

A Glimpse into the Past, Present, and Future of Engineering Interactive Computing Systems

JUDY BOWEN, The University of Waikato, New Zealand

JEAN VANDERDONCKT, Université catholique de Louvain, Belgium

MARCO WINCKLER, University Côte d'Azur, France

The area of Engineering Interactive Computing Systems (EICS) is often defined as the intersection between Software Engineering (SE) and Human-Computer Interaction (HCI). In this paper, we provide an overview of what EICS is and how it is positioned with respect to other venues in HCI, such as CHI, UIST, and IUI, highlighting its legacy and paying homage to past scientific events from which EICS emerged. We also take this opportunity to deliver a critical analysis of the past, present, and perhaps future research and development questions raised in EICS. In particular, we exemplify these questions by referring to scientific contributions which have been published in this area in this last issue of PACMHCI.

CCS Concepts: • **Human-centered computing** → **Human computer interaction (HCI); Interactive systems and tools;** • **Software and its engineering;**

Additional Key Words and Phrases: Interactive systems; Model-based design of user interfaces; Model-driven Engineering; Software Engineering; User Interfaces; Design; Formal methods; Verification and Validation.

ACM Reference Format:

Judy Bowen, Jean Vanderdonckt, and Marco Winckler. 2020. A Glimpse into the Past, Present, and Future of Engineering Interactive Computing Systems. *Proc. ACM Hum.-Comput. Interact.* 4, EICS, Article 71 (May 2020), 32 pages. <https://doi.org/10.1145/3394973>

1 WHAT EICS IS

The scientific field and community of Engineering Interactive Computing Systems (put shortly, “EICS”) are typically delineated at the overlapping of Human-Computer Interaction (HCI) and Software Engineering (SE). Therefore, scientific and technical contributions that address the development life-cycle of interactive systems and associated user interfaces are relevant to EICS. If Software Engineering can be defined as the discipline where engineering techniques are applied to systematically develop a part or whole of a software product, EICS covers the engineering applied to systematically specify, design, develop, and evaluate that specific part, *i.e.* the user interface, or the entirety of the interactive system, prototype, or system.

Consequently, *EICS focuses on models, languages, notations, methods, techniques, and tools that support the development life-cycle of interactive systems at any stage, from specification and requirements elicitation to validation.*

Authors' addresses: Judy Bowen, The University of Waikato, Department of Computing and Mathematical Sciences, Private Bag 3105, Hamilton, 3240, New Zealand, jbowen@waikato.ac.nz; Jean Vanderdonckt, Université catholique de Louvain, Louvain Research Institute in Organizations and Management, Place des Doyens, 1, Louvain-la-Neuve, 1348, Belgium, jean.vanderdonckt@uclouvain.be; Marco Winckler, University Côte d'Azur, I3S | SPARKS-WIMMICS team, Templiers 1, 930 Route des Colles, BP 145, Sophia Antipolis Cedex, 06903, France, winckler@unice.fr.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

© 2020 Copyright held by the owner/author(s).

2573-0142/2020/5-ART71

<https://doi.org/10.1145/3394973>

2 WHAT EICS IS NOT

While the EICS conference partially overlaps with other HCI venues, which is normal since the user interface is the common ground, there are other aspects to it that are rarely covered by those other venues. For this reason, it is important for newcomers to the field of EICS and especially to new submitters to PACMHCI/EICS to clearly understand the differences between EICS and other HCI venues so that their work can be submitted to, evaluated by, and presented in the right community.

- CHI, the ACM SIGCHI flagship Conference on Human Factors in Computing Systems, focuses on end-users, user interface designers and practitioners, and sometimes developers, whereas EICS targets all stakeholders that are involved in the software development life-cycle of an interactive system with explicit focus on analysts, developers, programmers, and system builders. While CHI typically targets the audience of user interface designers and end users, EICS targets the audience of those people who actually build the user interface.
- UIST, the ACM Symposium on User Interface Software and Technology, focuses on technological innovations regarding interaction techniques and user interfaces. When such techniques are introduced to the overall HCI community, UIST is the relevant sub-community to present them. However, when the innovations can be systematically specified, designed, developed, validated, and verified, they also become relevant to Engineering Interactive Computing Systems. When the implementation of these interaction techniques requires specific considerations, it falls into the scope of EICS.
- IUI, the ACM International Conference on Intelligent User Interfaces, is a venue and a community located at the intersection of HCI and Artificial Intelligence (AI). The relationship between EICS and IUI was very close at the beginning of IUI when, in March 1988, a group of researchers gathered in Monterey, California to participate in a workshop titled “Architectures for Intelligent Interfaces,”¹. The proceedings of the first edition in 1993 were published in [18]. When intelligent techniques for user interfaces are also relevant and employed for the systematic design, development, and evaluation of interactive systems, they become relevant to EICS as well. This intersection is often represented by model-based approaches for user interface design or development [28], later extended to the area of model-driven engineering of user interfaces [39].

3 A BRIEF HISTORY OF EICS

The EICS conference started as a sponsored ACM SIGCHI conference in 2009, but its journey as a scientific venue is probably one of the longest in the scientific field of Human-Computer Interaction. EICS was created as the union of several conferences (see Fig. 1 for a visual illustration):

- DSV-IS, the Design, Specification, and Verification of Interactive Systems International Conference, organized annually from 1994 [33] to 2008 [16]. This conference was mainly sponsored by Eurographics and its proceedings were published in the Lecture Notes in Computer Science series from Springer, most of the time. The first editions were specifically organized as an interactive workshop where delegates spent some days discussing current issues. See for instance [6, 31, 33].
- CCL, the IFIP Working Conference on Command Languages, with the first edition organized in 1974.
- TAMODIA, the Annual Conference on Task Models and Diagrams, organized annually from 2002 [34] to 2009 [13]. This venue was focusing on the usage of task models in the user interface development life cycle.

¹See <https://dl.acm.org/event.cfm?id=RE306>

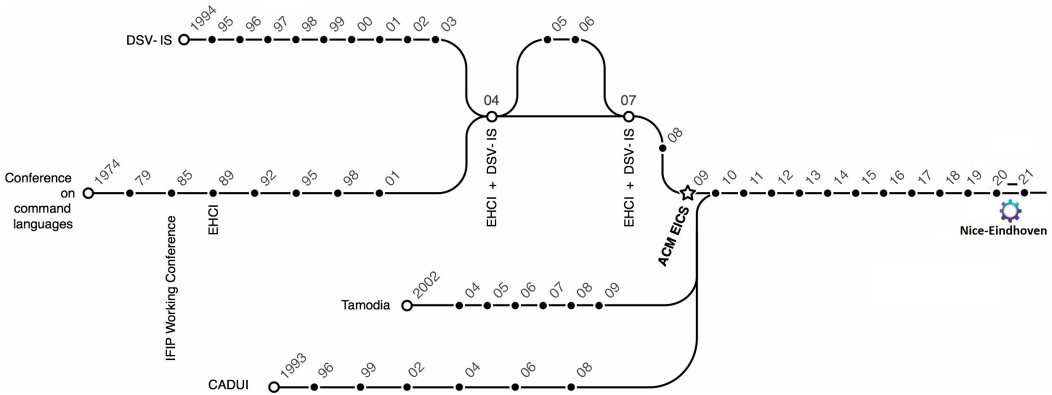


Fig. 1. A visual illustration of the history and legacy of the EICS community and conference showing past events, organized from as early as 1974, that contributed to EICS.

- CADUI, the International Workshop on Computer-Aided Design of User Interfaces, organized every two years since its first edition in Ulm in 1993. Its first published online proceedings were in 1996 [38] until 2008 [23]. This venue was more focusing on automating the user interface development life cycle, particularly on automated generation, verification, and validation of interactive systems.

Some of these events were sometimes organized jointly as several of their topics were overlapping, such as CADUI '96 [38] and DSV-IS '96 [6] or EHCI-DSVIS '04 [2] and EHCI-DSVIS '07 [19]. In parallel, a subset of EICS topics have started to be addressed for a particular type of interactive systems, Distributed User Interfaces (DUI), where the user interface is distributed across users, tasks, devices, platforms, and environments. DUI was organized as an international workshop along various conferences between 2011 and 2016; see [14, 15, 24, 25, 37].

Since all these scientific events shared similar topics and goals regarding engineering aspects of Human-Computer Interaction, *e.g.*, techniques, tools, and the practice of the software engineering life-cycle of interactive systems, merging them into one single event aimed at building a larger community was a natural thing to do. Since 2009 [17], EICS has been organized annually, reaching its twelfth edition in 2020 [32]. Three years ago, EICS full papers were published as part of PACMHCI, the Proceedings of the ACM on Human-Computer Interaction, a scientific journal for disseminating results that are relevant for the intersection between human factors and computing systems. This year, the fourth volume of PACMHCI/EICS has been published, featuring the latest advances on engineering interactive computing systems representing the best work of our community. Besides full papers, the EICS conference also welcomes Tech Notes and Late-Breaking Results, which are traditionally published in the Proceedings of the EICS conference.

4 EICS 2020

EICS 2020 was planned as the 12th edition of the ACM SIGCHI Symposium on Engineering Interactive Computing Systems, to be held during June 23-26 2020. EICS 2020 would have hosted by the University Côte d'Azur (Polytech Nice, SophiaTech Campus). The organizing team benefits from a wide experience in software development using the model-driven paradigm; the organizing team from the University Côte d'Azur has been working for many years on aspects regarding usability addressed from the perspective of conceptual models, model-based design of user interfaces, and web engineering. Due to the unexpected consequences of the COVID-19 pandemic EICS 2020 was unfortunately cancelled. All of the papers accepted for the PACMHCI/EICS journal were published

as usual, but authors who would normally have presented their work during the 2020 conference have instead been invited to do so at EICS 2021.

EICS 2020 received 61 full paper submissions during three reviewing rounds (Q3 and Q4 2019 and Q1 2020), of which 18 were accepted to be included in the 2020 volume of the PACMHCI journal, representing an acceptance rate of 30%, which is similar to last year's edition [32]. The EICS 2020 conference was also scheduled to include 4 Doctoral Consortium presentations, one tutorial entitled "Task Models Based Engineering of Interactive Systems", one workshop entitled "Entrain, Exploring New Territorial User Interfaces", 11 Late-Breaking Research papers (from a total of 18 submissions, representing an acceptance rate of 61%). Overall, the EICS 2020 conference comprises a programme composed of 18 selected presentations published in the proceedings [41], as well as 18 full papers published in the corresponding issue of PACMHCI (May 2020). Due to the postponement of EICS2020 the doctoral consortium and the workshop are organised on-line in a one day live event. EICS '2020 will therefore become part of EICS '2020/21!

5 A SUBMITTER'S GUIDE TO ENGINEERING INTERACTIVE COMPUTING SYSTEMS

While EICS 2021 is in progress to be held in Eindhoven (The Netherlands), future EICS conferences await contributions from the community. As the community is growing and attracting new researchers and practitioners interested in aspects that relate to engineering interactive systems, we believe that it is important to present them with an overview, in the form of a practical guide, regarding the types of contributions that are usually expected by the members of the EICS community for the PACMHCI journal as well as by the Programme Committee members and audience of the annual EICS conference.

Our main goal in this section is to provide a guide to newcomers to the field towards making their PACMHCI/EICS submissions successful and impactful.

Work intended to be presented at EICS covers the *full range of aspects that come into play when engineering interactive systems, such as innovations in the design, development, deployment, verification, and validation of interactive systems*. Consequently, topics of interest to the EICS audience include the design and development of systems incorporating new interaction techniques, multimodal interaction, multi-user, multi-device, multi-screen, and multi-environment interaction, mobile, wearable, and pervasive systems, large-scale and big data applications, Augmented and Mixed Reality, deployment of interactive systems, as well as novel methods and techniques to improve the development of interactive systems.

Overall, EICS focuses on models, languages, notations, methods, techniques, and tools that support designing and developing of interactive systems. The EICS conference aims to bring together researchers and practitioners that study or practice the engineering of interactive systems, drawing from the fields of Human-Computer Interaction, Software Engineering, Requirements Engineering, Conceptual Modeling, Computer Supported Cooperative Work, Artificial Intelligence, Augmented Reality, and Ubiquitous and Pervasive Systems. For instance, one topic that fits EICS well is *modeling interaction*.

Although there are several well-known models to represent persistence (such as the UML class diagrams) and behavior (the UML sequence model and use case diagrams), more innovations, consolidation, and development are welcome. For instance, conceptual programming is not feasible if persistence, behavior, and interaction are not adequately integrated with a sound conceptual basis [12]. Also, as new I/O devices become available, each device featuring diverse and different interaction modalities and characteristics, the study of alternative I/O modalities and multimodal interaction is timely and relevant to EICS.

Consequently, EICS topics include but are not limited to:

- Modelling and analysis of interaction and interactive systems
- Processes for engineering interactive systems (e.g., design, implementation, prototyping, evaluation, verification and validation, testing)
- Integrating interaction design into the software development process
- Requirements engineering for interactive systems
- Specification of interactive systems (methods, principles and tools)
- Software architectures for interactive systems
- Frameworks, toolkits, and APIs for interactive systems (e.g., API usability, interaction-driven API design)
- Domain-specific languages for interactive systems
- Formal methods within interactive systems engineering
- Modelling and analysis of users' activities
- Engineering innovative interactive applications (e.g., adaptive, tangible, touch and multitouch input, voice, gesture, EEG, multimodal input, mobile and wearable systems)
- Engineering hardware/software integration in interactive systems (e.g., fabrication and maker processes, physical computing, etc.)
- Engineering user experience (e.g., fun, affective)
- Engineering complex interactive systems (e.g., large datasets, large communities, enterprise systems, collaborative systems)
- Engineering interactive systems for various user categories (e.g., children, elderly, people with disabilities)
- Certification issues of interactive systems
- New datasets and evaluation data relevant for engineering interactive systems

Topics described above may be described in contributions of one, or more, of the following types:

- Proof of Concept - demonstration of the applicability of an approach/tool/method/architecture/framework/design space
- Case Study - collect detailed project data to demonstrate the benefits/shortcomings of the approach/artefacts used. May include datasets derived from the study.
- Comparative Study - monitor several projects to collect data on impact of the technology/method
- Formal Analysis - uses a formal analysis method or tool to analyse an artefact, e.g formal proof, model-driven approach, graph-based approach, user, UI and/or interaction modelling. Applying an approach.
- Theory/Formal Method - describes a new method/engineering approach for addressing research questions in EICS.
- Literature Review/Survey - Evaluate published studies that analyse the behaviour of similar methods/tools. Systematic mapping, systematic literature review etc.
- Experiment - perform a qualitative/quantitative analysis of a project/artefact.

6 REPRESENTATIVE EICS PAPERS

We provide in the following a list of contributions relevant to EICS together with examples of papers published at previous EICS events, as well as for this year's edition:

- (1) **Contributions to software engineering for interactive systems** by following sound software development methods, in the same way that UML is widely employed for data persistence and behavior. Several model-based approaches [28] exist to support the user interface development life-cycle and particularly design and code generation. Model-driven Engineering (MDE) is different from model-based approaches in that all models have to

be compliant with a meta-model, which also applies to transformations between models (Model-to-Model) and code generation (Model-to-Code transformations) [39].

✍ **Example:** Sandra Trullemans, Lars Van Holsbeeke and Beat Signer. 2017. The Context Modelling Toolkit: A Unified Multi-layered Context Modelling Approach. *Proc. ACM Hum.-Comput. Interact.* 1, EICS, Article 7 (June 2017), 16 pages. DOI: <http://dx.doi.org/10.1145/3095810>

✍ **Example:** Kai Breiner, Marc Seissler, Gerrit Meixner, Peter Forbrig, Ahmed Sef-fah, and Kerstin Klöckner. 2010. Pattern-driven engineering of interactive computing systems (PEICS). In *Proceedings of the 2nd ACM SIGCHI symposium on Engineering interactive computing systems (EICS '10)*. ACM, New York, NY, USA, 367-368. DOI: <http://dx.doi.org/10.1145/1822018.1822085>

- (2) **Novel interactive artifacts** include prototypes, software architectures, applications, and toolkits that contribute to the practice of engineering of interactive systems.

✍ **Example:** Jean-Luc Hak, Marco Winckler, and David Navarre. 2016. PANDA: prototyping using annotation and decision analysis In *Proceedings of the ACM SIGCHI symposium on Engineering interactive computing systems (EICS '16)*. ACM, New York, NY, USA, 171-176. <https://doi.org/10.1145/2933242.2935873>

✍ **Example:** Miriam Greis, Hendrik Schuff, Marius Kleiner, Niels Henze and Al-breicht Schmidt. 2017. Input Controls for Entering Uncertain Data: Probability Dis-tribution Sliders. *Proc. ACM Hum.-Comput. Interact.* 1, EICS, Article 3 (June 2017), 17 pages. DOI: <https://doi.org/10.1145/3095805>

- (3) **Formal methods** contribute approaches to specify, design, develop, verify, and validate the user interfaces of interactive systems. The topic of formal methods, located at the intersection of HCI and SE, is helpful for proving quality properties for interactive systems overall and safety-critical systems in particular.

✍ **Example:** Judy Bowen, Steve Reeves. 2017. Generating Obligations, Assertions and Tests from UI Models. *Proc. ACM Hum.-Comput. Interact.* 1, EICS, Article 5 (June 2017), 18 pages. DOI: <http://dx.doi.org/10.1145/3095807>

✍ **Example:** Philippe Palanque, Eric Barboni, Célia Martinie, David Navarre, and Marco Winckler. 2011. A model-based approach for supporting engineering usability evaluation of interaction techniques. In *Proceedings of the ACM SIGCHI Symposium on Engineering Interactive Computing Systems (EICS '11)*. ACM, New York, NY, USA, 21-30. DOI: <https://doi.org/10.1145/1996461.1996490>

- (4) **Design spaces and pattern spaces** cover the construction of design options in structured and systematic ways. A design space represents the combination of various *design dimensions* (e.g., user interface properties, high-level specifications, etc.) and *process parameters* that contribute to the systematic analysis of interactive systems and to their exploration. When a new design space is proposed, its systematic exploration falls under the specific area of interest called *Design Space Exploration* (DSE) [7], referring to the systematic analysis and pruning of design options based on the dimensions of interest. Design spaces are particularly useful to address the following three modeling virtues [3]: (1) *descriptive* (any interactive system

should be consistently described using the properties of the design space), (2) *comparative* (two or more systems can be compared using the properties of the design space), and (3) *generative* (the systematic comparison of existing interactive systems leads to identifying new, previously uncovered areas, under-represented design options, and potential avenues for future research and development).

✍ **Example:** Pragma Kar, Samiran Chattopadhyay, and Sandip Chabkraborty. 2020. Gestatten: Estimation of User's Attention in Mobile MOOCs From Eye Gaze and Gaze Gesture Tracking. *Proc. ACM Hum.-Comput. Interact.* 4, Article 72 (June 2020). DOI: <https://doi.org/10.1145/3394974>

- (5) **Conceptual modeling of interaction features** to abstractly represent the interaction. Model-to-Model and Model-to-Code transformation rules can be applied in a model-driven development paradigm. The conceptual modeling of interaction [1] remains an open challenge addressed not only by EICS, but which has recently raised interest in other communities, as well, such as the International Conference on Conceptual Modeling (ER).

✍ **Example:** Ivan Logre and Anne-Marie Déry-Pinna. 2018. MDE in Support of Visualization Systems Design: A Multi-Staged Approach Tailored for Multiple Roles. *Proc. ACM Hum.-Comput. Interact.* 2, EICS, Article 14 (June 2018), 17 pages. DOI: <https://doi.org/10.1145/3229096>

✍ **Example:** Marco Manca, Parvaneh Parvin, Fabio Paternò, and Carmen Santoro. 2017. Detecting anomalous elderly behaviour in ambient assisted living. In *Proceedings of the ACM SIGCHI Symposium on Engineering Interactive Computing Systems (EICS '17)*. ACM, New York, NY, USA, 63-68. <https://doi.org/10.1145/3102113.3102128>

- (6) **Empirical research in engineering interactive systems** contributes with findings gained by means of observation and data collection to create knowledge and inform new research and development.

✍ **Example:** Thomas Kunding, Celena Mayr, and Andreas Riener. 2020. Towards a Reliable Ground Truth for Drowsiness: A Complexity Analysis on the Example of Driver Fatigue. *Proc. ACM Hum.-Comput. Interact.* 4, EICS, Article 78 (June 2020). DOI: <https://doi.org/10.1145/3394980>

- (7) **Requirements elicitation and visualization of interaction features** to improve the usability and accessibility of interactive systems. Requirements that relate to interaction need to be considered from the earliest stages of the software development life-cycle.

✍ **Example:** Benoît Duhoux, Bruno Dumas, Hoo Sing Leung, and Kim Mens. 2019. Dynamic Visualisation of Features and Contexts for Context-Oriented Programmers. In *Proceedings of the ACM SIGCHI Symposium on Engineering Interactive Computing Systems (EICS '19)*. ACM, New York, NY, USA, 1-6. <https://doi.org/10.1145/3319499.3328240>

- (8) **Methodological contributions** describing the way in which the different facets of engineering interactive systems should be addressed. These contributions regard analysis, design, development, and measuring aspects that relate to the engineering of interaction.

✍ **Example:** José C. Campos, Camille Fayollas, Célia Martinie, David Navarre, Philippe Palanque, and Miguel Pinto. 2016. Systematic automation of scenario-based testing of user interfaces. In *Proceedings of the 8th ACM SIGCHI Symposium on Engineering Interactive Computing Systems (EICS '16)*. ACM, New York, NY, USA, 138-148. DOI: <https://doi.org/10.1145/2933242.2948735>

✍ **Example:** Alexandru Moldovan, Vlad Nicula, Ionut Pasca, Mihai Popa, Jaya Krishna Namburu, Anamaria Oros, and Paul Brie. 2020. OpenUIDL, a User Interface Description Language for Runtime Omni-Channel User Interfaces. *Proc. ACM Hum.-Comput. Interact.* 4, EICS, Article 2850 (May 2020), 30 pages. DOI: <https://doi.org/10.1145/3397874>

- (9) **Theoretical contributions in the engineering of interactive computing systems** consider novel concepts, definitions, models, principles, or frameworks.

✍ **Example:** Reinout Roels, Arno De Witte, and Beat Signer. 2018. INFEX: A Unifying Framework for Cross-Device Information Exploration and Exchange. *Proc. ACM Hum.-Comput. Interact.* 2, EICS, Article 2 (June 2018), 26 pages. DOI: <https://doi.org/10.1145/3179427>

- (10) **Survey contributions** include analysis and meta-analysis of developments and results on a specific topic in the form of systematic mapping studies, systematic literature reviews, and thematic synthesis. This type of contribution is particularly welcome since EICS has accumulated considerable knowledge, which is now in the need of consolidation. A recent survey [36] of EICS approaches revealed that the pool of entries is large and challenging.

✍ **Example:** Paulo Pinheiro Da Silva. 2000. User interface declarative models and development environments: a survey. In *Proceedings of the 7th international conference on Design, specification, and verification of interactive systems (DSV-IS'00)*, Philippe Palanque and Fabio Paternò (Eds.). Springer-Verlag, Berlin, Heidelberg, 207-226. DOI: https://doi.org/10.1007/3-540-44675-3_13

7 EVOLUTION OF EICS TOPICS OVER TIME

Since its first appearance in 2009, EICS has traditionally covered topics listed in Section 5, with more or less emphasis over time. In this section, we review some of the most frequent EICS topics over years and we suggest a few more topics for future editions.

7.1 Research Method

Our main research question is the following: what are the topics covered by papers published in EICS editions over years? Since the goal of our examination is to provide a consolidated overview of the EICS research, a Systematic Literature Review (SLR) is one typical research instrument to achieve this goal. Several SLR methods exist in the literature, among them is [20], a leading SLR method in software engineering. Such SLR methods are often aimed at extensively covering a large domain, like experiments in medicine [22]. A standard SLR procedure [20] was applied [9] to our main research question, based on the PRISMA (Preferred Reporting Items for Systematic reviews and Meta-Analyses) four-phase flow diagram [22], used in health care.

We selected the major Computer Science digital library where EICS paper were published from 2009 to 2019: ACM Digital Library². We retained all EICS papers written in English that underwent a peer-review process, and for which the full text was available. These included research papers (full papers, short papers, extended abstracts, posters, keynote papers, doctoral consortium papers) published in the peer-reviewed proceedings and journals, but not the associated symposiums, workshops, and reports.

We employed the following tools to create a collection of papers and generate summary statistics:

- (1) ZOTERO,³ a multi-platform bibliography management software tool use for managing our corpus of papers.
- (2) PAPERMACHINES,⁴ a Zotero extension for visualization of results.
- (3) PDF2TEXT,⁵ software for automatic extraction of text from PDF files. The extracted text was submitted to automatic language processing and analysis.

All the words contained in the metadata of each paper, such as in their title, abstract, internal and external keywords, and all the terms contained in their corresponding PDF papers were extracted to feed a database of indexed keywords and tags. A visualization tool exploits this database to produce several graphics and to browse the corpus along several structural dimensions, such as year, keyword, area, author, etc.

7.2 Results

In this section, we report the most significant observations made on the material resulting from the aforementioned process. Detailed figures are provided in respective appendices.

7.2.1 Large Word Clouds. To determine important terms in the corpus, we built two global word clouds: one based on Simple Term Frequency (see Fig. 2) and one based on Term Frequency-Inverse Document Frequency (TF-IDF)⁶ (see Fig. 3), the most frequently applied weighting scheme in recommender systems [5].

Besides discipline-oriented common terms found in these clouds (e.g., software, engineering, development, application, implementation, interaction), other terms suggest that “User (s)” is the most frequent term (“human” to a lesser extent), which reveals the wish to follow a user-centered approach in all circumstances, even if the target audience aims at software engineers, developers, and other stakeholders of the development team, with the ultimate benefit going to the end users of the application. As for any software development life cycle (SDLC), different phases concerning the user interface in particular or concerning the whole interactive application/system in general are covered: primarily the analysis and design phases, the programming/coding activity at large (e.g., via object-oriented programming, component-based approaches, but less so other phases).

To support these phases of the SDLC, “model(s)” is a pre-eminent term and is often referred to as a simplified characterisation, representation, often more formal than semi-formal or informal, that enables reasoning about the user interface and its related interactive application. Typical reasoning activities involve formal methods (see [42] for the most recent volume on this aspect), such as validation and verification of properties, which is particularly vital for complex systems [29], safety-critical systems [30], very large systems[40], (semi-)automated generation and/or evaluation/testing of user interfaces. Potential benefits from relying on a model-based approach include [27]: benefits

²<https://dlnext.acm.org>

³<http://www.zotero.org/>

⁴<http://www.papermachines.org/>

⁵<http://www.pdf2text.com/>

⁶This frequency is calculated as the product of the simple term frequency by the inverse term frequency, where the last factor is the logarithm of the ratio “total number of references where the term appears” by the number of references containing the term

various instruments, such as frameworks, design space, toolkits, etc. Rapid prototyping and composition of widgets are also popular themes. Rapid prototyping or prototyping of any kind of user interface, not just graphical, but also multimodal and gesture-based, remain constant.

Although Artificial Intelligence (AI) is more relevant to the IUI conference, as explained in Section 2, it appears for the first time in 2011 with various techniques, ranging from rule-based systems to language engineering and intelligent systems.

Model checking is particularly well addressed during the 2012 year, in particular for web interfaces. From 2015 and onward, gesture-based interfaces become a particular topic of interest, in particular due to the rise of sensors like the Microsoft Kinect for full-body gesture acquisition and recognition. All these themes remain constantly addressed with more advanced questions, with some peak on input devices, techniques, software architectures in 2017.

7.3 Discussion and Suggestions

Based on the aforementioned observations, we may formulate a series of suggestions for current and future topics, such as, but not limited to:

- (1) Consider other phases of the software development life cycle.

 **Suggestion:** Consider papers dealing with less covered development phases, such as deployment, training/tutoring, testing (evaluation is considered as a part of it), maintenance, but especially all aspects relevant to requirement engineering (e.g., requirements gathering, elicitation, formalization), a phase that is totally absent.

- (2) Consider other quality factors and sub-factors of user interfaces.

 **Suggestion:** Consider papers dealing with quality factors other than usability, as defined in the ISO 25010 standard⁷ on software quality: functional stability, performance efficiency, compatibility, reliability, security, maintainability, and portability. In this standard, the usability factor is itself decomposed into several sub-factors which also deserve specific treatments: appropriateness, recognizability, learnability, operability, user error protection, user interface aesthetics, and accessibility. Note that these two last sub-factors are sometimes covered in EICS papers.

- (3) Consider explicitly the dimensions of the context of use.

 **Suggestion:** Consider papers mentioning or addressing explicitly a particular context of use, as opposed to generic aspects of users, platforms, and environments. The environmental dimension should be better addressed as it is often left untouched in context-aware user interfaces. The environment could cover the psychological, physical, and organisational aspects of the environment in which the end users are carrying out their tasks, and not just the location. Smart environments, like smart homes, smart factories, smart cities are often published in other venues regarding their technical aspects, but not yet much in EICS regarding their user interfaces.

- (4) Consider high levels of data abstraction in tasks.

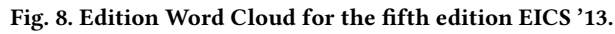
 **Suggestion:** Consider papers addressing higher levels of abstraction regarding data manipulated by end users in their tasks. While “data” and “information” are addressed, “knowledge” and “wisdom”, which are located on top of data and information, are not explicitly handled.

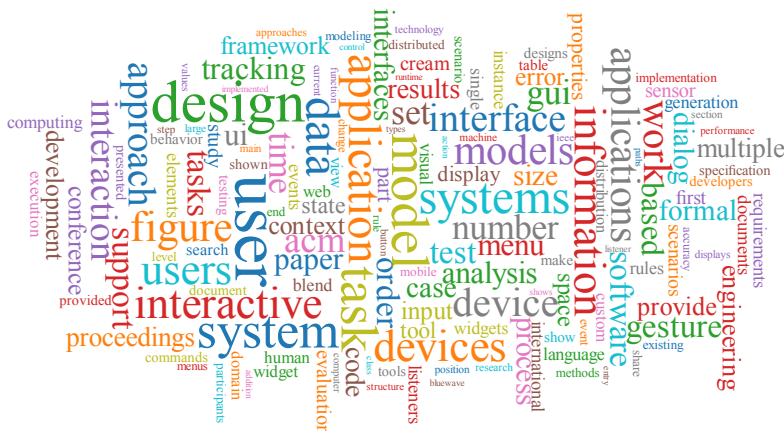
- (5) Consider programming activities of user interfaces more explicitly.

✍ **Suggestion:** Consider papers explaining how to effectively and efficiently program a particularly difficult or challenging user interface, such as those user interfaces with advanced modalities (*e.g.*, tangible interaction, gesture-based interface, augmented reality). For the moment, the badges for artifact review defined by the Association for Computing Machinery⁸ are not yet activated for EICS papers, but they are very appropriate and might become applicable in a short-term future. Without waiting for this activation, it is time to produce user interface code that satisfies these properties, such as “Artifacts Evaluated – Functional”, “Artifacts Evaluated – Reusable”, and “Artifacts Available”. Any support for these properties should be better appreciated, such as code availability on GitHub repositories, online documentation, running examples, etc.

- (6) Consider user interfaces for Internet-of-Things, smart environments, and Cyber-Physical Systems more profoundly.

✍ **Suggestion:** As the development of IoT and software for smart environments increases in prevalence the importance of understanding user needs for such systems suggest the need for engineering principles specific to this domain. This includes the ability for users to set up and control smart systems (*e.g.*, in the manner proposed by Manca et al. [26]) as well as understand the effects on users of living in, and interacting with, smart environments [8]. Likewise, increases in AI-based software brings with it the necessity for users to understand and control such software. This requires methodologies to ensure such requirements are considered fully during the development process [43].





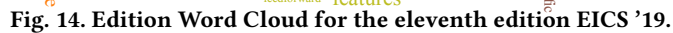
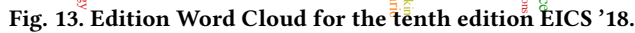
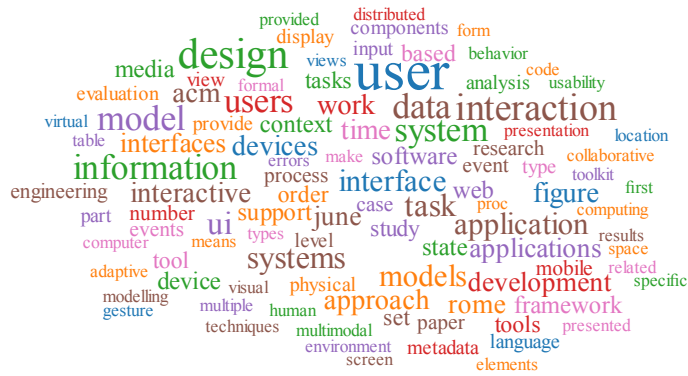
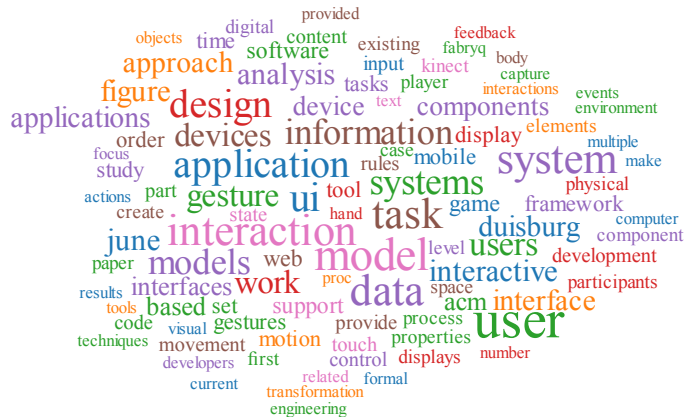
[illegible]

Fig. 15. Chronological Word Cloud for the first period.



2013/07/14-2014/07/14

Fig. 19. Chronological Word Cloud for the fifth period.



2014/07/14-2015/07/14

Fig. 20. Chronological Word Cloud for the sixth period.



2015/07/14-2016/07/13

Fig. 21. Chronological Word Cloud for the seventh period.



Fig. 22. Chronological Word Cloud for the eighth period.

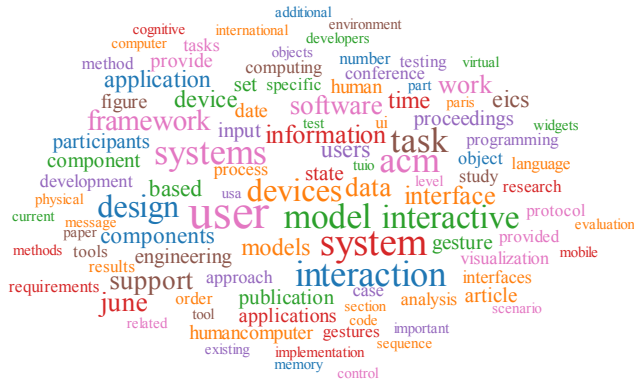
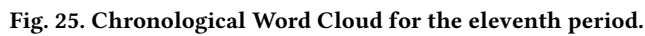


Fig. 23. Chronological Word Cloud for the ninth period.



Fig. 24. Chronological Word Cloud for the tenth period.



This section reproduces chronological word clouds obtained by similarity computed by Mann-Whitney correlation.





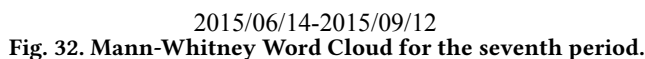
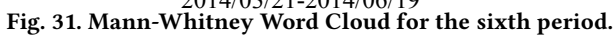
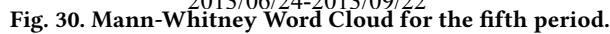




Fig. 36. Mann-Whitney Word Cloud for the eleventh period.



Fig. 37. Mann-Whitney Word Cloud for the last period with links to previous editions. Note that these keywords are greyed because not all EICS '20 papers are yet included in this word cloud.

APPENDIX B. PHRASENET APPLICATIONS

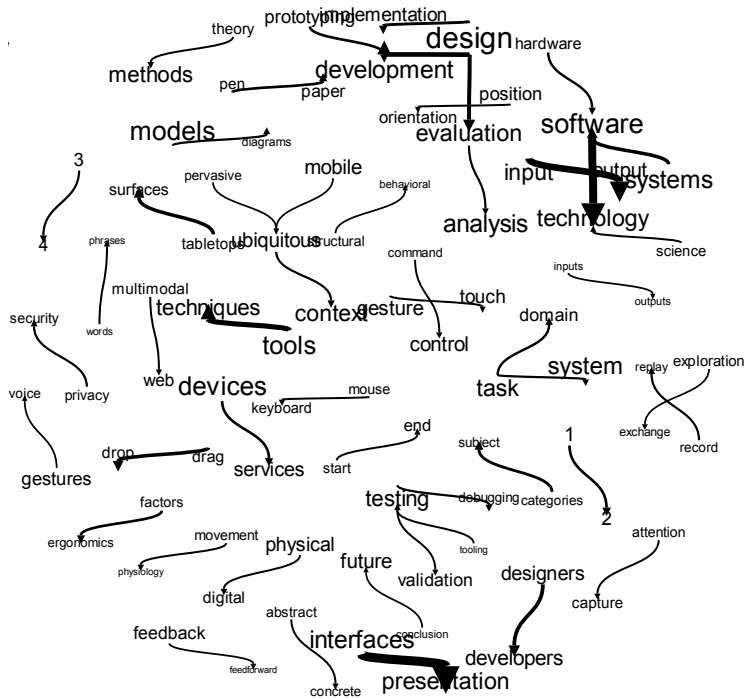


Fig. 38. PhraseNet application on conjunction: X and Y.

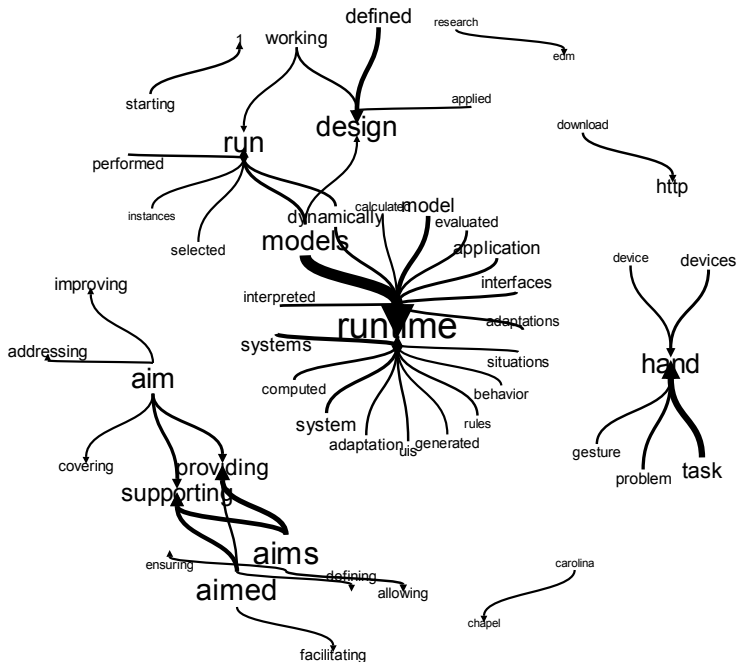


Fig. 39. PhraseNet application on determinant: X at Y.

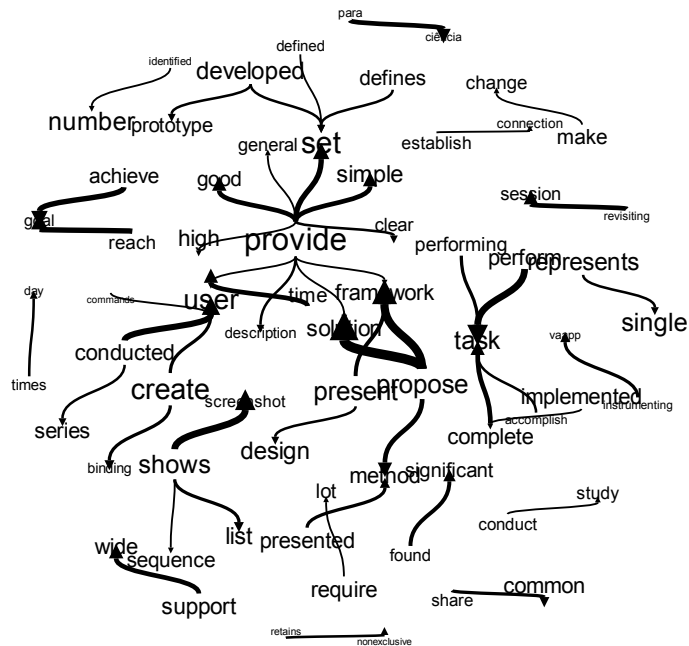
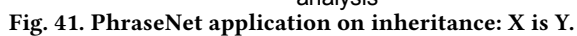


Fig. 40. PhraseNet application on expression: X a Y.



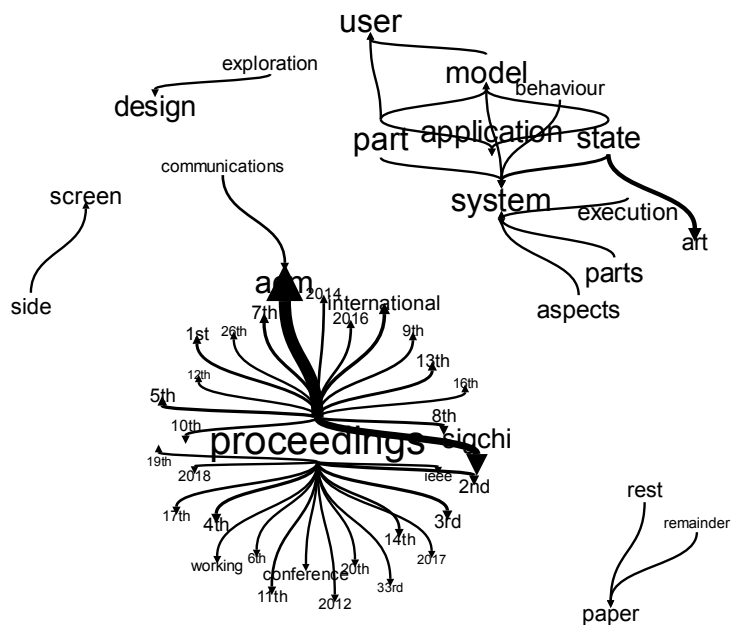


Fig. 42. PhraseNet application on determinant: X of the Y.

APPENDIX C. TOPIC MODELLING

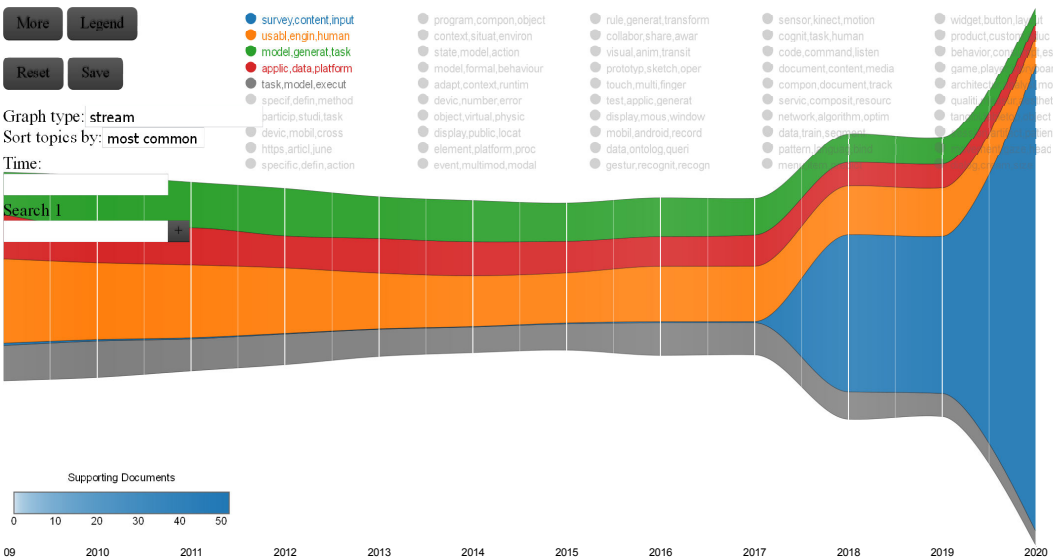
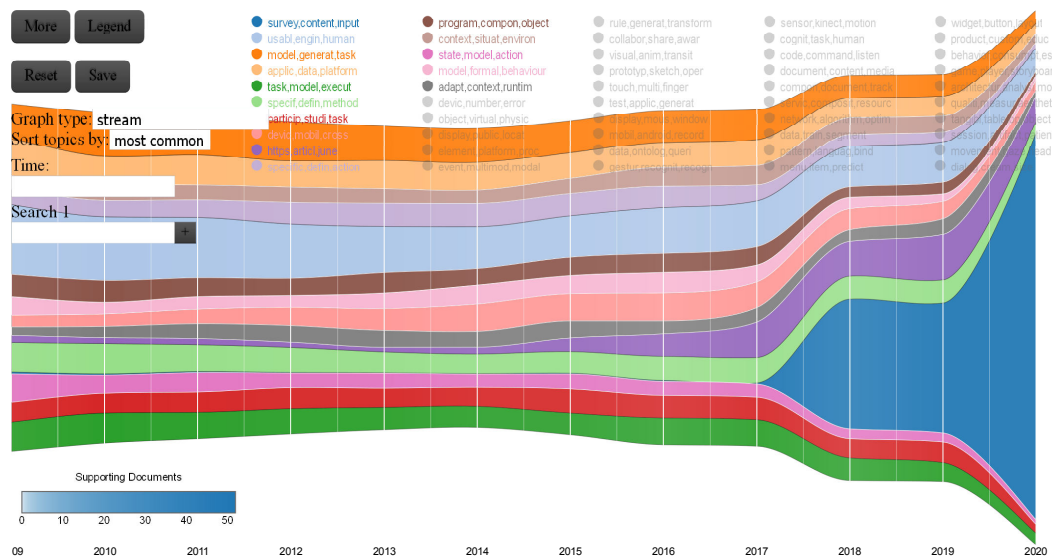
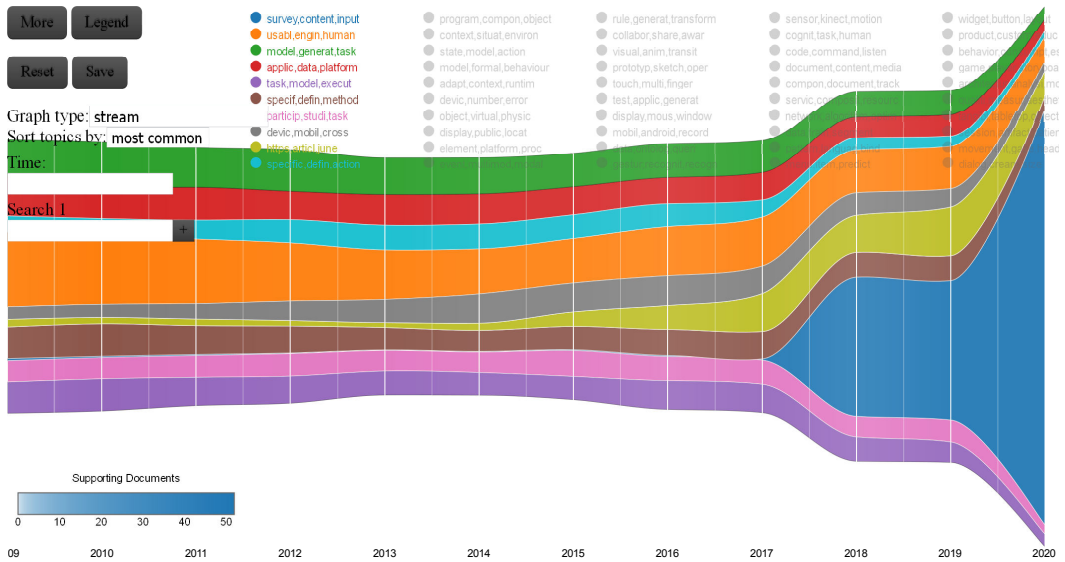


Fig. 43. EICS Topic Modelling: the main streams.



ACKNOWLEDGMENTS

We acknowledge the support of *University Côte d’Azur*⁹, Polytech Nice, and the *French National Centre for Scientific Research*¹⁰ as the main sponsors of EICS 2020. We would like to thank the Chairs of all the tracks of the EICS 2020 conference, the members of the local organization team, and the web master of the EICS 2020 web site. The planning for EICS 2020 would not have been possible without the commitment of the Programme Committee members and external reviewers, who did a particularly challenging work in spite of the COVID-19 crisis.

⁹<http://www.univ-cotedazur.fr/>

¹⁰ <http://www.cnrs.fr/en/cnrs/>

REFERENCES

- [1] Nathalie Aquino, Jean Vanderdonckt, Jose Ignacio Panach, and Oscar Pastor. 2011. *Conceptual Modelling of Interaction*. Springer, Berlin, Heidelberg, 335–358. https://doi.org/10.1007/978-3-642-15865-0_10
- [2] Rémi Bastide, Philippe A. Palanque, and Jörg Roth (Eds.). 2005. *Proceedings of Joint Working Conferences on Engineering Human Computer Interaction and Interactive Systems (EHCI-DSVIS '04, Hamburg, Germany, July 11-13, 2004)*. Lecture Notes in Computer Science, Vol. 3425. Springer, Berlin.
- [3] Michel Beaudouin-Lafon. 2004. Designing Interaction, Not Interfaces. In *Proceedings of the Working Conference on Advanced Visual Interfaces (AVI '04)*. ACM, New York, NY, USA, 15–22. <https://doi.org/10.1145/989863.989865>
- [4] Michel Beaudouin-Lafon. 2004. Designing Interaction, Not Interfaces. In *Proceedings of the Working Conference on Advanced Visual Interfaces (AVI '04)*. Association for Computing Machinery, New York, NY, USA, 15–22. <https://doi.org/10.1145/989863.989865>
- [5] Joeran Beel, Bela Gipp, Stefan Langer, and Corinna Breiter. 2016. Research-paper recommender systems: a literature survey. *International Journal on Digital Libraries* 17, 4 (01 Nov 2016), 305–338. <https://doi.org/10.1007/s00799-015-0156-0>
- [6] François Bodart and Jean Vanderdonckt (Eds.). 1996. *Proceedings of the Third International Eurographics Workshop on Design, Specification and Verification of Interactive Systems (DSV-IS '96)*, June 5-7, 1996, Namur, Belgium. Springer. <https://doi.org/10.1007/978-3-7091-7491-3>
- [7] Judy Bowen and Anke Dittmar. 2016. A Semi-Formal Framework for Describing Interaction Design Spaces. In *Proceedings of the 8th ACM SIGCHI Symposium on Engineering Interactive Computing Systems (EICS '16)*. Association for Computing Machinery, New York, NY, USA, 229–238. <https://doi.org/10.1145/2933242.2933247>
- [8] Judy Bowen and Annika Hinze. 2019. Smarter Software Engineering Methods for Smart Environments. In *Joint Proceedings HCI Engineering 2019 - Methods and Tools for Advanced Interactive Systems and Integration of Multiple Stakeholder Viewpoints co-located with 11th ACM SIGCHI Symposium on Engineering Interactive Computing Systems (EICS 2019)*, Valencia, Spain, June 18, 2019. 3–9. http://ceur-ws.org/Vol-2503/paper1_1.pdf
- [9] Pearl Brereton, Barbara A. Kitchenham, David Budgen, Mark Turner, and Mohamed Khalil. 2007. Lessons from applying the systematic literature review process within the software engineering domain. *Journal of Systems and Software* 80, 4 (2007), 571–583. <https://doi.org/10.1016/j.jss.2006.07.009>
- [10] Gaëlle Calvary, Joëlle Coutaz, David Thevenin, Quentin Limbourg, Laurent Bouillon, and Jean Vanderdonckt. 2003. A Unifying Reference Framework for multi-target user interfaces. *Interacting with Computers* 15, 3 (6 2003), 289–308. [https://doi.org/10.1016/S0953-5438\(03\)00010-9](https://doi.org/10.1016/S0953-5438(03)00010-9)
- [11] Antonio Delgado, Antonio Estepa, J.A. Troyano, and Rafael M. Estepa. 2016. Reusing UI elements with Model-Based User Interface Development. *International Journal of Human-Computer Studies* 86 (2016), 48–62. <https://doi.org/10.1016/j.ijhcs.2015.09.003>
- [12] David W. Embley, Stephen W. Liddle, and Oscar Pastor. 2011. *Conceptual-Model Programming: A Manifesto*. Springer Berlin Heidelberg, Berlin, Heidelberg, 3–16. https://doi.org/10.1007/978-3-642-15865-0_1
- [13] David England, Philippe A. Palanque, Jean Vanderdonckt, and Peter J. Wild (Eds.). 2010. *Proceedings of the 8th International Workshop on Task Models and Diagrams for User Interface Design (TAMODIA '09)*, Brussels, Belgium, September 23-25, 2009. Lecture Notes in Computer Science, Vol. 5963. Springer, Berlin. <https://doi.org/10.1007/978-3-642-11797-8>
- [14] José A. Gallud, Victor M. Ruiz Penichet, María Dolores Lozano, Ricardo Tesoriero, Jean Vanderdonckt, and Habib M. Fardoun. 2016. Proc. of 5th Workshop on Distributed User Interfaces: Distributed Interactions DUI '16. In *ICWE Workshops (Lecture Notes in Computer Science)*, Sven Casteleyn, Peter Dolog, and Cesare Pautasso (Eds.), Vol. 9881. Springer, Berlin, 157–207.
- [15] Jose A. Gallud, Ricardo Tesoriero, Jean Vanderdonckt, María Lozano, Victor Penichet, and Federico Botella. 2011. Distributed User Interfaces. In *CHI '11 Extended Abstracts on Human Factors in Computing Systems (CHI EA '11)*. Association for Computing Machinery, New York, NY, USA, 2429–2432. <https://doi.org/10.1145/1979742.1979576>
- [16] T. C. Nicholas Graham and Philippe A. Palanque (Eds.). 2008. *Proceedings of the 15th International Workshop on Design, Specification, and Verification of Interactive Systems (DSV-IS '08)*, Kingston, Canada, July 16-18, 2008. Lecture Notes in Computer Science, Vol. 5136. Springer, Berlin.
- [17] T.C. Nicholas Graham, Gaëlle Calvary, and Philip D. Gray (Eds.). 2009. *Proceedings of the 1st ACM SIGCHI Symposium on Engineering Interactive Computing System (EICS '09, Pittsburgh, PA, USA, July 15-17, 2009)*. ACM, New York, NY, USA. <https://dl.acm.org/citation.cfm?id=1570433>
- [18] Wayne D. Gray, William E. Hefley, William E. Hefley, and Dianne Murray (Eds.). 1993. *Proceedings of the 1st ACM International Conference on Intelligent User Interfaces* (January). Association for Computing Machinery, New York, NY, USA. <https://dl.acm.org/doi/proceedings/10.1145/169891>
- [19] Jan Gulliksen, Morten Borup Harning, Philippe A. Palanque, Gerrit C. van der Veer, and Janet Wesson (Eds.). 2008. *Proceedings of Joint Working Conference on Engineering on Interactive Systems (EIS '07), joining EHCI '07, DSV-IS '07, HCSE '07, Salamanca, Spain, March 22-24, 2007*. Lecture Notes in Computer Science, Vol. 4940. Springer, Berlin.

- [20] Barbara Kitchenham, Riallette Pretorius, David Budgen, O. Pearl Brereton, Mark Turner, Mahmood Niazi, and Stephen Linkman. 2010. Systematic literature reviews in software engineering – A tertiary study. *Information and Software Technology* 52, 8 (2010), 792–805. <https://doi.org/10.1016/j.infsof.2010.03.006>
- [21] Xin Li, Dan Roth, and Yuancheng Tu. 2003. PhraseNet: Towards Context Sensitive Lexical Semantics. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003 - Volume 4 (CONLL '03)*. Association for Computational Linguistics, Stroudsburg, PA, USA, 87–94. <https://doi.org/10.3115/1119176.1119188>
- [22] Alessandro Liberati, Douglas G. Altman, Jennifer Tetzlaff, Cynthia Mulrow, Peter C. Gøtzsche, John P.A. Ioannidis, Mike Clarke, P. J. Devereaux, Jos Kleijnen, and David Moher. July 2009. The PRISMA statement for reporting systematic reviews and meta-analyses of studies that evaluate health care interventions: explanation and elaboration. *PLoS Medicine* 6, 7 (July 2009), 1–22. <https://doi.org/10.1371/journal.pmed.1000100>
- [23] Víctor López-Jaquero, Francisco Montero Simarro, José Pascual Molina Massó, and Jean Vanderdonckt (Eds.). 2009. *Proceedings of the Seventh International Conference on Computer-Aided Design of User Interfaces (CADUI '08)*, Albacete, Spain, June 11–13, 2008. Springer, Berlin. <https://doi.org/10.1007/978-1-84882-206-1>
- [24] María Dolores Lozano, José A. Gallud, Ricardo Tesoriero, Victor M. Ruiz Penichet, Jean Vanderdonckt, and Habib Fardoun. 2013. 3rd workshop on distributed user interfaces: models, methods and tools. In *EICS*. ACM, New York, NY, USA, 335–336. <http://dwi-workshop.org/2013/>
- [25] María Dolores Lozano, Jean Vanderdonckt, Habib M. Fardoun, Ricardo Tesoriero, José A. Gallud, and Victor M. Ruiz Penichet (Eds.). 2014. *Proceedings of the 2014 Workshop on Distributed User Interfaces and Multimodal Interaction (DUI'14)*, Toulouse, France, July 1, 2014. ACM, New York, NY, USA.
- [26] Marco Manca, Fabio Paternò, Carmen Santoro, and Luca Corcella. 2019. Supporting end-user debugging of trigger-action rules for IoT applications. *Int. J. Hum. Comput. Stud.* 123 (2019), 56–69. <https://doi.org/10.1016/j.ijhcs.2018.11.005>
- [27] Gerrit Meixner, Gaëlle Calvary, and Joëlle Coutaz. 2014. Introduction to Model-Based User Interfaces - W3C Working Group Note. (2014). <https://www.w3.org/TR/mbui-intro/>
- [28] Gerrit Meixner, Fabio Paternò, and Jean Vanderdonckt. 2011. Past, Present, and Future of Model-Based User Interface Development. *i-com* 10, 3 (2011), 2–11. <https://doi.org/10.1524/icom.2011.0026>
- [29] Philippe Palanque, Eric Barboni, Célia Martinie, David Navarre, and Marco Winckler. 2011. A Model-Based Approach for Supporting Engineering Usability Evaluation of Interaction Techniques. In *Proceedings of the 3rd ACM SIGCHI Symposium on Engineering Interactive Computing Systems (EICS '11)*. Association for Computing Machinery, New York, NY, USA, 214–230. <https://doi.org/10.1145/1996461.1996490>
- [30] Philippe A. Palanque, Sandra Basnyat, Regina Bernhaupt, Ronald L. Boring, Chris W. Johnson, and Peter Johnson. 2007. Beyond usability for safety critical systems: how to be sure (safe, usable, reliable, and evolvable)? In *Extended Abstracts Proceedings of the 2007 Conference on Human Factors in Computing Systems, CHI 2007, San Jose, California, USA, April 28 - May 3, 2007*, Mary Beth Rosson and David J. Gilmore (Eds.). ACM, 2133–2136. <https://doi.org/10.1145/1240866.1240966>
- [31] Philippe A. Palanque and Rémi Bastide (Eds.). 1995. *Proceedings of the Second Eurographics Workshop on Design, Specification and Verification of Interactive Systems (DSV-IS '95)*, Toulouse, France, June 7–9, 1995. Springer. <https://doi.org/10.1007/978-3-7091-9437-9>
- [32] José Ignacio Panach, Jean Vanderdonckt, and Oscar Pastor (Eds.). 2019. *Proceedings of the ACM SIGCHI Symposium on Engineering Interactive Computing Systems, EICS 2019, Valencia, Spain, June 18–21, 2019*. ACM, New York, NY, USA. <https://dl.acm.org/citation.cfm?id=3319499>
- [33] Fabio Paternò (Ed.). 1994. *Proceedings of the First International Eurographics Workshop on Design, Specification and Verification of Interactive Systems (DSV-IS '94)*, Bocca di Magra, Italy, June 8–10, 1994. Springer, Berlin.
- [34] Costin Pribeanu and Jean Vanderdonckt (Eds.). 2002. *Proceedings of the First International Workshop on Task Models and Diagrams for User Interface Design (TAMODIA '02)*, 18–19 July 2002, Bucharest, Romania. INFOREC Publishing House, Bucharest. <https://dl.acm.org/citation.cfm?id=646617>
- [35] P. Riehmann, M. Hanfler, and B. Froehlich. 2005. Interactive Sankey diagrams. In *IEEE Symposium on Information Visualization, 2005. INFOVIS 2005*. 233–240. <https://doi.org/10.1109/INFVIS.2005.1532152>
- [36] Jenny Ruiz, Estefanía Serral, and Monique Snoeck. 2018. Evaluating user interface generation approaches: model-based versus model-driven development. *Software & Systems Modeling* (17 Oct 2018). <https://doi.org/10.1007/s10270-018-0698-x>
- [37] Ricardo Tesoriero, María Dolores Lozano, Jean Vanderdonckt, José A. Gallud, and Victor M. Ruiz Penichet. 2012. Distributed user interfaces: collaboration and usability. In *CHI Extended Abstracts*. ACM, New York, NY, USA, 2719–2722. <http://dwi-workshop.org/2012/>
- [38] Jean Vanderdonckt (Ed.). 1996. *Proceedings of the Second International Workshop on Computer-Aided Design of User Interfaces (CADUI '96)*, Namur, Belgium, June 5–7, 1996. Presses Universitaires de Namur, Namur.
- [39] Jean Vanderdonckt. 2005. A MDA-Compliant Environment for Developing User Interfaces of Information Systems. In *Advanced Information Systems Engineering*, Oscar Pastor and João Falcão e Cunha (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 16–31. https://doi.org/10.1007/11431855_2

- [40] Jean Vanderdonckt and Pierre Berquin. 1999. Towards a Very Large Model-Based Approach for User Interface Development. In *1999 User Interfaces to Data Intensive Systems, UIDIS 1999, Edinburgh, Scotland, September 5-6, 1999*, Norman W. Paton and Tony Griffiths (Eds.). IEEE Computer Society, 76–85. <https://doi.org/10.1109/UIDIS.1999.791464>
- [41] Jean Vanderdonckt, Marco Winckler, and Judy Bowen (Eds.). 2020. *Proceedings of the 12th ACM SIGCHI Symposium on Engineering Interactive Computing System (EICS '20, Nice, France, June 23-26, 2020)*. ACM, New York, NY, USA.
- [42] Benjamin Weyers, Judy Bowen, Alan Dix, and Philippe A. Palanque (Eds.). 2017. *The Handbook of Formal Methods in Human-Computer Interaction*. Springer International Publishing. <https://doi.org/10.1007/978-3-319-51838-1>
- [43] Jürgen Ziegler. 2019. Challenges in User-Centered Engineering of AI-based Interactive Systems. In *Joint Proceedings HCI Engineering 2019 - Methods and Tools for Advanced Interactive Systems and Integration of Multiple Stakeholder Viewpoints co-located with 11th ACM SIGCHI Symposium on Engineering Interactive Computing Systems (EICS 2019), Valencia, Spain, June 18, 2019*. 51–55. http://ceur-ws.org/Vol-2503/paper1_8.pdf

Received March 2020; revised April 2020; accepted May 2020