Modelling Human-Centric Aspects of End-Users with iStar

Harshita Singh[a], Hourieh Khalajzadeh[a], Sahba Paktinat[b], Ulrike M. Graetsch[a], John Grundy[a]

[a] *HumaniSE Lab, Monash University, Melbourne, Australia*
[b] *TOSAN Corporation, Tehran, Iran*

Abstract

Purpose: Human-centric characteristics of the end-users of software systems, such as gender, age, emotions, personality, language, culture, and physical and mental impairments, play an essential role in the uptake and usage of the software. Current software tools suffer from the lack of in-depth elicitation and understanding of these human-centric requirements during the design and modelling of the system. This can lead to ineffective and hard to use software for some users. Methods: In this paper, to be able to account for contextual variables in human interaction with diverse characteristics, we propose an approach for using personas and contexts to model human-centric aspects of the software in goal models. To achieve this, we select the iStar language due to its ability to model social, intentional and strategic dimensions, and propose an extension of it to model human-centric aspects of the software. Our novel approach is illustrated with two examples. Results: We conducted user evaluation studies to understand how users model human aspects, and also to measure the effectiveness of our approach. Results show the lack of consideration of the human-centric aspects in existing modelling frameworks and how our extended model can simplify the understanding and addressing of such aspects. Conclusions: This shows and encourages that more research on modelling human aspects of the end-users is required to achieve human-centred modelling.

Email address: `hourieh.khalajzadeh@monash.edu` (Hourieh Khalajzadeh)

Keywords: modelling, human-centric aspects, istar extension, persona, contextual modelling, PRISE guidelines

1. Introduction

Software is used by diverse end-users with different characteristics including differing ages, gender, culture, language, personalities, technical proficiency, emotional reactions to software systems, socioeconomic status, and preferences. Even when users belong to one category, e.g. same age group, and have a general objective, what works for one person might not be acceptable for another since each individual has a different perspective, attributes and expectations. We call these diverse characterises as human aspects of the end-users of the software systems [1]. Such characteristics need appropriate considerations in different aspects of the software [2, 3, 4].

Possible consequences for the omission of human aspects may include huge economic costs, inefficiencies, not fit-for-purpose solutions, or dangerous and potentially life threatening situations [5]. Negligence of human aspects leads to the software – primarily designed and built to solve human needs – not fulfilling the end-users' expectations and causing frustration [6, 7, 8]. It can further lead to dissatisfaction and extra costs if the user wishes to resolve these issues [5].

When handling human-centric aspects in software design, developers need to be aware and carefully consider the characteristics, limitations, and abilities of the end-users, such as their age, satisfaction, preferences, working environment, and gender [9]. To better support the human-centric aspects of stakeholders and end-users of the software, it is essential to incorporate their human-centric aspects into the software engineering processes from early modelling and design stages. However, the existing software requirements and design models do not support the modelling of human end user characteristics. This often leads to important considerations being missed and diverse end users having serious challenges, such as accessibility and usability, using the resultant software systems [10].

Domain-Specific Languages are used to assist the development of complex software systems with implementation concerns such as usability, security, persistence, and business rules independently of the platform and the coding technology [11]. Existing requirements modelling languages, including iStar 2.0 [12], SysML and BPMN, have been designed to model software functional and non-functional requirements. A number of extensions of iStar have been made to explore different domains and concepts [13]. However, although there have been works on modelling emotions and interactions of the users [14, 15, 16, 17], there is no provision to model diverse human-centric aspects of software end-users.

To address this gap, we have used the concept of contextual modelling and personas to represent user groups with differing attributes, e.g. age, gender, culture and specific goals, e.g. book the quickest flight. We aimed to be able to model human-centric aspects of the software users in early requirement engineering stages. We selected iStar given it has the ability to model social, intentional and strategic dimensions, and was assumed to have a better potential to model human-specific dimensions. We extended the iStar 2.0 language by adding elements of persona contextual modelling to enable the modelling of human-centric aspects of diverse users that need to be considered during software design. We use persona model as a representation of the end-users of the system, that represents their characteristics, behaviour, goals and motives. Contexts refer to any human-centric information or requirement that can be used to characterise the situation of the persona [18].

We applied the iStar extension process per "PRocess to support iStar Extensions" (PRISE) guidelines [19], as the only extensive set of guidelines to extend iStar, and then conducted user evaluation studies to evaluate the extension. As part of the PRISE framework, we motivate this research using two examples. One is "a Flight Booking system", and the other is a "Smart Home" for elderly [6]. The main contributions of this paper include:

- A proposed new method for modelling human-centric aspects of end-users based on contextual modelling and personas.

- A proposed extension of iStar 2.0 language to model human-centric aspects of end-users.
- Evaluation of these extensions using real-world examples and a user evaluation.

The rest of the paper is structured as follows. Section 2 presents the background and Section 3 the motivation of the research. Section 4 presents our approach, following PRISE guidelines. Section 5 presents our evaluation setting, results, and the threats to the validity of our results. Section 6 briefly reviews the related work. Section 7 discusses and reflects on the key findings. Finally, Section 8 draws conclusions and proposes avenues for future work.

2. Background

In this section, we provide a background of the study covering the definition and examples of human-centric aspects, and an introduction to iStar language and PRISE guidelines, as the only formal set of guidelines to extend iStar.

2.1. Human-centric aspects

Understanding end-user human-centric aspects play an essential role in designing software that meets the requirements of diverse users. Various human-centric aspects are explored in the literature. We consider human-centric aspects, as the characteristics of diverse end-users of the software, such as age, gender, culture, physical and mental impairments, and so on [1], and here, we summarise the works taking into account such aspects in their design. Different **age** groups have different expectations, challenges, and reactions to the same software [20]. The increase in the average age of internet users provides evidence for a need to better cater for elderly users of software systems [10]. **Gender** bias in software applications, such as smart living technologies discussed in different works [21, 22], reflect the importance of taking gender-related issues into account when designing a software system. **Cultural** differences significantly influence the uptaking of the software systems. Therefore, beliefs and behaviours of the end-users of the system need to be incorporated into the design of the system [23]. Users speak in different **languages** and access the software from various **locations** all around the world. Hence, languages spoken by end-users and access location should be considered as part of requirements elicitation and when designing software systems. **Physical and mental impairments** of end-users impact the ways they are able to access the software. Impaired mobility, sight, hearing, speech and other challenges [3, 24] need to be taken into account when designing software systems [25]. Different users have various **emotional** reactions to the software based on the user interface, colours, icons, the information they present and so on. Such emotional impacts can influence the uptake of applications and need to be carefully considered [14].

2.2. Persona

A persona is a model of a user with a specific purpose, as a tool for software and product design, emphasising on the individual's goals when using an artefact [26]. The persona model is a representation of real or potential users, and represents patterns of users' behaviour, goals and motives. Persona is compiled in a fictional description of a single individual, and contains made-up personal details to make the persona tangible and alive for the development team. The idea of personas is part of the Goal-directed design approach to software design, developed by Cooper [27]. The word persona illustrates a model of a user who has a personality – "a life-like character driven by personal motives" [27]. Cooper's idea was that personality is important because it is a solution to a common problem in the design process [27]. Cooper recommends using a very specific individual – a persona – and directing the design for this specific individual, rather than referring to the general term of "the user" in design. Cooper believes that the purpose of using a persona is not to give a precise description or a complete theoretical model of a user, but rather a simple and good enough description of the user to make it possible to design the system. A good definition of a persona is provided by [28], as "User models, or personas, are fictional, detailed archetypical characters that represent distinct groupings of behaviours, goals and motivations observed and identified during the research phase." These fictional details include features, such as name, picture and some personal background details [27].

2.3. Contextual Modelling

Context is defined as "any information relevant to an interaction between a user and an application", such as a user's location, activities, emotions, or social setting [18, 29]. According to Dey et al.'s definition [18], the context includes any information (human-centric aspects in our case) that can be used to characterise the situation of entities (an end-user in our case) that are considered relevant to the interaction between a user and an application. Context is typically considered as the location, identity and state of people, groups and computational and physical objects. In our work, we focus on the state of people, or in other words, characteristics of the end-users of the system, i.e. human-centric aspects. Ali et al. [30] propose the contextual goal model to accommodate the relationship between goals and context. Context modelling requires the design and creation of modelling constructs to represent software and user context [30]. Our idea of contextual modelling is inspired by the work of Murukannaiah et el. [29], where the context is treated as a cognitive notion and is systematically related to the other cognitive notions such as goals and plans.

2.4. The iStar language

The iStar language, originally developed in the mid-nineties [31], is a goal- and actor-oriented language proposed to fill the gap of conceptual modelling languages. As specified by Dalpiaz et al. [12], the iStar modelling language was introduced to "fill the gap in the spectrum of conceptual modelling languages, focusing on the intentional (why?), social (who?), and strategic (how? how else?) dimensions." Dalpiaz et al. [12] introduced the iStar 2.0 core language, evolving the basic concepts of iStar into a consistent and clear set of core concepts. The iStar 2.0 language has been applied in various domains, such as healthcare systems, eCommerce, business modelling, security analysis, and so on [31]. It is used to model system requirements in the early phases of software development. We selected the iStar 2.0 language as it already provides the ability to model social, intentional and strategic dimensions [12], and we could add the human aspects dimension on top of the other dimensions.

The iStar 2.0 language represents social characteristics of systems in terms of *Actors*, their intentions, and relationships. In iStar 2.0, actors which are autonomous entities, can be of type *Role* or *Agent*. Actors aim to achieve their goals in collaboration with other actors and are graphically represented as circles. Role is an abstract characterisation of the behaviour of a social actor within some specialised context or domain of endeavour, and is represented by a curved line in the lower part. Agent is an actor with concrete and physical manifestations and is represented by a straight line being added in the top part of the actor circle. Actors are considered to have intentions that are modelled as goals to be achieved, tasks to be performed, resources to be used or levels of quality to be achieved. These intentions are modelled using *intentional elements*. An intentional elements included in the language are 1) *Goal*: what actor wants to achieve, 2) *Quality*: the level of achievement the actor desires, 3) *Task*: actions that an actor wants to be executed to achieve the goals, and 4) *Resource*: the entities that the actor needs to perform a task. Graphically, goals are represented as ovals, qualities as more curved cloud-like shapes, tasks as hexagons, and resources as rectangles. There are four types of links between intentional elements: Refinement, Needed-by, Contribution and Qualification. There are also dependencies or associations between actors (is a, part-of, plays, covers, occupies, instantiates), which represent the social aspect. Actors' intentions and relationships are contextualised within an actor's boundary. A boundary is a language concept with a graphical representation similar to a container that limits the scope of the actor [12].

2.5. iStar extensions guidelines (PRISE)

There are a number of prior studies where iStar 2.0 [12] and its predecessor iStar were extended by researchers to explore different domains and concepts

[13]. Although there are no formal extension mechanisms within the iStar 2.0 meta-model, a PRocess to support iStar Extensions (PRISE) was proposed by [19], guided by input from iStar researchers [32]. PRISE provides an end to end process and associated guidelines to define, evaluate, develop, validate, and publicise iStar extensions. The aim of PRISE is to make extensions as complete, consistent and without conflict as possible through a systematic process. PRISE involves three roles: 1) Extender, i.e., the researcher who develops the extension, 2) Expert in iStar extensions, and 3) Experts in domain/application area. When the experts in iStar or domain/application area propose the iStar extension, they play the role of Extender. The process consists of six sub-processes, as below:

Analyse the need for extension (Sub-process 1): The whole process starts with verifying the need for an extension. Based on the conclusions of this step, the process execution continues with the proposal. The outcome of this step is an Extension specification [Analysed]. This sub-process consists of 11 tasks (task 1.1-1.11), explained and used in the next section.

Describe concepts of the iStar extension (Sub-process 2): This step describes the concepts identified in the previous step in detail, and makes sure to reuse the existing constructs (if possible) and relate the constructs with the original iStar constructs. The outcome of this step is an Extension specification [Concepts described]. This sub-process consists of 7 tasks (task 2.1-2.7), explained and used in the next section.

Develop iStar extension (Sub-process 3): This is the main part of PRISE that helps to design the extension based on a set of guidelines. The outcome of this step is the Extension specification [Developed]. The guidelines, partially related to Moody's principles [33] are: 1) Preserve the language (iStar) original syntax; 2) Carry out consistent, complete and without conflicts extensions and follow a process/method to do them; 3) Perform a literature review, consider the participation of domain experts and iStar experts, and model systems of application area before extending; 4) Describe a clear definition of the extension concepts; 5) Propose concrete and abstract syntax of the extension; 6) Check consistency between abstract and concrete syntaxes; 7) Relate concepts introduced by the extensions with the iStar concepts; 8) Define extensions with a smallest possible number of modifications and new representations in order not to complicate the use of the modelling language (iStar); 9) Propose careful and simple graphical representations, able to be drawn on paper without a tool. The guidelines are detailed by Gonçalves et al., in [19]. This sub-process consists of 6 tasks (task 3.1-3.6), explained and used in the next section.

Validate and evaluate the iStar extension (Sub-process 4): This sub-process involves illustrating the usage and validating, refining and evaluating the extension with the help of experts. The outcome of this step is the Extension specification [Validated/evaluated]. This sub-process consists of 7 tasks (task 4.1-4.7), explained and used in the next section.

Check other new constructs to be introduced (Sub-process 5): This task is performed in parallel with sub-processes 2-4, in order to generate the list of concepts to be introduced through an iterative process. If new constructs are identified in this step, the execution of PRISE returns to task 2, otherwise, it continues to the next sub-process. This step does not include any specific tasks.

Publicise the iStar extension (Sub-process 6): Finally, the iStar extension is completed and needs to be made accessible to the community. This sub-process consists of 3 tasks (task 6.1-6.3).

3. Motivation

The aim of our work is to be able to model the system based on the requirements of individual users rather than the users in general. For example, a user might have a comfort specific requirement (comfort contexts) and wants to pay extra to be able to have a more comfortable option, while another user has more monetary specific requirements (monetary context) and wants to get a more affordable option. Our aim is to model such human-specific requirements of the end-users rather than the general functional and non-functional requirements of the system, or all the users, such as usability, reliability, effectiveness and performance. The motivation of our study is to explore how we can extend the existing goal modelling approaches to be able to model requirements specific to the human-centric aspects of the end-users of the software in early requirement engineering and design stages.

In this section, we present two examples to motivate our study, the Flight Booking example, and the Smart Home system for the elderly, presented in [6]. The reason for selecting two examples is to make sure we are not biased toward one example and are exposed to different scenarios. Also, we were able to discuss and model diverse human aspect characteristics. The Flight Booking system represents a hypothetical scenario that comprehensively covers all the aspects that a customer might require while travelling or booking a flight. This example gave us the option to have cultural diverse users with various requirements. Smart Home system's objective is to provide older adults with technology-based solutions to cater to physical and mental challenges so that they can independently take care of their health and feel safe. A person using the system can have different personal aspirations and goals. This example gave us the option to model a system that was already being modelled from the emotional aspects for elderly users, from the human centric aspects as well. The main goal of modelling these two examples with iStar language was to identify whether it is possible to specify all the desired domain concepts with iStar without creating new representations.

3.1. Flight Booking

To elaborate our approach, we chose two settings where different users can have different human-centric requirements. A Flight booking system is one such comprehensive setting. Every user using the system has a similar ultimate functional requirement of booking their travel flight ticket but each user reaches that goal based on his specific human-centric requirements. For instance, some customers might want to book the cheapest flight possible, others might want to keep their travel time to a minimum and do not mind spending extra on the flight. Some passengers might consider the language spoken the flight entertainment and some might prioritise the care and support they receive in their journey more. The flight booking system allows us to focus and capture such requirements and helps us in formulating our approach.

Figure 1 shows an iStar 2.0 model of two system functionalities: a) Booking a flight b) Saving user details. We determined that the iStar 2.0 successfully modelled all the system's functional and non-functional requirements, to be able to achieve the listed functionalities, but it had no scope for human-centric aspects.

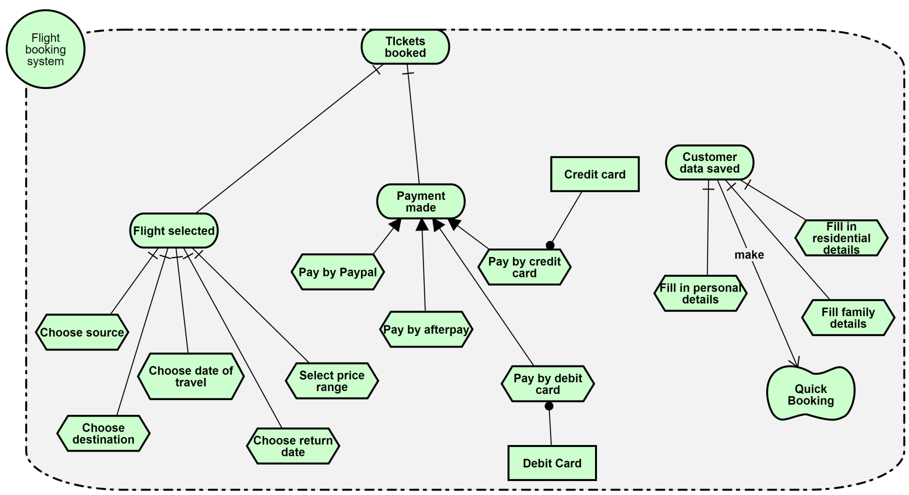

Figure 1: Flight Booking system using iStar 2.0

3.2. Smart Home

The Smart Home is a technologically enhanced home for the elderly that aids them in living independently. A person living in a smart home might have different conditions specific to them and require focus on that particular aspect. Some users might require sensors for fall detection, some might use a notification system in case of emergency. Some users might need multiple assistance at once. Hence, Smart home example allow capturing various individual needs.

Figure 2 shows an iStar 2.0 model of five system functionalities: a) Night time activity tracking, b) Fall detection, c) Emergency help, d) Reminders and e) Helping special needs of the Smart Home system. Similar to the previous example, we observed that iStar 2.0 was able to model the functional and non-functional requirements well to be able to achieve the listed functionalities, but did not cater for all possible human-centric aspects.

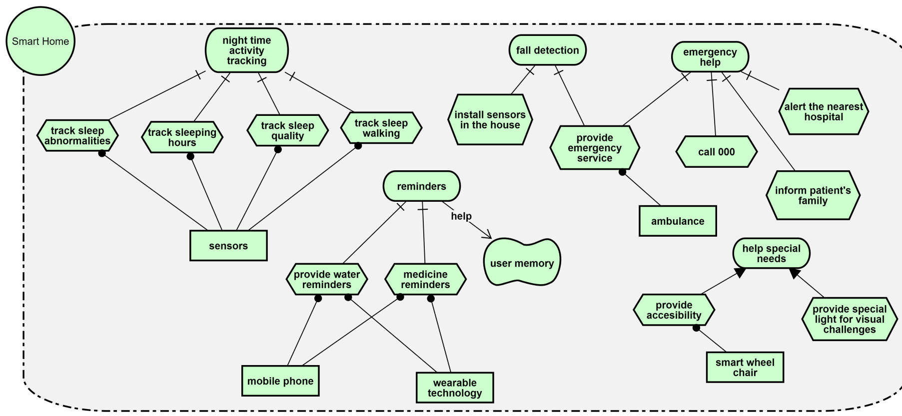

Figure 2: Smart Home system using iStar 2.0

4. Our Approach

In this paper, we followed the PRISE guidelines to propose a new extension of iStar. PRISE, as a reference process to support iStar extensions, can be adjusted based on particular situations [19]. In this paper, we illustrate a customisation of PRISE to the usage of an extender who is an expert in the application area, as shown in Figure 3. In this customisation we maintained relevant tasks which are the core of PRISE and highlighted the steps that are skipped. The following presents our approach using the PRISE stages [19]. By following this steps, we aim to be able to model the system for a specific individual, i.e. persona, rather than a general user. Our iStar extension enforces the modeller to come up with personas of the users and define the contexts for the specific persona, and then model the system using the contexts specific to the persona.

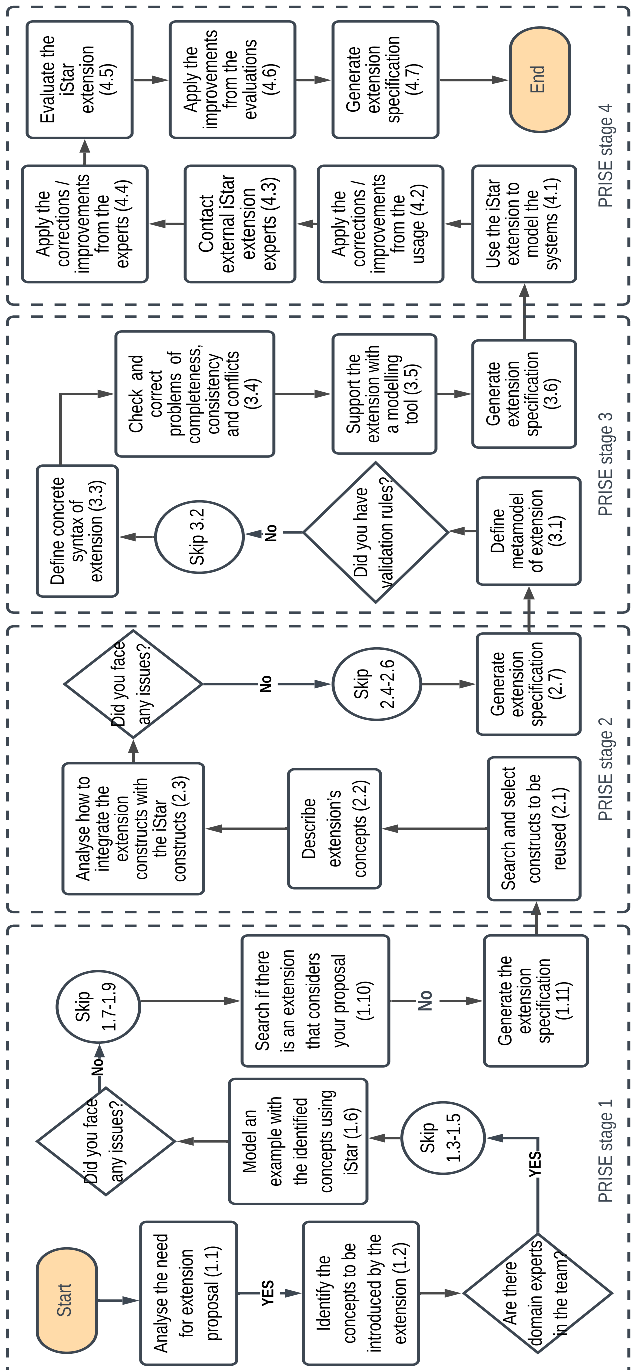

Figure 3: An overview of our iStar extension method

4.1. Analyse the need for extension proposal — study/review a domain/application area (task 1.1)

As discussed in the introduction, human-centric aspects play an essential role in the uptake of software by diverse users, and they need to be taken into account starting in early stages of software development. The human-centric aspects are architectural in the sense that they have a fundamental impact on the design of the software and, consequently, cannot be easily retrofitted. The PRISE process suggests that additional domain expertise may need to be consulted. In fact, our team had a number of researchers experienced in human-centric aspects of the software systems. Although we did not contact additional experts in the field, we conducted a comprehensive literature review of domain-specific modelling languages proposed for modelling human-centric aspects. We present a summary of the most related works in Section 6.

4.2. Identify the concepts to be introduced by the extension (task 1.2)

To model both the system and **human-centric requirements**, we combined the elements of **persona contextual modelling** along with iStar 2.0. The persona contextual modelling helps enhancing user understandability and provides a provision to model the persona's goals and attributes in various contexts [34].

Given the extender team included experts in the application area, it was not necessary to consult an expert in application area [19]. Thus, the tasks 1.3, 1.4, 1.5 which are related to consulting experts in the application area were not needed to be performed. However, we had several internal discussions about the concepts and the application area. We then continued from step 1.6.

4.3. Model an example with the identified concepts using iStar (task 1.6)

We were uncertain if it was possible to model human-centric aspects, such as age and gender-specific, cultural, personality related and emotional requirements with iStar without doing the extension. Consequently, as a validation to the conclusions in step 1.2., guided by PRISE, we tried to model some examples using iStar 2.0. To avoid problem-related biases, we selected the two examples presented in Section 3.

Observations: From these two examples, we can observe that the above models can capture the system functional/non-functional requirements effectively to a very granular level. For example, iStar 2.0 was able to capture the flight booking system's functional/non-functional requirements like selecting the flight, making the payment, and payment security well, but could not cater to individual human-centric aspects like variety and luxury. Similarly, for the Smart home system, iStar 2.0 was able to capture the functional/non-functional requirements about tracking sleep, receiving medicine reminders, and keeping private data encrypted, but was not able to capture every individual human-centric need of self-sufficiency, user authority/power and privacy.

Apart from the primary offerings that the overhead Flight Booking and Smart Home systems offer, users might also want some additional features based on their nationality, language preferences, health state, affinity with technology, comfort requirement and special needs. These requirements might vary from person to person and need to be modelled to achieve maximum user satisfaction. The current framework of iStar 2.0 has provision to only model the system goals, soft goals (qualities in iStar 2.0), resources and tasks. Our research in this paper is trying to bridge this gap and create an iStar 2.0 modelling extension that successfully captures both system goals and human-centric aspects of the user requirements.

In this step, if the extender encounters an issue, they need to contact experts in iStar extensions (task 1.7), the expert in iStar extensions receives and mitigate the issues (task 1.8), and the extender receives a response (task 1.9). Since we did not have any issues, we continued to the the next step (task 1.10).

4.4. Search if there is an extension that considers your proposal (task 1.10)

We searched the catalogue of iStar extensions[1] [35] for existing extensions modelling human-centric aspects. We also checked with the catalogue manager to make sure the catalogue is up to date. No extension has been proposed for our purposes, as discussed in Section 6, and therefore, we generated the extension specification [Analysed] (task 1.11).

4.5. Search and select constructs to be reused (task 2.1)

We conducted two different searches and analysed results to define suitable extensions. First, we searched for extensions that model contextual information. Second, we looked for extensions using colour in their notations, to help with conveying emotions, adding variety and interest to our designs [36].

4.5.1. Context modelling

We found extensions that model contextual aspects [37, 38, 39, 30, 40, 29]. Ali et al. introduce contextual goal models to relate goals and contexts [30]. In this work, context refers to a partial state of the world that is relevant to an actor's goals, and is represented by annotating the notations. Our context notation was inspired by the work of Murukannaiah et al. [29] since they are also creating abstract context and then diving into plans and goals. In their modelling, they have attached the context diagram to goal, quality notations and plan. Their dependency link notation is different and is called context means. The context notation presented in this paper is oval, however, we selected a circle shape to resemble the actor. Upon analysis, none of the other contextual goal modelling extensions represent the constructs we intend to represent and we did not consider them further.

4.5.2. Use of colours

Colours convey emotions, add variety and interest to our designs, and separate distinct areas of a page [36]. Use of colours can facilitate recognition, re-

[1] `https://istarextensions.cin.ufpe.br/catalogue/publication/list`

duce memory load and improve navigation and scanning speed. When selecting colours, conditions such as colourblindness, and users susceptibility to migraine or those suffering certain health conditions should be considered. Photosensitive seizures can be triggered by certain colours, such as red and blue [41]. Physics of Notation (PoN) principles [33] provided a useful guide on how to assess colour choices. We identified a number of potential iStar extensions that used colour: Colour schemes is used in several works [42, 43, 44, 45], to differentiate states of notations, but not to differentiate the actual elements. Amyot et al. use a colour scheme to highlight intentional elements to visually differentiate the goals that are satisfied (green), neutral (yellow), or denied (red) [43]. Ribeiro et al. use colours for the stereotypes containing the name of the specialised construct [44]. Specific colours are used by Morandini et al. for different notations [46]. The proposed colours pose a challenge because they are not differentiable for red-green and Monochromacy/Achromatopsia colour-blindness types[2]. Different colours are used in [47]. However, colour is used as the only element differentiating different notations, i.e., environment and users. This is insufficient to meet PoN guidelines. We used the original iStar 2.0 colours and a new, differentiable colour (orange) for our new notation element.

4.6. Describe extension's concepts (task 2.2)

Here we identify the description of the concepts we identified in task 1.2, persona contextual modelling.

> **Definition 1 - Persona:** A persona represents a user category and consists of human factors, attributes and goals specific to that user group.

> **Definition 2 - Context:** The persona's human-centric objectives can be classified into various contexts depending on the nature of the goal.

[2] `https://www.color-blindness.com/coblis-color-blindness-simulator/`

An example of a persona is an imaginary woman within her 60s, who does not speak English and wants to book a flight, with several human-centric objectives, i.e. contexts such as comfort context (getting an aisle seat), cultural context (receiving translated instructions, food preferences), and so on. The persona goals should be based on the persona attributes and the system that persona is going to use. The goals are then classified into various contexts. These context act like the broader scope of the goals based on which the persona would make decision or have certain requirements. The context will enhance the documentation and modelling, especially when there are many personas. Each persona will have specific goals. These goals can be classified into broader context. Their individual needs can then be mapped as a task to fulfil each persona's individual needs. This makes capturing specific individual needs easier and produces a detailed specification. Each context depends on the problem that is being modelled. Different modelling problems might have altogether different contexts. The context should reflect a broader concept to capture all possible user requirements in the model.

4.7. Analyse how to integrate the extension constructs with the iStar constructs (task 2.3)

Contexts, identified in task 1.2, can be incorporated in the iStar language using the "task" construct. Task constructs in iStar 2.0 describe the actions/solution that will help the user reach their goals. From the design viewpoint, the tasks can effectively be designed into features. Therefore, we decided to use the task construct of iStar 2.0, for modelling contextual human-centric aspects. While implementing human-centric requirements, the goal of the context would be realised by executing supporting tasks. One context can signify multiple tasks. Single or multiple tasks can contribute to goal fulfilment.

In case of any issues with the integration, the extender needs to contact experts in iStar extensions (task 2.4), the expert in iStar extensions receives and mitigates the issues (task 2.5), and the extender receives a response about the issues (task 2.6). Since we did not have any issues, we generated extension specification [Concepts described] (task 2.7).

4.8. Define metamodel of extension (task 3.1)

Figure 4 shows a meta-model of our iStar 2.0 constructs. The metamodel lays out the primary construct and defines the classes and attributes of the modelling tool. Our metamodel gives the underlying schema of our iStar 2.0 extension. We used Sirius Desktop tool[3] to create our metamodel and modelling tool. Our metamodel consists of metaclasses in Sirius 2.0 and also our newly added Task Context construct. The yellow colour blocks in the domain model refer to classes. The grey coloured blocks are the abstract classes. This metamodel was then used to model the scenarios.

The model consists of 15 classes. The diagram container represents the root node of the model and used by the Eclipse design file to refer to the domain model. The iStar diagram class can contain 0 to many actor elements and intentional elements. The actor elements class also can contain 0 to many intentional elements classes. The intentional element class is the superclass of resource, quality, task context (new notation) and goal task elements (goals and task). A quality class can have a qualification relationship with any of the intentional element classes, and all the other classes can have contribute relationship with the quality class. The goal and task elements of iStar 2.0 are kept separately under another class as these entities can have AND and OR refinement relationships between them. In addition to the relationships mentioned above, the domain model also has needed by and signifies relationships. Needed by relationship in the model is the relation between the task and the resource and can be used when a particular resource requires a specific resource to execute the task. The signifies relationship is between task and the task context attribute.

The actor element class is the supertype class of the agent class, actorIsAElement and roleIsAElement class. The Agent class has participated in a relationship with all the other courses of the actor elements. Actors and roles are kept

[3] `https://www.eclipse.org/sirius/`

separately under separate abstract classes as both constructs can have an isA relationship only with similar constructs. For instance, a role can have an isA relationship with another role. However, it can not have an isA relation with an agent or actor.

Since we did not have any validation rules which could not be represented with the meta-model, it was not necessary to add validation rules in this extension (task 3.2). Hence, we continued with the next task (task 3.3).

4.9. Define concrete syntax of extension (task 3.3)

Systems will become more complex as they accommodate human-centric requirements. User expectations and number of requirements will increase, to address various contexts. Not having a clear indicative notation might lead to confusion and issues in understanding a task's purpose. Having an explicit notation will also help the development process by allowing the development team to create acceptance criteria, i.e. conditions that software products need to meet to be accepted by the users, closely aligned with different contexts and associated user requirements. To allow better interpretation of the iStar 2.0 model, we have added a new notation in the iStar construct called "Task Context". Each task is attached to this new notation through a new "signifies" connection to signify the context they are fulfilling. The purpose of the new notation is to communicate additional context relevant to the task, and thereby enhance understanding of its purpose. The new notation, i.e. "Task Context", is a dotted circle attached to each task via a new dotted line, i.e. "signifies", as shown in Figure 5.

The notation selection was made according to the Physics of Notations (PoN) guidelines [33], as summarised in Table 1. According to the Complexity Management and Graphic Economy guidelines, we kept the count of notations to a minimum and only introduced two new notations for contexts and connecting them to the tasks. While choosing the notation, we wanted to keep a similar shape notation to that of actor, as the contexts were based on the user persona. The shape and the connector connecting it to the task also resemble a

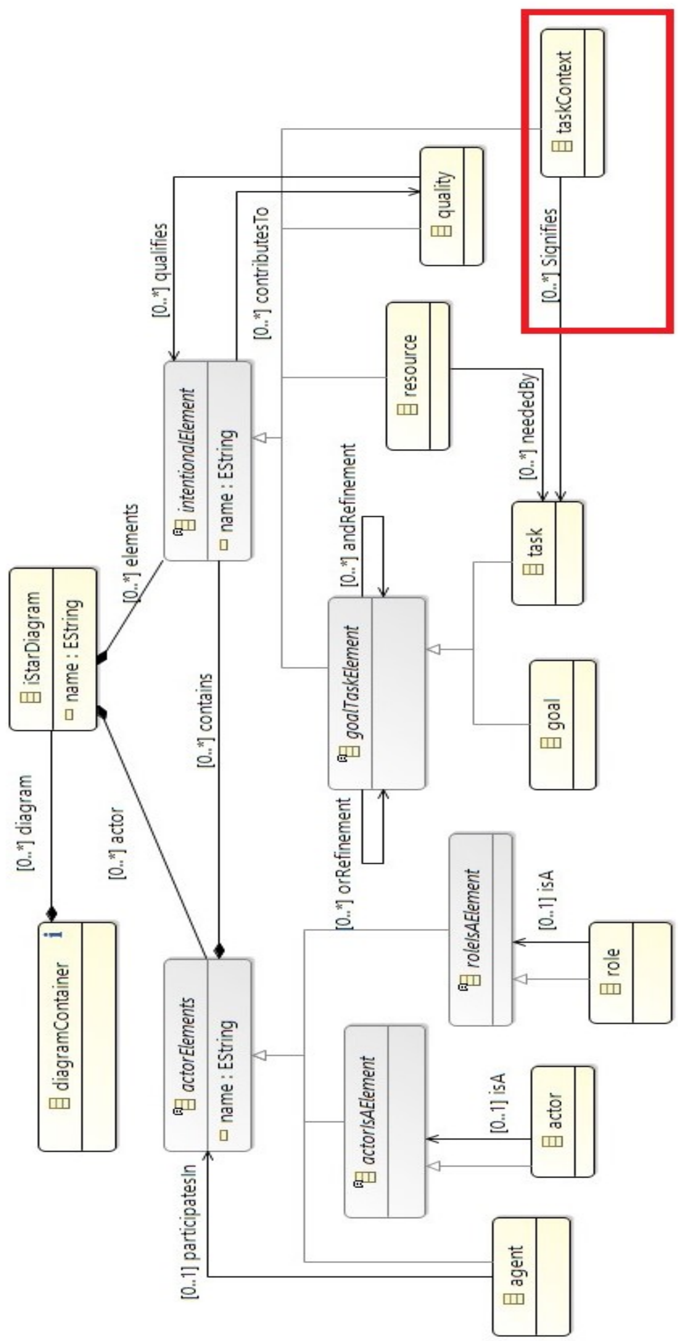

Figure 4: An overview of our meta-model

(a) Task context (b) Signifies

Figure 5: Our new - Task context and Signifies - notations

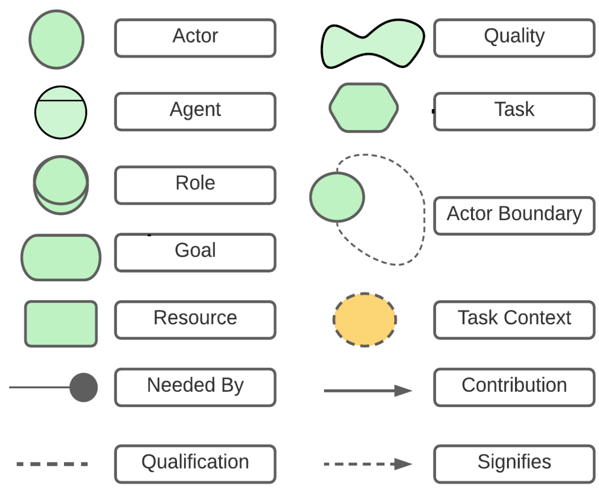

Figure 6: Our Notations

magnifying glass, and emphasises the significance of the human-centric aspect as a context. The shape's design can also be drawn easily by hand without any tools required. The dotted circle contains the context name. We maintained the other symbols proposed in the original version of iStar 2.0 as shown in Figure 6. We maintained the green colour of the iStar 2.0 original notations, as this is commonly used. However, to avoid the issues red colour might cause, such as photosensitive seizures [41], we opted for a pastel and opaque shade of yellow, for the new notations, as shown in Figure 6. We confirmed that users suffering various colour blindness types can differentiate the colours of the notations.

Table 1: How our iStar extension meets PoN Guidelines

PoN Guideline	*Definition*	*Our approach*
Semiotic clarity	A diagram should avoid symbol redundancy, overload, excess and deficit	All our visual symbols have 1:1 correspondence to their referred concepts
Perceptual discriminability	It is primarily determined by the visual distance between symbols.	All symbols we used have different shapes as their main visual variable, plus redundant coding such as colour and/or textual annotation. As colour is only used redundantly, our notations are suitable for handwritten diagrams and users with colour blindness
Semantic transparency	Identifies the extent to which the meaning of a symbol can be inferred from its appearance	We maintained the iStar default notations to make it easy for iStar users to remember the notations
Complexity management	Restricts a diagram to have as few visual elements as possible	We introduced only two new notations to depict all different human-centric aspects, rather than a notation for each aspect. This meets the goals of complexity management and also enables generalisation for emerging human-centric aspects
Cognitive integration	Identifies that the notations should support the user to assemble information from separate diagrams into a coherent mental representation of a system	We follow the iStar construct and therefore, each actor has its own actor boundary, which makes it easy to maintain the perceptual integration between different actors
Visual expressiveness	Refers to the use of a range of visual variables to result a perceptually enriched representation that exploits multiple visual communication channels	We used shape, size, brightness and colour to distinguish symbols and convey meaning
Dual coding	This means that textual encoding should also be used	All our visual symbols have a textual annotation
Graphic economy	Requires that the number of different visual symbols should be cognitively manageable	Only two extra visual symbols are added to the default iStar notations
Cognitive fit	Refers to having different visual dialects for different tasks or users	This is a limitation of iStar, however, as we borrow concepts and symbols from default iStar notations, the iStar users are already familiar with the concepts

4.10. Check and correct problems of completeness, consistency and conflicts (task 3.4)

We assessed our proposed symbols and concepts according to the *Checklist for verification of problems*. The checklist aims to ensure that proposed constructors are well-defined, are represented in the abstract syntax, do not introduce inconsistencies with the existing iStar 2.0 syntax. It also avoids conflicts by ensuring that each proposed concept only has a single concrete syntax representation [19]. In terms of completeness, our iStar extension was applied to the *concepts definition* and *concrete syntax levels*. The completed checklists covering consistency and presence of the iStar 2.0 syntax is shown in Table 2. In this table, X shows whether a concept is an iStar default syntax, and whether it is consistent with its definition, meta model, and concrete syntax. We only have two new concepts, that are consistent, and we did not make any changes to the existing iStar concepts. Regarding the conflicts, none of the concepts present any conflicts including: one construct with two or more symbols, two or more constructs with one Symbol, wrong representation of iStar constructs, and construct which is not part of the extension.

4.11. Support the extension with a modelling tool (task 3.5)

Our extension is implemented on Eclipse Sirius, an Eclipse foundation's open-source software project. It allows the creation of custom graphical modelling workbenches by using the Eclipse Modelling technologies. The modelling workbench created is composed of a set of Eclipse editors (diagrams, tables and trees) which allow the users to create, edit and visualise model. The meta model and design file generated from Sirius were used to automatically generate JSON code using the Acceleo plugin. We created the domain model of our iStar extension in Sirius to help create a framework that users can use to develop scenario-specific models. Our Sirius based modelling tool can be downloaded from [48].

Table 2: Consistency and presence of nodes and links of iStar default syntax

Consistency				
Concept	*iStar default syntax*	*Metamodel*	*Concrete syntax*	*Concept's definition*
Actor	[X]	[X]	[X]	[X]
Agent	[X]	[X]	[X]	[X]
Role	[X]	[X]	[X]	[X]
Resource	[X]	[X]	[X]	[X]
Quality	[X]	[X]	[X]	[X]
Task	[X]	[X]	[X]	[X]
Goal	[X]	[X]	[X]	[X]
Actor Boundry	[X]	[X]	[X]	[X]
Task Context	[ ]	[X]	[X]	[X]
Signifies	[ ]	[X]	[X]	[X]
Needed by	[X]	[X]	[X]	[X]
Qualification	[X]	[X]	[X]	[X]
Contribution	[X]	[X]	[X]	[X]

4.12. Generate extension specification (task 3.6)

In this step, we generated the Extension specification [Developed], and added the metamodel and syntax representations to the Extension specification [Concepts described].

4.13. Use the iStar extension to model the systems (task 4.1)

We now demonstrate how the iStar extensions can be used to model persona context aspects in our previously introduced Flight Booking and Smart Home scenarios.

4.13.1. Flight Booking

To further elaborate the scenario, we consider the hypothetical persona of Mary Andrews, shown in Table 3, who wants to use the Flight Booking system. Using our persona contextual modelling, the goals of the above persona can be classified into a) Time Context, b) Comfort Context, c) Monetary Context, d) Cultural Preference Context, and e) Age Context. The airline booking system is an agent actor that passengers will utilise to fulfil their functional, non-functional and human-centric goals. Each of the goals such as variety, luxury, book flight will be satisfied by one or more tasks. The task will directly be linked to the context problem that it is trying to resolve or serve. Each goal in the persona presented in Table 3 is a fact in the contexts mentioned above. These contextual goals can now be modelled by a single or group of tasks, as shown in Figure 7.

Table 3: Persona - Flight booking system

Attributes	Name: Mary Andrews
	Nationality: Australian
	Spoken Language: English
	Date of Birth: 19/01/1987
	Date of Boarding: 10/01/2021
Goals	Book a return flight from Melbourne to Christchurch, New Zealand
	Would prefer an aisle seat
	Wants vegan food option
	Can upgrade to business class if being offered a discount

4.13.2. Smart Home

We used the hypothetical persona in Table 4 to elaborate the system further using our extension model. Based on Mike's goals, the requirements can

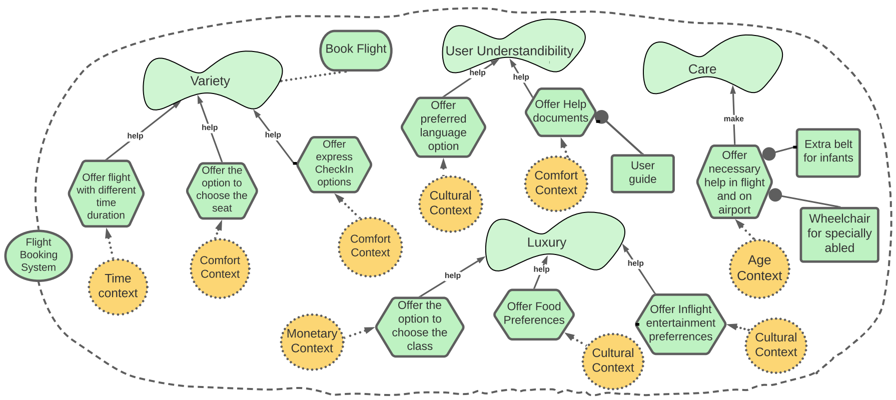

Figure 7: Flight Booking system using our iStar extension

be framed from the following contexts: a) Independent Living, b) Tracking Context, c) Control Context, d) Care Context, e) Reassurance Context and f) Responsive Context. The diagram in Figure 8 uses these contexts and models the requirements.

Observations: Using our iStar extension helped us to be able to identify human-centric aspects, convert them to the contexts and model them in an easy and straightforward way. Therefore, we were able to not only model the system goals, but also the users' human-centric goals, such as the additional features users require based on their nationality, language preferences, health state, comfort requirement and so on for different people.

We applied all the corrections/improvements from the usage in this step (task 4.2). Task 4.3 related to consulting experts in the application area was not needed to be performed. However, we contacted external iStar extension experts (authors of [13, 19]) to mitigate issues about integration. Through email communications, the feedback we received was: *"The context of your extension has a vast application and fits well in industrial projects."* The experts expressed their interest in the domain, and confirmed that we correctly followed the PRISE steps. We also received helpful feedback on using several other iStar extensions to compare with our work, reuse of the original iStar 2.0 notations and colours,

Table 4: Persona - Smart Home

Attributes	Name: Mike Ross
	Nationality: British
	Spoken Language: English
	Date of Birth: 10/04/1943
	Health problems: Arthritis, Low blood pressure, Diabetes
Goals	Wants to live alone without bothering his daughter
	Wants to have some support service provided in case of fall
	Wants to keep his medical data safe
	Wants reminders for taking medicines

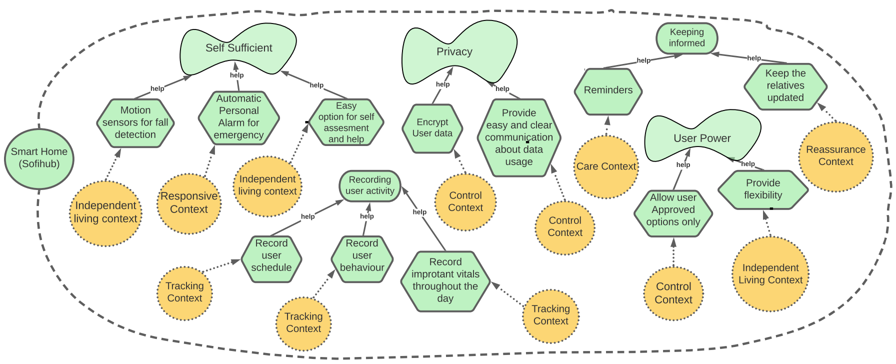

Figure 8: Smart Home using our iStar extension

and use of a different colour for our new notation. We updated the models according to the feedback we received. We considered the feedback received from iStar experts on the usage and made improvement/modifications accordingly (task 4.4). We conducted evaluations (task 4.5) and made modifications based on the results (4.6). The evaluations are detailed in Section 5. Finally, we generated the extension specification [Validated/evaluated]. Given new constructs were identified during sub-processes 2-4, we did not need to return back to sub-process 2 and were able to continue with sub-process 6. However, sub-process

6 is not discussed in this paper since we have not publicised our iStar extension yet.

5. Evaluation

We designed a control experimental design with two groups of participants, with a background in modelling, to understand how they currently model human aspects (control group) and whether our iStar language can help them in modelling the human aspects (experimental group). Therefore, we measured the time taken for both groups to complete the modelling and also whether they modelled any human aspect in their final model. Moreover, we asked the experimental group to rate our iStar extension and share their feedback with us for future improvements. Therefore, we aimed to answer the following research question:

> RQ: Can our extended iStar language encourage and facilitate the modelling of human aspects?

5.1. User evaluation study setting

To validate our extended model, we conducted a qualitative user evaluation study. The study consisted of two groups of 5 participants each - one control group and one experimental group. To avoid any bias, we kept the groups separated by running a separate Zoom session for each group. Our user evaluation study participants were limited to those with prior experience with modelling tools and modelling frameworks, since we wanted to avoid technical proficiency biases, e.g. not being able to model or rating the modelling process rather than modelling the human aspects. All participants were asked to provide details about modelling frameworks they were familiar with. Participants in both groups were asked to develop an imaginary persona with specific characteristics like age, gender, nationality, physical/mental impairments, and differing preferences. They were then asked to select a problem of their own or a given hypothetical situation and model the problem using any modelling tool of their choice [control group] and our iStar extension [experimental group]. The experimental group only were then given an introduction to our iStar extension framework. All participants were asked to draw the diagrams on a Lucidchart template and share the diagrams through an anonymous link. We recorded the total time taken for each participant to model their problem. Once participants completed their diagramming tasks, they were asked to fill in a survey that assessed their experience while modelling the problem for their imaginary persona. The experimental group received additional questions to evaluate their understanding and interpretation of the new notation for modelling human-centric aspects. The surveys consisted of both multiple-choice and subjective questions. The user evaluation study surveys as well as the personas and diagrams participants created can be accessed online [49]. The multiple-choice questions had five options in compliance with the 5 points Likert scale [50]. No personal details were collected, and the survey was kept anonymous.

5.2. User evaluation study results

5.2.1. Control Group

The control group participants consisted of two end-users, one system developer, one security engineer and one system designer. Their self-assessed modelling experiences were rated as one novice, two intermediates, and two experts. The participants ranged from 22–46 years old, and listed Flowchart, BPMN, UML, Lucidchart, ER diagram, workload automation as the modelling frameworks they were familiar with. For the experiment, two participants used Flowchart, two used UML, and one used BPMN to model their problem. Each member of the control group confirmed that they were able to model the problem for the persona that they created. The two Flowchart modellers did not identify any limitations of modelling the human-centric requirements of the persona. However, the UML and BPMN modellers did identify some limitations. Notations used by the control group and how they rated their difficulty of modelling human aspects is listed in Table 5. An interesting finding was that none of the diagrams, created by the control group contained any reference or notation to human-centric aspects particular to the persona they created. Even those participants who rated the difficulty of modelling human-centric aspects as "very easy" did not model human-centric aspects. All the diagrams, created by the participants, can be accessed online [49]. An example of a flight booking system modelled by a user, using BPMN, is shown in Figure 9. As it can be seen, even though the user was asked to model human-centric aspects, only the functional and non-functional aspects are modelled.

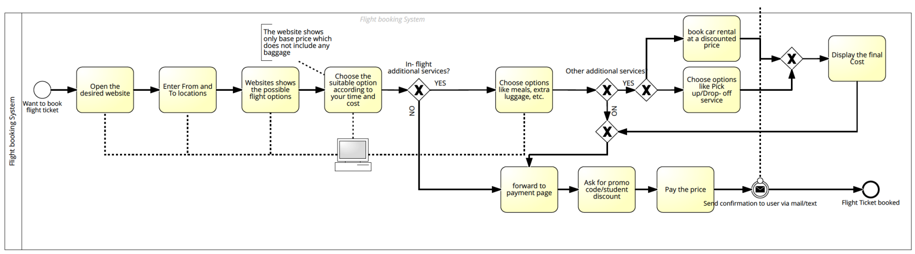

Figure 9: An example BPMN diagram created for flight booking

5.2.2. Experimental Group

The experimental group consisted of three end-users, one system developer, and one system designer. Their modelling experiences were rated as one novice, three intermediates, and one expert. Experimental group participants ranged between 24-52 years old, and listed UML, Flowchart, Data Flow Diagrams, and ER diagrams, as the modelling frameworks with which they were familiar. The experimental group created a persona and used our iStar extension to model a problem for their persona, after which they completed their survey. Diagrams created by the experimental group, using our iStar extension, can be accessed on [49]. The colours/notations used in the user evaluation study are updated in the latest version, according to the participants, and iStar experts' feedback.

Survey results are shown in Figures 10 and 11. The feature participants liked the most about the modelling extension include the graphical representation of the persona's goals and tasks, the goals and context, live diagram authoring,

Table 5: Notations used by the control group and their difficulty of modelling human aspects

modelling	*Difficulty of modelling (1: difficult, 5: easy)*	*Reason*
Flow chart	5	flowchart is a very basic modelling language and hence it was easy to map everything out as a process.
UML	2	It was very hard to come up with ways to model the persona goals.
Flow chart	5	I just mapped the system flows and requirements thinking about what my persona might do once he started using the system. The step by step mapping seemed pretty easy.
UML	4	Had to really think about the functionalities to fulfilling the persona goals. I created a high level diagram which was easy but creating detailed functionality based on persona's goal related to its attribute can get a little tough.
BPMN	3	The simple process was easy to map but took a while to figure out how to try and add persona goals in the process. The total process hence became quite lengthy.

easy approach to collaborate or discuss with other designers or developers, easy to understand notations, as well as the ability to show how both Quality and Context can influence the Task. They all replied "yes" to use this tool in future or recommend it to the others. The reasons provided for this recommendation were: it is nice and simple, it enables capturing the requirements, it is a useful next step in capturing human aspects in requirements engineering, and ease of communication and collaboration. One participant mentioned the modelling notation is good, however, Lucidchart is not ideal (tedious to move actor boundaries, can not use for code generation). We only used Lucidchart for the experiment, and plan to develop a web-based version of the tool using Sirius-web to make future experiments easier.

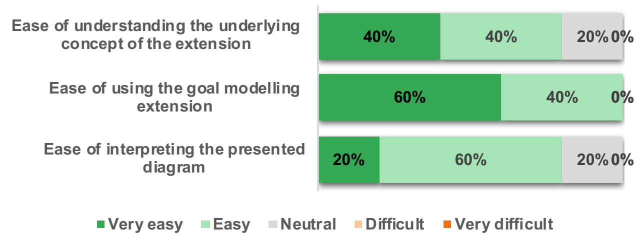

Figure 10: Difficulty of using our iStar extension

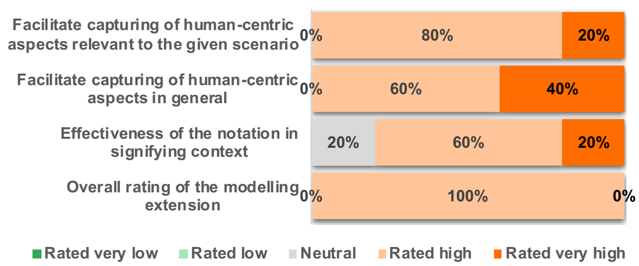

Figure 11: Rating of our iStar extension

The experimental group provided valuable feedback about the features. One participant thought the task context signifier lacked clarity and suggested to use a plain language concept. We actioned this feedback after the user evaluation study and included two separate notations to simplify the concepts. Another participant stated that the notation was overwhelming initially because there are many connectors, and suggested the use of different colours for the Context and Quality. We actually had used different colours for Context and Quality (i.e., soft goal), however, the participant might not have perceived the difference. One participant mentioned that “Soft goals and Context can sometimes influence each other in non-trivial ways, but it can be difficult to capture this complexity”. This participant suggested additional features including the ability to add further details (e.g. as a comment / meta-data) in case of complex interrelationships (e.g. “because I have a broken arm, I can only type slowly and want to minimise unnecessary movement"). Other requests included the ability to distinguish notations between positive (help, make) and negative (break, hurt) factors. More general diagram tooling features were also suggested - including support for auto-layout, when connecting a task to multiple context and routing the links to be non-overlap, and interactions such as showing tooltip or highlight of elements on mouse hover.

An example of a blood donation system modelled by a user, using our iStar extension, is shown in Figure 12. As it can be seen, our context notation has encouraged the user to model the human-centric aspects on top of the functional and non-functional requirements. These human aspects include work/life balance for time schedule, memory ability due to the age for appointment reminders, and so on.[4]

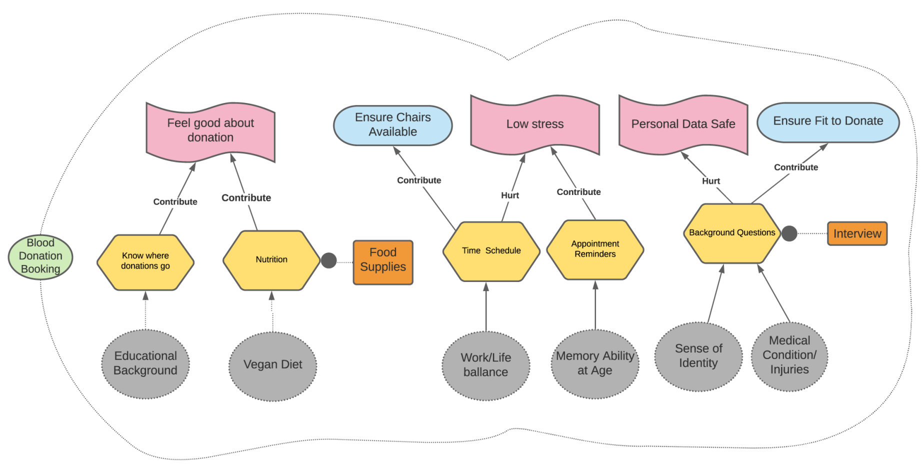

Figure 12: An example of our iStar extension diagram created for blood donation

5.2.3. Comparing the groups

We compared the two groups from two aspects: 1) whether they modelled the human aspects of the system, and 2) the time it took for them to model

[4] Note: The colours are different from the final colours since we changed the colours based on the user evaluation study feedback.

Table 6: The comparison of the time taken (in minutes) for control (and the notations they used) and experimental groups.

Experimental group	34	26	27	21	25
Control Group	70	30	40	55	83
	BPMN	Flowchart	Flowchart	UML	UML

the system. As shown in the examples, even though both groups were asked to model the human aspects of the system according to the persona they created, the models generated by the control group did not include modelling of any human aspects, while the context notation in the experiment group made the users think of and try to model human aspects.

Table 6 shows that participants in the experimental group spent less time (on average 26.6 minutes versus 55.6 minutes) in modelling the problem for their imaginary persona than the control group. It indicates that our tool helped modellers save the modelling time. The results also indicate it made modelling the human-centric aspects easier and more straightforward for them. The 5 participants who used our iStar extension (M = 26.6, SD = 4.72) compared to the 5 participants in the control group (M = 55.6, SD = 21.54) spent significantly less time to model their chosen problem for their imaginary persona. We used the Mann-Whitney U test [51], which is specifically designed for small samples [52], to understand the significance of the differences between the two groups. Mann-Whitney two-tailed U test results indicate that the U-value is 1. Given the critical value of U at $p < .05$ is 2, therefore, the result is significant at $p < .05$. Moreover, the z-score is -2.29783, and the p-value is .02144. Therefore, the result is significant at $p < .05$. It suggests that the reduction in modelling time experienced with the iStar extension is significant (p-value $< .05$). The experimental group were able to include human-centric aspects of the problem in the iStar model, and they were able to do this faster than the group that used alternative modelling notation and did not include human-centric aspects in their models. We believe that the decreased time of modelling is due to the assistance of our iStar extension which gives participants an easy starting point for modelling human-centric aspects of a persona through contexts. It needs to be noted that this quantities are for orientation purposes only, based on a relatively small sample and are not considered as hard values.

5.3. Threats to the validity

There are threats that may have affected the results and conclusion of our study. We considered controls to reduce the threats to validity and their impact on our user evaluation studies. In this section, we discuss these threats and the controls applied to them.

5.3.1. Internal validity

There are currently no other suitable baseline modelling languages to model the human aspects of the users. Hence, we were not able to compare our extension with any existing tool. Therefore, we asked the control group to select a modelling language of their own. Selected languages, such as UML, are general-purpose notations, and might not be comparable with the i-Star extension. However, this helped us to understand what current approaches they use to be able to model human aspects. We aim that our approach acts as a benchmark for the future research on modelling human aspects of the end-users. Moreover, to avoid biasing toward the projects' complexity, we provided the participants with the freedom to select the project of their own. This might impact the time taken and make them not comparable. However, this allowed us to evaluate the extension under different complexities, and let the participants focus on the modelling part rather than understanding a new project. Finally, there is a risk of participants experiencing boredom and slowness as the user evaluation study progressed over time. However, participants were advised that the evaluation was voluntary and there was no obligation for them to complete the evaluation study and they could leave whenever they wanted. Also, we made sure to keep the session as short as possible and quite interactive to avoid boredom.

5.3.2. External validity

One of the threats to the validity of our results is the small sample size. The restrictions imposed by the COVID-19 pandemic limited us in running more interactive face-to-face user evaluation studies, and recruiting more participants. However, we had a representative sample of participants with diverse roles, ages, genders, and so on. We successfully recruited a range of novice to expert modellers. Overall, we had two novice, five intermediate, and three expert modellers. The novice modellers had basic experience with modelling. This helped us to be able to evaluate our iStar extension's ability to model human aspects without worrying about the modelling challenges the participants might come across due to their technical proficiency. Moreover, all our participants had some prior knowledge in modelling, and therefore, our number of expert participants is in the accepted range [53, 54, 55]. Finally, to limit the impact of the small number of participants on the results of the study, we applied the Mann-Whitney U test, which is designed for small sample sizes.

6. Related Work

6.1. Modelling human-centric aspects of software

Miller et al. [14] extended Sterling and Taveter's agent-oriented modelling notation to capture emotions for emergency systems mobile apps. Miller et al. introduced the People Oriented Software Engineering (POSE) method. POSE is an approach to capture emotional desires by using emotional goals. The authors classify emotional goals into personal emotions and context-specific emotions. They used a survey to evaluate and compare their emotional model against iStar by implementing two domain models in both iStar and POSE. The survey results identified that participants preferred the use of POSE models because they are clearer, easier to understand, and not complicated to interpret. Participants mentioned that they were not confident to make modifications to iStar models, and would prefer to modify POSE model if required. Moreover, Lopez-Lorca et al. [15] used personas and scenarios to explore the diversity of users, and ensure that emotional desires are met. They described their experience in three projects in emergency mobile apps for older adults, diagnosis of depression, and a web application for self-managed treatment of psychosis. Data are gathered form interviews and other ethnographic studies and personas are built based on composing textual description of personality traits. Emotional scenarios are script templates which explore how different personas react in identical situations. These two works mainly focus on the emotions of the users.

Curumsing et al. [6] used an emotion-oriented approach to improve identifying, modelling and evaluating emotional goals in requirements engineering. The authors demonstrate how they used this approach to develop a Smart Home platform for elderly people. They used several modelling supports for the emotion-oriented requirements engineering such as role model, goal model, motivational scenarios, interaction model, scenario model, and behavioral model. In order to obtain emotional requirements, they adopted two existing techniques: content analysis and affinity diagram. Personas were used to assist applying captured requirements in the designing phase. Evaluations were done through the Attrakdiff questionnaire, which assess users' feelings. Watson et al. [16] examined human and system considerations using the system modelling language (SysML). Laurenzi et al. [17] proposed a modelling language to support user interactions in heterogeneous physical scenarios. The model helps designers identify the services that will be required by the users to support their activities. It is assumed that systems modelled by this language are structured as Human-centric Wireless Sensor Network (HWSN). The nodes participating in an HWSN can be human-based sensors, regular sensors, mules, witness units and actuators. Finally, a wire-frame extension method is presented in [10], to incorporate end-user diverse ages into the design of the software. However, these works focus on the emotional and interaction aspects of the users. However, these works focus on user's interaction with the system. We focus on modelling the human factors associated with the end-users of the system, rather than the user's interactions. To the best our knowledge, our work has been the first attempt to model human-centric aspects of the end-users of systems.

6.2. The iStar extensions

A systematic literature review of the iStar extensions is presented in [13]. Human-centric aspects are discussed in only a few of the extensions. Guzman et al. presented a methodology for modelling Ambient Intelligence applications including user and technology interactions using iStar [47]. A case study with different users such as older adult and family is presented. However, requirements specific to the user groups are not taken into account. Piras et al. [56] proposed a generic framework for designing gamification solutions for acceptance requirements depending on the context (cognitive and social) elements of the intended user community. However, the focus is more on modelling the users' usage acceptance, rather than the users themselves. An extension of iStar [57] is used to investigate the working relationships between people and the fit of technology within work groups. This focuses on modelling the way technology should support groups of people collaborating together rather than single-user stakeholders.

7. Discussion

We selected the iStar 2.0 language, which is one of the most influential requirements engineering notations as noted by Caire et al. [58], for our extension. The reason we selected iStar 2.0 is that it is a recognised goal-based method to model the social characteristics of systems. Therefore, we expected it to be suitable and ready to be extended for modelling human characteristics. We believe that it should be possible to apply our notation and approach in other modelling languages and frameworks, but future research is required to explore and confirm this. The iStar 2.0 notation overall has some limitations from a PoN point of view - particularly in terms of Cognitive Fit. Prior research has explored different notations for iStar that could offer better adherence to PoN [59, 58]. However, we used the original iStar 2.0 notations as this is the accepted notation in the iStar community and our purpose was to research how human-centric aspects can be incorporated in the design of goal-modelling frameworks. We leave research on how to improve the notation visually to future work.

In this work, we took the human-centric aspects as general, and converted all such issues into the category of context. The reason for this was to be able to capture all the possible human-centric aspects through our ongoing user evaluation studies and use cases and do not limit it or bias it toward our assumptions. Using our method can help to recognise the specific categories of human-centric aspects and to classify them into meaningful categories. It also facilitates modelling them by using dedicated modelling concepts and notations.

PRISE guidelines were followed in this work, as a structured way of extending iStar. Although the method is very helpful and getting iStar experts advice has been an essential step toward building our method, there is still space for improvement. Contacting experts in the domain is an optional step if the expertise exists in the team. Although some of the authors have recognised expertise in the field of human-centric software design, input from other experts in the domain would certainly be beneficial.

Furthermore, we followed the PoN guidelines to design our new notations. However, all the guidelines developed for designing domain-specific visual languages, such as [33, 60, 61] have their limitations in terms of taking diverse end-users into account. These guidelines, although very helpful for designing the visual notations and diagrams, do not take the requirements of diverse users of the software into the account. We plan to work on extending these guidelines to provide more rigorous principles for human-centric domain-specific visual languages. Moreover, we hope that this work will also encourage other researchers in the domain.

Lastly, we believe that "one size fits all" is an approach to software design that we need to fix. We used a different colour to highlight our new notation, however, as a future goal, we plan to provide the users the option to select the notation or colour they prefer. The tool can also be equipped with a recommender to suggest options and store the users preferences, if they allow it to do so, for future usage. This will help us with designing a human-centred modelling framework.

8. Conclusions

In this paper, we discussed the importance of taking human-centric aspects into account from the early stages of software development, and reflected on the fact that current modelling languages and frameworks are inadequate for dealing with the complexity of human-centric aspects. To this end, we presented an extension of iStar 2.0 language, following PRISE guidelines, to model human-centric aspects of end-users. Human aspects refer to any individuals specific characteristics, age, gender, physical/mental impairments, culture, technical proficiency, to name a few. We used the concept of personas and contextual modelling to show the specific users' characteristics and model these using contexts, such as comfort context, monetary context, cultural preference context, age context, and so on. Experiment results confirmed the lack of understanding and capturing of such human-centric aspects, and how our extended model can simplify the understanding and addressing of such issues. More research is required in developing methods and corresponding languages to deal with this inadequacy. We hope this work encourages future research on a human-centred modelling. In future, we plan to develop a web-based version of our tool, and run more experiments with domain experts and requirement engineers to evaluate our iStar extension, and to use it for real-world applications. We also plan to focus on specific contexts to identify the relationships or interactions among the several contexts, and eventually refine the contexts. Finally, we plan to apply our extension to other modelling languages, to make modelling human aspects a common practice for other modelling languages as well.

Acknowledgment

Support for this work from ARC Laureate Program FL190100035 is gratefully acknowledged. We also greatly appreciate the insightful and detailed feedback from Professor Bran Selić and Professor Enyo Gonçalves on this work.

References

[1] J. Grundy, H. Khalajzadeh, J. McIntosh, T. Kanij, I. Mueller, Humanise: Approaches to achieve more human-centric software engineering, in: International Conference on Evaluation of Novel Approaches to Software Engineering, Springer, 2020, pp. 444–468.

[2] K. Hartzel, How self-efficacy and gender issues affect software adoption and use, Communications of the ACM 46 (9) (2003) 167–171.

[3] S. E. Stock, D. K. Davies, M. L. Wehmeyer, S. B. Palmer, Evaluation of cognitively accessible software to increase independent access to cellphone technology for people with intellectual disability, Journal of Intellectual Disability Research 52 (12) (2008) 1155–1164.

[4] S. Wirtz, E.-M. Jakobs, M. Ziefle, Age-specific usability issues of software interfaces, in: Proceedings of the IEA, Vol. 17, 2009.

[5] J. Grundy, H. Khalajzadeh, J. Mcintosh, Towards human-centric model-driven software engineering., in: ENASE, 2020, pp. 229–238.

[6] M. K. Curumsing, N. Fernando, M. Abdelrazek, R. Vasa, K. Mouzakis, J. Grundy, Emotion-oriented requirements engineering: A case study in developing a smart home system for the elderly, Journal of systems and software 147 (2019) 215–229.

[7] R. Prikladnicki, Y. Dittrich, H. Sharp, C. De Souza, M. Cataldo, R. Hoda, Cooperative and human aspects of software engineering: Chase 2013, SIGSOFT Softw. Eng. Notes 38 (5) (2013) 34–37. `doi:10.1145/2507288.2507321`.
URL `https://doi.org/10.1145/2507288.2507321`

[8] N. S. M. Yusop, J. Grundy, R. Vasa, Reporting usability defects: A systematic literature review, IEEE Transactions on Software Engineering 43 (9) (2016) 848–867.

[9] O. Kulyk, R. Kosara, J. Urquiza, I. Wassink, Human-centered aspects, in: Human-centered visualization environments, Springer, 2007, pp. 13–75.

[10] A. Y. Jim, H. Shim, J. Wang, L. R. Wijaya, R. Xu, H. Khalajzadeh, J. Grundy, T. Kanij, Improving the modelling of human-centric aspects of software systems: A case study of modelling end user age in wirefame designs., in: International Conference On Evaluation of Novel Approaches to Software Engineering (ENASE), 2021, pp. 68–79.

[11] A. Iung, J. Carbonell, L. Marchezan, E. Rodrigues, M. Bernardino, F. P. Basso, B. Medeiros, Systematic mapping study on domain-specific language development tools, Empirical Software Engineering 25 (5) (2020) 4205–4249.

[12] F. Dalpiaz, X. Franch, J. Horkoff, istar 2.0 language guide, arXiv preprint arXiv:1605.07767.

[13] E. Gonçalves, J. Castro, J. Araújo, T. Heineck, A Systematic Literature Review of iStar extensions, Journal of Systems and Software 137 (2018) 1–33. doi:10.1016/j.jss.2017.11.023.

[14] T. Miller, S. Pedell, A. A. Lopez-Lorca, A. Mendoza, L. Sterling, A. Keirnan, Emotion-led modelling for people-oriented requirements engineering: the case study of emergency systems, Journal of Systems and Software 105 (2015) 54–71.

[15] A. A. Lopez-Lorca, T. Miller, S. Pedell, A. Mendoza, A. Keirnan, L. Sterling, One size doesn't fit all: diversifying" the user" using personas and emotional scenarios, in: Proceedings of the 6th International Workshop on Social Software Engineering, 2014, pp. 25–32.

[16] M. E. Watson, C. F. Rusnock, J. M. Colombi, M. E. Miller, Human-centered design using system modeling language, Journal of Cognitive Engineering and Decision Making 11 (3) (2017) 252–269.

[17] Á. Monares, S. F. Ochoa, V. Herskovic, R. Santos, J. A. Pino, Modeling interactions in human-centric wireless sensor networks, in: Proceedings of the 2014 IEEE 18th International Conference on Computer Supported Cooperative Work in Design (CSCWD), IEEE, 2014, pp. 661–666.

[18] A. K. Dey, G. D. Abowd, D. Salber, A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications, Human–Computer Interaction 16 (2-4) (2001) 97–166.

[19] E. Gonçalves, J. Araujo, J. Castro, PRISE: A process to support iStar extensions, Journal of Systems and Software 168 (2020) 110649. doi:10.1016/j.jss.2020.110649.
URL https://doi.org/10.1016/j.jss.2020.110649

[20] A. Hussain, M. N. F. Abd Razak, E. O. Mkpojiogu, M. M. F. Hamdi, Ux evaluation of video streaming application with teenage users, Journal of Telecommunication, Electronic and Computer Engineering (JTEC) 9 (2-11) (2017) 129–131.

[21] C. C. Perez, Invisible women: Exposing data bias in a world designed for men, Random House, 2019.

[22] Y. Strengers, J. Kennedy, The Smart Wife: Why Siri, Alexa, and Other Smart Home Devices Need a Feminist Reboot, MIT Press, 2020.

[23] T. Alsanoosy, M. Spichkova, J. Harland, Cultural influence on requirements engineering activities: a systematic literature review and analysis, Requirements Engineering (2019) 1–24.

[24] D. Zhao, Z. Xing, C. Chen, X. Xu, L. Zhu, G. Li, J. Wang, Seenomaly: Vision-based linting of gui animation effects against design-don't guidelines, in: 42nd International Conference on Software Engineering (ICSE'20). ACM, New York, NY, 2020.

[25] Z. Rashid, J. Melià-Seguí, R. Pous, E. Peig, Using augmented reality and internet of things to improve accessibility of people with motor disabilities

in the context of smart cities, Future Generation Computer Systems 76 (2017) 248–261.

[26] S. Blomkvist, The user as a personality-using personas as a tool for design, KTH-Royal Institute of Technology, Stockholm Www. Nada. Kth. Se/tessy/Blomkvist. Pdf.

[27] A. Cooper, The inmates are running the asylum, in: Software-Ergonomie'99, Springer, 1999, pp. 17–17.

[28] S. Calde, K. Goodwin, R. Reimann, Shs orcas: The first integrated information system for long-term healthcare facility management, in: Case studies of the CHI2002— AIGA Experience Design FORUM, 2002, pp. 2–16.

[29] P. K. Murukannaiah, M. P. Singh, Xipho: Extending tropos to engineer context-aware personal agents, in: Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems, Citeseer, 2014, pp. 309–316.

[30] R. Ali, F. Dalpiaz, P. Giorgini, A goal-based framework for contextual requirements modeling and analysis, in: Requirements Engineering, Vol. 15, Springer, 2010, pp. 439–458.

[31] E. Yu, Modeling strategic relationships for process reengineering., Social Modeling for Requirements Engineering 11 (2011) (2011) 66–87.

[32] E. Gonçalves, M. A. de Oliveira, I. Monteiro, J. Castro, J. Araújo, Understanding what is important in istar extension proposals: the viewpoint of researchers, Requirements Engineering 24 (1) (2019) 55–84.

[33] D. Moody, The "physics" of notations: Toward a scientific basis for constructing visual notations in software engineering, IEEE Transactions on Software Engineering 35 (6) (2009) 756–779. doi:10.1109/TSE.2009.67.

[34] G. N. Rodrigues, C. J. Tavares, N. Watanabe, C. Alves, R. Ali, A persona-based modelling for contextual requirements, in: International Working Conference on Requirements Engineering: Foundation for Software Quality, Springer, 2018, pp. 352–368.

[35] E. Gonçalves, T. Heineck, J. Araújo, J. Castro, Catie: A catalogue of istar extensions, Cadernos do IME-Série Informática 41 (2018) 23–37.

[36] T. Y. Siang, The building blocks of visual design, https://www.interaction-design.org/literature/article/the-building-blocks-of-visual-design.

[37] R. Ali, F. Dalpiaz, P. Giorgini, A goal modeling framework for self-contextualizable software, in: Enterprise, Business-Process and Information Systems Modeling, Springer, 2009, pp. 326–338.

[38] T. Li, J. Horkoff, J. Mylopoulos, Integrating security patterns with security requirements analysis using contextual goal models, in: IFIP Working Conference on The Practice of Enterprise Modeling, Springer, 2014, pp. 208–223.

[39] A. Lapouchnian, J. Mylopoulos, Modeling domain variability in requirements engineering with contexts, in: International Conference on Conceptual Modeling, Springer, 2009, pp. 115–130.

[40] R. Crook, D. C. Ince, B. Nuseibeh, Using i* to model access policies: Relating actors to their organizational context. (2011).

[41] L. Plunkett, The banned pokémon episode that gave children seizures, https://kotaku.com/the-banned-pokemon-episode-that-gave-children-seizures-5757570.

[42] A. Shamsaei, D. Amyot, A. Pourshahid, E. Braun, E. Yu, G. Mussbacher, R. Tawhid, N. Cartwright, An approach to specify and analyze goal model families, in: International Workshop on System Analysis and Modeling, Springer, 2012, pp. 34–52.

[43] D. Amyot, S. Ghanavati, J. Horkoff, G. Mussbacher, L. Peyton, E. Yu, Evaluating goal models within the goal-oriented requirement language, International Journal of Intelligent Systems 25 (8) (2010) 841–877.

[44] M. Ribeiro, J. Castro, J. Pimentel, istar for safety-critical systems., in: iStar@ ER, 2019.

[45] F. B. Aydemir, P. Giorgini, J. Mylopoulos, Multi-objective risk analysis with goal models, in: 2016 IEEE Tenth International Conference on Research Challenges in Information Science (RCIS), IEEE, 2016, pp. 1–10.

[46] M. Morandini, L. Penserini, A. Perini, A. Marchetto, Engineering requirements for adaptive systems, Requirements Engineering 22 (1) (2017) 77–103.

[47] A. Guzman, A. Martinez, F. V. Agudelo, H. Estrada-Esquivel, J. P. Ortega, J. Ortiz, A methodology for modeling ambient intelligence applications using i* framework., in: iStar, 2016, pp. 61–66.

[48] Anonymous, Modelling Human-Centric Aspects of End-Users with iStar - code (May 2021). `doi:10.5281/zenodo.4757565`.
URL `https://doi.org/10.5281/zenodo.4757565`

[49] Anonymous, Modelling Human-Centric Aspects of End-Users with iStar - dataset (May 2021). `doi:10.5281/zenodo.4765606`.
URL `https://doi.org/10.5281/zenodo.4765606`

[50] A. Joshi, S. Kale, S. Chandel, D. K. Pal, Likert scale: Explored and explained, Current Journal of Applied Science and Technology (2015) 396–403.

[51] M. P. Fay, M. A. Proschan, Wilcoxon-mann-whitney or t-test? on assumptions for hypothesis tests and multiple interpretations of decision rules, Statistics surveys 4 (2010) 1—39. `doi:10.1214/09-ss051`.

[52] S. Ma, C. Chen, H. Khalajzadeh, J. Grundy, Latexify math: Mathematical formula markup revision to assist collaborative editing in math q&a sites, Proceedings of the ACM on Human-Computer Interaction 5 (CSCW2) (2021) 1–24.

[53] F. Amini, N. H. Riche, B. Lee, A. Monroy-Hernandez, P. Irani, Authoring data-driven videos with dataclips, IEEE transactions on visualization and computer graphics 23 (1) (2016) 501–510.

[54] A. Satyanarayan, J. Heer, Authoring narrative visualizations with ellipsis, in: Computer Graphics Forum, Vol. 33, Wiley Online Library, 2014, pp. 361–370.

[55] H. O. Obie, C. Chua, I. Avazpour, M. Abdelrazek, J. Grundy, T. Bednarz, Authoring logically sequenced visual data stories with gravity, Journal of Computer Languages 58 (2020) 100961. `doi:https://doi.org/10.1016/j.cola.2020.100961`.

[56] L. Piras, P. Giorgini, J. Mylopoulos, Acceptance requirements and their gamification solutions, in: 2016 IEEE 24th International Requirements Engineering Conference (RE), IEEE, 2016, pp. 365–370.

[57] A. Sutcliffe, Analyzing the effectiveness of human activity systems with i*. (2011).

[58] P. Caire, N. Genon, P. Heymans, D. L. Moody, Visual notation design 2.0: Towards user comprehensible requirements engineering notations, in: 2013 21st IEEE International Requirements Engineering Conference (RE), IEEE, 2013, pp. 115–124.

[59] R. F. de Oliveira, A. M. Barrio, J. P. D. S. Santos, A. d. P. A. Oliveira, J. C. S. do Prado Leite, A critical view over istar visual constructs., in: iStar@ ER, 2019.

[60] S. Pissierssens, J. Claes, G. Poels, The "physics of diagrams": Revealing the scientific basis of graphical representation design, arXiv preprint arXiv:1903.05941.

[61] T. R. Green, Cognitive dimensions of notations, People and computers V (1989) 443–460.