

Sketching is the practice of drawing a rough outline or rough draft version of a final piece of art. Sketching is an aid to thought. Sketches are used as a mean of designing. Design by sketching has its foundations on the participatory design approach [37] in which a person not trained, qualified or experienced is an active and essential participant in the design process. As a communication tool, sketching can be used as a way of graphically specifying abstract ideas. It is a message from the designer to stakeholders. It can be understood by the receiver, misinterpreted, or ignored, but there is definitely a message. This message must be validated when there is a consensus to be achieved between the designer and someone else, for which designers often use limited or scaled versions of what is being designed.

A common ground to disciplines relying on is that stakeholders, particularly designers and end-users, whatever their background and skills in sketching and design are, feel actively engaged [5]. End-users help in materializing some requirements such as usability [50]. Designers then annotate original sketches introduce by end-users, add illustrations, and further develop them. They usually proceed by iteratively sketching a concept at different levels of abstraction [45]. Ambler [1] defines UI Prototyping as an iterative analytical technique in which end users are actively involved, namely by providing feedback since the early development stages and continuously afterwards.

Sketching covers many domains of human activity and inside these domains, there are several works exploiting sketching for one or many sub-activities such as for example: problem analysis in general [44], computer science [17] (e.g., user experience support [8], [18], user interface design, prototyping, and recognition [23], [13], [14], [39], [3], [2], [47], cross-device UI design [25], user-centered design in agile projects [32], [7], system walkthrough [24]), system development (e.g., QUILL [16] for model-based design of web applications), flexible modelling [40] (e.g., FlexiSketch [54], [55] for model sketching), RAPIDO [35] for web API development, sketching UML models (e.g., TAHUTI for sketching UML Class diagrams [19] and SketchML for various UML diagrams [3]), distributed software design [29], [33], task modelling [41], notation creation [55]), computer-supported collaborative work [21] (e.g., stakeholders’ meetings [20], collocated tables for meetings [51] and interactive design spaces [6]), product and service design (e.g., sketching in design [21], extreme designing [15], industrial design [52], shape-changing products [42]), public displays [36], learning (e.g., classroom design studio [26], teaching geometry [11]), ideation [4] and concept generation [56], knowledge design, capture, and sharing [31], design in any area of engineering [56] (e.g., knot diagramming [43]). Many techniques have been successfully reported for expressing sketch grammars [10], with the need to take into account the context of use (i.e., the user, the devices/platforms, and the environment) [9] to get context-aware sketching [11].

III. A DESIGN SPACE FOR CROSS-SURFACE COLLABORATION BY SKETCHING

We determined the greatest common denominator in terms of properties between the 41 aforementioned references, independently of their domain, provided that collaborative sketching is involved to some extent. This identification resulted into CROSSIDE, a design space expressing collaborative sketching according seven dimensions represented clockwise in Fig. 1: user configurations, surface configurations, input interaction techniques, work methods, tangibility, layout, and device configurations. Each dimension is organized according to a progressive degree of sophistication: each step starts from the simplest value found in the literature until the most sophisticated degree. We considered this representation as adequate to satisfy three virtues: *descriptive* (the design space should be able to describe any work on collaborative sketching based on these seven dimensions), *comparative* (the design space should be able to compare two or more works on collaborative sketching to identify their similarities and differences) and *generative* (once a comparison is performed, the design space should enable to identify undercovered areas and generate new and interesting configurations).

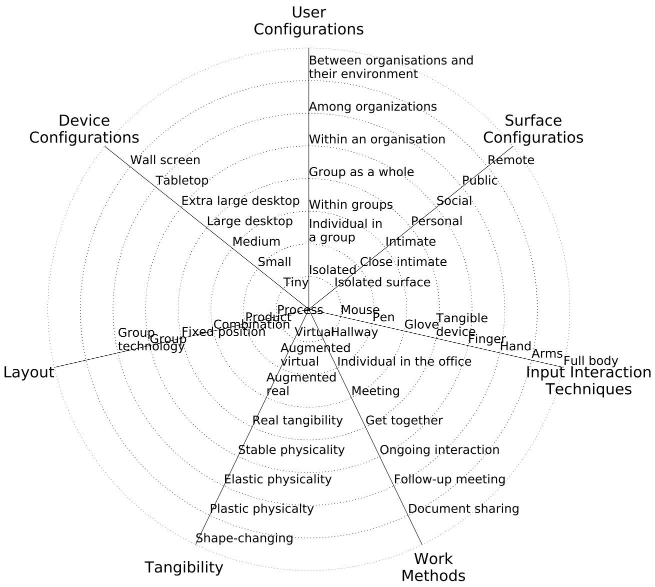

Fig. 1. A design space for Cross-Surface Collaboration by Sketching.

These seven dimensions are not intended to be completely independent of each other. Rather, they are aimed at serving these three virtues. Moreover, a radar chart can be effectively used to graphically represent the values along these dimensions by representing them on axes starting from the same origin. The steps on each dimension are not intended to be aligned or to be corresponding by concentric level. Therefore, steps joined on a same circle are not necessarily dependent, they only represent some progression.

A. User Configurations

The *user configurations* provide multiple ways to carry out a task by different stakeholders. Since many people could be located in different places, possibly at different levels of various organisations, the design space should consider the group configuration. Fig. 2a depicts various configurations among stakeholders involved in a distributed task [5]: *individual* (one task is carried out by one person in a group), *within groups* (one task is distributed across persons of a same group in the organisational structure), *group as a whole* (one task is carried out by one group of persons, independently of its internal organisation), *among groups* (one task is passed from one group to another), *within organisation* (one task is distributed across entities of the organisational structure), *among organisations* (when one task is distributed across several different organisations, all having their own internal structure), and *between organisations and their environment* (when tasks are exchanged between organisations and their common environment). Collaborative sketching requires stakeholders within groups because sketching tasks are distributed across people of a same group of the same organisational structure or not.

Fig. 3 graphically depicts an evolutive scale of the user configuration for different user configurations with respect to number of sessions and scenarios: an isolated stakeholder working alone, an individual stakeholder working as a group member, a single group of stakeholders, multiple groups of different stakeholders, multiple groups within the same organization (e.g., a group of representative end-users in a bank vs a group of interaction designers in the same bank), several groups across organizations (e.g., an association of interaction designers in a particular domain of human activity) or the whole world (e.g., a community of practice could be developed that gathers people involved in a particular family of designs).

B. Surface Configurations

The *surface configurations* express a hypothetical distance between a stakeholder working alone and multiple ones collaborating in a personal, social, public or remote environment, thus touching the notion of territoriality. Fig. 2c differentiates configurations depending on concentric spatial zones centered around the stakeholder. The distance at which users feel comfortable interacting with others in tabletop environments depend on age and culture [51]. This study has contemplated the type of task or activity in which the users are engaged to influence tabletop design. In a completely isolated or close intimated space, a user works with personal interaction surfaces [12].

For example, a collaborative sketching system enables users testing UI prototypes on any surface size if the user is completely isolated. When users require a close intimate space, tiny or small surfaces will be used instead because they facilitate face-to-face collaboration and eye-contact [52]. Stakeholders who are not designers or developers usually come to meetings with their own interaction surface (e.g., a smartphone, a tablet) and would like to see the UI prototype on their very right devices, even if they drew something on another, possibly larger, surface. Personal spaces offer more ample possibilities than intimate spaces because users sitting around a table or standing side-by-side can share work-spaces by using multiple displays, large, or extra large desktops, tabletops, and wall screens. The social space takes place in a business-based interaction. Stakeholders and final users could interacting in shared work-spaces by using large tabletop display or wall screens.

For example, a social space is adopted when some kind of multiple user's interactions in taking place in a space between 1.2–3.6 m. Public space allows the user to interact with a sizable audience. In this context, the users make use of multiple display. Small surfaces allow manipulation of the work-space, while large surfaces serve to display the information to the entire audience. Informational wall surfaces, like public displays [36], provide shared views to the users standing far away from the display. Interactive walls enable users to walk up to the display and interleave interaction and discussion among participants. Finally, in remote spaces, users share work-spaces with same or different times. A collaborative sketching system could offer most surface configurations. Stakeholders interact simultaneously with any device. In a collaborative session, users can sketch scenes in a private way or separated into small groups, perform a task privately and then communicate the results to all stakeholders [45].

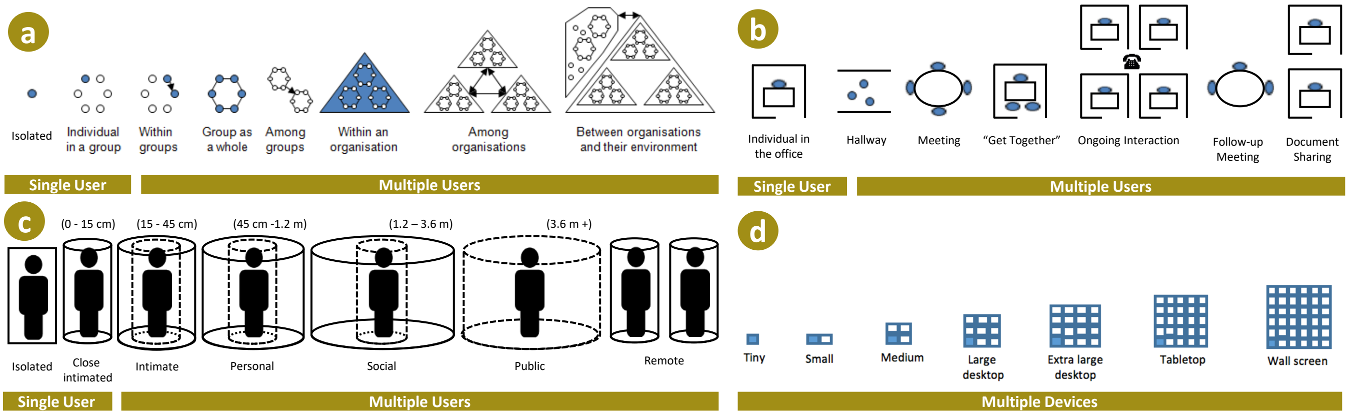

Fig. 2. CROSSIDE dimensions: (a) the "user" dimension, (b) the "work method" dimension, (c) the "surface" dimension, and (d) the "device" dimension.

They can interact using integrated environments composed of horizontal or vertical displays regardless of the dimensions of these devices.

C. Input Interaction Techniques

The *input interaction techniques* express the sophistication degree with which sketching is supported, ranging from an indirect manipulation to full direct manipulation. This dimension takes into account the use of materials and the naturalness of the interaction. A classical indirect pointing device is the "mouse" because its locus of control (i.e., the physical space in which actions occur) is different from or outside the locus of application (i.e., where the actions are applied). "Pen-based" interaction enables the user to interact with the device by using a passive or active stylus rather than a mouse. While pen devices initially supported indirect manipulation, they now support direct manipulation, namely by touch technology, regardless the fingers or the pointers involved. The "glove" considers the family of lightweight and stretchable devices that combines hand posture sensing and tactile pressure. Manipulation is direct, by device intermediation, like with smartwatches, armbands, and rings. "Tangible" devices naturally offer direct manipulation by connecting objects and surfaces to digital information. Tangible UIs typically work on tabletop surface and embed the tracking mechanism inside or outside the surface [46]. The "finger" considers situations where the end user interacts directly with the surface by finger tracking. This dimension could be extended to using the whole hand, but requires technologies for hand pose recognition. The last step, i.e. "full body", involves full-body gesture recognition, but its benefits for sketching are yet to be demonstrated.

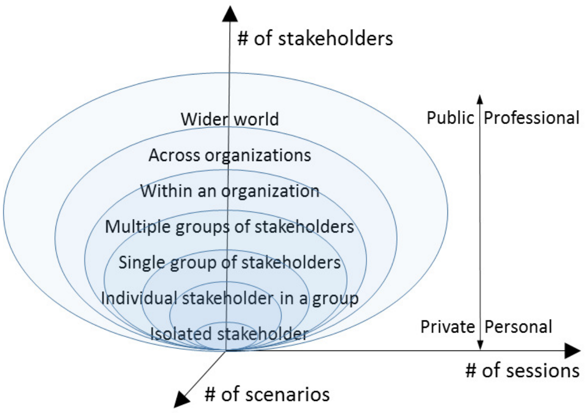

Fig. 3. User configuration with sessions, stakeholders, and scenarios.

D. Work Methods

The *work methods* characterise how interaction surfaces are spatially arranged according to a territoriality in the environment and how these environments are connected together (e.g., through Wi-Fi, LAN, WAN). Interaction surfaces could be *tiled*, *coupled*, *uncoupled*, or *positioned side-by-side* [12]. This dimension subsumes the physical location of each surface and how it is positioned (i.e., vertically, horizontally, in an oblique way). The dimension also considers the devices configurations of the five categories of technologies for collocated collaborative work classified by Wang et al. [52] namely: horizontal displays, large vertical displays, multiple displays, tangible interfaces, and integrated environments. Fig. 2b depicts typical setups of single and multiple users. Users can work alone in their own organisations. Multiple users work in corporate environments [5]: *hallway* (when an environment consists of any informal place where users could meet), *individual in the office* (when an environment only accommodates one user at a time, although this user can change over time), *meeting* (when the environment accommodates several users at a time for conducting a meeting), *get together* (when the environment accommodates several users for collaboration in general), *ongoing interaction* (when different environments involve many different users).

E. Tangibility

The *tangibility* dimension expresses to what extent the materialization of collaboration space is digital, physical, or mixed, ranging from applications in digital environments to the use of shape-changing devices. Digital environments define most classical applications where UI can be 1D (based on lines), 2D (based on surfaces), 2D1/2 (based on a space projected onto a surface) or 3D (in space). 1D UIs are

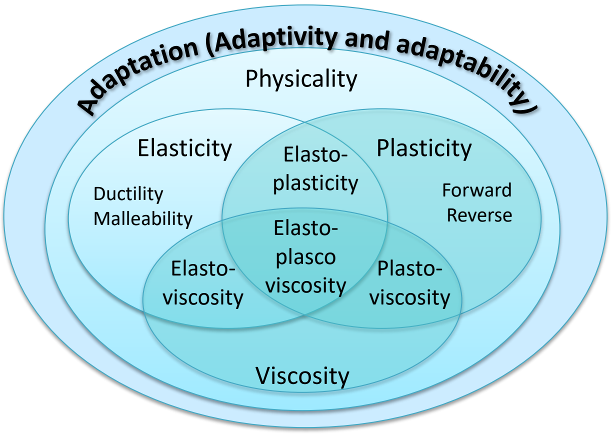

Fig. 4. Properties of physicality, such as plasticity and elasticity.

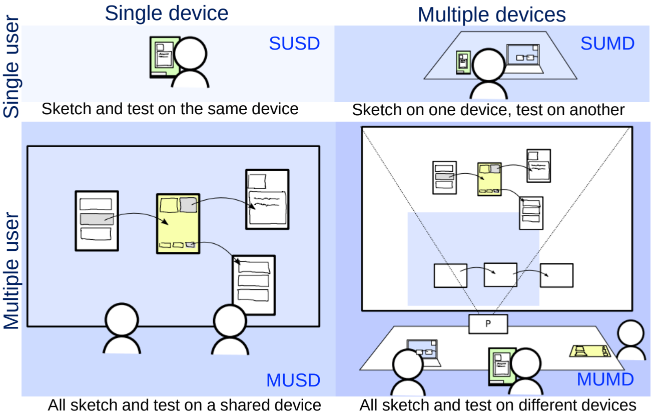
Single device
Multiple devices
Single user
Multiple user
SUSD
SUMD
Sketch and test on the same device
Sketch on one device, test on another
MUSD
MUMD
All sketch and test on a shared device
All sketch and test on different devices

Fig. 5. Quadrant of the four configurations.

typically based on command lines or instructions. In digital environments, users interact with a minimal immersion degree. Information is presented as a stream of characters in a text terminal. 2D UIs are incorporated in an environment based on conventional bi-dimensional representations where the interaction techniques are supported by events. Their widgets and their composition enable creating complex interfaces. The 2D1/2 and 3D interfaces extend classic GUIs with the notion of overlap and depth perceptions.

Virtual describes real environments simulated by a computer where users are involved with a high immersion degree. Users interact with virtual objects in an infinite 3D space. On the opposite of the dimension, applications are developed in real environments and use of mechanisms to increase the perception of the user. *Augmented Virtual* are applications including virtual worlds generated by a computer. These applications incorporate virtual reality to replace the physical world and the virtual world predominates over the real. Augmented Virtual extends the physical reality perceived by incorporating virtual objects into the physical world, thus increasing the degree of immersion. In contrast, *Augmented Real* considers an otherwise real environment augmented by means of virtual objects [34]. In the Augmented Real, virtual objects increase the real world, the dominant medium.

Tangible UIs give physical form to digital information, employing physical artifacts both as representations and controls for computational media [48]. This is further refined into four steps depending how tangible objects are materialized: real tangibility, stable physicality, elastic physicality, and plastic physicality. Real tangibility attempts the reproduce the physical behavior of a real world object into the tangible object, or a sub-set of it. Stable physicality occurs when tangible objects never change their behavior, whether the are expected to mimic some real work or not. Elastic physicality groups all environments that use both physical materials with computational analysis and simulation. These environments are used to understand and represent the world. Plastic physicality occurs in the area of shape-changing UI, where the devices are artifacts whose surface and/or volume can be articulated and modulated with their spatial domain [42]. In physics and materials science, the physicality property is one form of adaptation, which could be decomposed into three major properties (Fig. 4): *plasticity*, which describes the deformation of a material undergoing (non-reversible) changes of shape in response to applied forces; *elasticity*, which is the tendency of solid materials to return to their original shape after being deformed; and *viscosity*, which expresses to what extent a material is resistant with respect to deformation. Solid objects will deform when forces are applied on them ; if the object is elastic, it will return to its initial shape and size when these forces are released. These two terms can be used for tangibility.

F. Device Configurations

The *device configurations* refer to the surface size of device independently of their density and orientation. Fig. 2d depicts the different dimensions for the device configurations. The sizes are categorized by generalised dimensions ranging from tiny to wall screen. A system could run on a variety of devices offering different sizes, performing scaling and resizing to accommodate variations induced by different screens. Four typical configurations based on stakeholders (or users) and their surfaces therefore emerge (Fig. 5):

1) *Single user-Single device* (SUSD): a single stakeholder, such as an end-user, is working in isolation on one device only to sketch a UI (see figure 5 - top left). This device could be the very right device on which the final UI should run or another one. Therefore, this configuration is appropriate for conducting the sketching, prototyping, and testing activities as defined.
2) *Single user-Multiple devices* (SUMD): a single stakeholder is sketching on multiple devices either simultaneously or asynchronously in order to assemble the various sketches into a coherent design scenario (see figure 5 - top right). Therefore, this configuration is appropriate for conducting the sketching, prototyping, and sharing/testing activities.
3) *Multiple user-Single Device* (MUSD): several stakeholders share a same device in order to interact together and to come up with an agreement on some design questions (see figure 5 - bottom left). This configuration is expected to mimic the classical whiteboard

configuration where different stakeholders are together in front of a whiteboard and sketching things all together. Therefore, this configuration is appropriate for supporting the sharing/testing and discussing/reflecting activities, while it could be also used for the sketching and prototyping activities, but not primarily.

4) *Multiple users-Multiple devices* (MUMD): several stakeholders exploit private and public devices to apply any decided modification on the design scenario (see figure 5 - bottom right). They could perform these actions first on their own private device (e.g., their tablet), and then propagate modifications to the whole scenario (displayed on a wall screen). They straightforwardly interact on the public device to collect immediate feedback from other stakeholders. Therefore, this configuration is appropriate for supporting the sharing/testing and discussing/reflecting activities.

G. Building process of the Design Space

The design space has been built by identifying the greatest common denominator of software features described from forty-one references in various domains. The process followed to build the design space was basically a middle-out approach that combines two sub-processes, while trying to satisfy the principle of separation of concerns regarding the three main aspects of a context of use (e.g., users and their interactive tasks, their platforms and devices, their physical and organisational environments [9]):

1) A *bottom-up approach* consisting of browsing each reference at a time, identifying any high-level factor supporting collaborative sketching by separating them for users, platforms, and environments. All values found for each factor were collected in the same set.
2) A *top-down approach* consisting of determining such high-level factors by looking after a theoretical definition in the literature.

For example, the "User configurations" factor was identified among several references, therefore encouraging us to find a taxonomy that is relevant enough to express values found for this factor. We discovered a similar concept in Mandviwalla and Olfman [27]. To make it suitable for our purpose, we expanded the possible values from values collected in the references and we adapted the definition accordingly to make it categorical. Fig. 2a graphically represents the resulting factor presented as a progressive dimension.

Another example is related to the "Surface configurations" factor. By extracting from each use case found in every reference, we wanted to categorise the surface projected on the ground that delineates the interaction space. When only one person is involved in a sketching, there are various comfort zones from social psychology. When several persons are involved in a collaborative sketching, the way they interact with their system depends on what they want to do with their sketch. The sketching activity itself is more frequently found in small surfaces, while sharing and discussing the sketching is more frequently found in medium to large surfaces. The scope of the input and the output determines the surface.

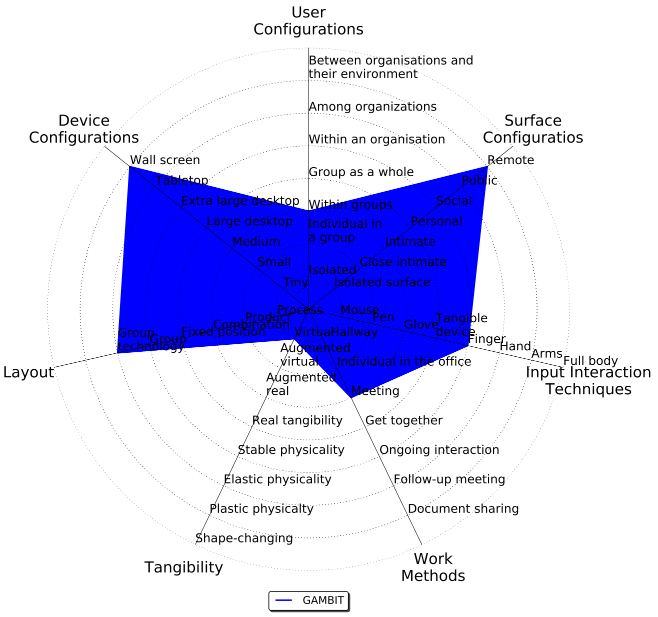

Fig. 6. CROSSIDE instantiation for GAMBIT.

IV. INSTANTIATION OF THE DESIGN SPACE

This section first instantiates CROSSIDE on GAMBIT, one of our systems for collaborative sketching, to check how the descriptive virtue is addressed. It then performs the instantiations for the external tools: BELONGINGS [38], FLEXISKETCH [54], [55], [53], CALICO [28], SKETCHML [3] and EVE [47]. For the instantiations of each of the tools, we superimpose the first drawing to address the comparative virtue. At the end, all instances are presented in the same design space allowing to see their similarities and differences.

A. *The* CROSSIDE *instantiation for* GAMBIT

GAMBIT[1] (Gatherings And Meetings with Beamers and Interactive Tables) [45] is a freely accessible[2] multi-platform HTML5 environment for collaborative design by sketching for UI design, but not only, by supporting (Fig. 6):

- *Multi-fidelity*: it should enable stakeholders to provide material at any level of fidelity and to easily switch from one level of fidelity to another.
- *Multi-format*: while sketching remains at the heart of the service, it should accommodate various formats of data both as input and as output.
- *Multi-session*: it should offer the possibility to create, edit, one or many sessions simultaneously.
- *Multi-scenario*: within a single session, it should be able to manage several alternate scenarios at once, namely by representing explicitly the different scenarios and

[1] The word "gambit" is used in chess in reference to a movement in which the player sacrifices a piece, usually a pawn, in exchange to some future advantage; in user interface design the gambit would be made by the designer who sacrifices sketches in exchange to know in advance how the interface would function in the hands of the end-users, allowing her to learn from the eventual observation of this usage.

[2] Accessible at http://gambitsketch.appspot.com/

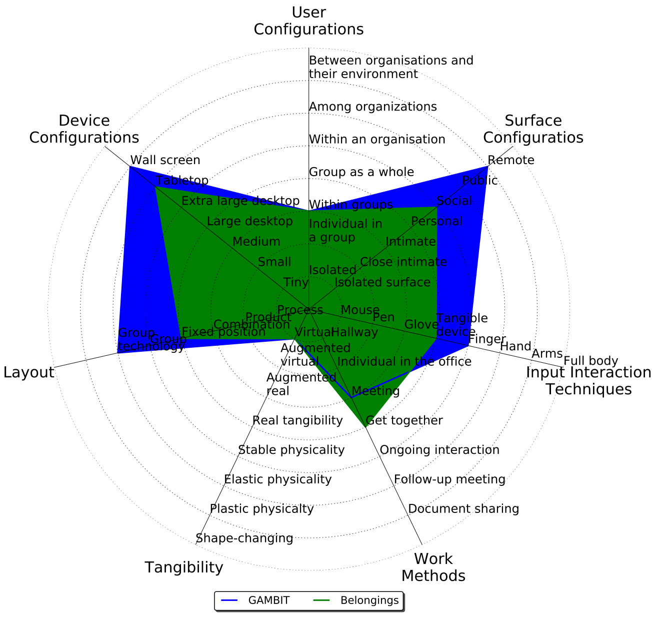

Fig. 7. CROSSIDE instantiation for BELONGINGS superimposed to the instantiation for GAMBIT.

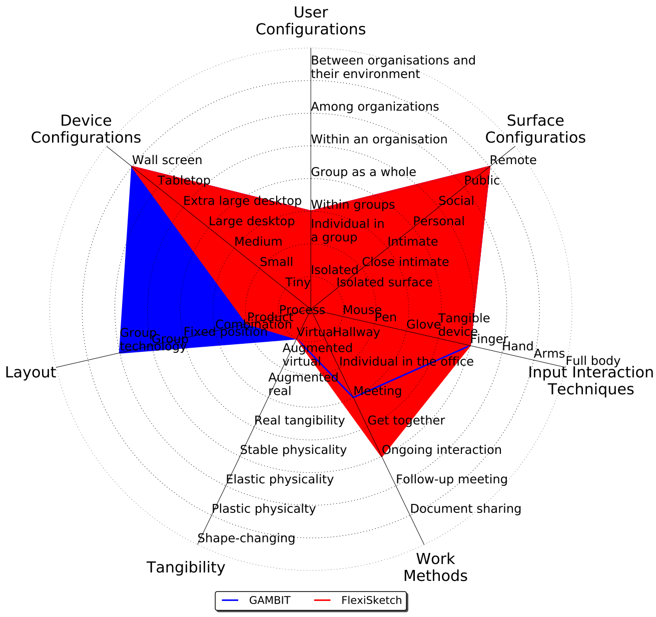

Fig. 8. CROSSIDE instantiation for FLEXISKETCH on GAMBIT.

by linking them. For instance fork and join mechanisms could be used to map scenarios together and create different design paths for idea writing. Different stakeholders make explicit more design ideas so as to generate new design paths that were remained under explored before. This is very important to avoid the "Do not fall in love of your own design" observation: once a design alternative is selected as an official one, some stakeholders tend to keep it (they fall in love of their design) and to spend remaining time in refining this design as opposed to exploring alternate designs.

- *Multi-stakeholder*: a same session could be created to involve a wide community of stakeholders whose roles and responsibilities may evolve over time. When the role or the responsibility of a stakeholder changes depending on the context of the problem, the tool should accommodate these changes flexibly.
- *Multi-access*: each stakeholder should contribute to a design session with her own devices. End-users as well as designers tend to create smaller sketches on small screen devices and larger sketches when the interaction surface increases, without necessarily adding more details. Multi-device is equally important to support cross-device interaction, e.g., when a stakeholder may switch from one device to another while participating in the same session, but with different roles.

B. *The* CROSSIDE *instantiation for* BELONGINGS

BELONGINGS [38] consists of an interactive tangible tabletop surrounded by multiple interactive surfaces and smaller devices for collaborative query by sketching. The table was installed in a museum and designed to communicate indigenous traditional knowledge and cultural values. Fig. 7 depicts the design space instantiated for this environment superimposed on the previous one (Fig. 6) to compare the coverage of both tools. BELONGINGS can be operated in a fixed tabletop by two users working together in a social context. Unlike GAMBIT, BELONGINGS supports the use of tangible devices, thus being better with that respect on the dimension. Both software share a participatory design [37] approach as part of the development process: designers and final users work together to produce the final application.

C. *The* CROSSIDE *instantiation for* FLEXISKETCH

FLEXISKETCH [54], [55], [53] comes in two versions: the first uses tablets as sketching media and supports inexpensive, mobile sketching at any time and in any place, while the second version, known as "FlexiSketch Desktop", runs on electronic whiteboards and provides a wide screen for collocated meetings. Multiple tablets can be connected to the desktop version over Wi-Fi, thus enabling end users to collaborate and simultaneously work in the same workspace regardless the multiple screens they are using. FLEXISKETCH enables participants to edit the workspace simultaneously, offers shared and private views, and allows group work and/or facilitating individual work. Fig. 8 depicts how FlexiSketch is instantied on the design space. As a way for comparison, the instantiation of Gambit is also superimposed. FlexiSketch and Gambit have the similarities of considering the participatory design as part of the development process. Both tools allow multiple concurrent accesses through the simultaneous use of multiple devices.

D. *The* CROSSIDE *instantiation for* CALICO

CALICO [28] is a free hand rapid design tool supporting early software design activities to be used with touch screen interfaces, such as interactive whiteboards and tablet PCs.

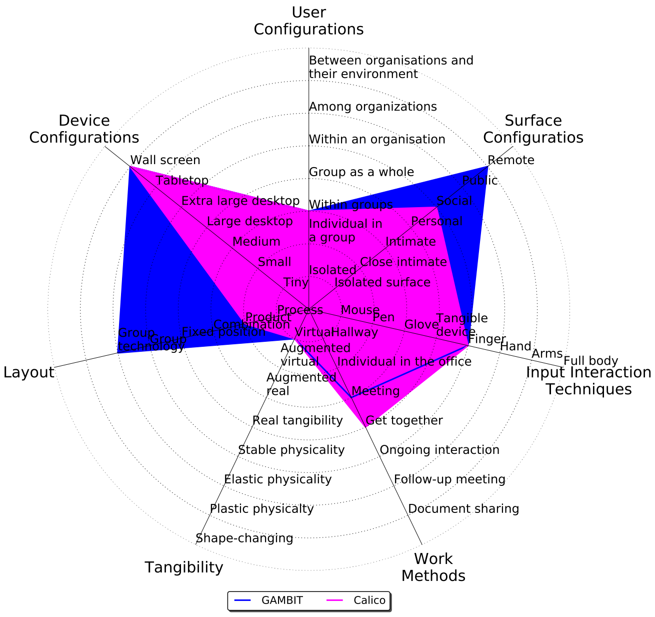

Fig. 9. CROSSIDE instantiation for CALICO superimposed to the instantiation for GAMBIT.

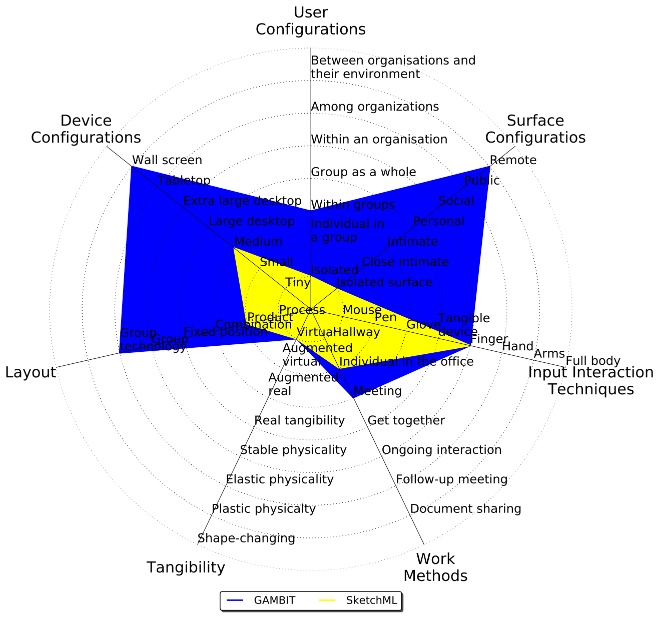

Fig. 10. CROSSIDE instantiation for SKETCHML superimposed to the instantiation for GAMBIT.

CALICO enables designers creating designs on multiple canvases based on a client-server architecture, supporting up to 20 simultaneously active users [30]. A CALICO client is portable, supporting computers connected to electronic whiteboards, laptops, and tablets. Thus, CALICO supports collaborative work across multiple devices, allowing multiple designers to work synchronously on the same canvas or asynchronously on different canvases. This allows designers working in a group to branch off to their own canvas. Fig. 9 combines the instantiations of CALICO and GAMBIT.

E. *The* CROSSIDE *instantiation for* SKETCHML

SKETCHML [3] is a framework that offers the ability to define and recognize every kind of 2D graphical library, by using freehand drawing, to be used in the construction of user interfaces. The framework uses an empirical language based on XML. It language allows the compatibility of SketchML with other applications and services through various devices. Fig. 10 combines the instantiations of SKETCHML and GAMBIT.

F. *The* CROSSIDE *instantiation for* EVE

EVE [47] is a Sketch-based prototyping workbench that facilitates end-users to define their design through a set of low-fidelity sketches. The Low-fidelity representations are recognized and translated in medium fidelity representations, as well as in high fidelity prototypes. End-users realize the representations in a canvas of two dimensions. End-users can navigate through the three levels of loyalty. At each level it is possible to make the desired changes. For each of the fidelity, end-users can operate three modes. The design functionality, the configuration of the interaction and the preview of the prototype. Fig. 12 combines the instantiations of EVE and GAMBIT.

G. *The comparative virtue of* CROSSIDE

Fig. 13 combines all the instantiations of tools studied insofar in a radar diagram to facilitate the visual comparison of the tools (provided that they are not too numerous), such as the similarities and differences. The goal is to satisfy the comparative virtue of the design space, not to promote an ideal tool being the best along all dimensions. Rather, some tools are more advanced along some dimensions, say the “Device configurations”, while others are targeting more flexibility for other dimensions, like “User configurations”. For example, the range of “Device configurations” is well covered by both GAMBIT and CALICO, up to the three first steps of “Work methods”. Fig. 11 illustrates the “Meeting” work method, where two designers share a tabletop for sketching a prototype by pen that is rendered in real-time on the smartphone of an end-user. Sketching is in direct manipulation: the designer is sketching directly on the tabletop and manipulating the sketch in the same way.

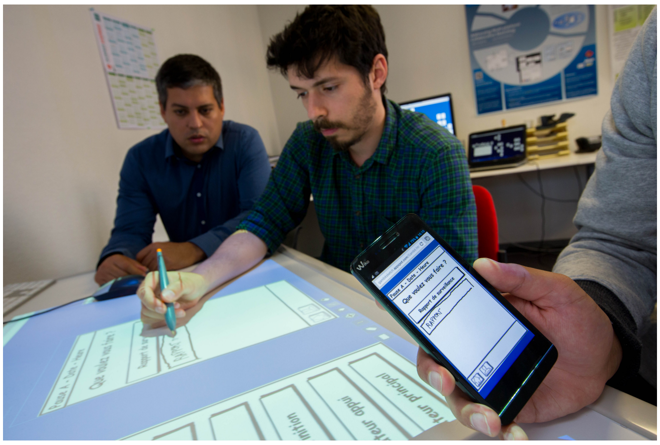

Fig. 11. Meeting work method for GAMBIT.

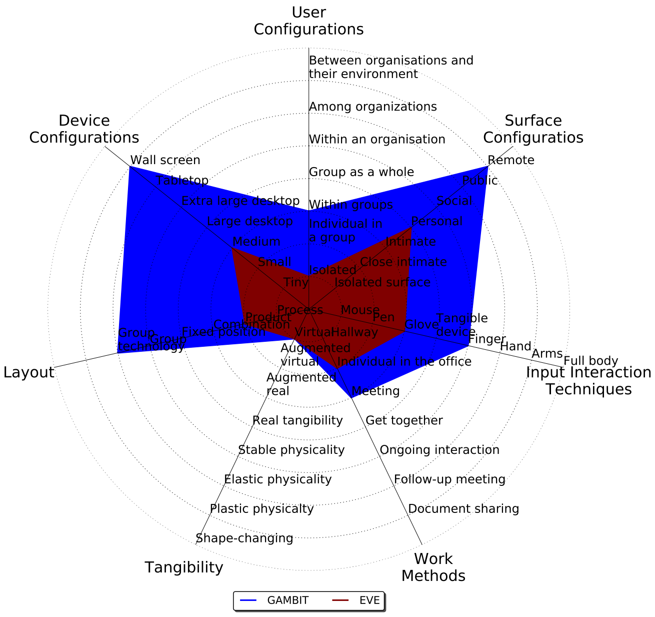

Fig. 12. CROSSIDE instantiation for EVE superimposed to the instantiation for GAMBIT.

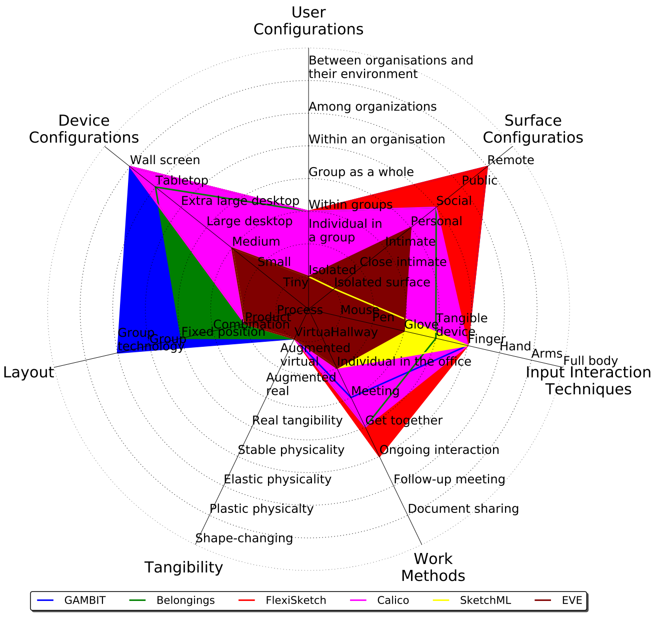

Fig. 13. CROSSIDE instantiations for BELONGINGS, FLEXISKETCH, CALICO, SKETCHML, EVE and GAMBIT all combined at once.

H. *The generative virtue of* CROSSIDE

A close examination of Fig. 13 leads to the observation that some dimensions are pretty well covered, like “Device configurations” and “Surface configurations”, some others are moderately covered like “Layout”, “Work methods”, and “Input interaction techniques” while the remaining ones are limited, such as “User configurations” and “Tangibility”. This suggests that these dimensions are welcome to be explored in further activities of a research agenda. Indeed, tangibility is very limited in all tools examined so far: most of them represent digital solutions where sketching is achieved on a 2-D surface with limited beautification performed in this space. There are probably other tools for collaborative sketching in 3D exhibiting the capability to virtually augment the real world, but they do not belong to our initial list of references. For the practitioners, this means that no tool is available today to fulfill these needs.

The design space obtained so far only reflects some significant dimensions identified among a set of forty-one references considered as representative instances of collaborative sketching in various domains of human activity, ranging from learning to industrial design. While this set of references covers several domains, we do not argue that its coverage is complete or representative enough of the vast majority of tools of interest.

Therefore, our next step consists of conducting a Systematic Literature Review (SLR) [22] for identifying references relevant to collaborative sketching for multiple purposes:

1) To expand the coverage of reviewed works from our 41 selected references to a larger panel.
2) To address the reproductibility of the procedure for guaranteeing the coverage of the design space.
3) To address explicitly the generative virtue by discussing the most promising configurations on this design space which may serve for a research agenda in the near future.

For the moment, we can superimpose the instantiations of the design space performed for the 41 references. On one hand, this superimposition enables us to identify portions of each dimension that are more or less frequently covered, or not covered at all. But this analysis considers only one dimension at a time, which may be considered as reductive. On the other hand, the superimposition also enables us to identify the configurations that are the most or the least frequently adopted by tools. This does not mean that they are appropriate or not, but simply the coverage could be discussed. We prefer to perform this analysis on a set of references resulting from the SLR instead of our initial set.

V. CONCLUSION AND FUTURE WORK

We presented CROSSIDE, a design space for representing capabilities of a software for collaborative sketching in a cross-surface setting. This design space consists of seven dimensions (i.e., user configurations, surface configurations, input interaction techniques, work methods, tangibility, layout, and device configurations) resulting from a comparative analysis of 41 references in the domain. Each instantiation of these 41 references on the design space is graphically depicted as a radar diagram, which visually supports three virtues: descriptive, comparative, and generative.

ACKNOWLEDGMENTS

The authors thank the anonymous reviewers for their constructive and patient comments on ealier versions of this manuscript.

References

[1] S. W. Ambler. Agile adoption rate survey results: March 2007, 2007. [Online; accessed 05-April-2019].

[2] D. Avola, L. Cinque, and G. Placidi. Sketchspore: A sketch based domain separation and recognition system for interactive interfaces. In A. Petrosino, editor, *Image Analysis and Processing – ICIAP 2013*, pages 181–190, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[3] D. Avola, A. Del Buono, G. Gianforme, S. Paolozzi, and R. Wang. Sketchml a representation language for novel sketch recognition approach. In *Proceedings of the 2Nd International Conference on PErvasive Technologies Related to Assistive Environments*, PETRA '09, pages 31:1–31:8, New York, NY, USA, 2009. ACM.

[4] R. Bellamy, M. Desmond, J. Martino, P. Matchen, H. Ossher, J. Richards, and C. Swart. Sketching tools for ideation (nier track). In *33rd Int. Conference on Software Engineering*, ICSE '11, pages 808–811, New York, NY, USA, 2011. ACM.

[5] E. Berglund and M. Bång. Requirements for distributed user interface in ubiquitous computing networks. In *Proc. of Conf. on Mobile and Ubiquitous MultiMedia*, 2002.

[6] J. Bowen and A. Dittmar. A semi-formal framework for describing interaction design spaces. In *Proceedings of the 8th ACM SIGCHI Symposium on Engineering Interactive Computing Systems*, EICS '16, pages 229–238, New York, NY, USA, 2016. ACM.

[7] T. Buchmann. Towards tool support for agile modeling: Sketching equals modeling. In *Proceedings of the 2012 Extreme Modeling Workshop*, XM '12, pages 9–14, New York, NY, USA, 2012. ACM.

[8] B. Buxton. *Sketching User Experiences: Getting the Design Right and the Right Design*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2007.

[9] G. Calvary, J. Coutaz, D. Thevenin, Q. Limbourg, L. Bouillon, and J. Vanderdonckt. A Unifying Reference Framework for multi-target user interfaces. *Interacting with Computers*, 15(3):289–308, 06 2003.

[10] G. Costagliola, V. Deufemia, and M. Risi. Sketch grammars: A formalism for describing and recognizing diagrammatic sketch languages. In *Eighth International Conference on Document Analysis and Recognition (ICDAR 2005), 29 August - 1 September 2005, Seoul, Korea*, pages 1226–1231. IEEE Computer Society, 2005.

[11] G. Costagliola, M. D. Rosa, and V. Fuccella. Local context-based recognition of sketched diagrams. *J. Vis. Lang. Comput.*, 25(6):955–962, 2014.

[12] J. Coutaz, C. Lachenal, and S. Dupuy-Chessa. Ontology for multi-surface interaction. In M. Rauterberg, M. Menozzi, and J. Wesson, editors, *Human-Computer Interaction INTERACT '03: IFIP TC13 International Conference on Human-Computer Interaction, 1st-5th September 2003, Zurich, Switzerland*. IOS Press, 2003.

[13] A. Coyette, S. Faulkner, M. Kolp, Q. Limbourg, and J. Vanderdonckt. SketchiXML: Towards a multi-agent design tool for sketching user interfaces based on UsiXML. In *Proc. of 3rd Int. Conf. on Task Models and Diagrams*, TAMODIA '04, pages 75–82, NY, 2004. ACM.

[14] A. Coyette, S. Kieffer, and J. Vanderdonckt. Multi-fidelity prototyping of user interfaces. In *Proc. of 11th IFIP*.

[15] B. S. da Silva, V. C. O. Aureliano, and S. D. J. Barbosa. Extreme designing: binding sketching to an interaction model in a streamlined HCI design approach. In *7th Brazilian symposium on human factors in cs*, pages 101–109. ACM, 2006.

[16] V. Genaro Motti, D. Raggett, S. Van Cauwelaert, and J. Vanderdonckt. Simplifying the development of cross-platform web user interfaces by collaborative model-based design. In *Proceedings of the 31st ACM International Conference on Design of Communication*, SIGDOC '13, pages 55–64, New York, NY, USA, 2013. ACM.

[17] C. Gonzalez-Perez. Filling the voids - from requirements to deployment with open/metis. In *Proc. of 5th Int. Conf. on Software and Data Technologies, Vol. 1, Athens, Greece, July 22-24, 2010*, page 19, 2010.

[18] S. Greenberg, S. Carpendale, N. Marquardt, and B. Buxton. *Sketching User Experiences - The Workbook*. Academic Press, 2012.

[19] T. Hammond and R. Davis. Tahuti: A geometrical sketch recognition system for uml class diagrams. In *AAAI Spring Symposium on Sketch Understanding*, pages 59–68, 2002.

[20] M. Johansson and M. Arvola. A case study of how user interface sketches, scenarios and computer prototypes structure stakeholder meetings. In *Proc. of the 21st British HCI Group Annual Conference on People and Computers, vol. 1*, BCS-HCI '07, pages 177–184, Swinton, UK, 2007. British Computer Society.

[21] G. Johnson, M. D. Gross, J. Hong, and E. Y. Do. Computational support for sketching in design: A review. *Foundations and Trends in Human-Computer Interaction*, 2(1):1–93, 2009.

[22] B. Kitchenham, R. Pretorius, D. Budgen, O. P. Brereton, M. Turner, M. Niazi, and S. Linkman. Systematic literature reviews in software engineering a tertiary study. *Information and Software Technology*, 52(8):792 – 805, 2010.

[23] J. A. Landay and B. A. Myers. Interactive sketching for the early stages of user interface design. In *Proc. of the CHI'95*, pages 43–50.

[24] C. Lewis, P. G. Polson, C. Wharton, and J. Rieman. Testing a walkthrough methodology for theory-based design of walk-up-and-use interfaces. In *Proc. of the ACM Conf. on Human Factors in Computing Systems*, CHI '90, pages 235–242, New York, USA, 1990. ACM.

[25] J. Lin and J. A. Landay. Employing patterns and layers for early-stage design and prototyping of cross-device user interfaces. In *Proc. of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, pages 1313–1322, New York, NY, USA, 2008. ACM.

[26] D. Loksa, N. Mangano, T. D. LaToza, and A. v. d. Hoek. Enabling a classroom design studio with a collaborative sketch design tool. In *Proc. of the Int. Conf. on Software Engineering*, ICSE '13, pages 1073–1082, Piscataway, NJ, USA, 2013. IEEE Press.

[27] M. Mandviwalla and L. Olfman. What do groups need? a proposed set of generic groupware requirements. *ACM Trans. Comput.-Hum. Interact.*, 1(3):245–268, Sept. 1994.

[28] N. Mangano, A. Baker, and A. Van Der Hoek. Calico: a prototype sketching tool for modeling in early design. In *Proceedings of the 2008 international workshop on Models in software engineering*, pages 63–68. ACM, 2008.

[29] N. Mangano, M. Dempsey, N. Lopez, and A. van der Hoek. A demonstration of a distributed software design sketching tool. In *Proc. of the 33rd Int. Conf. on Software Engineering*, ICSE '11, pages 1028–1030, New York, NY, USA, 2011. ACM.

[30] N. Mangano, T. D. LaToza, M. Petre, and A. van der Hoek. Supporting informal design with interactive whiteboards. In *Proc. of the ACM Conf. on Human Factors in Computing Systems*, pages 331–340, 2014.

[31] D. McCrickard. *Making Claims:The Claim as a Knowledge Design, Capture, and Sharing Tool in HCI*. Morgan & Claypool, 2012.

[32] P. McInerney and F. Maurer. Ucd in agile projects: dream team or odd couple? *interactions*, 12(6):19–23, 2005.

[33] J. Melchior, J. Vanderdonckt, and P. Van Roy. A model-based approach for distributed user interfaces. In *Proceedings of the 3rd ACM SIGCHI Symposium on Engineering Interactive Computing Systems*, EICS '11, pages 11–20, New York, NY, USA, 2011. ACM.

[34] P. Milgram and F. Kishino. A taxonomy of mixed reality visual displays. *IEICE TRANSACTIONS on Information and Systems*, 77(12):1321–1329, 1994.

[35] R. Mitra. Rapido: A sketching tool for web api designers. In *24th Int. Conference on World Wide Web*, WWW '15 Companion, pages 1509–1514, Republic and Canton of Geneva, Switzerland, 2015. Int. World Wide Web Conf. Steering Committee.

[36] J. Müller, F. Alt, D. Michelis, and A. Schmidt. Requirements and design space for interactive public displays. In *Proceedings of the 18th ACM International Conference on Multimedia*, MM '10, pages 1285–1294, New York, NY, USA, 2010. ACM.

[37] M. J. Muller and S. Kuhn. Participatory design. *Communications of the ACM*, 36(6):24–28, June 1993.

[38] R. Muntean. Considering collaboration in ?eleẁk̀w—belonging. In *WorkShop Cross-Surface 2016: Third International Workshop on Interacting with Multi-Device ecologies" in the wild"*, 2016.

[39] v. Obrenovic and J.-B. Martens. Sketching interactive systems with sketchify. *ACM Trans. Comp.-Hum. Interact.*, 18(1):1–38, May 2011.

[40] H. Ossher, A. van der Hoek, M.-A. Storey, J. Grundy, and R. Bellamy. Flexible modeling tools (flexitools2010). In *32nd ACM/IEEE International Conference on Software Engineering-Volume 2*, pages 441–442. ACM, 2010.

[41] J.-L. Pérez-Medina, S. Dupuy-Chessa, and A. Front. A survey of model driven engineering tools for user interface design. In M. Winckler, H. Johnson, and P. Palanque, editors, *Task Models and Diagrams for User Interface Design*, volume 4849 of *LNCS*, pages 84–97. Springer Berlin Heidelberg, 2007.

[42] M. K. Rasmussen, G. M. Troiano, M. G. Petersen, J. G. Simonsen, and K. Hornbæk. Sketching shape-changing interfaces: Exploring vocabulary, metaphors use, and affordances. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, pages 2740–2751, New York, NY, USA, 2016. ACM.

[43] M. D. Rosa, A. Fish, V. Fuccella, R. Saleh, S. Swartwood, and G. Costagliola. A toolkit for knot diagram sketching, encoding and re-generation. In G. Polese and V. Deufemia, editors, *The 22nd International Conference on Distributed Multimedia Systems, DMS 2016, Salerno, Italy, November 25-26, 2016.*, pages 16–25. KSI Research Inc. / Knowledge Systems Institute Graduate School, 2016.

[44] P. Sachse, W. Hacker, and S. Leinert. External thoughtdoes sketching

assist problem analysis? *Applied Cognitive Psychology*, 18(4):415–425, 2004.

[45] U. B. Sangiorgi, F. Beuvens, and J. Vanderdonckt. User interface design by collaborative sketching. In *Proceedings of the Designing Interactive Systems Conference*, DIS '12, pages 378–387, New York, NY, USA, 2012. ACM.

[46] O. Shaer and E. Hornecker. Tangible user interfaces: past, present, and future directions. *Foundations and Trends in Human-Computer Interaction*, 3(1–2):1–137, 2010.

[47] S. Suleri, V. P. Sermuga Pandian, S. Shishkovets, and M. Jarke. Eve: A sketch-based software prototyping workbench. In *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI EA '19, pages LBW1410:1–LBW1410:6, New York, NY, USA, 2019. ACM.

[48] B. Ullmer and H. Ishii. Emerging frameworks for tangible user interfaces. *IBM systems journal*, 39(3.4):915–931, 2000.

[49] R. van der Lugt. Functions of sketching in design idea generation meetings. In *Proc. of the 4th Conf. on Creativity & Cognition*, C&#38;C '02, pages 72–79, New York, USA, 2002. ACM.

[50] J. Vanderdonckt and A. Beirekdar. Automated web evaluation by guideline review. *J. Web Eng.*, 4(2):102–117, June 2005.

[51] J. R. Wallace and S. D. Scott. Contextual design considerations for co-located, collaborative tables. In *Horizontal Interactive Human Computer Systems, 2008. TABLETOP 2008. 3rd IEEE International Workshop on*, pages 57–64. IEEE, 2008.

[52] H. Wang and E. Blevis. Concepts that support collocated collaborative work inspired by the specific context of industrial designers. In *Proc. of the ACM Conf. on Computer Supported Cooperative Work*, CSCW '04, pages 546–549, New York, USA, 2004. ACM.

[53] D. Wüest, N. Seyff, and M. Glinz. Collaborative sketching and notation creation with FlexiSketch Team. *Software Engineering 2017*.

[54] D. Wüest, N. Seyff, and M. Glinz. Flexisketch: A mobile sketching tool for software modeling. In D. Uhler, K. Mehta, and J. Wong, editors, *Mobile Computing, Applications, and Services*, volume 110 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pages 225–244. Springer, 2013.

[55] D. Wuest, N. Seyff, and M. Glinz. Flexisketch team: Collaborative sketching and notation creation on the fly. In *Software Engineering (ICSE), 2015 IEEE/ACM 37th IEEE Int. Conf. on*, volume 2, pages 685–688, May 2015.

[56] M. C. Yang. Observations on concept generation and sketching in engineering design. *Research in Engineering Design*, 20(1):1–11, 2009.