HCI part 7

Giorgio Locicero

March 2023

1 Introduction

The task for this week is to read two articles, these are **Towards new web services based supervisory systems in complex industrial organizations: Basic principles and case study**[1] and **Agent-based architecture for interactive system design: Current approaches, perspectives and evaluation** [2], and describe what I have learned in each section of each document(around 1 page and a half for each article).

2 Towards new web services based supervisory systems in complex industrial organizations: Basic principles and case stud

2.1 Introduction section

The introduction section discusses the challenges of **providing a human-machine interface for industrial supervisory systems**, which require data from various sources and structures. The article proposes **integrating process supervision systems with web technologies** to enable remote access and improve efficiency. The authors suggest using a service-oriented architecture model and web services to create new interaction models for different organizational human actors. Design issues of traditional supervisory HMIs are presented and the motivations and foundations of the service-oriented approach are introduced. Additionally, the section presents a global methodological framework for integrating the development of service-oriented interactive applications and HMI specifications. Finally, the section concludes by showing the **feasibility** of the proposed approach through a real case study in a sugar refinery supervisory HMI that will be revisited in other sections.

2.2 Related work and new motivations

This section discusses the main design **issues** and **new motivations** for the design of supervisory human-machine interfaces (HMI) in the industrial supervision domain. The article argues that with the newest technological environment in which supervision is evolving, newer functional and organizational issues have appeared, leading to new motivations regarding supervisory HMI design. The section presents a **move towards more global and wider supervision**, the **impact** of new peripheral devices and communications means on supervision, and the **importance** of taking the context of the supervisory HMI into account. It also discusses the different human operator profiles involved and their various tasks dependent on new means, the intra-organization and extra-company mobility, and the need for a unified view of the supervision HMI.

2.3 Service oriented approach for the HMI design

This article section discusses a **service-oriented approach** to design HMI. It explains the concept of service orientation and service-oriented architecture (SOA), which breaks down software design into **agile services dealing with specific needs**, making them available as independent services accessible in a standardized way. The benefits of this approach include reducing development costs and complexity, as well as making business processes and logic accessible via relevant web technologies such as WSUI and WSRP. The section also discusses web services, which are auto-descriptive, modular, and **loosely coupled** applications that can deliver functionalities such as simple requests or sophisticated business processes. They can be deployed over multiple distribution channels, and relevant user interface specifications towards multi-channel user access and aggregation of web services using a single presentation page may be used. Overall, the service-oriented approach can enable complex organizations to make their business processes and logic seamlessly accessible via various web technologies, thereby reducing development costs and complexity.

2.4 Overall methodological framework

This section describes the overall methodological framework for designing the HMI, considering the specificities of web services and the new needs of supervision operators. The framework is based on the principles of a generic Unified Process and emphasizes three main design activities referred to as phases. The design process follows the general principles of a user-centered design methodology and is illustrated using a case study applying the service-oriented based approach.

The framework described is comprised of the following phases and steps

1. **Phase 1**: Overall supervisory organization analysis, which involves studying the existing organization to identify the business problems and objectives of the global human-machine system. This phase includes three steps: **business analysis, human task analysis**, and **user analysis**. Business analysis involves pinpointing the users' expectations and needs from a utility point of view. Human task analysis involves studying how users perform their tasks in a given environment to achieve a business goal. User analysis involves defining the profiles of users who will use the future HMI by gathering relevant data and information for decisions in the design of the interactive system.

2. **Phase 2**: Requirements analysis and elicitation. The aim of this phase is to capture, analyze, and understand stakeholders' requirements and translate them into different types of requirements such as service requirements in terms of functions, tasks, sub-tasks, and services. The phase involves understanding how stakeholders are involved regarding the identified services and task analysis outcome so as to derive their requirements from their mobility and cooperative task issues. The phase involves elicitation in terms of

services, stakeholders' mobility, and cooperation. Finally, the phase involves defining the human actors implied in the complex organization. Overall, the phase aims to identify and derive a web service categorization seen from a high level of abstraction (business-oriented) to understand the use of the future system.

3. **Phase 3**: HMI specification. This involves designing and constructing mock-ups and prototypes for the HMI components that will interact with the web service-based user task model that was developed in the previous phase. This phase includes **identifying the user interaction objects** that are required for all the identified requirements related to information, mobility, and cooperation issues. The HMI components are exchanged between the user and the service managing the supervision HMI as XML-based message descriptions, which are processed by a presentation engine and presented to the user. The HMI specification phase involves **two main steps**. In step 1, the singular and aggregated web services are identified and modeled, and the interface objects are specified as the user perceives and manipulates them. This step also involves identifying the presentation units that correspond to the underlying interaction aspects (input, dialogue, and display) of a sub-task of an interactive task. In step 2, the HMI mock-ups and prototypes are deduced for all the information needs identified during the user and task analyses. This step helps to visualize, test, simulate, and validate the user interface according to user supervisory scenarios. User participation in the design and evaluation of the interface during the mock-up and prototyping phase enables the ergonomic evaluation process to reduce the risks. Ergonomic guidelines and principles guide the software or service component specification. The evaluation phase consists of measuring the application's usability, which can be carried out by different methods. A mock-up can be used as support for user testing so as to validate the understanding and visibility of the different screen zones or areas.

In the conclusion of this section about the design framework, the authors **summarize their approach for designing service-oriented supervisory HMI** based on principles from software engineering, human-machine interaction, and web technology. The framework uses a user-centered approach and takes into account the diversity of technologies and contexts of use. The authors suggest that this approach is a preliminary attempt towards a service-oriented model for HMI, as there is currently no methodology for developing software components for a specific layer of service-oriented HMI. The framework is used as the basis for a project designing a sugar refinery supervisory HMI for a multi-site industrial organization.

2.5 Case study: supervision of a sugar refinery

This article section discusses a case study focused on **supervising a sugar refinery to address mobility and cooperative task issues** with the techniques seen previously.

2.6 Conclusion and perspectives

See 4 for a generalized conclusion of the two articles.

3 Agent-based architecture for interactive system design: Current approaches, perspectives and evaluation

3.1 introduction section

This introduction section provides an **overview of the history of architecture research in the Human-Computer Interaction domain**, which began with defining recommendations for developers and has evolved to include tools for designing, developing, and validating interactive systems. The paper focuses on **agent-based architectures** and provides a survey of the different models available in the literature. The paper is divided into four main parts, which include an introduction to architecture models and general agent-based approaches, agent-based approaches for Computer Supported Cooperative Work (CSCW), the connection between agent-based architecture and web services domains, and an evaluation of interactive systems based on an agent-based architecture. Additionally, the paper introduces the first version of a dedicated evaluation tool.

3.2 From Seeheim Model to Agent-Based Architectures

This section of the article discusses **different approaches to architecture models** in the **Human-Computer Interaction domain**. The **Seeheim model**, which is a global model that defines a fixed structure with precisely defined components, is introduced. The drawbacks of global models are also discussed, such as the difficulty in applying an object-oriented approach to define elementary interaction classes. To solve this problem,

agent-based architecture models like **PAC, AMF**, and **AoMVC** were developed, which define elementary software bricks composed of fixed or non-fixed parts and define the relations between them. The section also introduces hybrid models that combine global and object-oriented approaches to refine some of the main components like Presentation or Controller. Examples of hybrid models include PAC-Amodeus and H4. The section concludes by mentioning related research on architecture models concerning distributed and plastic UI.

3.3 Agent-Based Architectures: Approaches Dedicated to CSCW Systems

This section discusses the architecture of **Computer-Supported Cooperative Work** (CSCW) systems. CSCW systems are **multi-user distributed systems** and their architecture needs to meet specific requirements, such as taxonomy of collaborations, awareness, and nature of cooperation activities. CSCW systems are inspired by interactive systems architectures, such as layered, agent, and hybrid architectures. Recent developments in CSCW systems relate to the mobility of actors, and adaptation techniques for different interaction devices and environments. Model-based approaches are identified as the most powerful, where a high-level and abstract representation can be instantiated later in the development lifecycle to meet specific usability requirements. The article concludes that an interaction model is needed to link together all the models and usability attributes.

3.4 Web Services and Agent-Based Architectures

This section discusses the impact of web services on distributed system design and how **agent-based architectures provide new possibilities and challenges**. Traditional web services follow a client/server architecture, but agent-based architectures utilize autonomous and proactive behaviors of agents, which offers new perspectives in this field. The article describes a technical framework for agent-based web services (AWS, even though this acronym is ambiguous for the amazon web services acronym) that captures, models, and implements service functionalities with autonomous and dynamic interactions. The article also mentions **DAML-S**, a semantic markup language for describing web services and related ontologies, which has been **superseded by OWL-S**. The article further discusses dynamic web-service invocation by agents and how agents are used to **specify service providers** and **service customers** in a **hybrid peer-to-peer model infrastructure**. The JADE (Java Agent Development Environment) framework is used to publish web services as a JADE agent service and agent services as web service endpoints. The article emphasizes that while these proposals offer new perspectives for service-oriented interactive systems, they must be considered with attention regarding agent-based architecture perspectives.

3.5 Agent-Based Architectures: The Evaluation Problem

This section discusses the evaluation of interactive systems, with a particular focus on systems that use an agent-based architecture. The authors highlight the need for automated or semi-automated tools to assist in evaluating such systems and introduce an **electronic informer** (EI) as one such tool. This EI **captures interactions between the user and the interface agents**, as well as interactions between the agents themselves. The tool is composed of seven modules, including **event capture**, **task determination**, **data analysis**, **Petri net generation** and comparison, and **criteria association**. The authors suggest that this tool represents a new generation of evaluation tools for agent-based interactive systems, but note that **more research is necessary** to adapt the tool to different application fields and architecture models and to provide real-time assistance to evaluators.

3.6 Conclusion and Perspectives

The article's conclusion discusses the various models and approaches proposed in the literature for distributed or agent-based architectures of interactive systems since the 1980s. The section provides a brief overview of four domains: general agent-based architecture models, models dedicated to CSCW systems, interactive systems based on web services, and evaluation of interactive systems using agent-based architecture. The article suggests that **general agent-based architecture models need to be more widely used** at the implementation level and **integrated** into development environments to allow the development of tools for software design, simulation, and evaluation. The section also highlights the importance of context adaptation in capillary cooperative systems and the advantages of autonomous behavior and independence of agent-based systems. Finally, the article suggests that many research problems need to be studied and solved regarding the evaluation of agent-based interactive systems.

A generalized conclusion of the two articles is given in 4

4 Overall conclusions

As the conclusion of this homework, I have added this additional section to take the conclusion and objectives of the two articles seen and give an overall meaning to both of them in the same context. The first article reflects on the case study and the proposed framework for designing a web service-based human-machine interface (HMI) for supervising complex industrial processes and focuses on the design of a supervisory web-based HMI that addresses concerns such as mobility, cooperation, and information access. The article suggests that the proposed framework can address the new and complex design issues that arise in such applications, especially those related to operator mobility and cooperative tasks. The article proposes a service-oriented architecture approach, which separates different layers of services for better development and maintenance of HMIs. The article concludes by highlighting the need for further research and validation of the proposed framework in other application domains.

The second article proposes a survey on agent-based architectures of interactive systems. The survey covers general agent-based architectures, agent-based approaches dedicated to CSCW systems, and new agent-based approaches required for web services. The authors also highlight the need for new evaluation tools for agent-based interactive systems and present an example of a representative evaluation tool.

The two articles address different aspects of designing interactive systems for complex industrial organizations using web-based technologies.

Overall, both articles emphasize the need for new approaches and tools to design and evaluate interactive systems for complex industrial organizations using web-based technologies. The first article proposes a specific approach based on a service-oriented architecture and web services, while the second article provides a broader survey of different agent-based architectures and their evaluation tools. Both articles highlight the importance of considering mobility, cooperation, and information access when designing interactive systems for complex industrial organizations.

References

[1] Djilali Idoughi, Moussa Kerkar, and Christophe Kolski. Towards new web services based supervisory systems in complex industrial organizations: Basic principles and case study. *Computers in Industry*, 61(3):235–249, 2010.

[2] Christophe Kolski, Peter Forbrig, Bertrand David, Patrick Girard, Chi Dung Tran, and Houcine Ezzedine. Agent-based architecture for interactive system design: Current approaches, perspectives and evaluation. In *Human-Computer Interaction. New Trends: 13th International Conference, HCI International 2009, San Diego, CA, USA, July 19-24, 2009, Proceedings, Part I 13*, pages 624–633. Springer Berlin Heidelberg, 2009.