algorithms

MDPI

Article

A Multi-Process System for Investigating Inclusive Design in User Interfaces for Low-Income Countries

Yann Méhat [1,2], Sylvain Sagot [2,3], Egon Ostrosi [2,*] and Dominique Deuff [1]

[1] Orange Innovation, 2 Av. Pierre Marzin, 22300 Lannion, France; yann.mehat@utbm.fr (Y.M.); dominique.deuff@orange.com (D.D.)
[2] ELLIADD-ERCOS, Université de Bourgogne Franche-Comté, UTBM, 90010 Belfort, France; ssagot@esta-groupe.fr
[3] ESTA Belfort, School of Business and Engineering, 3 Rue du Dr Frery, 90000 Belfort, France
* Correspondence: egon.ostrosi@utbm.fr

Abstract: Limited understanding exists regarding the methodologies behind designing interfaces for low-income contexts, despite acknowledging their potential value. The ERSA (Engineering design Research meta-model based Systematic Analysis) process, defined as a dynamic interactive multi-process system, proposes a new approach to constructing learnings to succeed in designing interfaces for low-income countries. ERSA is developed by integrating database searches, snowballing, thematic similarity searches for corpus of literature creation, multilayer networks, clustering algorithms, and data processing. ERSA employs an engineering design meta-model to analyze the corpus of literature, facilitating the identification of diverse methodological approaches. The insights from ERSA empower researchers, designers, and engineers to tailor design methodologies to their specific low-income contexts. Our findings show the importance of adopting more versatile and holistic approaches. They suggest that user-based design methodologies and computational design can be defined and theorized together.

Keywords: interface design; inclusive user interfaces; network; intelligent process; modeling; conceptual models; computational models

check for updates

Citation: Méhat, Y.; Sagot, S.; Ostrosi, E.; Deuff, D. A Multi-Process System for Investigating Inclusive Design in User Interfaces for Low-Income Countries. *Algorithms* **2024**, *17*, 232. https://doi.org/10.3390/a17060232

Academic Editor: Frank Werner

Received: 14 April 2024
Revised: 9 May 2024
Accepted: 20 May 2024
Published: 27 May 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

While the design of user interfaces is a field driven by people of Western origin and culture [1], most users of IT tools come from much more diverse backgrounds and cultures. This gap can give rise to ergonomic problems in the way interfaces are used. The Western technological and cultural context is reflected in interfaces, which may be much less suited to use in other technological contexts and cultures [2]. Certain criteria—such as low literacy rate, low level of computer literacy, the high prevalence of local languages (often oral) compared with national languages, and the lack of access to IT tools—are specific to low-income countries and make it more difficult to master these interfaces.

A striking example of these issues appears in an experiment of an interactive voice response service. Khullar et al. [3] have identified usability issues by rural users of India in using keypress-based interfaces. While their vocal counterparts still have limitations, the mental load of manipulating both verbal and written numbers was too high for some low-literate users, and they were confused on how to act. Moreover, the keypad of their phones could be so worn out that the numbers were barely legible. Finally, a date of birth question was asked using the English calendar, unbeknownst to the Hindi culture users who are more familiar with the Hindi calendar.

Knowledge about the end users and their context of interaction is therefore mandatory for designing a suitable user interface. Since a user interface is an abstraction designed for the users, it is very important to understand users' mental models and the capabilities given by their devices. User-Centered Design (UCD) is a framework that has such a goal, by

putting the knowledge of the end user at the forefront during the design work [4]. However, the distance, both physically and culturally, between teams in charge of the development of such IT tools and the target users is so high that the promise of the UCD framework is never quite reached. Physically, the distance between the users makes it difficult for designers to understand end users' living context [5–7]. Culturally, since their living context is so distant from the one usually experienced by the development team, it would require a thorough study to understand it in detail [5,7–13]. However, such a study would require time and resources, being even more costly to the development of the system. Moreover, users sometimes have little experience with technology, which may need an adaptation of the tools and techniques usually used [14]. The UCD framework is therefore hard to apply to such projects, as experienced in the literature previously cited.

User-Centered Design provides no guidelines regarding the application of a UCD methodology within a developing world design context, underscoring an existing research gap on this topic [10]. Moreover, authors state that while existing user-centered analysis tools and techniques have proven to be useful and applicable in developing world design initiatives, no study has focused on the development of guidelines or frameworks within UCD that address developing world specific issues.

Recently, some researchers have begun to address this problem using different approaches. Tyukala [15] has interest in the user requirements part of the design methodology, by using the model of participatory design. This contribution proposes a methodology that allowed a community of Southern African users to come together and find common ground on their requirements for information systems. Unfortunately, it has not yet been tested in other communities. Adapting the participatory design methodology, in this case to the Ethiopian community, is another proposed strategy [16]. The authors highlight that what could work in this specific context may not work in other low-income countries, and that this participatory design "should include an appropriation of the design process itself to new cultural contexts". Maunder [17] studies specifically the suitability of the UCD framework in low-income countries. The author concludes that the empathic approach of the UCD framework is not ideal, whereas the usage of a technology probe is more suitable. Puri et al. [18] identifies that due to the very different nature of the context and specificities of each use case or region, "there is no single algorithmic best practice regarding participatory design in information systems which is applicable to all situations." This study details three interventions in three different regions and contexts and explains how the design team is able to include the end users in the design methodologies. Winschiers-Theophilus [19] focuses on the adaptation of design tools to include cultural specificities of the target user group. For example, in a workshop involving users of Namibian culture, the authors highlighted that the Namibian culture is a high collectivistic culture, using Hofstede's [20] cultural model. Considering this characteristic, the users have tested the proposed service in small groups instead of one-on-one testing sessions. The session was also started by directly asking the users what their definition of a "qualitative" system was. It allows the authors to understand what is valuable to the users and then focus the subsequent evaluation on those aspects.

While beneficial in their respective design projects, contexts of work, or specific research domains, existing methodologies have drawbacks and cannot be universally applied in low-income countries. The pertinent research question then arises: *Can we identify new patterns, trends, and paradigms that hold promise for application in interface design for low-income countries?*

In response to this research question, this paper proposes the development of the Engineering design Research meta-model based Systematic Analysis (ERSA) process, which was developed in collaboration between the ERCOS research Pole of the ELLIADD laboratory at UTBM and Orange Innovation. ERSA is a new process for acquiring learnings to succeed in designing interfaces for low-income countries.

An engineering design meta-model proposed by Ostrosi et al. [21] is used as a tool for analyzing a collection of research papers to extract valuable insights and identify emerging

patterns in design methodologies. A meta-model framework provides a structured and systematic approach to analyzing research papers. It allows researchers to organize and categorize information, making it easier to identify commonalities, differences, and gaps in existing design methodologies. It involves multiple interconnected models distributed in layers, providing a holistic understanding of the design process and its implications. Clustering algorithms within the meta-model framework enables the grouping of similar methodologies or approaches. This clustering facilitates the identification of trends and patterns across different studies, leading to a deeper understanding of the design of interfaces. It also streamlines data processing techniques, allowing researchers to extract relevant information efficiently and accurately. By employing a meta-model framework, we can identify and evaluate a wide range of design methodologies, enabling the development of more tailored and effective solutions. By understanding the strengths and limitations of different approaches, we can adapt and customize methodologies to suit specific contexts of low-income countries, ultimately leading to more impactful and sustainable solutions. It also can facilitate the transfer of knowledge and best practices across different multidisciplinary teams. By synthesizing and organizing information from diverse sources, it helps bridge the gap between theory and practice, fostering collaboration and innovation in the field of interface design.

This article is organized as follows: Section 2 presents the engineering design meta-model for interface design. Section 3 outlines the construction of the proposed ERSA (Engineering design Research meta-model based Systematic Analysis). Section 4 applies and examines the results of the ERSA process concerning interface methodologies adapted for users in low-income countries. Section 5 discusses the results and findings. Section 6 concludes and explores research perspectives.

2. Engineering Design Meta-Model for Interface Design

In this research, a multimodal user interface is considered and conceived as an evolving living product in complex interaction with its users. The engineering design meta-model proposed by Ostrosi et al. [21] is used and adopted for interface design to categorize the design methodologies employed in the selected articles. This meta-model provides a framework for the analysis design process of products. This section introduces a novel application of the engineering design meta-model for the design of user interfaces. Moreover, a network-based formalization is presented to establish the formal basis for the ERSA process.

2.1. Engineering Design Domains

To systemize the engineering design knowledge for the inclusive design of a multimodal user interface, the concept of domains is used. A domain in engineering design interface refers to a specific aspect or area of the design problem. It encompasses different kinds of design activities and represents a distinct subset of knowledge. The world of design modeling of a multimodal user interface as a living product can be defined in four domains: (i) *user domain*, (ii) *function domain,* (iii) *physical domain*, and (iv) *process or task domain*.

User domain. In interface mobile phone design, user domain refers to the full range of user characteristics and needs that designers must consider when creating interfaces. This includes but is not limited to users' physical and cognitive abilities, cultural and linguistic backgrounds, technical skills, and goals and motivations related to using an interface on a mobile device. Overall, the user domain is a complex and multifaceted concept that requires designers to consider a wide range of user characteristics, needs, preferences, and desires in order to create interfaces that are inclusive, accessible, and engaging for all users. The user domain is characterized by attributes a user desires. With this viewpoint, an interface is seen for what it gives to the user, without or with knowledge about the interface of a mobile phone or how the interface, as a "living product", works. The user domain includes attributes related to convenience, accessibility, multilingual support, and accuracy.

The user domain also includes factors such as user preferences, expectations, and context of use. For example, some users may prefer a minimalist interface, while others may want a more visually rich and stimulating interface. The context of use may also vary, which can affect the design of the interface to ensure usability and accessibility. In essence, the user domain represents the human-centered approach to design, which prioritizes the needs and desires of the user to create interfaces that are intuitive, efficient, and satisfying to use.

Function domain. The function domain is broad and encompasses a wide range of technical and functional specifications that the interface must fulfill to ensure accessibility, usability, and effectiveness for all users. Essentially, functions represent abstractions of the interface's behavior. In the context of mobile phone interface design, the functional domain comprises the set of functional requirements from a design and engineering point of view that the interface must satisfy to provide a complete and satisfactory user experience to all users, including those with diverse needs and abilities. The functional domain is concerned with specifying the desired performances of the interface as a living product. It encompasses the various actions and interactions that the interface engages in with its users over its lifetime. These interactions are complex, dynamic, and multifaceted, and they define the functions of the interface.

Physical domain. Designers and engineers need to imagine and define solutions in the physical domain to meet the functional requirements in the functional domain. The physical domain encompasses the physical components and structure of the interface as a living product, which are strategic elements in mobile development. It includes hardware components like the mobile device and software components such as the interface layout, typography, color, and visual design elements. The physical domain refers to the product's physical, tangible characteristics. The design of a product is studied in this domain. Interaction between the user and the interface is an important consideration in the physical domain, such as the configuration of the interface. Moreover, accessibility is also an essential aspect of the physical domain, including considerations like the size and color contrast of text and visual elements and support for assistive technologies. Considering the physical domain is crucial in inclusive design for conversational mobile service interfaces, as it plays a significant role in the usability, accessibility, and overall user experience of the interface.

Process or task domain. The process or task domain is a critical aspect of designing a conversational mobile service interface. It refers to the user journey that can be carried out using the product. Therefore, the process domain sees the product as a whole, including other elements of context that could interact with it. It involves the various steps and actions that a user takes to interact with the interface and achieve their goals. This includes the flow of information and tasks between the user and the interface. In addition to the actual interaction, the process domain also includes any feedback or guidance provided to the user during the interaction. To ensure inclusive design, it is important to design these various steps and actions to be intuitive and efficient for all users, regardless of their cognitive or physical abilities. A well-designed process can significantly enhance the overall user experience and make the interface more accessible and usable for everyone.

2.2. Engineering Design Models

Two cross-referencing modeling levels with four domains are proposed: (1) *Conceptual models* and (2) *Concrete models*. Concrete models give a fine and acute description of knowledge for interface modeling. Quite systematically, they use a formal description with a codified and specific language. We can distinguish three subclasses of such models: (a) *mathematical models*, (b) *computational models*, and (c) *experimental models*.

A conceptual model refers to design models that are represented by concepts or related concepts for interface modeling. A conceptual model is an abstract representation that describes the essential components, concepts, and relationships of an interface. A conceptual model can be used to clarify ideas, identify problems, and develop solutions. It is a simplified representation of an interface that allows people to understand, analyze, and

communicate about that interface. A conceptual model is an important tool for visualizing and communicating ideas in a way that is easy to understand and analyze. In interface design, a conceptual model is a representation of the designer's understanding of how users will interact with the interface. It is a high-level abstraction that captures the essential features of the interface and its functionality. An example of a conceptual model in the context of user interfaces is a proposal for an interaction method or a set of specifications that an interaction method must follow. The conceptual model can be represented using different forms such as written descriptions, diagrams, and sketches. The purpose of the conceptual model is to provide a shared understanding among designers, developers, and stakeholders about the design of the interface. This shared understanding can then be used as a basis for building more detailed models.

A mathematical model represents empirical objects, phenomena, and physical processes encountered during interface modeling in a mathematical or logical way. Mathematical models are often used in interface modeling to provide a more precise and quantitative representation of the system being analyzed. They can be used to simulate the behavior of the interface under different conditions and to make predictions about how the interface will behave in the future. Mathematical models can also be used to optimize the design of the interface by identifying the most efficient or effective parameters and configurations. The choice of mathematical model depends on the specific problem being analyzed as well as the available data and resources. These make it possible to carry out computing and measurements to study the interface in its context. An example of a mathematical model in the context of user interfaces is a model representing the contrast between two elements in a graphic interface. Defining the contrast mathematically enables its validation according to a set of standards.

A computational model is a representation of an interface that is implemented in a computational way, typically using computer simulation. While some computational models are based on mathematical models, others are derived from empirical data or observations. In interface design, analytical solutions are often not available, making computational models a useful tool for simulating and analyzing the behavior of the interface. An example of a computational model in the context of user interfaces is the virtual prototype of a user interface. Designers can experiment with different design parameters by adjusting inputs in the computer and observing the resulting outputs. While some computational models are treated as "black boxes", meaning that the internal workings are not necessarily understood, others can be transparent or interpretable, allowing for a deeper understanding of the underlying mechanisms. Computational models are not only used for simulation purposes but also for optimization, prediction, and data analysis during interface design. The computational model thus provides a computational representation of the interface, which facilitates simulations in a controlled virtual environment.

An experimental model is a way of studying the behavior of an interface by observing real-interface phenomena in a controlled environment. This can involve conducting experiments on the actual interface or on a physical model that represents some of its characteristics. An example of an experimental model in the context of user interfaces is an interactive prototype that simulates the behavior of the user. By gathering data on how users interact with the interface, experimental models can help designers identify problems and suggest improvements. They also offer the possibility of discovering new interactions or issues that were not previously identified or modeled.

2.3. Engineering Design Meta-Model Network for Methodology Representation

The meta-model is represented mathematically as a network structure composed of layers as well as nodes and edges (Figure 1).

Definition 1. *The meta-model is represented as a network structure composed of layers as well as nodes and edges. The engineering design multilayer network is defined as a quadruplet*

$M = (V_M, E_M, V, L)$ where V is the set of nodes, L is the set of layers, $V_M \subseteq V$ is the set subset of active-nodes, and $E_M \subseteq V_M \times V_M$ is the set of active edges.

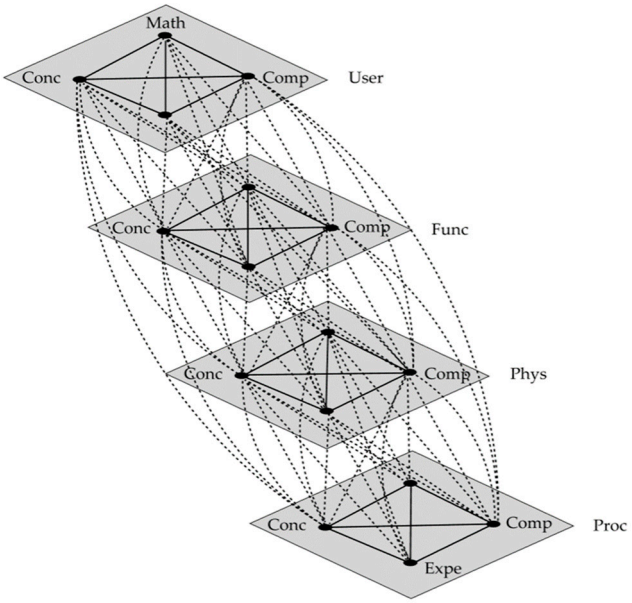


Figure 1. The set $V \times L = \{(v \in V, l \in L)\}$. For readability purposes, all possible combinations of nodes were not represented in the figure.

The set of nodes V in the network of the engineering design meta-model is composed of four models: *Conceptual model, Mathematical model, Computational model*, and *Experimental model*. The nodes of the set V are denoted as follows: $V = \{Conc, Math, Comp, Expe\}$, representing the Conceptual model, Mathematical model, Computational model, and Experimental model, respectively.

The engineering design meta-model network has one dimension named the *domain dimension*. The domain set of the engineering design meta-model consists of the following elements: *User domain, Function domain, Physical domain*, and *Process domain*. A layer is used for every element of this set. The nodes of the set L are denoted as follows: $L = \{User, Func, Phys, Proc\}$, representing the User domain, Function domain, Physical domain, and Process domain, respectively.

The set $V \times L = \{(v \in V, l \in L)\}$ encompasses all possible combinations of model-domain pairs. If a model is absent in a given design methodology, it means it is also absent in a given domain layer. This implies that for every node and layer pairing, it is necessary to specify whether the node is active within that layer. If a model is used in a given design methodology, then it is represented by an *active-node*.

Definition 2. *The subset of active nodes $V_M \subseteq V$ comprises solely those combinations of nodes and layers where a node is present in the corresponding layer.*

In the meta-model multilayer network, edges represent pairwise connections between all possible combinations of nodes in the set of models {*Conceptual, Mathematical, Computational, Experimental*} and layers in the set of domains {*User, Function, Physical, Process*}, as illustrated in Figure 1.

A model node $v \in V$ in domain layer $l \in L$ can be connected to any node $v\prime \in V$, where $v\prime \neq v$, in any layer $l \in L$. This means that intra-relationships within a domain, as well as inter-relationships between domains, exist. Assuming that intra-domain relationships and inter-domain relationships are reciprocal, all edges are considered undirected.

Definition 3. *An edge set $E_M \subseteq V_M \times V_M$, referred to as active-edges, of the engineering design multilayer network is defined as pairs of node-layer combinations that exist within the set V_M.*

The intra-layer edges are represented using solid curves and inter-layer edges using dotted curves. Figure 2a shows an illustrative example, where M_{83} is the methodology employed in the 83th article studied [22] in the literature review. M_{83} is composed of the related quadruplet $V_{M_{83}}, E_{M_{83}}, V, L$ where $V_{M_{83}}$ (red color) is the set of active-nodes that are part of the methodology in place in the articles, and $E_{M_{83}}$ (red color) is the set of edges that orders the nodes in the methodology.

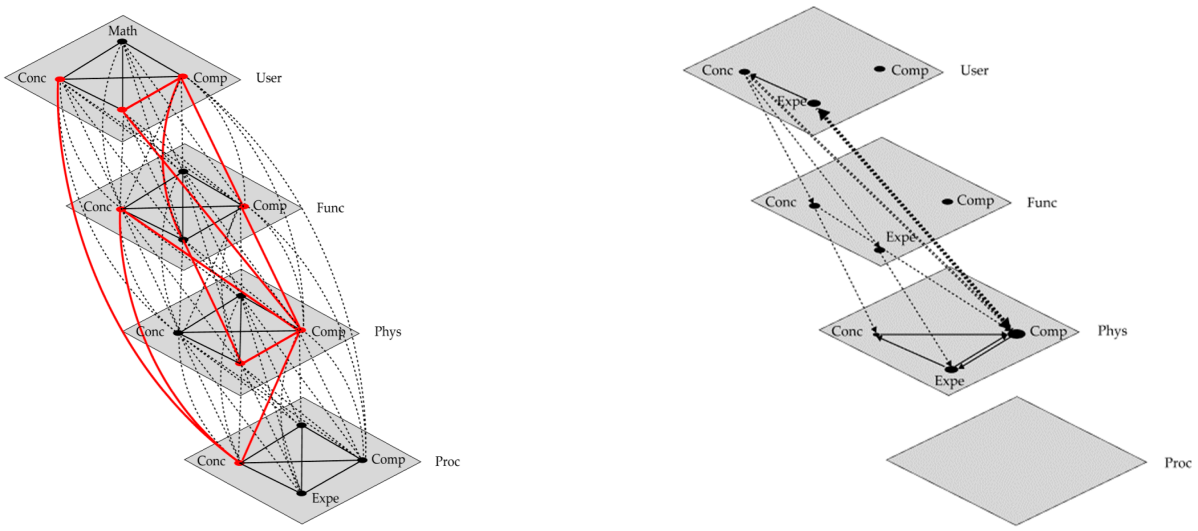


Figure 2. (**a**) The engineering design multilayer network (in red) and (**b**) the engineering design multilayer directed network.

In design methodology, the design phases may not operate bidirectionally. Assuming that intra-domain relationships and inter-domain relationships are not reciprocal, all edges are considered directed. Thus, a directed network is used to model such methodologies. The edges of the directed network are ordered pairs.

Definition 4. *The engineering design multilayer directed network is defined as a quadruplet $M = (V_M, E_M, V, L)$ where V is the set of nodes, L is the set of layers, $V_M \subseteq V$ is the set subset of active-nodes, and $E_M \subseteq V_M \times V_M$ is the set of ordered active edges.*

Figure 2a shows the engineering design multilayer directed network of the methodology employed in the 83th article studied [22] in the literature review.

Then, the following analyses can be conducted: *Model Utilization, Model Utilization Frequency*, and *Model Chronology.*

Model Utilization: This analysis investigates the specific models chosen for each methodology (M_i) employed in the i article. It will not only reveal the set of models but also shed light on their functional roles within the methodology, their underlying theoretical foundations, and their practical implications. Furthermore, examining the interactions between these models can offer valuable insights into the evolution of problem-solving approaches, the integration of interdisciplinary knowledge, and the emergence of innovative solutions. For model usage analysis, we propose to use the attribute named *Indicator of Model Utilization*, denoted as $u(v, M_i)$, where v represents the specific model and M_i represents the methodology employed in the i article. This attribute takes the value 0 or 1, indicating whether the model v is chosen for the methodology employed in the i article.

Model Frequency Utilization: The distribution of usage frequencies for specific models within each methodology (M_i) employed in the i article is of interest. The frequent use of a model suggests a focus on iterative improvement, complexity management, or adaptation to changing conditions. For model usage analysis, we propose to use the attribute named *Frequency of Model*, denoted as $f(v, M_i)$, where f represents the frequency f for the model v within the methodology employed in the i article.

Model Chronology: Understanding the temporal sequence of a model within a methodology (M_i) employed in the i article provides insights into the temporal dynamics, efficiency, and critical path of the methodology itself. For model usage analysis, we propose to use the attribute named *Temporal Sequence of Model Utilization*, denoted as $t(v, M_i)$, where v represents the specific model and i represents the methodology. This attribute indicates the temporal sequence of model v within the methodology employed in the article i.

Considering the previous analysis, the engineering design multilayer directed network is extended as follows:

Definition 5. *The engineering design multilayer attributed network is defined as a quintuplet* $M = (V_M, E_M, A, V, L)$ *where* V *is the set of nodes,* L *is the set of layers,* $A = \{u, f, t\}$ *is the set of attributes,* $V_M \subseteq V$ *is the set subset of active-nodes,* $V_M = \{(v, a) | v \in V, a \in A\}$*, and* $E_M \subseteq V_M \times V_M$ *is the set of ordered active edges.*

In the context of the engineering design multilayer attributed network, the set of attributes plays an important role in defining the characteristics and behavior of the network's nodes and edges. While the provided definition presents a specific set of attributes, it is important to note that the flexibility to extend or adapt the attribute set depends on the specific problem. Moreover, while the definition focuses on attributes associated with nodes, there is potential utility in extending attributes to edges within the network. Introducing edge attributes can enrich the network representation by encoding additional information about the relationships between models. However, while recognizing the possibility of extending attributes to edges or introducing new attribute types, this paper will primarily explore the predefined set of attributes $A = \{u, f, t\}$ within the context of the engineering design multilayer attributed network $M = (V_M, E_M, A, V, L)$.

3. ERSA (Engineering Design Research Meta-Model Based Systematic Analysis) Process

ERSA (Engineering design Research meta-model based Systematic Analysis) is a new dynamic process that utilizes the engineering design meta-model for user interface design, along with its network-based formalization developed in Section 2. ERSA is formally defined as a dynamic interactive multi-process system. The definition of the ERSA system, denoted as E, is adopted from [23–25] and is presented as follows:

$$E = \langle \Pi, I, P, O, \Phi \rangle \tag{1}$$

where Π is a set of processes, I is a set of interactions, P is a set of roles that processes can perform, O is a set of organizations, and Φ is a set of functions in each process, respectively.

The behavior β of each process π is given formally by:

$$\beta_\pi = \left\langle \Phi_{\Delta(\pi)}, \Phi_{\Delta(\pi)}, \Phi_{\Gamma(\pi)}, K_\pi \right\rangle \tag{2}$$

where:

$\Phi_{\Omega(\pi)} : (E_\pi \cup I_\pi) \times \Sigma_\pi \to \Omega(\pi)$ is function of perception, where $E_\pi, I_\pi, \Sigma_\pi, \Omega(\pi)$ E_π is the finite set of observed events of a process π, I_π is the finite set of interactions of a process π, Σ_π is the finite set of states of a process π, and $\Omega(\pi)$ is the finite set of perceptions of a process π;

$\Phi_{\Delta(\pi)} : \Omega(\pi) \times \Sigma_\pi \to \Delta_\pi$ is function of decision of a process π, Δ_π is the finite set of decisions of a process π;

$\Phi_{\Lambda(\pi)} : \Delta_\pi \times \Sigma_\pi \rightarrow \Lambda_\pi$ is function of action of a process π, Λ_π is the finite set of actions of a process π;

K_π is the knowledge of a process π, which includes key words, decision rules, and learnings.

The proposed and developed ERSA process consists of six dynamic phases, as illustrated in Figure 3, facilitating the collection of relevant articles and the construction of learnings on a specific topic. Subsequent sections will refer to this figure to describe each of these phases.

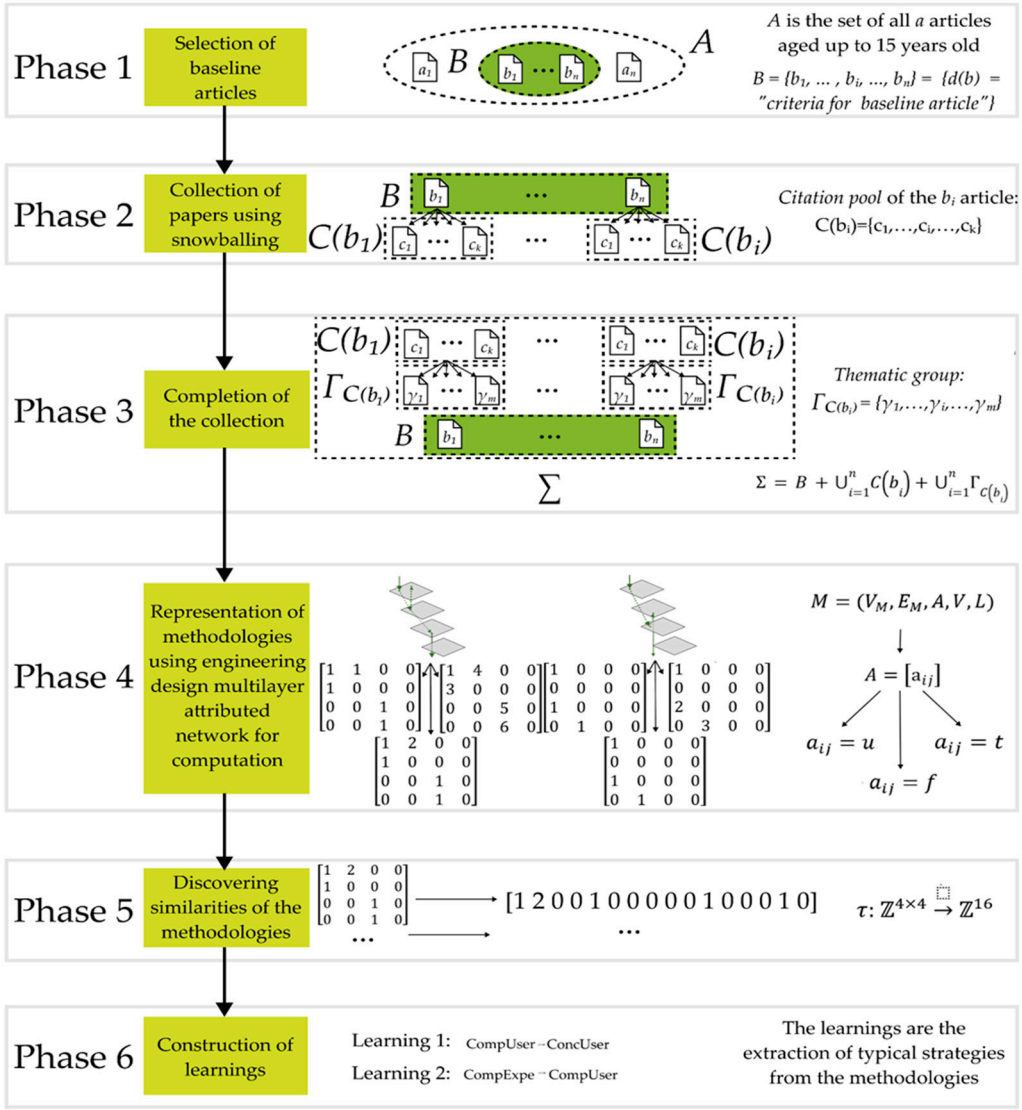


Figure 3. The ERSA (Engineering design Research meta-model based Systematic Analysis) process composed of six phases.

3.1. Phase 1–Selection of the Baseline Articles

The first phase of the ERSA process involves selecting the articles that are relevant to the topic, which will help gather comprehensive knowledge. Two main strategies are usually considered when conducting a literature review: snowballing and database searches. The first consists of studying articles preceding and following those identified as relevant. The second involves searching on one or multiple highly populated and cross-sourced databases using keywords to extract a set of articles. Performance-wise, Badampudi et al. [26] states that both strategies are comparable but highly dependent on the baseline article for the snowballing and the keywords for the database search. The

snowballing technique enables a more precise approach by leveraging specific articles rather than relying solely on various keywords related to the topic. A set of baseline articles needs to be identified as the first phase of the backward snowballing process [27]. Articles that have set a reference in the field(s) that is the aim of the research are recommended, eventually selecting multiple articles to cover different point of views regarding a specific topic. The selection of articles aged up to θ years old, for instance, $\theta = 15$, from the time of collection has been chosen in the application, to avoid the processing of too many research articles and to avoid older articles that may not be relevant considering the recent adoption of smartphones in low-income countries. Let A denote the set of all articles a aged up to 15 years old and $\delta(b) \in \Phi_{\Delta(\pi)}$ represent the function that identifies baseline articles B. Then, the set of baseline articles B is define as follows:

$$B = \left\{ b \middle| \delta(b) \in \Phi_{\Delta(\pi)} \right\} \tag{3}$$

3.2. Phase 2–Collection of Related Articles Using Snowballing

The second phase of the ERSA process involves collecting more recent articles through a snowballing process. Once the baseline articles $B = \{b_1, \ldots, b_i, \ldots, b_n\}$ are identified, for each baseline article b_i, a search to find articles citing b_i is conducted to gather a selection of recent articles on the topic. For each baseline article b_i, the citing articles are collected into a set $C(b_i) = \{c_1, \ldots, c_i, \ldots, c_k\}$ named *citation pool* of the baseline article b_i. The *citation pool* is limited to articles that are aged as κ years or less, for instance $\kappa = 5$, which allows for a smaller set of more relevant articles. Moreover, and since various authors may have cited the baseline articles, some articles collected could be off-topic. To keep the collection on the topics, filters are established. Google Scholar [28] can be used to select articles citing a baseline article b_i. If a baseline article a happen to be less than 5 years old, it will then be selected as a baseline article b.

3.3. Phase 3–Completion of the Collection

The third phase of the process consists of completing the collection by incorporating related articles for each *citation pool* $C(b_i)$. For every *citation pool* $C(b_i)$, a set of new articles is identified through a search for thematic similarity, denoted as $\Gamma_{C(b_i)} = \{\gamma_1, \ldots, \gamma_i, \ldots, \gamma_m\}$ and named *thematic group* related to the *citation pool* $C(b_i)$. As stated in Wohlin's [27] work on snowballing, it is always relevant to use different strategies to cover the literature as comprehensively as possible. The set of articles is therefore extended to include more recent works and to include works that may have been overlooked in the previous phase of the process. The Research Rabbit process [29], which uses a list of works to visualize their interconnections through their common citations, permits to find studies on similar topics. In this case, the list of articles collected in snowballing phase is used in Research Rabbit process. The criteria applied for the research in Research Rabbit process are the same as those applied in the snowballing phase. The list is completed by articles suggested by Research Rabbit, contributing to its continuous growth. This phase stops when Research Rabbit has no other studies to suggest than the one already present in the list or those not considered in the focus of the study.

The complete collection of articles Σ is defined as the union of three sets: the baseline articles set (B), encompassing all initial articles aged up to θ years old; the set of all citation pools ($C(b_i)$), representing the citing articles for each baseline article (b_i); and the set of thematic similarity articles ($\Gamma_{C(b_i)}$), comprising new articles identified through thematic similarity searches for each citation pool:

$$\Sigma = B + \bigcup_{i=1}^{n} C(b_i) + \bigcup_{i=1}^{n} \Gamma_{C(b_i)} \tag{4}$$

3.4. Phase 4–Representation of Methodologies Using Engineering Design Multilayer Attributed Network for Computation

Each paper's methodology is represented by the *engineering design multilayer attributed network* $M = (V_M, E_M, A, V, L)$. Then, to simplify computation, ERSA use the matrix $A = [a_{ij}]$, which represents the set $V \times L = \{(v \in V, l \in L)\}$ encompassing all possible combinations of model-domain pairs in the engineering design multilayer attributed network, with $a_{ij} \in \{u, f, t\}$.

The matrix (5) shows the set $V \times L = \{(v \in V, l \in L)\}$ encompassing all possible combinations of model-domain pairs.

$$\begin{bmatrix} ConcUser & ConcFunc & ConcPhys & ConcProc \\ MathUser & MathFunc & MathPhys & MathProc \\ CompUser & CompMath & CompPhys & CompProc \\ ExpeUser & ExpeMath & ExpePhys & ExpeProc \end{bmatrix} \tag{5}$$

The matrix (6) illustrates the case where $a_{ij} = u$. It indicates that the employed methodology incorporates the models *ConcUser*, *ConcFunc*, *ConcPhys*, *ConcProc*, *MathPhys*, *CompPhys* and *ExpePhys*.

$$\begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{6}$$

The matrix (7) depicts the case where $a_{ij} = f$. It indicates that the frequency f for the models *ConcUser*, *ConcFunc*, *ConcPhys*, *ConcProc*, and *MathPhys* within the employed methodology is 1, while the frequency f for the models *CompPhys* and *ExpePhys* within the employed methodology is 2.

$$\begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 2 & 0 \end{bmatrix} \tag{7}$$

The matrix (8) displays the case where $a_{ij} = t$, indicating the temporal sequence of models as follows: 1: *ConcUser*, 2: *ConcFunc*, 3: *ConcPhys*, 4: *MathPhys*, 5: *CompPhys*, 6: *ExpePhys*.

$$\begin{bmatrix} 1 & 2 & 3 & 0 \\ 0 & 0 & 4 & 0 \\ 0 & 0 & 5 & 0 \\ 0 & 0 & 6 & 0 \end{bmatrix} \tag{8}$$

Through matrix representation and analysis, ERSA can unlock a deeper understanding of engineering design methodologies. This powerful tool allows us to identify the starting and ending models within each methodology. It also enables us to recognize recurring patterns, highlighting similarities between different approaches. Furthermore, we can use matrices to compare methodologies and evaluate which aspects of the engineering design meta-model receive the most focus.

3.5. Phase 5–Discovering Similarities of the Methodologies

To extract similarities and striking elements in the methodologies in order to identify and interpret meaningful patterns, trends, or insights, computation is required. Clustering algorithms, a subset of unsupervised machine learning techniques, group data points based on similarities. They provide a systematic way to identify commonalities and differences among methodologies. By organizing methodologies into clusters, researchers gain insights into the underlying structures of employed methods. Instead of analyzing an unstructured set of methodologies, coherent clusters sharing similar models can be identified. This simplifies trend identification, best practice determination, and the recognition of

emerging paradigms. Clustering results can prioritize relevant methodologies for further investigation and can reveal gaps in existing research, aiding knowledge synthesis and new research directions.

Careful consideration of data representation, algorithm choice, and result interpretation ensures the validity and reliability of clustering outcomes. To apply clustering algorithms, ERSA transform the matrices presented in Section 5 into vectors. Let us denote the transformation function as τ, where τ represents the transformation from the matrix A to column vector X. Mathematically, we can define this function as:

$$\tau : \mathbb{Z}^{4\times 4} \rightarrow \mathbb{Z}^{16} \tag{9}$$

where $\mathbb{Z}^{4\times 4}$ denotes the set of all 4×4 matrices with integer entries, and $\mathbb{Z}^{16}$ represents the set of all 16-dimensional row vectors with integer values.

The transformation function can be defined as:

$$\tau(A) = X \tag{10}$$

where A is a 4×4 matrix with integer entries representing the input from phase 5, and X is a 1×16 row vector with integer entries representing the output. The function τ maps each integer element of the input matrix A to its corresponding position in the output column vector τ while preserving the order of elements. Thus, the function τ transforms a 4×4 matrix with integer entries into a 16-dimensional row vector with integer values.

For instance, the transformation of the matrix (6) is represented below (11):

$$\tau\left(\begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}\right) = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \tag{11}$$

Thus, each article in the complete collection of articles Σ is represented by a vector X. All these vectors are then assembled into a matrix Y, where each row represents an article with employed methodology and each column represents 16 models. The assembly function f operates on n row vectors ($n = card(\Sigma)$), each with 16 integer elements, producing a resulting matrix Y with n rows and 16 columns. The assembly function f is defined as:

$$f : \left(\mathbb{Z}^4\right)^n \rightarrow \mathbb{Z}^{n\times 16} \tag{12}$$

$$f(X_1, X_2, \ldots, X_n) = Y \tag{13}$$

where f is the function that assembles row vectors into matrix; X_i represents the i-th row vector with 16 integer elements; and Y is the resulting matrix, also with n rows and 16 columns. The elements of Y are obtained by assembling the row vectors $X_1, X_2, \ldots, X_n$ as rows of the matrix Y with $y_{ij} = x_{ij}$, where y_{ij} represents the element in the i-th row and j-th column of Y, and x_{ij} represents the j-th element of the i-th row vector X_i.

The matrix $Y = [y_{ij}]$ (14) with dimension $n \times 16$ illustrates the assembly of the vector X in the Equation (7).

$$Y = \begin{bmatrix} y_{11} & y_{12} & y_{13} & y_{14} & y_{15} & y_{16} & y_{17} & y_{18} & y_{19} & y_{110} & y_{111} & y_{112} & y_{113} & y_{114} & y_{115} & y_{116} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ y_{n1} & y_{n2} & y_{n3} & y_{n4} & y_{n5} & y_{n6} & y_{n7} & y_{n8} & y_{n9} & y_{n10} & y_{n11} & y_{n12} & y_{n13} & y_{n14} & y_{n15} & y_{n16} \end{bmatrix} \tag{14}$$

ERSA comprehends the application of interactive processes for discovering similarities based on the characteristics of the defined problem. Here, in order to discover the cluster of methodologies, we show the adoption k-means with the elbow method among the various

clustering processes. The dataset for k-means with the elbow method is represented by the matrix $Y = [y_{ij}]$. K-means stands out for its simplicity, efficiency, and effectiveness in many real-world applications. However, in determining the optimal number of clusters of methodologies, k, is crucial for the success of k-means clustering. In this context, the elbow method is a commonly used technique for selecting the optimal number of clusters in k-means clustering. The elbow method utilizes the explained variance ratio to identify the optimal number of clusters k. It looks for the "elbow" point on the graph representing the explained variance ratio as a function of k, where increasing k leads to a minimal increase in the explained variance ratio, suggesting diminishing returns in explaining additional variance. The rationale behind this method is that adding more clusters beyond the elbow point does not lead to a significant improvement in clustering quality. Thus, the elbow method provides an objective criterion for selecting the optimal number of clusters, allowing data analysts to make informed decisions based on the characteristics of the dataset. Additionally, the elbow method is robust and intuitive. It does not rely on assumptions about the underlying data distribution and can be applied to various types of datasets. The elbow plot offers a straightforward visualization of the trade-off between model complexity (number of clusters) and clustering quality (intra-cluster variation). While k-means clustering itself is computationally efficient, the elbow method adds minimal computational overhead.

3.6. Phase 6–Construction of the Learnings

The last phase identifies learnings that will help the design of methodologies on the topic. These learnings should be generic and easy enough to be understood and put into place by future researchers while they are designing their methodology. The learning process in ERSA is currently assisted by human-based learning, where human expertise and guidance play a central role in comprehending and processing results effectively. While this approach allows for good understanding and adaptability, there is substantial potential in integrating machine learning techniques to advance ERSA's capabilities.

The next section describes the application and results obtained through the developed ERSA process for each phase in the context of user interfaces design adapted for users in low-income countries.

4. Application and Results

The results of each phase are described below:

4.1. Phase 1–Selection of the Baseline Articles

For this phase of ERSA process, the specificities that made the interaction for users in low-income countries different from high-income countries are used as a starting point to identify baseline articles. According to the literature, four criteria are specific to low-income countries: (1) low literacy rate [30], (2) low level of computer literacy [31], (3) high rate of multilingualism in those regions with a high number of local languages, often oral [32], (4) and the impossibility of accessing IT tools, either for financial reasons, such as telecom charges and the cost of acquiring and maintaining the device, or structural reasons. These structural reasons can be because of white or poorly covered areas or the distance of the user from the electricity network. Then there is a more general criterion: (5) the integration of user culture into the design of user interfaces. This criterion comes into play when the target users have a different culture than that of the design team [33]. At least one baseline study was collected on each of the five criteria previously described, as described in Table 1.

Table 1. List of the baseline studies in each criterion. Each article has been selected as a reference in their respective criterion or in related user interfaces solutions to take into account the criteria.

	Criteria	Low-Income Countries Versus High-Income Countries	Baseline Studies
1	Low literacy rate	In low-income countries, illiteracy is almost three times as high as in lower-middle and upper-middle income countries. The user interfaces based on text are therefore not suitable for users in low-income countries.	Medhi Thies [34]
2	Low level of computer literacy	Low-income countries exhibit extremely low levels of digital literacy. The user interfaces that assume users are proficient on using a computer are therefore not suitable for users in low-income countries.	Medhi Thies [34]
3	High rate of multilingualism with oral languages	As a whole, high-income areas have low linguistic diversity, while low-income areas have high linguistic diversity geographically. The user interfaces made for only a few major worldwide languages are therefore not suitable for users in low-income countries	Danis et al. [35]
4	Impossibility to access IT tools	In around 50% of the low-income countries, fewer than 50% of inhabitants report owning a smartphone. Energy access is low in poorer countries, and increases as incomes increase. A decent internet connection is out of reach for 90 per cent of people in low- and middle-income countries. The user interfaces designed on devices that are not used and/or requires frequent recharging or high connection quality is therefore not suitable for users in low-income countries	Blumenstock and Eagle [36]
5	User culture	As most user interfaces decisions are made in high-income countries, the need to understand the culture of the end user is utmost important. The user interfaces designed with the culture of the designers' country in mind is therefore not suitable for users in low-income countries.	Heimgärtner [37–39]

4.2. Phase 2–Collection of Related Articles Using Snowballing

For this second phase of ERSA process, studies that were conducted in high-income countries were withdrawn from the selection to keep only low-income countries. The choice was made to include in this definition the African continent, South Asia (Bangladesh, India, Indonesia, Pakistan), and South America. A first set of articles was collected using this strategy. A filter was also applied to keep only articles published after 2018. Figure 4 depicts the results from the Research Rabbit process [29] when given the list of the articles collected during this phase on criterion (1), low literacy rate. Each node represents an article, while a connection indicates that there is a citation between two articles. Some groups of citations are formed at the top and bottom, while in the middle sits a set of articles that are not interconnected.

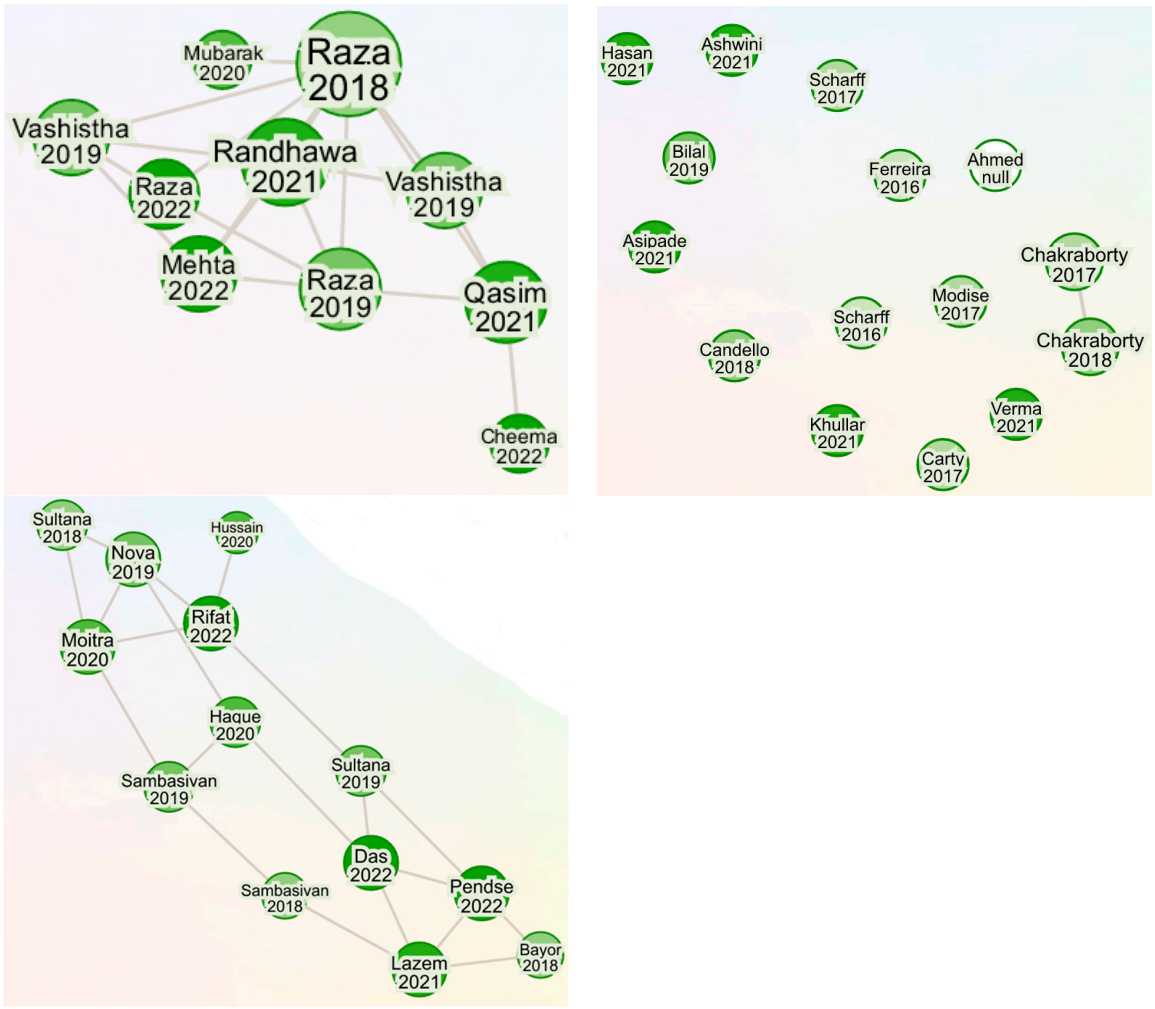


Figure 4. Results of the snowballing work from articles citing the work of Medhi-Thies [34] and being dated from 2018 to 2023.

4.3. Phase 3–Completion of the Collection

Using the four previous lists based on the four baseline articles and the Research Rabbit process, new articles are listed, allowing for the extension of each list, which is now put in common for the rest of the study. The detailed number of articles collected in this phase can be found in Figure 5.

There were 174 articles selected to form a representative set of articles on the topic of interface design for users in low-income countries (see Table S1).

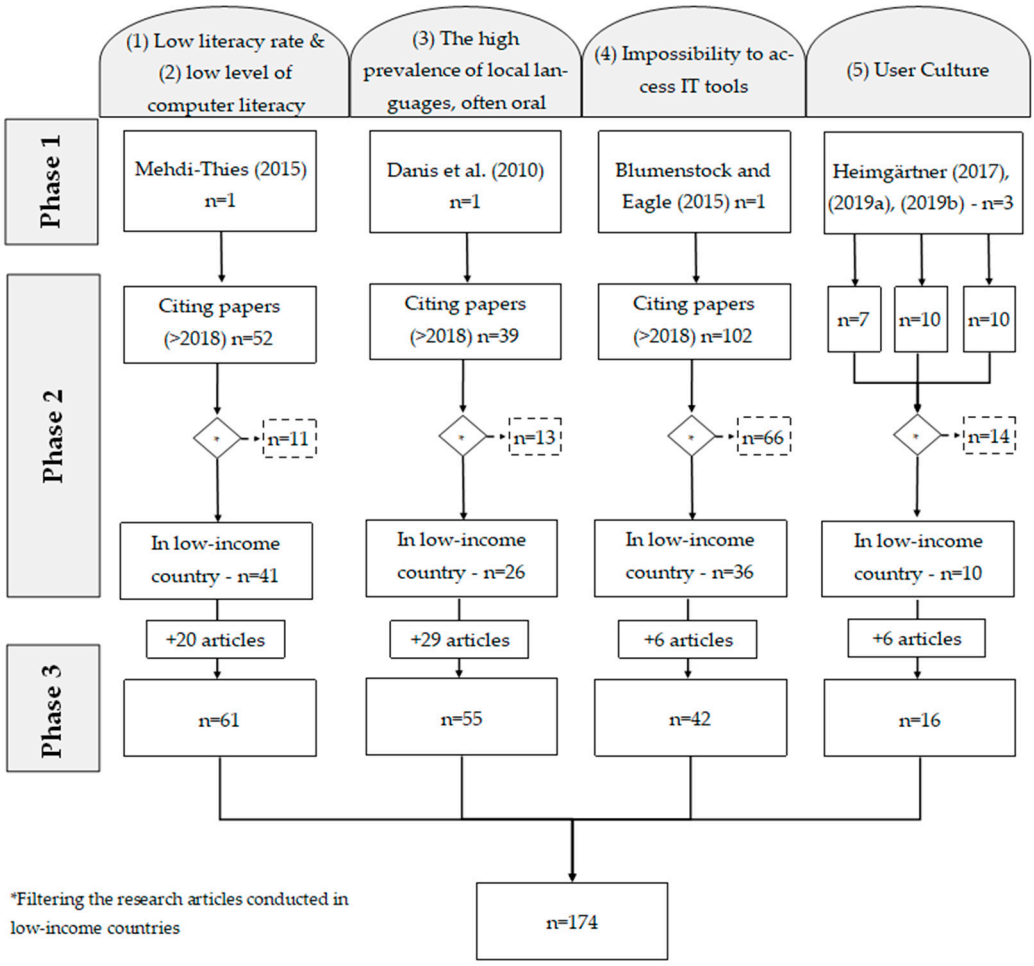


Figure 5. The first three phases of the ERSA process applied to methodologies for interface design for users in low-income countries. The articles cited in the phase 1 are Mehdi-Thies [34], Danis et al. [35], Blumenstock and Eagle [36] and Heimgärtner [38–40].

4.4. Phase 4–Representation of Methodologies Using Engineering Design Multilayer Attributed Network for Computation

In this phase, ERSA process applies the engineering design meta-model. The 174 articles were therefore put in perspective of the engineering design meta-model, which generated 174 matrices with $a_{ij} \in \{u, f, t\}$ (see Equations (6)–(8)).

4.5. Phase 5–Discovering Similarities of the Methodologies

In this phase, ERSA builds the matrix $Y = [y_{ij}]$ with dimension 174×16. The elements of Y are obtained by assembling the row vectors $X_1, X_2, \ldots, X_{172}$ as rows of the matrix Y with $y_{ij} = x_{ij}$, where y_{ij} represents the element in the i-th row and j-th column of Y, and x_{ij} represent the j-th element of the i-th row vector X_i. The k-means algorithm, utilizing the elbow method, was chosen among different clustering processes in ERSA for the application to the matrix $Y = [y_{ij}]$ where $a_{ij} \in \{u, f, t\}$. In this paper, we will consider the results for $a_{ij} = t$, which yields findings that are particularly pertinent and insightful. Fifteen clusters were found, represented in Figure 6. The clusters 1, 3, 5, 7, 9, 11, and 14 are specific to a model, while the clusters 4, 6, 12, 13, and 15 are grouping similar methodologies. Finally, the clusters 8 and 10 only include a limited number of methodologies.

- **Cluster 1:** The 1st cluster is the most populated cluster, with 95 methodologies entering it. It is also one of the simplest ones, since almost all the methodologies have only one

step, except for one. Most of the articles also studied the conceptual model, except for two that are interested in the experimental model of the physical domain (ExpePhys).

- **Cluster 2:** The 2nd cluster is formed of 2 methodologies that both have around four or five steps, start from the conceptual model of the user domain (ConcUser), and finish on ExpePhys. During the process, both consider aspects of the conceptual models and the user domain, each in a different way.
- **Cluster 3:** The 3rd cluster is composed of 15 methodologies that, except for two, all have one step only, on the experimental model of the user domain (ExpeUser). The two exceptions both end on this step while starting on the physical domain.
- **Cluster 4:** The 4th cluster is made of 5 clusters that all have around four to six steps. The difference between this and the 2nd cluster is that while both start on conceptual models and end on experiments, the 4th cluster is considering the computational model of the physical domain (CompPhys).
- **Cluster 5:** The 5th cluster is a simple cluster formed by 21 methodologies, 19 being simple step methodologies over the conceptual model of the functional domain (ConcFunc). The two exceptions are no less than three steps and also take into account ConcFunc.
- **Cluster 6:** The 6th cluster is a cluster of 14 methodologies that all have two or three steps and end on ExpePhys, while starting on different models but mainly on models of the conceptual model. Most of the methodologies also consider CompPhys.
- **Cluster 7:** Cluster 7 is only formed of one six-step methodology that starts on ConcUser, ends on ExpePhys, and iterates over this same ExpePhys step.
- **Cluster 8:** Cluster 8 is formed by two methodologies that start by ConcUser and the conceptual model of the physical domain (ConcPhys). One stops here, while the other ends with ExpePhys.
- **Cluster 9:** Cluster 9 has 5 one-step methodologies that have interest in the mathematical model of the use domain (MathUser). It is the simplest cluster, the only one having all its methodologies being one step.
- **Cluster 10:** Cluster 10 has 2 three-step methodologies that both consider CompPhys and ConcFunc, while adding a step in the conceptual model before or after those two steps.
- **Cluster 11:** Cluster 11 is only formed of one eight-step methodology that iterates three times over ConcPhys and CompPhys before ending on the mathematical model of the physical domain (MathPhys) and ExpePhys.
- **Cluster 12:** Cluster 12 is composed of 3 methodologies that consider three to four steps, all ending on ExpUse. Two have just before considered MathPhy and ConcPhy.
- **Cluster 13:** Cluster 13 has 3 two-step methodologies that all end on MathUser. Two of the three start on ConcUser, while the last one starts on the conceptual model of the process domain (ConcProc)
- **Cluster 14:** Cluster 14 is composed of only one methodology. It is the longest with 17 steps and highly iterative, and it has already been identified in the previous phase.
- **Cluster 15:** The last cluster is formed of 3 methodologies that all consider ConcPhys while being no more than two steps long.

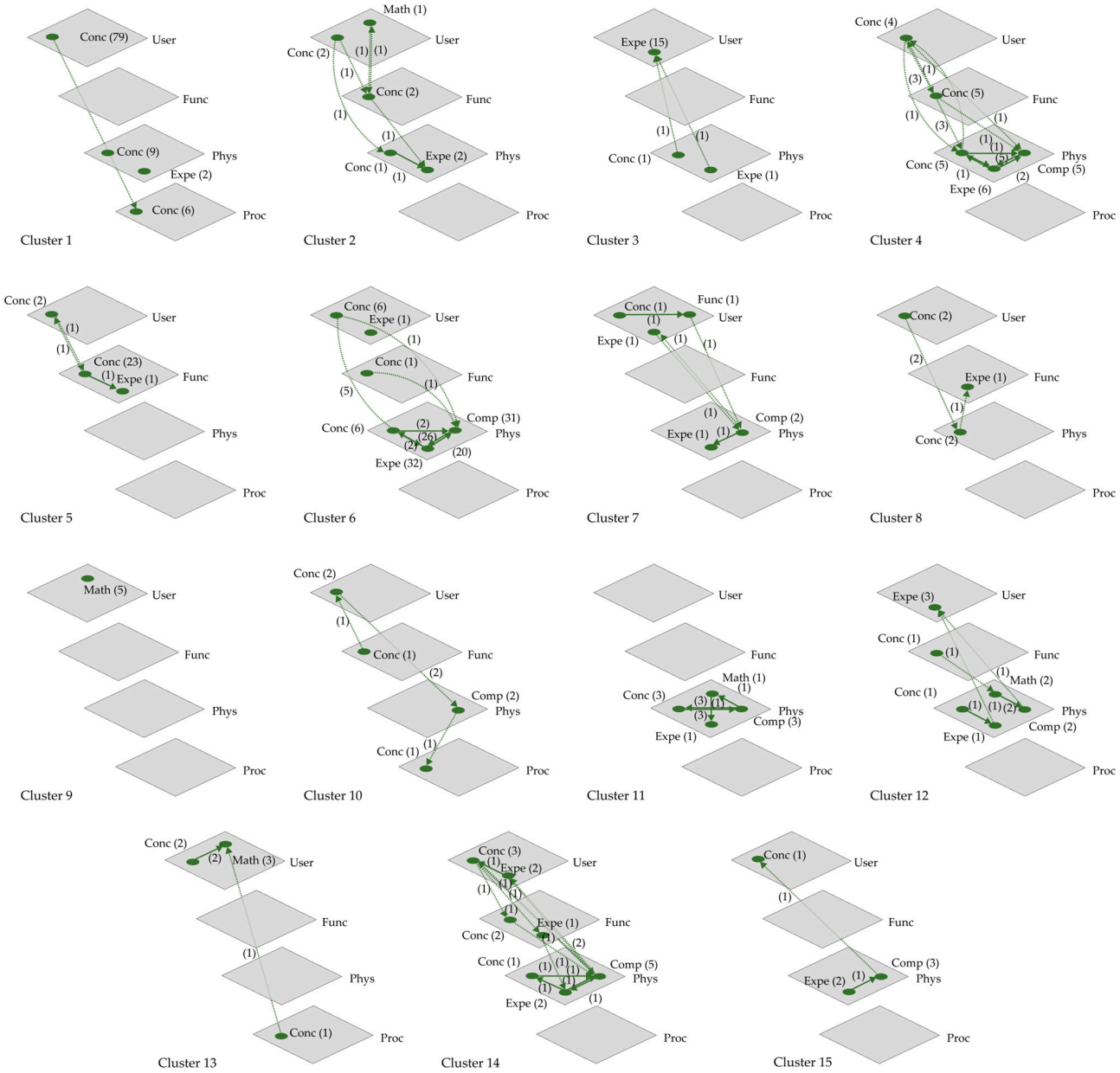


Figure 6. Representation of the 15 clusters using a multilayer representation.

Several emerging patterns resulting from cluster analysis and data processing are the following:

- **Clusters with one-step methodologies:** The clusters n°1, 3, 5, and 9 are part of this first family. The research articles present in those methodologies do not propose a true methodology, since they focus only on one part of the design work. The high number of methodologies could be explained by the fact that the algorithm has tried to group together methodologies that were heavily focused on one model of the engineering design meta-model: the 1st cluster revolves around the conceptual domain of the user model (ConcUser), the 3rd cluster is focused on the experimental model of the user domain (ExpeUser), the 5th cluster is interested in the conceptual model of the functional domain (ConcFunc), and the 9th cluster is composed only of one-step methodologies that consider the mathematical model of the physical domain (MathPhys).
- **Clusters with two to five steps:** The 4th, 6th, 12th, 13th, and 15th clusters are part of the second family. This family of clusters shows similarities to existing methodologies;

for example, the 4th cluster is similar to the UCD methodologies: the methodologies developed in this cluster start by learning about the users but also working on the specifications the end product should have. Finally, a product is developed and is tested. The 6th cluster is more likely to represent one sprint in the context of a SCRUM framework [40]; this small part of a project allows the final product to be more suited to the command instead of basing the whole process on one version of the product. The 15th cluster is similar because the methodologies found in this cluster propose a user interface without testing it. These small iterations can also be found in the 13th cluster, although it is more oriented toward MathUser. The 12th cluster, however, was more likely to experiment with the end users instead of testing a product.

- **Clusters with only one methodology:** These clusters form the third family. The clusters with only one methodology (the 7th, 11th, and 14th clusters detailed in Section 3.6 are populated with long methodologies that show specific approaches while retaining aspects of UCD methodologies: the users are usually a focus of interest at the start, and an experiment is conducted at the end of the methodology. These methodologies being conducted in different contexts, and requiring a different number and types of iterations, are in the end very different, which is why the algorithm has ranked them into different clusters. The 2nd cluster is of similar shape, but of medium length, which has allowed us to group together two similar methodologies.
- **Exceptions or original methodologies:** The 8th and 10th clusters are methodologies that are exceptions, in the sense that they are mainly focused on the conceptual model without testing the solution proposed. In this sense, they hardly follow an existing methodology.
- **Identification of single-step methodologies**: 75.8% (132 of 174) of the methodologies are single-step methodologies. These articles include interventions in a general topic such as a mode of interaction or a specific study of digital adoptions in certain regions.
- **Identification of the most commonly utilized models as an initial step***:* 82.9% (34 of 41) of the multiple-step methodologies have a first step that is part of the conceptual model. Moreover, 51.2% (21 of 41) of the process methodologies have a first step that is part of the user domain. This means that overall, researchers were more likely to obtain knowledge about the end users in an abstract way before diving into more concrete aspects.
- **Identification of models frequently utilized as the final step:** 65.9% (27 of 41) of the process methodologies end with a step that is part of the physical domain. Moreover, 70.7% (29 of 41) of the process methodologies end with a step that is part of the experimental model. This shows that there is a will on the researchers' part to test out the user interface (physical domain) as the final part of their methodology. All the methodologies that ended with the experimental model have put in place experiments, such as about the user interface they developed, the product they tested, or a strategy they needed to test out. In the context of user interface design, experimentation is a crucial step, since it forms the testing of the newly developed interface.
- **Iterations**: 22% (9 of 41) of the multiple-step methodologies presented iterations, meaning that the methodologies went back to a step that was previously developed. Of these iterations, 55% (5 of 9) happened over the computational model of the physical domain and the experimental model of the physical domain (CompPhys -> ExpePhys). All five of these iterations were conducted over different versions of a user interface, making it possible to carry out heavy testing on the interface developed, with each new version trying to improve the previous one.
- **Conceptualization of design methodologies:** In their article, De Silva et al. [22] adapted the Design Science Research methodology theorized by Hevner [41] for specificities of interface design for users in low-income countries. The specificities taken into account on the methodology are to enable collaborations at different times and locations. The result is a six-step methodology that is highly iterative, and where each step has their identifier, which is numbered according to the iteration number

associated with each step. According to the authors, they "strongly recommend this model for any large and complex research project aiming to develop an artifact". From the 174 research articles, it is the only one that put such a focus on the methodology.

- **Coverage of the meta-model:** Matrix (15) sums up the coverage of the models by the process methodologies; for example, over the 41 multiple-step methodologies, 58.5% covered the ConcUser model. The sum of percentages for the models is greater than 100% because each methodology covers more than one model. The models never encountered were MathFunc, MathProc, CompUser, CompFunc, CompProc, and ExpeProc. Considering the focus of the application on the design of interfaces adapted for users in low-income countries, it makes sense that aspects such as the knowledge of the end users (ConcUser) and aspects of the features of the app (ConcFunc) are more developed than mathematical or computational models because of the nature of the topic, which is more likely to focus on the understanding of the users or the solution proposed in a conceptual way.

$$\begin{bmatrix} ConcUser & ConcFunc & ConcPhys & ConcProc \\ MathUser & MathFunc & MathPhys & MathProc \\ CompUser & CompFunc & CompPhys & CompProc \\ ExpeUser & ExpeFunc & ExpePhys & ExpeProc \end{bmatrix} = \begin{bmatrix} 58.96 & 18.50\% & 13.87\% & 4.62\% \\ 5.20\% & 0\% & 1.73\% & 0\% \\ 0\% & 0\% & 15.61\% & 0\% \\ 12.14\% & 1.73\% & 16.18\% & 0\% \end{bmatrix} \quad (15)$$

4.6. Phase 6–Construction of the Learnings

The developed process has led to the generation of the following key learnings.

- **Learning n°1 (Core models inclusion)**–The analysis reveals that conceptual and experimental models are prominently featured across most clusters. Notably, the 9th cluster diverges by excluding both models. Additionally, despite the widespread prioritization of the conceptual and experimental models, the utilization of the mathematical model remained surprisingly low.
- **Learning n°2 (Adapting iterative strategies)**—The analysis demonstrates that the most common strategies include iterative processes, especially involving the physical user interface design and subsequent testing. This adaptation enables continuous user feedback when initial models are incomplete. Despite the potential benefits of iterative methodologies, the results highlight that research articles from low-income countries exhibit a low rate of iterations. This may be explained by the challenges associated with field testing and distance from end users.
- **Learning n°3 (Incomplete modeling)**—The analysis indicates that the methodologies examined in this study fail to fully incorporate all models of the engineering design meta-model. Indeed, none of the 15 clusters included all the models. This result underscores the incompleteness within the field and a gap in current research methodologies.
- **Learning n°4 (Mathematical Modeling Gap)**—The analysis exposes a reluctance in adopting mathematical modeling. Among the clusters analyzed, only clusters 2 and 13 employ the mathematical model. The results suggest that designers may be limited in their ability to take advantage of the enhanced precision, predictive capabilities, optimization potential, and validation opportunities afforded by mathematical models.
- **Learning n°5 (Community-based Design)**—The analysis suggests a transition from user-based design to community-based design. To better addresses local cultural needs, the results show that it is preferable to focus on, or conduct, studies on community-wide studies rather than individual users.
- **Learning n°6 (Real-time dynamic solutions)**—The analysis outlines the importance in incorporating cultural sensitivity across multiple cultures. The physical models should not exclude users due to socio-economic and literacy disparities. The results reveal that designs should be tested in real-time using dynamic solutions to ensure they meet the varied capacities and computer literacy levels of users.

- **Learning n°7 (Versatility)**—The analysis allowing for identification of 15 clusters, demonstrating the diversity of the methodologies employed. The results show that including varying levels of literacy, computer literacy, and potential future literacy is crucial to make designs of user interface versatile. More than in any other context, the results suggest that designers should correctly include users in all phases.
- **Learning n°8 (Model-based Interface design)**—The design of an interface can be seen as a process of building engineering models from the user domain to the process domain.

5. Discussion and Limitations

The study presents a new approach to analyzing design methodologies for inclusive user interfaces in low-income countries. ERSA, formally defined as a system of interactive processes, allows for dynamic learning and developing design methodologies for inclusive user interfaces in low-income countries.

The ERSA process applied to Orange allowed for a comprehensive examination of the characteristics, patterns, and implications of different design methodologies, contributing to a deeper understanding of design practices in inclusive user interfaces for low-income countries.

The results indicate that although design methodologies vary, common principles and practices support effective design processes across different contexts. The presence of both simple one-step methodologies and more elaborate, iterative approaches highlights the need for versatile design strategies to accommodate diverse needs and constraints. The widespread occurrence of single-step methodologies suggests a common tendency to focus on specific aspects of the design process, potentially overlooking broader considerations.

The findings indicate that researchers prioritize user-centered design principles, usability testing, or cultural adaptation. The presence of iterative approaches and experimentation in design methodologies reflects a commitment to user-centered and iterative design principles, which, though essential for ensuring the effectiveness and usability of solutions, highlight the challenges in achieving convergence toward optimal solutions.

Conceptual and experimental models are central to user-based design methodologies, while mathematical models are poorly represented, and computational models are uniquely used in the physical domain. While computational modeling techniques exist for certain aspects of interface design, they are not utilized in design practices for inclusive user interfaces in low-income countries, neither for the user domain, functional domain, nor processes domain. Surprisingly, computational predictive modeling of user behavior is lacking.

The results indicate that designers may be hesitant to adopt computational modeling. While computational modeling can offer numerous benefits, it also comes with its own challenges, including the complexity of developing and implementing models, integrating them into the design process, and interpreting the results. Indeed, computational modeling is complex and technically challenging. This perceived complexity may discourage designers from incorporating computational modeling into design processes, opting for simpler and more familiar methods instead.

The rather large absence of mathematical models can also lead to a deficiency in computational modeling within the context of interface design. Mathematical models can enable designers to quantitatively describe and analyze various aspects of interface design, such as user behavior, interaction patterns, and usability metrics. Without these formal mathematical representations, it becomes challenging to systematically analyze and predict the performance of interface designs. In addition, without appropriate mathematical models, designers may lack the necessary tools to systematically evaluate the effectiveness, usability, and user satisfaction of interface designs. Mathematical models enable the integration of advanced techniques, such as machine learning, data analytics, and predictive modeling, into interface design. Without mathematical frameworks for representing and analyzing data, designers may be limited in their ability to utilize these advanced techniques for inclusive user interfaces in low-income countries.

Designers often use intuition, creativity, and user feedback to iteratively refine and improve interfaces. Inclusive interface design involves subjective elements related to aesthetics, usability, and human perception, which are challenging to quantify and model computationally. While mathematical and computational models can provide valuable insights into many aspects of interface design, they may not capture the full richness and complexity of human-centered design.

In this context, our findings reveal the importance of adopting more holistic approaches. They suggest that user-based design methodologies, mathematical models, computational models, and experimental models can be defined and theorized together. Future research could explore additional factors influencing the effectiveness of design methodologies in low-income countries, such as diversity of cultural norms but also the dynamic evolution of these countries. In this context, computational design can be used to assist designers in predicting the future of interfaces for low-income countries.

While the learning process in ERSA currently relies on human-based learning, there is recognition of the significant potential in integrating machine learning techniques to improve ERSA's capabilities. This transition toward machine learning can be seen as a future direction toward developing more objective learnings.

The ERSA process has only undergone testing within the topic of design methodologies for interfaces adapted for users in low-income countries. While its efficacy has been demonstrated in this specific context, its transferability to other topics remains open. However, the generic approach of the snowballing process, the flexibility of the engineering design meta-model over different product methodologies, and the similarity of the constructions of research articles suggest that the process could easily be used in other contexts.

6. Conclusions

The ERSA process developed in this article proposes a new approach to construct learnings from the literature by using both a collection of research articles regarding a topic integrating design methodologies and an analysis through both algorithmic clustering and data processing. By aggregating and analyzing the methodologies conducted by the research articles, the systematic approach allowed for the creation of learnings from the literature on the topic of methodologies to, in our case, design interfaces adapted for users in low-income countries.

The learnings gained from the application of the process on the topic of design methodologies for interfaces adapted for users in low-income countries will also enable researchers, designers, and engineers who want to construct design methodologies that suit their context of work, independently of the low-income country that is targeted.

Our findings indicate the importance of adopting more holistic approaches. Particularly, they suggest that user-based design methodologies and computational design can be defined and theorized together.

This process could be used on other topics that focus on design methodologies descriptions to extract learnings from a set of research articles, applying design methodologies in their works. The proposed ERSA process is opening new opportunities for data collection in any engineering-related field, allowing researchers to gather information using existing the methodologies of research articles. Because of these three factors—the adaptability of the snowballing process, the flexibility of the engineering design meta-model, and the structural similarities found in research articles—the described process could be effectively utilized in diverse contexts beyond its original application.

Supplementary Materials: The following supporting information can be downloaded at: https://www.mdpi.com/article/10.3390/a17060232/s1, Table S1: Collection of 174 articles focused on the topic of interface design for users in low-income countries.

Author Contributions: Conceptualization, E.O., Y.M., S.S. and D.D.; methodology, E.O., Y.M., S.S. and D.D.; software, Y.M.; validation, S.S., E.O. and D.D.; formal analysis, E.O., Y.M. and S.S.; investigation, Y.M., E.O., S.S. and D.D.; resources, S.S. and D.D.; data curation, Y.M. and S.S.; writing—original

draft preparation, Y.M., E.O., S.S. and D.D.; writing-review and editing, Y.M., E.O., S.S. and D.D.; visualization, Y.M., S.S., E.O. and D.D; supervision, E.O., S.S. and D.D.; project administration, D.D., E.O. and S.S.; funding acquisition, D.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research is funded by Orange Innovation, France.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

Acknowledgments: The authors thank Orange Innovation for their valuable support in this research.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Pearson, J.; Robinson, S.; Reitmaier, T.; Jones, M.; Joshi, A. Diversifying Future-Making Through Itinerative Design. *ACM Trans. Comput.-Hum. Interact.* **2019**, *26*, 1–21. [CrossRef]
2. Alsswey, A.; Al-Samarraie, H. The Role of Hofstede's Cultural Dimensions in the Design of User Interface: The Case of Arabic. *AIEDAM* **2021**, *35*, 116–127. [CrossRef]
3. Khullar, A.; Hitesh, P.; Rahman, S.; Kumar, D.; Pandey, R.; Kumar, P.; Tripathi, R.; Prince, P.; Jha, A.A.; Himanshu, H.; et al. Costs and Benefits of Conducting Voice-Based Surveys Versus Keypress-Based Surveys on Interactive Voice Response Systems. In Proceedings of the ACM SIGCAS Conference on Computing and Sustainable Societies (COMPASS), Virtual Event, Australia, 28 June 2021; pp. 288–298. [CrossRef]
4. *ISO 9241-210:2019*; Ergonomics of Human-System Interaction. International Organization for Standardization: Geneva, Switzerland, 2019. Available online: https://www.iso.org/standard/77520.html (accessed on 31 January 2024).
5. Backhaus, N.; Brandenburg, S.; Trapp, A. Positive Technology and User Experience for Human Needs in Developing Countries: Some Considerations. In *Proceedings of the DUXU 2014*; Marcus, A., Ed.; Springer International Publishing: Cham, Switzerland, 2014; Volume 8519, pp. 400–410. [CrossRef]
6. Chetty, M.; Tucker, W.D.; Blake, E.H. Developing Locally Relevant Applications for Rural Areas: A South African Example. In *Proceedings of the SAICSIT '04; SAICSIT*; ACM: New York, NY, USA, 2004. [CrossRef]
7. Dearden, A.; Rizvi, S. Participatory IT Design and Participatory Development: A Comparative Review. In *Proceedings of the PDC'08*; Indiana University: Indianapolis, IN, USA, 2008; pp. 81–91. [CrossRef]
8. Heeks, R. Information Systems and Developing Countries: Failure, Success, and Local Improvisations. *Inf. Soc.* **2002**, *18*, 101–112. [CrossRef]
9. Teka, D.; Dittrich, Y.; Kifle, M. Contextualizing User Centered Design with Agile Methods in Ethiopia. In *Proceedings of the 2017 IEEE AFRICON*; IEEE: Piscataway, NJ, USA, 2017; pp. 911–916. [CrossRef]
10. Maunder, A.; Marsden, G.; Gruijters, D.; Blake, E. Designing Interactive Systems for the Developing World—Reflections on User-Centred Design. In Proceedings of the 2007 International Conference on Information and Communication Technologies and Development, Bangalore, India, 15–16 December 2007; IEEE: Piscataway, NJ, USA, 2007; pp. 1–8. [CrossRef]
11. Sherwani, J.; Palijo, S.; Mirza, S.; Ahmed, T.; Ali, N.; Rosenfeld, R. Speech vs. Touch-Tone: Telephony Interfaces for Information Access by Low Literate Users. In *Proceedings of the 2009 International Conference on Information and Communication Technologies and Development (ICTD)*; IEEE: Piscataway, NJ, USA, 2009; pp. 447–457. [CrossRef]
12. Winschiers, H. The Challenges of Participatory Design in a Intercultural Context: Designing for Usability in Namibia. In Proceedings of the PDC 2006, Trento, Italy, 1 August 2006; pp. 73–76.
13. Winschiers-Theophilus, H.; Bidwell, N.J. Toward an Afro-Centric Indigenous HCI Paradigm. *Int. J. Hum.-Comput. Interact.* **2013**, *29*, 243–255. [CrossRef]
14. Doerflinger, J.; Dearden, A. Evolving a Software Development Methodology for Commercial ICTD Projects. *Inf. Technol. Int. Dev.* **2013**, *9*, 43–60.
15. Tyukala, M. *A Framework to Elicit User Requirements for Information Systems: A Localised Participatory Approach from Southern Africa*; Nelson Mandela Metropolitan University: Gqeberha, South Africa, 2014.
16. Zewge, A.; Dittrich, Y.; Bekele, R. Software Designing Methodology for ICT4D Domain. In *Proceedings of the 13th Participatory Design Conference: Short Papers, Industry Cases, Workshop Descriptions, Doctoral Consortium Papers, and Keynote Abstracts—Volume 2; Windhoek Namibia, October 6th–14th 2014*; Association for Computing Machinery: New York, NY, USA, 2014; pp. 237–241. [CrossRef]
17. Maunder, A.J. *Assessing the Suitability of User Centric Design Methods When Applied in a Developing World Context*; University of Cape Town: Rondebosch, South Africa, 2010.
18. Puri, S.K.; Byrne, E.; Nhampossa, J.L.; Quraishi, Z.B. Contextuality of Participation in IS Design: A Developing Country Perspective. In *Proceedings of the PDC 04*; ACM Press: New York, NY, USA, 2004; Volume 1, p. 42. [CrossRef]
19. Winschiers-Theophilus, H. Cultural Appropriation of Software Design and Evaluation. In *Handbook of Research on Socio-Technical Design and Social Networking Systems*; IGI Global: Hershey, PA, USA, 2009; pp. 699–710. [CrossRef]
20. Hofstede, G. Cultural Differences in Teaching and Learning. *Int. J. Intercult. Relat.* **1986**, *10*, 301–320. [CrossRef]

21. Ostrosi, E.; Choulier, D.; Kurth, M. A Meta-Model for Engineering Analysis in Product Design. In *Concurrent Engineering Approaches for Sustainable Product Development in a Multi-Disciplinary Environment*; Stjepandić, J., Rock, G., Bil, C., Eds.; Springer: London, UK, 2013; pp. 545–556. [CrossRef]
22. De Silva, L.; Ginige, T.; Di Giovanni, P.; Mathai, M.; Goonetillake, J.; Wikramanayake, G.; Sebillo, M.; Vitiello, G.; Tortora, G.; Tucci, M.; et al. Interplay of Requirements Engineering and Human Computer Interaction Approaches in the Evolution of a Mobile Agriculture Information System. In *Proceedings of the UsARE 2014*; Ebert, A., Humayoun, S.R., Seyff, N., Perini, A., Barbosa, S.D.J., Eds.; Springer International Publishing: Cham, Switzerland, 2014; pp. 135–159. [CrossRef]
23. Ostrosi, E.; Fougères, A.-J.; Zhang, Z.-F.; Stjepandić, J. Intelligent Modular Design with Holonic Fuzzy Agents | Advances in Manufacturing. *Adv. Manuf.* **2021**, *9*, 81–103. [CrossRef]
24. Fougères, A.-J.; Ostrosi, E. Fuzzy Agent-Based Approach for Consensual Design Synthesis in Product Configuration. *Integr. Comput.-Aided Eng.* **2013**, *20*, 259–274. [CrossRef]
25. Ostrosi, E.; Fougères, A.-J.; Ferney, M. Fuzzy Agents for Product Configuration in Collaborative and Distributed Design Process. *Appl. Soft Comput.* **2012**, *12*, 2091–2105. [CrossRef]
26. Badampudi, D.; Wohlin, C.; Petersen, K. Experiences from Using Snowballing and Database Searches in Systematic Literature Studies. In Proceedings of the 19th International Conference on Evaluation and Assessment in Software Engineering (EASE 2015), Nanjing, China, 27–29 April 2015; ACM Press: Beijing, China, 2015. [CrossRef]
27. Wohlin, C. Guidelines for Snowballing in Systematic Literature Studies and a Replication in Software Engineering. In Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering, London, UK, 13–14 May 2014; ACM: New York, NY, USA, 2014; pp. 1–10. [CrossRef]
28. Google Scholar. Available online: https://scholar.google.com/ (accessed on 4 March 2024).
29. Research Rabbit. Available online: https://researchrabbitapp.com/ (accessed on 5 March 2024).
30. Unesco World Bank Open Data. Available online: https://data.worldbank.org (accessed on 4 January 2024).
31. Digital Skills in the Global South: Gaps, Needs, and Progress. Available online: https://iap.unido.org/articles/digital-skills-global-south-gaps-needs-and-progress (accessed on 11 January 2024).
32. Eberhard, D.M.; Simons, G.F.; Fenning, C.D. *Ethnologue: Languages of the World*, 26th ed.; SIL International: Dallas, TX, USA, 2024; Available online: https://www.ethnologue.com/browse/countries/ (accessed on 4 January 2024).
33. Sagot, S.; Ostrosi, E.; Lacom, P. Computer-Assisted Culturalization Process Integration into Product-Website Design. *J. Ind. Inf. Integr.* **2022**, *26*, 100252. [CrossRef]
34. Medhi Thies, I. User Interface Design for Low-Literate and Novice Users: Past, Present and Future. *FNT Hum.-Comput. Interact.* **2015**, *8*, 1–72. [CrossRef]
35. Danis, C.M.; Ellis, J.B.; Kellogg, W.A. Mobile Phones for Health Education in the Developing World: SMS as a User Interface. In *Proceedings of the ACM DEV '10*; Association for Computing Machinery: New York, NY, USA, 2010. [CrossRef]
36. Blumenstock, J.E.; Eagle, N. Divided We Call: Disparities in Access and Use of Mobile Phones in Rwanda. *Inf. Technol. Int. Dev.* **2012**, *8*, 1–16.
37. Heimgärtner, R. Using Converging Strategies to Reduce Divergence in Intercultural User Interface Design. *J. Comput. Commun.* **2017**, *5*, 84. [CrossRef]
38. Heimgärtner, R. Towards a Toolbox for Intercultural User Interface Design. In Proceedings of the 3rd International Conference on Computer-Human Interaction Research and Applications, Vienna, Austria, 20–21 September 2019. [CrossRef]
39. Heimgärtner, R. IUID Method-Mix: Towards a Systematic Approach for Intercultural User Interface Design (IUID). *J. Comput. Commun.* **2019**, *7*, 162–194. [CrossRef]
40. Schwaber, K. SCRUM Development Process. In *Proceedings of the Business Object Design and Implementation*; Sutherland, J., Casanave, C., Miller, J., Patel, P., Hollowell, G., Eds.; Springer: Cham, Switzerland, 1997; pp. 117–134. [CrossRef]
41. Hevner, A.R.; March, S.T.; Park, J.; Ram, S. Design Science in Information Systems Research. *MIS Q.* **2004**, *28*, 75–105. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.