HCI part 2

Giorgio Locicero

February 2023

1 Introduction

The task for this week is to read two articles, that are **A user-centered approach for the design and implementation of KDD-based DSS: A case study in the healthcare domain**[1] and **A human-centered methodology applied to decision support system design and evaluation in a railway network context** [2], and describe what I have learned in each section of each document(around 1 page and a half for each article).

2 A human-centered methodology applied to decision support system design and evaluation in a railway network context

2.1 Introduction section

Methodology and software design is an evergreen topic in human-machine interaction, and over the years, different models have been built to enhance the process of building complex systems that have some human interaction. Although, these models have proven to be unreliable and unstable when the system is interactive and lacks the possibility of taking into account and making the main focus these interactions between humans and machines. The article's aim is to provide HCI-enriched models analysis and to see a case study for one of these models.

2.2 The limitations of the software engineering models and HCI enriched cycles

Brief description of the classical models used in software engineering (waterfall, spiral and V-model) and how, over the years, they expanded and integrated with HCI with prototyping and by integrating the interaction in the design a-priori. Follows an introduction to these models with:

- **Waterfall**: sequential, returns to previous stages are only possible for the directly connected previous stage, no HCI is taken into account, at least for the standard model, while it was taken into account most of the time in the real integration of this model.

- **V-model**: sequential, divides the stages into two main categories: specification and design, validation and tests, integration between the two categories is directly implemented in the design, meaning that even in the first stages of development and design, the evaluation of the system is available. Although returns are also limited as well, like the waterfall model. No HCI is taken into account even though validation is so early in the design, so it should probably be in consideration.

- **spiral**: iterative, specifications are analyzed and the tasks are organized and created progressively. Also, risks are taken into account and analyzed when they are encountered. Prototyping is also integrated directly in the spiral, although no explicit thoughts are given to the user perspective and interaction itself

After this introduction to these models, the enrichment process is presented and the classical models introduced serve as the basis to include HCI directly into the design with enrichment. The enrichment that are used is to make the evaluation process an intermediate phase between the various stages to take the end-users directly into the product and see if they think that is feasible enough at that point of the development. Another type of enrichment used in a design called **nabla** uses a reference model to emulate the ideal human-machine system and confront the model currently designed with this reference model to identify the needs of the users and the modules to add for user support. The enrichment process and the resulting enriched models have their own limitations as well, like limited returns or unclear and ambiguous definitions(as for the prototype or for the reference model itself in the case of nabla). The evaluation process is also divided in two types, formative and summative, the former during implementation and design, and the latter when a full working system is deployed.

2.3 A U-model for the design and evaluation of an interactive system

The U-model considers the human factor at defined stages during the development, the model itself is structured into two phases:

- **descending**: design and implementation, take into consideration already an existing framework for consistency and speed of implementation, to use tested and evaluated designs to also have an example for taking into consideration for the specific task or module related to the human interaction and need. The steps are analyzing the application domain, developing interface prototypes, defining tasks, and modeling tasks;

- **ascending**: evaluation, getting feedback from the users and adding constraints or needs, the operator that will use the system is capable of using it properly and somewhat easily, taking into account **usefulness** and **usability**. The steps of the ascending phase are evaluating the system's usefulness and usability, testing user behavior, and comparing specified tasks with real tasks.

2.4 Case study: the INFRAFER project

This section pertains to the development of a software system that would aid the owner and manager of the French railway network in managing its investments in its infrastructures.

The needs, preferences, and behaviors of the people who will be using the system will be the primary focus of the design process. The owner and manager of the railway infrastructure have the responsibility to ensure that the infrastructure is safe, efficient, and effective. One of the key challenges in managing the railway infrastructure is making strategic investments in its maintenance, repair, and expansion. The system must also be user-friendly, and simplify data entry. It must be simple to implement and also present its results as clearly as possible so that non-experts are able to understand them. The systems and methods which exist in these investments must be made with limited resources and must take into account a range of factors, including current infrastructure conditions, projected usage, and the needs of stakeholders. To help the French railway network owner manage its investments in infrastructure, a software system could be designed that incorporates data on the current state of the infrastructure, projected usage, and the needs of stakeholders. The system could provide insights and recommendations to help decision-makers make informed decisions about where to allocate resources.

Overall, the aim of this project was to develop a software system that helps the French railway network owner to manage its investments in infrastructure more effectively, taking into account the needs and preferences of the people who will be using the system. The human-centered approach to design will help to obtain the objectives of usability, accessibility, and user satisfaction, leading to better outcomes for all stakeholders involved. The U-model was used to design the system, and the coordination of the activities performed by the designers and experts was also key.

3 A user-centered approach for the design and implementation of KDD-based DSS: A case study in the healthcare domain

3.1 introduction section

This article describes the research on developing an interactive decision support system (DSS) based on a user-centered design approach. The system is based on a process called Knowledge Discovery from Data (KDD), which is a data mining process that extracts information and new knowledge from large sets of data.

The article discusses the challenges of developing a DSS that meets the needs of end-users who may not have expertise in computer science or statistics. To overcome these challenges, the authors propose an approach that combines methods from Software Engineering (SE) and Human-Computer Interaction (HCI) to create a KDD-based DSS. They validate their approach through a case study involving physicians from an Intensive Care Unit (ICU) in Tunisia, who need a DSS to predict and prevent hospital infections.

The article provides an overview of a proposed approach for designing and developing decision support systems (DSS) based on the Knowledge Discovery from Data (KDD) process.

The authors argue that a user-centered approach is critical to the success of interactive DSS, as the system's design must account for the needs, preferences, and behaviors of the end-users. The proposed approach integrates SE and HCI methods to improve the reliability, evolvability, reusability, and portability of the software while also explicitly and systematically taking the end-users into account.

3.2 DSS and HCI: key concepts for KDD

This section describes the three phases in the decision-making process: search for information, design process, and choice of a solution. The need for synergy between human and automatic processing in a decision support system (DSS) is highlighted, and the importance of the communication capabilities and user interface in the success of a DSS is emphasized. The concept of interactivity in a DSS underscores the crucial role of humans, and the need for system developers to design cooperative DSS systems that distribute competencies between the user and the computer is stressed. The use of data is also of great importance to facilitate decision-making. Tools such as On-Line Analytical Processing (OLAP) and data mining (DM) are used for multidimensional analysis and to explore new knowledge in large amounts of data. Data mining refers to the most important stage of a multistage process known as **KDD (Knowledge Discovery from Databases)**, which includes problem formulation, data retrieval, data selection, data cleansing, data transformation, data mining, result evaluation, and knowledge integration. A KDD-based DSS can require up to six modules for its implementation(select data, cleanse data, transform them, mine them, evaluate, interpret patterns, and manage extracted knowledge). The end-users' acceptance is taken into consideration. Adapting HCI activities is also of great importance to each decision-maker.

3.3 Development Approaches

In this section, the traditional models in the fields of SE and the HCI are seen briefly. The models presented are almost the same as the previous section 2, and it adds the following models:

- **Agile models**: this is a new type of models seen in this article, these models prioritize quick software design with heavy customer involvement and responsiveness to their requests. The objective of these models is customer satisfaction rather than just meeting the terms of a contract. Examples of such models include Adaptive Software Development (ASD), Dynamic Systems Development Method (DSDM), Extreme Programming (XP), and Rapid Application Development (RAD). However, these models do not have systematic processes like traditional models and often encounter resistance from systems developers.
- **Unified Process**: a process pattern that can be adapted to various software systems, fields of applications, companies, qualification levels, and project sizes. UP is controlled by use cases, centered on system architecture, and iterative and incremental. It has four phases: initialization, development, construction, and transition. UP allows costs to be limited, makes it possible to limit the risks of delaying the installation of the application, and permits potential problems to be identified in the first stages of development. However, the Rational Unified Process (RUP), a commercial version of UP, is not user-centered according to the standard ISO 13407. The use of use cases is not sufficient to make a design process user-centered.

The article continues with a discussion of how traditional software engineering (SE) models tend to focus too much on technical aspects of the system and not enough on user needs. The Unified Process (UP) is mentioned as an exception, as it involves the user throughout the project life cycle, but this involvement is not formally explained. The passage notes that the trend in software development is towards iterative processes and integrating re-use concepts, but user involvement is often not accompanied by clear explanations of their involvement. The importance of expressing user characteristics in interactive system development is emphasized, and the passage notes that about fifteen years ago, traditional SE models began incorporating human-computer interaction (HCI) principles to better address user needs(as stated in the previous section). Then, different HCI-enriched development models are described.

3.4 Proposed approach: UP/U

This section focuses on the adaptation of the U model to include both the practices of SE and HCI enrichment into consideration and also discusses the reasons for adapting the U model for use in the context of a decision support system (DSS). The adapted U model is the same as described in the previous article 2 involves.

The proposed user-centered approach (UP/U) aims to put the human-computer interaction (HCI) at the center of the KDD process. The approach follows the **UP principle** of iterative and incremental development, where each task is evaluated as soon as the first iterations of the development process have been completed. The approach executes several complete UP iterations, from initialization to transition phase. The adapted U model is applied at each iteration level, which incorporates the continual presence and constant participation of the user throughout the project. Each activity of the adapted U model is divided into sub-activities that model the HCI of the DSS. The activities include **Needs Assessment, Analysis, Design, Implementation, and Testing**. The **Needs Assessment** activity allows defining the user's functional needs and non-functional technical needs, and the **Analysis** activity helps in understanding customer needs and requirements. The **Design** activity provides a more accurate understanding of the constraints related to the programming language and the operating system, and the **Implementation** activity results in planning the integration of the components and producing the classes and providing the source code. Finally, the **Testing** activity allows verifying the results, and it must be carried out simultaneously with the U model activities, notably tests with the users and the comparison of the tasks initially specified by the designer and the tasks really accomplished by the system.

3.5 Healthcare case study

This section presents a real project case study about the integration of a **Decision Support System** in healthcare. I don't want to go too much into the details since a case study is quite complex, but I will give a brief description of the keypoints:

- A case study in the healthcare domain could involve the development of a DSS to assist healthcare professionals in the diagnosis and treatment of a particular condition or disease.
- The user-centered approach would involve working closely with healthcare professionals to identify their specific needs and preferences for the DSS, as well as their workflows and decision-making processes.
- System needs to be user-friendly.
- The proposed approach was the U-model

4 Overall conclusions

As the conclusion of this homework, I have added this additional section. It can be said that a user-centered approach to the design and implementation would focus on delivering a system that meets the unique needs and preferences of its end-users, providing them with valuable insights that can improve the outcomes and experience. A user-centered approach to design and implementation aims to place the needs of the end-users at the center of the system design process. The approach presented tries to obtain the significance that the system is designed to meet, with the unique needs and preferences of its users.

Integrating a human-centered approach into the design of a system means putting the needs, preferences, and behaviors of the people who will be using the system at the center of the design process. It involves understanding the context in which the system will be used and the goals and motivations of the people who will interact with the system.

A human-centered approach typically involves a set of design practices and techniques that help designers to empathize with the end-users, understand their needs, and involve them in the design process. For example, designers may conduct user research to gather insights about the end-users' behaviors, goals, and pain points. They may also conduct user testing to evaluate the usability and effectiveness of the system in a real-world setting.

Some key principles of human-centered design include:

- **Involvement:** End-users should be involved in the design process, providing feedback on early prototypes, testing the system in a real-world setting, and collaborating with designers to refine the system over time.
- **Iteration:** Human-centered design is an iterative process, which means that designers must be willing to iterate on their design based on user feedback and insights gathered through testing.

By integrating a human-centered approach into the design of a system, designers can create systems that are more intuitive, effective, and enjoyable to use. This can lead to higher user adoption, increased user satisfaction, and improved business outcomes.

References

[1] Mounir Ben Ayed, Hela Ltifi, Christophe Kolski, and Adel M Alimi. A user-centered approach for the design and implementation of kdd-based dss: A case study in the healthcare domain. *Decision Support Systems*, 50(1):64–78, 2010.

[2] Sophie Lepreux, Mourad Abed, and Christophe Kolski. A human-centred methodology applied to decision support system design and evaluation in a railway network context. *Cognition, Technology & Work*, 5:248–271, 2003.