A Holonic Human Cyber-Physical System in Healthcare

by
Travis Wade Defty

Thesis presented in partial fulfilment of the requirements for the degree
of Master of Engineering (Mechatronic)
in the Faculty of Engineering at Stellenbosch University

Supervisor: Dr Karel Kruger
Co-supervisor: Prof Anton Herman Basson

December 2022

The financial assistance of the National Research Foundation (NRF) towards this research is hereby acknowledged. Opinions expressed and conclusions arrived at are those of the author and are not necessarily to be attributed to the NRF.

Declaration

By submitting this thesis electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the sole author thereof (save to the extent explicitly otherwise stated), that reproduction and publication thereof by Stellenbosch University will not infringe any third party rights and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

Date: 4 September 2022

Abstract

A Holonic Human Cyber-Physical System in Healthcare

T.W. Defty

Department of Mechanical and Mechatronic Engineering
Stellenbosch University
Thesis: MEng (Mechatronic Engineering)
December 2022

The fourth industrial revolution (I4.0) aims to enhance operational performance and promote interconnectivity between system components. Labour-intensive industries which are reliant on humans to perform tasks along with automation and integrated equipment are typically classified as Human Cyber-Physical Systems (HCPSs). Many I4.0 developments within HCPSs lack the appropriate consideration and integration of humans, resulting in performance inefficiencies, quality concerns, and under-utilisation. Healthcare is one such industry, with challenges due to a lack of integration developments which are exacerbated by limited resources and high patient volumes.

The objective of this thesis is the development of a holonic HCPS for healthcare. These holonic HCPS developments address challenges in healthcare through the application of Human-System Integration (HSI) principles and requirements. HSI aims to enhance human intelligence and cooperation within digitised environments. HCPSs within this study are defined by human, physical and digital components that interact in a complex manner to achieve overall system goals. An ambulatory clinic case study was selected to study the complex interactions between system components and actors.

This thesis presents a Representation-Communication-Interfacing (RCI) framework to better define HSI and guide the development and evaluation of the HSI maturity in HCPSs. A holonic system approach is selected, using the RCI framework, to improve the maturity of HSI developments in healthcare. This thesis presents a structured design process for holonic systems using the Activity-Resource-Type-Instance (ARTI) architecture and the Biography-Attribute-Schedule-Execution (BASE) architecture. The design process is presented in response to a lack of available implementation details expressed in reviewed holonic applications.

The developed holonic HCPS is evaluated experimentally, to showcase how the system meets the requirements and enhances the HSI maturity of the ambulatory case study. The evaluation shows that the holonic HCPS effectively integrates humans, resulting in improved operational efficiencies and lower workloads experienced by humans. The RCI framework offers valuable guidance for elevating the human component to the cyber layer, improving the autonomy and cooperability of humans in the system and easing the reconfigurability of the system. The holonic design process guides the system development by standardising the partitioning of system components and interactions into distinct holons. Furthermore, this design process reduces the development complexity and time. The holonic HCPS demonstrates how HSI developments can improve clinic workflow efficiencies, aid decision-making, improve the traceability of activities and reduce the workload for healthcare practitioners.

Uittreksel

'n Holoniese Menslike Kuber-Fisiese Stelsel in Gesondheidsorg

T.W. Defty

Departement Meganiese en Megatroniese Ingenieurswese
Universiteit Stellenbosch
Tesis: MIng (Megatroniese Ingenieurswese)
Desember 2022

Die vierde industriële revolusie (I4.0) het ten doel om operasionele prestasie te verbeter en interkonnektiwiteit tussen stelselkomponente te bevorder. Arbeidsintensiewe nywerhede wat op mense staatmaak om take saam met outomatisering en geïntegreerde toerusting uit te voer, word tipies geklassifiseer as menslike kuber-fisiese stelsels (*Human Cyber-Physical Systems*, HCPS'e). Baie I4.0-ontwikkelings binne HCPS'e het nie die gepaste oorweging en integrasie van mense nie, wat lei tot prestasie-ondoeltreffendheid, kommer oor kwaliteit en onderbenutting. Gesondheidsorg is een so 'n bedryf, wat deur uitdagings geteister word weens 'n gebrek aan integrasie-ontwikkelings wat vererger word deur beperkte hulpbronne en hoë pasiëntvolumes.

Die doel van hierdie tesis is die ontwikkeling van 'n holoniese HCPS vir gesondheidsorg. Hierdie holoniese HCPS-ontwikkeling spreek uitdagings in gesondheidsorg aan deur die toepassing van die beginsels en vereistes van mens-stelsel-integrasie (*Human-System Integration*, HSI). HSI het ten doel om menslike intelligensie en samewerking binne gedigitaliseerde omgewings te verbeter. HCPS'e word binne hierdie studie gedefinieer deur menslike, fisiese en digitale komponente wat op 'n komplekse wyse in wisselwerking tree om algehele stelseldoelwitte te bereik. ’n Ambulante kliniekgevallestudie is gekies om die komplekse interaksies tussen sisteemkomponente en akteurs te bestudeer.

Hierdie tesis bied 'n raamwerk aan vir verteenwoordiging-kommunikasie-koppelling (*Representation-Communication-Interfacing*, RCI) om HSI beter te definieer en die ontwikkeling en evaluering van die HSI-volwassenheid in HCPS'e te verbeter. 'n Holoniese-sisteembenadering word gekies, met behulp van die RCI-raamwerk, om die volwassenheid van HSI-ontwikkelings in gesondheidsorg te verbeter. Hierdie tesis bied 'n gestruktureerde ontwerpproses vir holoniese stelsels aan deur gebruik te maak van die Aktiwiteit-Hulpbron-Tipe-Instansie (*Activity-Resource-Type-Instance,* ARTI) argitektuur en die Biografie-Kenmerk-Skedule-Uitvoering (*Biography-Attribute-Schedule-Execution*, BASE) argitektuur. Die ontwerpproses word aangebied in reaksie op 'n gebrek aan beskikbare implementeringsbesonderhede wat in die oorsig van holoniese toepassings uitgevind word.

Die ontwikkelde holoniese HCPS word eksperimenteel geëvalueer, om te wys hoe die stelsel aan die vereistes voldoen en die HSI-volwassenheid van die ambulante gevallestudie verbeter. Die evaluering bewys dat die holoniese HCPS mense effektief integreer, wat lei tot verbeterde operasionele doeltreffendheid en laer werkladings wat deur mense ervaar word. Die RCI-raamwerk bied waardevolle leiding om die menslike komponent na die kuberlaag te verhef, die outonomie en samewerking van mense in die stelsel te verbeter en die herkonfigureerbaarheid van die stelsel te vergemaklik. Die holoniese ontwerpproses lei die stelselontwikkeling deur die verdeling van stelselkomponente en -interaksies in afsonderlike holons te standaardiseer. Verder verminder hierdie ontwerpproses die ontwikkelingstyd en kompleksiteit. Die holoniese HCPS demonstreer hoe HSI-ontwikkelings kliniekwerkvloeidoeltreffendheid kan verbeter, besluitneming kan help, die naspeurbaarheid van aktiwiteite verbeter en die werklading vir gesondheidsorgpraktisyns kan verminder.

Acknowledgements

Many people have been instrumental in supporting me and helping me to continue to prioritise various aspects of my life while I worked and wrote this thesis. Although not everyone is mentioned, I hope to honour some in this section.

Giving my life to Jesus Christ and desiring to follow all that God has commanded, out of my love for Him because of His love for me, I have been able to trust Him for all provision and direction. God, I could have never chosen my family, where I grew up, or selected the capabilities that I have and do not have. God, during my early years of schooling you healed my eyesight and provided me with the support to catch up on my lost academic progress. You guided my every decision and pathed the way for me to be where I am today. I know you heard the prayers of many faithful people throughout my life, some of whom I might never know of. You are faithful and steadfast. All that I have done for this thesis is to Your glory.

My parents, mom and dad, thank you for always protecting, providing, teaching and loving me. Words cannot describe how thankful I am for you both. My brother, Gareth, thank you for your love and for every single chat which always left me thinking and better off – I value you. My soon-to-be bride, Elsjé, thank you for your endless love and support throughout the past two years. Your resilience, patience and dependence on God inspired me to draw closer to Him for my strength. To my flatmates, thank you for always being willing to support me, have critical discussions and dream with me about how we can change the world with the few talents we have been given. My friends, thank you for all the laughter, advice and care that you have shown me throughout this journey.

Dr Kruger and Prof Basson, thank you for all your guidance and sacrificial effort to help me write this thesis. You both show an immense amount of passion for your work and family. Thank you for always being willing to discuss an idea with me and encouraging me to pursue excellence in my field. Dale, thank you for the time you sacrificed to guide me and encourage me. Prof Allwood, thank you for your guidance, encouragement and your effort in making me feel valued every time I visit – you have inspired me to live out my calling through my work. To the MAD research group members, thank you for all the laughs and advice over the past two years – each of you played a crucial role.

Dedication

To those struggling to maintain work-life or without work - I hope and pray that technology and research efforts will help bridge the gap to dignified work so that you may be contributors to your family, community and society in the way God intends

Table of contents

Page

List of Figures

Page

List of Tables

Page

Abbreviations

3SAL	Three Stage Activity Lifecycle
AC	Administrative Cognition
ADACOR	Adaptive Holonic Control Architecture
AI	Activity Instance
AL	Administrative Logistics
AP	Analysis Plugin
AR	Augmented Reality
ARTI	Activity-Resource-Type-Instance
AT	Activity Type
BASE	Biography-Attribute-Schedule-Execution
BMI	Body Mass Index
CPPS	Cyber-Physical Production System
CPS	Cyber-Physical System
DFD	Data Flow Diagram
DT	Digital Twin
ECM	Enterprise Content Management
EHR	Electronic Health Record
EP	Execution Plugin
FEV1	Forced Expiratory Volume (in one second)
FR	Functional Requirement
FVC	Forced Vital Capacity
HCA	Holonic Control Architecture
HCPS	Human Cyber-Physical System
HDT	Human Digital Twin
HF	Human Factors
HitL	Human-in-the-Loop
HMS	Holonic Manufacturing System
HP	Healthcare Practitioner
HSI	Human-System Integration
I4.0	Fourth Industrial Revolution
IoT	Internet of Things
JADE	Java Agent Development
MAS	Multi-Agent System
MCPS	Medical Cyber-Physical System
MR	Mixed Reality
NDP	National Development Policy

NFR	Non-Functional Requirement
NHLS	National Health Laboratory Service
OTP	Open Telecommunication Platform
PACS	Picture Archiving and Communication System
PROSA	Product-Resource-Order-Staff Architecture
RI	Resource Instance
RP	Reflection Plugin
RT	Resource Type
SAR	Surface-Mounted Augmented Reality
SEBoK	Systems Engineering Body of Knowledge
SoS	System of Systems
SP	Scheduling Plugin
TLX	Task Load Index
TMS	Time Motion Study
VM	Virtual Machine

1 Introduction

This chapter presents background information regarding related research to guide the readers' understanding of the thesis objectives. Furthermore, the objectives are motivated by considering the industry and healthcare needs as well as research opportunities. Lastly, the research methodology discusses the approach to fulfilling the objectives of this thesis.

1.1 Background

Labour-intensive industries, such as manufacturing, mining, agriculture and healthcare services, are aiming for improved efficiencies in workflow and quality assurance within their operational processes (Department of Labour, 2019). In response, these industries are adopting innovative technologies and approaches giving rise to the fourth industrial revolution (I4.0). This technological revolution is novel due to its claims of improved interconnectivity and intelligence for both machines and humans. Nevertheless, I4.0 developments remain in their infancy with many claims still to be realised.

Operational processes typically utilise humans (as workers), equipment, and digital system components. These have been termed Human Cyber-Physical Systems (HCPSs), a refinement of the Cyber-Physical System (CPS) concept (Zhou, Zhou, Wang, *et al.*, 2019). HCPSs and CPSs highlight the importance of implementing technologies and methods to improve interconnectivity and information exchange, throughout various business layers, between system components.

Humans are regarded as intelligent and crucial resources for executing tasks in operational processes – offering a unique skillset of physical dexterity, situational awareness, decision-making, and communication (Becker & Stern, 2016). Equipment, information systems and control systems can reduce human workload, provide task automation, and ensure a high level of accuracy. A common challenge in I4.0 developments is the integration of non-human system components (e.g. machines, information systems, control systems) with humans in operational processes. Often physical system components are limited in their ability to react to and cooperate with the actions of humans due to insufficient awareness. Furthermore, humans often have limited knowledge and access to information about the physical and digital systems by which to improve their situational awareness and decision-making. These technical challenges are addressed by Human-System Integration (HSI) developments that aim to improve the exchange of knowledge and coordination between humans and systems (Defty, Kruger & Basson, 2022).

The general HSI definition for systems engineering is the "management and technical discipline of planning, coordinating and optimizing all human-related considerations during system design, testing, production, use and disposal of systems, sub-systems, equipment and facilities” (SEBoK, 2021a). Within this context, this thesis focuses on the elevation of humans to the cyber-layer, through technological developments, to optimise the human considerations during system operation. This requires the improvement of human integration with other digital systems beyond *factory-floor* developments, such as human-machine interfaces. HCPSs must incorporate HSI developments to integrate the advantage of human intelligence and machine intelligence (i.e. sensing, cognition, decision-making and control) (Zhou *et al.*, 2019).

The healthcare industry must adopt I4.0 technologies and methods to overcome challenges related to data integration, workflow optimisation and improved clinical decision-making to provide a better quality of care to everyone, everywhere (Emanuele, J., & Koetter, 2007; Aceto, Persico & Pescapé, 2020). These challenges have been exacerbated by staffing shortages, high patient volumes and under-resourced health facilities. The healthcare industry is starting to experience the convergence of automation, health informatics and biomedical engineering (Pang, Yang, Khedri, *et al.*, 2018). These efforts have resulted in the terms such as *healthcare 4.0* and *smart health*, which aim to provide pervasive (anyone, anywhere, anytime), preventive, personalised, precision and patient-centred healthcare (Pang *et al.*, 2018).

Within healthcare, ambulatory facilities are those which provide out-patient medical services to patients (e.g. examinations, blood tests, x-rays, etc.) to diagnose and treat medical conditions. Workflow processes within healthcare, from the perspective of HCPS, require the integration of personnel (e.g. patients and doctors) and physical and digital systems (e.g. patient information systems and medical testing equipment) to handle the complex interactions. I4.0 challenges in healthcare are prominent in ambulatory facilities due to the high patient volumes and poor medical information systems. Therefore, workflow challenges within these ambulatory medical processes could find solutions from HCPS developments with greater attention to HSI.

Developing CPS and HCPS within complex environments pose unique challenges and have therefore received much research attention. The holonic systems concept, initially presented by Koestler (1967), was developed considering social, biological and organisational behaviours. Koestler (1967) argues that complex systems must be comprised of smaller units (holons) that retain autonomy while remaining collaborative to ensure stability. Over the past three decades, this concept has been introduced into complex manufacturing processes through various Holonic Control Architectures (HCAs) within the Holonic Manufacturing System (HMS) paradigm (Derigent, Cardin & Trentesaux, 2021). HMSs arguably present a means to overcome I4.0 challenges by improving the heterarchical control, robustness and scalability of the coordination systems.

Developments including holonic systems, HCPS and Human Digital Twins (HDTs), aim to improve the digital (or cyber) representation of humans in integrated environments. The Biography-Attribute-Schedule-Execution (BASE) architecture, developed by Sparrow (2021), aimed to improve human integration through digital representation. The BASE architecture provides a digital administration shell that represents the state of a human worker. In the case study by Sparrow (2020), the BASE architecture was used to advise and trace the manual operations of a worker through an assembly process. The BASE architecture has since been applied to various case studies to extend the architecture to improve system scalability and robustness, and to evaluate its performance in a manufacturing environment.

The BASE architecture was developed within the Mechatronics, Automation and Design research group at Stellenbosch University, which focuses on CPS and human integration architectures through the implementation of DTs. This thesis builds on the research efforts of the group by developing a holonic HCPS with a focus on human integration within complex environments.

1.2 Objectives

The objective of this thesis is the development of a holonic HCPS, using the BASE architecture, in a healthcare environment.

This thesis develops and presents the requirements for HSI, to better understand the technical consideration of the human component in HCPS and the evaluation thereof. The *holonic HCPS* is developed to automate and coordinate clinic-related processes to satisfy HSI and healthcare requirements. The development is focused on integrating existing or new technologies and does not focus on organisational management principles. The use of the *BASE architecture* entails the customisation of BASE digital administration shells and the development of the BASE platform for the holonic HCPS implementation. The *healthcare environment* provides the context for studying interactions between humans (i.e. patients and healthcare practitioners) and systems (i.e. machines, information systems, decision support systems).

This study considers the BASE architecture for the software design and implementation and does not explore alternative architectures. Furthermore, aspects of general HSI, such as human-centred design and human factor principles, are only superficially considered in this study and are not the focus of this thesis.

1.3 Motivation

The *magic human* describes a scenario in which physical components, in an HCPS, assume that the human component (in the system) is fully aware of their tasks (planned or in execution), the required interactions, and goals – which in reality is untrue (Trentesaux & Millot, 2016). Often, the *magic human* results in production errors, safety concerns and inefficiencies due to a lack of shared awareness between humans and other physical components. The healthcare environment is similar in this respect to production environments. There are many human errors reported in healthcare, such as healthcare practitioners (HPs) forgetting, mistreating and neglecting patients due to limited awareness of patient conditions and clinic decisions. This issue is further exacerbated by the lack of decision-support systems to aid HPs.

HSI developments aim to introduce *human-in-the-loop* (HitL) principles to avoid the emergence of the *magic human* in systems. Therefore, the HSI contributions of this thesis will benefit HPs and the clinical care process by reducing the frequency and impact of avoidable errors by improving the awareness and leveraging the intelligence of HPs.

Furthermore, HPs are under-resourced, required to multi-task and work for long durations in time-pressured and high-risk environments. Sensors, devices and alarms are common in healthcare facilities to improve HPs awareness of patient conditions. As an unfortunate result, HPs suffer from alarm fatigue in clinical environments, due to the vast range of different threshold alarms from medical sensors and devices. This has led to the failure of HPs to respond to critical alarms, caused by alarm desensitization (Sendelbach & Funk, 2013). Additionally, HPs are often tasked with data handling, transport, storage and formatting, which demand significant time and attention, leading to the depersonalisation and reduced quality of service of HPs (Cañadas-De la Fuente, Vargas, San Luis, *et al.*, 2015). The development of the holonic HCPS in this thesis can support HP tasks, improve their quality of care and reduce their experienced workload.

Healthcare facilities, especially in developing countries, experience high patient volumes, lack physical and computational resources, have poor record-keeping mechanisms and suffer regular adverse events (Maphumulo & Bhengu, 2019). Unfortunately, these facilities are not supported by flexible software systems that aim to optimise workflows and resource utilisation. Chapter 10 of the National Development Goals for South Africa motivates "*Healthcare for all"*, with a priority on improved healthcare systems and health information systems (National planning commission, 2012). According to the National Health Strategic Plan, during 2018-2019, a recorded 31% of patient complaints were about long waiting periods, while 8% were related to poor access to patient information. Patients are often required to wait for long periods during clinic visits with little feedback on the availability of HPs. This thesis explores HSI developments that could aid HPs and support the transparency and traceability of clinic processes.

Data integration within healthcare remains a high priority. Often, patient data is distributed in data silos with limited accessibility by patients and HPs. The Tygerberg Hospital has multiple ambulatory clinics and serves a large population in the Western Cape, South Africa. During patient consultations, doctors must access multiple data sources through different web portals, each of which require different login credentials. Patients' data is also inaccessible to patients unless it is accessed through a verified HP. The National Digital Health Strategy for SA 2019-

2024 motivates the research and development in this thesis (National Department of Health, 2019) by prioritising:

- Complete electronic health records to improve patient management.
- Digitisation of health system business processes at an institutional level to improve efficiency and quality of human resources.
- Person-centred developments, meaning, "individuals and families [must be] be more involved in and able to influence the healthcare required".
- Innovative developments for increased automation and improvement of healthcare decision-making.

HCPS research has grown in popularity over the past decade and has contributed to many solutions to fulfil the I4.0 requirements. HCPS developments prioritise interconnectivity and networking between various system components. This thesis furthers research in HCPS developments, with a unique application of HCPS concepts to healthcare processes. Furthermore, this thesis investigates the human component in HCPSs more thoroughly by better defining the requirements for HSI. These HSI considerations can be applied to other application domains to aid the development process of HCPS.

The BASE architecture has been applied to manufacturing, maritime and agriculture contexts, but this thesis aims to refine and evaluate the BASE architecture for the healthcare context with greater attention to HSI in the HCPS. The BASE architecture has predominantly focused on integrating physical components (e.g. sensors, manufacturing cells) with limited attention to complex interactions between multiple human actors and roles. This thesis provides a thorough application of the BASE architecture in a complex system with a higher dependence on different human components.

Furthermore, the BASE architecture was built using holonic systems principles. Holonic principles, developments and resulting HCAs have been thoroughly explored in production environments and Manufacturing Execution Systems (MES). Holonic systems principles promote flexibility and reconfigurability for control systems. This thesis presents an evaluation of an HCA for healthcare processes that includes the integration of humans. This offers new insights for holonic systems research and developments in terms of HSI. Additionally, the application of an HCA to a non-manufacturing domain supports the efforts for broader adoption of HMS concepts.

1.4 Methodology

This thesis departs from a literature review in which Sparrow (2021) highlighted a lack of research developments to aid workers in modern production systems. The literature review in Chapter 2 gives insight into the challenges and benefits of HSI developments for complex HCPS through research into current I4.0, HCPS, HDT and holonic concepts and contributions. The human system component and integration challenges for HCPS were further explored by the author to develop a better understanding of HSI requirements. The definition and requirements for HSI, explained in Chapter 3, were established to guide the selection and development of the holonic HCPS.

The holonic principles did satisfy the needs and requirements of HSI in HCPS, as argued in Chapter 3. The Activity-Resource-Type-Instance (ARTI) architecture (Valckenaers, 2019), developed within the HMS paradigm, and the BASE architecture were selected to develop the holonic HCPS for healthcare. The benefits of ARTI in conjunction with the BASE architecture were argued by Wasserman (2022). A design process for holonic systems was established (presented in Chapter 4) to guide the development of holonic HCPS using the ARTI and BASE architectures. The BASE architecture was implemented using Erlang due to its robustness, scalability and performance, as justified by Kruger & Basson (2019).

The scope of the healthcare domain was limited to that of a developing country, such as South Africa, which has barriers such as poor medical information systems and insufficient physical resources to handle the high patient volumes. Research insights into existing healthcare technologies and methods, such as hospital management systems, medical information systems and medical CPSs, were used to perform a needs analysis to guide the holonic HCPS development, as discussed in Chapter 5.

A case study approach was selected to focus the development of the holonic HCPS, as described in Chapter 6. Using a case study limited the development time and complexity to what is achievable within the scope of a thesis, while still allowing a detailed evaluation of the holonic HCPS in a representative context. An ambulatory clinic scenario was selected for the case study, as it portrayed operational complexity and HSI challenges reported in literature. The development was guided by the holonic system design process.

Evaluation of the case-specific holonic HCPS was performed through an experiment in a laboratory scenario, to allow for ethical experimentation and to improve the controllability of external factors during the system validation and evaluation. The results aid the discussion, in Chapter 7, of the performance of the holonic HCPS. A limitation of the case study approach is that the research results cannot be universally accepted and applied to all healthcare environments and cases. This thesis acknowledges that the research results are only proven for one specific case and the generalisation thereof, while valuable, would be guided by the results of this single case study.

This research was exempt from ethics clearance. No sensitive or human-related data was required during the development and evaluation of the holonic HCPS as the focus of this study was on the technologies (e.g. interface devices, web applications, IoT networks, etc.) in such contexts. Furthermore, no interaction with patients or medical procedures was required for this study.

2 Literature Review

This chapter presents a review of concepts, methods and technology found in literature that applies to the developments of this thesis. Key principles and technologies within I4.0 research are described as they provide foundational concepts on which to build an argument for the interconnectivity between system components within HCPS and HSI developments. Holonic system concepts and applications are reviewed, with specific consideration to HCAs, to highlight the requirements, design principles and key contributions for HMSs that are applied in the case study design. Proceeding from I4.0 technologies and holonic systems, a more in-depth review of the human component in HCPSs is presented, followed by an overview of HCPSs in healthcare. Finally, the BASE architecture and related concepts are discussed to aid in understanding the case study implementation decisions and details.

2.1 Industry 4.0

2.1.1 Definition

Oztemel & Gursev (2020) offered an extensive review of I4.0, defining it as "a manufacturing philosophy that includes modern automation systems with a certain level of autonomy, flexibility, and effective data exchanges encouraging the implementation of next-generation production technologies, innovation in design, and more personal and more agile production systems as well as customized products".

A review by Sony & Naik (2020) presented an integration-centric view of I4.0 – stating that it aims to improve the horizontal and vertical integration of production systems and product value chains. Horizontal integration promotes the interconnectedness of system components, such as software and control systems, machinery, workers and data flow on the production floor, between facilities and along the value chain. Vertical integration describes the transparency of data throughout the organisational structure, from the operational activities to the strategic management team and stakeholders, to promote data-driven decisions.

2.1.2 Design Principles

Growing and stochastic demand for customised products, with short waiting times, creates complex challenges for modern manufacturing (Leitão, Casais & Restivo, 2005). I4.0 in production systems is driven by the shift to meet these demands, leading to new production requirements (Neumann, Winkelhaus, Grosse, *et al.*, 2021).

A review of I4.0 developments highlighted a specific set of design principles. *System integration* is enabled by connecting sub-systems and system components to act cohesively as one overall system (Rüßmann, Lorenz, Gerbert, *et al.*, 2015).

Interoperability is the capability of the systems to exchange information with other systems, including humans, equipment and technologies to enable connection, communication and cooperability (Ruppert, Jaskó, Holczinger, *et al.*, 2018). *Virtualisation* promotes the creation of a digital representation (e.g. a DT) of factories and systems, using aggregated sensor data, in real-time, to enhance and optimise processes (Ghobakhloo, 2018). *Modularity* and *decentralisation* support the shift from rigid to more flexible systems by allowing greater autonomy within system resources for decision-making to achieve the overall goal of the system (Ghobakhloo, 2018). *Flexibility* describes the ability of a system to accommodate various products, while *adaptability* describes the ability of a system to handle disturbances.

Manufacturing processes designed for mass customisation, typically seen as production lines with repetitive tasks, are not capable of supporting such an industry shift. The need for decentralisation and interconnectedness of systems are some of the new production requirements to cope with these complex processes (Beier, Ullrich, Niehoff, *et al.*, 2020). In manufacturing, these principles enable smart products that make use of sensor networks and embedded sensors in and surrounding products, which enhance productivity in manufacturing processes. The improved accessibility and availability of product information can inform decision-making due to the awareness of the product's progress, and current and desired state. Smart factories aim to integrate all physical resources through the Internet of Things (IoT) infrastructure (Ghobakhloo, 2018).

2.1.3 Cyber-Physical Systems and Digital Twins

CPS developments aim to fulfil the I4.0 design principles of system integration, modularity, decentralisation and virtualisation. Furthermore, they aim to connect the physical world to the virtual (or cyber) world, through integrating networking infrastructure, computing resources and storage mechanisms (Pivoto, de Almeida, da Rosa Righi, *et al.*, 2021). CPSs are essentially systems which can exchange data between devices through a network. Cyber systems typically coordinate, supervise and analyse the overall system processes, while the physical systems provide the services and actors for producing the product or providing a service.

The 5C architecture, as seen in Figure 1, has received much research recognition when developing CPSs. The *connection* layer encompasses various sensors and networking infrastructure for data acquisition. The *conversion* layer encapsulates the use of algorithms and methods for making acquired data useful (i.e. information), such as in the maintenance management of equipment. The *cyber* layer enables information comparison between resources in the facility (or overall system), essentially processing the network of information. The *cognition* layer translates the network of information to knowledge for decision-making among users. Finally, the *configuration* layer completes the feedback loop from the cyber layer back to the physical layer, resulting in corrective actions based on the knowledge gained from the CPS. (Lee, Bagheri & Kao, 2015)

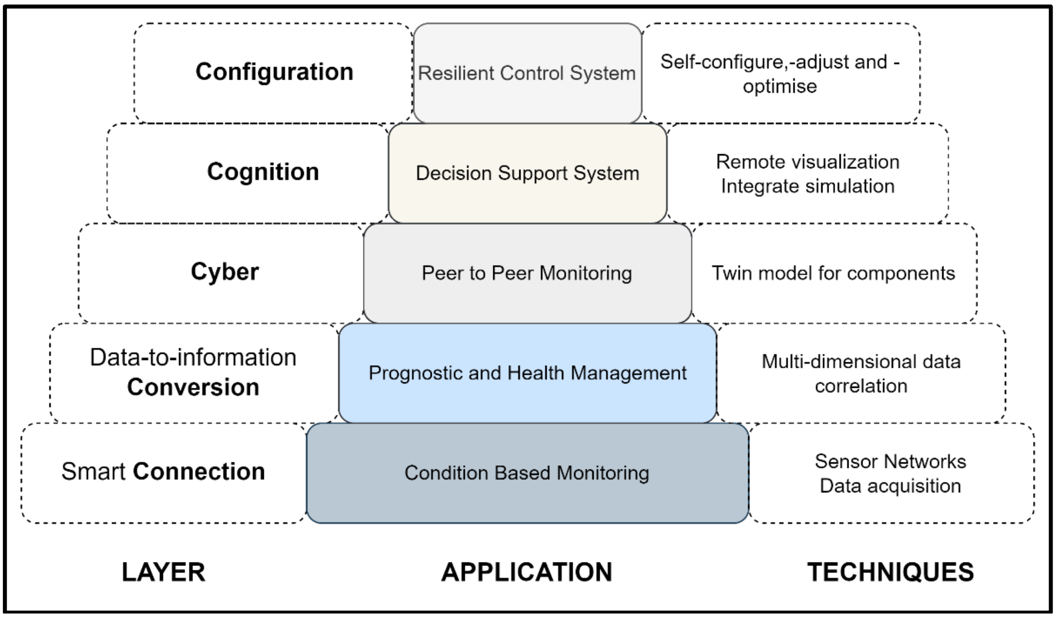


Figure 1: 5C architecture for developing a CPS for I4.0 (adapted from Lee *et al.* (2015)).

The DT concept originated from the National Aeronautics and Space Administration and has since gained popularity in many domains. Many definitions have since been published that attempt to explain the DT concept. The fundamental principles governing what a DT should entail or the services it should provide are still developing. DTs, according to Schluse & Rossmann (2016), are “virtual substitutes of real-world objects consisting of virtual representations and communication capabilities, making up smart objects that act as intelligent nodes inside the internet of things and services”. These “*virtual substitutes*” exist purely in software, but may be distributed on various hardware components. Furthermore, DTs allow for the simulation of physical objects and processes for prediction and optimisation during runtime (Gabor, Belzner, Kiermeier, *et al.*, 2016). DTs provide a means of reflecting the world of interest in a digital system from which real-time and predictive decisions can be made. In a production setting, there could be a DT of each physical component (e.g. product, machinery, person). A six-layer architecture for DTs, presented by Redelinghuys, Basson & Kruger (2020), supported the I4.0 goals of modularity, reconfigurability and flexibility, and satisfied all layers of the 5C architecture.

2.1.4 Reference Architecture Model Industry 4.0

The Reference Architecture Model Industry 4.0 (RAMI 4.0) (Zezulka, Marcon, Vesely, *et al.*, 2016) is an extension of the Smart Grid Architecture Model (SGAM) (CEN/CENELEC/ETSI Joint Working Group on Standards for Smart Grids, 2012). The SGAM framework was developed to support the design of smart electrical grids for decentralised power management in Europe. The three-dimensional model represents five interoperability layers – business, function, information, communication and component – in a technology-neutral manner. Basic connectivity is addressed in the component interoperability layer. The component layer addresses the physical distribution of system actors, applications, equipment and network infrastructure. Network interoperability, being a mechanism for data exchange, and syntactic interoperability, the understanding of exchanged data, are addressed in the communication layer of SGAM. This layer describes the protocols and mechanisms for data exchange between system components.

While SGAM was developed from a power management perspective, RAMI4.0 proposes small modifications to enable a more general perspective, as seen in Figure 2. RAMI4.0 describes six interoperability layers: business, functional, information, communication, integration and asset. The asset layer, like the component layer in SGAM, represents reality (e.g. physical system components). These are connected to the virtual reality layer, using hardware and software components described in the integration layer.

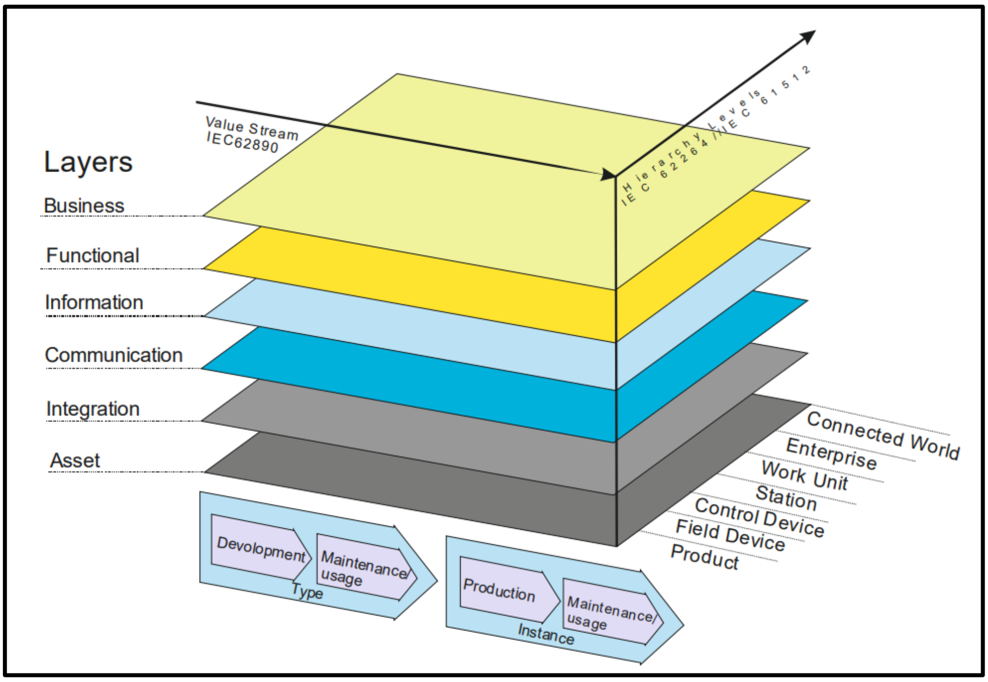


Figure 2: Reference Architecture Model for Industry 4.0 (Zezulka *et al.*, 2016).

Extending from RAMI4.0, an I4.0 component model, seen in Figure 3, was developed to improve the description of cyber-physical objects. The virtual representation of each system object must be data-structured, thus requiring an administration shell to contain specific information and enable standardised communication between system objects. The requirements for the component model are:

- I4.0 components must be structured in a network in such a manner to allow exchange between all components.

- The I4.0 concept must be adoptable in different focal areas of a business (i.e. shop floor, office floor, etc).
- The communication protocol must ensure that the data of the virtual representation be contained in the I4.0 component model itself or a higher-level layer.

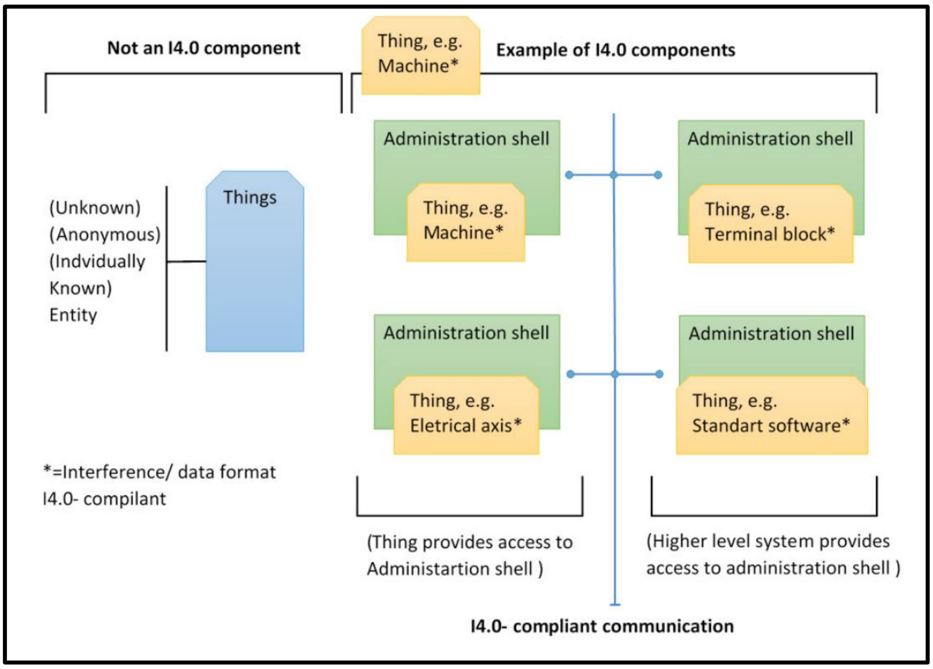


Figure 3: Industry 4.0 component model (Zezulka *et al.*, 2016).

2.2 Holonic Manufacturing Systems

This section explores HMS and MAS concepts and implementations, which support I4.0 design principles for system control and integration. Two HMS architectures are explored in-depth as they are closely related – the latter being used in the development of the holonic HCPS for this thesis. Although HMSs are focussed on manufacturing, the principles and control philosophies apply to a range of complex operational systems. MASs were explored along with HMSs due to the similar philosophies in both. HMSs and MASs promote modularity, scalability and robustness in control systems which better reflect the true reality of operational systems.

2.2.1 Holonic Systems Concept

A *holon*, termed by Arthur Koestler (1967), is a system entity that operates with autonomy and cooperability among a *holarchy* of entities to fulfil both independent and collective goals. A holon is regarded as a *whole*, requiring it to be self-sufficient, independent and cohesive between internal parts. The holon is a stable entity that manages internal communication and processes. Additionally, a holon is regarded as a *particle* or *part* within the greater system, with external communication capabilities and functional specialisation to enable cooperability between holons. This paradigm has been observed from biological, social and organisational structures and has been applied to production systems. Holarchies bring stability to complex systems that inherently display emergent behaviour. This is due to the decomposability of each holon, thereby avoiding reliance on the stability of the entire system.

2.2.2 Architecture and Implementation

Manufacturing Execution Systems (MESs) typically handle the control and execution of processes within manufacturing operations, adhering to Computer Integrated Manufacturing (CIM) principles and technologies (De Ugarte, Artiba & Pellerin, 2009). In an attempt to overcome some of the CIM challenges in MESs, such as barriers to flexibility and reconfigurability, HMSs were developed. These proved effective in managing complex system behaviours with many interacting elements (Leitão *et al.*, 2005). The development of HMSs is guided by HCAs which allocate various physical system components, processes and functions to specialised holons (Van Brussel, Wyns, Valckenaers, *et al.*, 1998).

HCAs adopt a heterarchical control philosophy, rather than the hierarchical control and centralised control philosophies which are common in CIM. HCAs reduce complexity through localising information and control for system entities. Furthermore, they improve the maintainability and reconfigurability of the control system. Rather than demanding a fault-free system, HCAs adopt a fault-tolerant approach. HCAs should prioritise globally mutual cooperability and independence with few predetermined interactions, but rather well-defined protocols. (Duffie, 1990)

The complexity of manufacturing systems has led to the aggregation of holons, which introduces some sort of a flexible hierarchy into the holarchy. Aggregation can be seen in DT research as well – offering modularity and reconfigurability to these systems (Redelinghuys, Kruger & Basson, 2020). Each system component within the production environment is represented digitally by a holon, depending on the HCA, which can be aggregated in the holarchy according to the specialisation and functional groupings of the various holons.

Similar to holonic systems and HCAs, the Multi-Agent System (MAS) paradigm has been pivotal in realising the driver of I4.0 by overcoming traditional production shortcomings (Wang, Wan, Li, *et al.*, 2016; Ruppert *et al.*, 2018). Agents represent decision-making or service-providing entities that perform tasks autonomously to meet their design objectives. According to Adeyeri *et al.* (2015), this paradigm has been typically used to optimize resource scheduling and production planning, and to evaluate worker performance. HMS and MAS can be realised through *actor* or *agent-based* software, such as Erlang and the Java Agent Development (JADE) Framework (Kruger & Basson, 2019).

2.2.3 Product-Resource-Order-Staff Architecture

The Product-Resource-Order-Staff Architecture (PROSA) (Van Brussel *et al.*, 1998) is one of the early HCAs developed for HMS. PROSA has been well documented and applied in many manufacturing cases, which has shown promising benefits for improving the control of manufacturing processes.

In PROSA, HMS components are specialised in *Product*, *Resource* and *Order* holons as seen in Figure 4. Resource holons provide a digital entity for physical components, such as conveyor belts, robots, workers and tools. The Resource holon provides information about and to the physical component, thus enabling control over it. The Product holon encapsulates all the information about the product and the manufacturing process required to make it. It does not represent a specific instance of the product being manufactured, like the Order holon, but rather stores the user requirements, design, process plans and quality assurance procedures information associated with the product. However, the Order holon is specific to a product being manufactured, i.e. there would be one Order holon for each product currently being manufactured. This holon assigns and coordinates the correct tasks to other holons by managing a specific workpiece as it moves through the factory. The authors of PROSA realised the possibility of a Staff holon which would act as an expert, providing Order holons with additional analysis information to assist the Order holon's decision making.

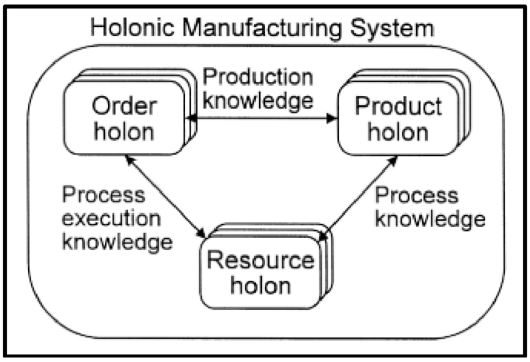


Figure 4: Basic building blocks and relations of an HMS (Van Brussel *et al.*, 1998).

2.2.4 Activity-Resource-Type-Instance Architecture

Developed more recently, the Activity-Resource-Type-Instance (ARTI) architecture (Valckenaers, 2019) offers a refinement of PROSA. The terminology presented in PROSA was considered too specific to manufacturing (i.e. product and staff holon); as such, the ARTI architecture introduces terminology that can be applied to many domains. In ARTI, *Product* and *Order* holons become *activity types* and *activity instances*, respectively. Additionally, *Resource* holons are segmented into *resource types* and *resource instances*.

Activity holons are distinct from resource holons, as activity holons should be able to use and contract any resource holons to fulfil the required tasks. Activity Type (AT) holons represent the class of activity and are equivalent to the Product holon in PROSA. AT holons contain the related information and knowledge about what needs to happen in the process to complete the activity. In terms of a product, this might include information such as the bill of materials, assembly requirements, process plans, etc. Similar to Product holons, AT holons do not represent specific instances of products. Rather, Activity Instance (AI) holons “inherit” the AT holons' knowledge for the specific activity being performed in the environment, but are unique for each activity that needs to be completed in the environment. AI holons coordinate with Resource Instances (RIs) and other AIs to complete certain processes.

Resource Type (RT) holons, like AT holons, refer to the class of resources. For example, specific workers, equipment and information systems would require RT holons. These holons contain standardised information across all instances of that type of resource. RI holons inherit certain standardised information from the RT, but further represent an individual resource with information as to its schedule, current execution state and history.

Humans, as discussed by Valckenaers (2019), are not seen as resource holons, but as *activity performers*. Valckenaers (2019) motivates an ARTI digital extension for humans to improve their integration, but provide limited detail on how such an extension for humans might be implemented. The authors state that “*humans will need a digital extension to be able to fly as fast and far as aircraft, metaphorically*” – implying that if other inanimate systems are integrated into ARTI, humans must be as well so as to effectively integrate all the system components.

2.3 Human Cyber-Physical Systems

2.3.1 Definition

Referring to humans and machines as entities and agents is not foreign in MAS and HMS research domains. Entities can be both physical components (e.g. humans or machines) or virtual components (e.g. software entities). In systems engineering, these entities can be regarded as system components, which may exist in a bigger system, more commonly known as a system of systems (SoS). These entities have *properties* and *behaviours* which contribute to the larger system. These entity behaviours may involve interactions with other entities, intentionally or unintentionally, which results in the overall behaviour of the bigger system.

The behaviour and interactions of humans, within systems, has gained research attention beyond that only social and ergonomic considerations. HCPSs are systems which aim to consider the holistic nature and integration of humans in the system. An HCPS, according to Yilma, Panetto & Naudet (2021), is “*a system of interconnected systems (i.e. computers, devices and people) that interact in real-*

time, working together to achieve the goals of the system – which are ultimately human goals”.

HCPSs differ from typical CPS in their integration of human attributes, such as *cognition*, *predictability* and *motivation* (Sowe, Simmon, Zettsu, *et al.*, 2016). *Cognition* describes how humans can perceive their environment through various modes (e.g. sight, sound, touch), process decisions and then act. In contrast, non-human entities (such as conveyor belts) do not have cognitive function; rather, they rely on their cyber representation to provide intelligence. *Predictability* describes how humans can adapt to changing environments and disturbances. This makes it both beneficial and challenging for digital systems to predict the intention of the worker. Finally, *motivation* promotes incentives for worker participation and encourages a high standard of work. Humans act as sensors, decision makers and actors. These attributes offer unique adaptions to the typical CPS model.

The Operator 4.0 concept, depicted in Figure 5, provides a conceptual definition of HCPS. It describes a human operator who has enhanced physical, sensing and cognitive capabilities through integration with machines (Stahre *et al.*, 2016). The Operator 4.0 can integrate with highly digital (cyber) systems effectively, as enabled by I4.0 technologies. These technologies aim to track and support the operator to support human-in-the-loop principles (Ruppert *et al.*, (2018); Cimini, Pirola, Pinto, *et al.*, (2020)). These principles ensure the integration of humans incorporated into the decision loops and activities of production systems. Though the Operator 4.0 concept is popular in research, few implementations have attempted to realise the idyllic concept.

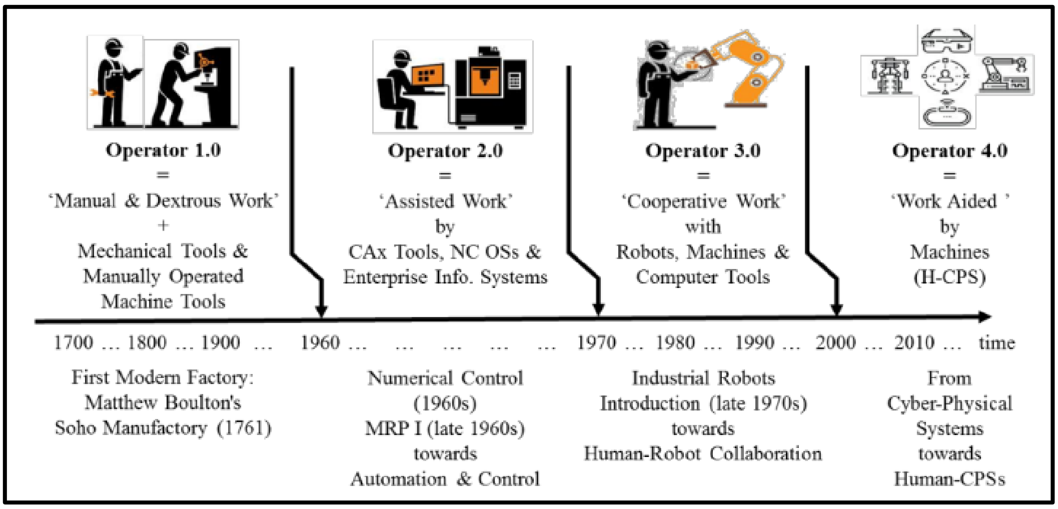


Figure 5: Evolution of HCPS for production systems (Romero *et al.*, 2016).

2.3.2 Human Intelligence Characteristics

Humans are a system in themselves, but most often they are part of a greater SoS. The system domain affects the consideration of different human properties and behaviours. Nevertheless, considering humans from a systems engineering perspective does aid the integration of humans into the SoS. Humans possess certain characteristics which should be understood when attempting to integrate them.

Human behaviour is unpredictable due to their flexible nature, thus leading to inconsistencies (Kahlen, Flumerfelt & Alves, 2016). Humans have various properties, such as age, location and skill, which can affect the way that they behave or how other humans or entities might interact with them. Human behaviour and functionality are highly unpredictable since their decision-making is inherently independent and autonomous. The same capabilities also make humans extremely flexible.

Humans are inherently adaptive due to their sense making, as opposed to a computer that only takes an input and produces an output (Kahlen *et al.*, 2016). This enables humans to learn from previous errors and feedback, retaining tacit knowledge, to improve their future behaviour. The adaptive nature of humans can also be seen in their reactive behaviour to external events. Humans are capable of sensing faults, opportunities and risks, and adjusting their behaviour accordingly to mitigate or capitalise on events.

The characteristics of an intelligent agent described by Romero *et al.* (2016) can be seen in human behaviour:

- **Purposeful:** Displays goal-seeking behaviour.
- **Perceptive:** Can observe information about the surrounding world and filter it according to relevance for orientation.
- **Aware:** Can develop situational awareness that is relevant for the agent’s purpose.
- **Autonomous:** Can decide on a course of action (plan) to achieve a goal.
- **Able to act:** Can mobilise its resources to act on its plan; these resources may include parts of the self or tools at the autonomous disposal of the agent and resources for physical action or information gathering/processing.
- **Reflective:** Can represent and reason about the abilities and goals of itself and those of other agents.
- **Adaptability and learning:** Can recognise the inadequacy of its plan and modify it, or change its goal.
- **Conversational and cooperative:** Can negotiate with other agents to enhance perception, develop common orientation, decide on joint goals, plans, and action; essentially participate in maintaining the ‘emergent’ agent created through joint actions of agents.

Humans already possess these characteristics, which inherently make them intelligent agents. The use of automation and machines might better improve the agenthood of humans (Romero *et al.*, 2016).

2.3.3 Operational Role of Humans

The unique capabilities of humans as system elements contribute to the unique roles which they can offer in HCPSs. The specific function that humans perform is dependent on the domain and system goal in which they work. Cimini *et al* (2020) described the role of humans, in a manufacturing context, as follows:

- **Data acquisition**: Humans can use their sensory and cognitive ability to provide a system with data on the system's properties or behaviour. This supports the idea of humans being sensors for a system.
- **State inference**: Humans have a cognitive ability to process and evaluate data on the system's properties and behaviour. The human can then provide feedback and aid decision-making for the system.
- **State/system influencing**: Humans the system's behaviour depending on their properties and behaviour. An example would be activating the safety stop for a robot when walking through the collision path.
- **Actuation**: Humans can behave (or act) within a system.

These unique functions can be performed by humans with varying levels of success, depending on the assistance of other systems. Furthermore, the unique role of humans can aid machines in their shortfalls.

2.3.4 Cooperation in Human Cyber-Physical Systems

Each system component, whether human or machine, must cooperate to achieve the overall goals of the system. Communication between system elements enables cooperation. Pacaux-Lemoine *et al.* (2017) refer to these system components as “a decisional entity”. Two agents are in cooperation when both strive to achieve individual goals while not interfering with the other agent’s goals. Furthermore, they should “try to manage such interference to make others’ activities easier”. According to Pacaux-Lemoine *et al.* (2017) and using the work of Millot & Lemoine (1998), agent cooperation requires the necessary *know-how* and *know-how-to-cooperate*, as seen in Table 1.

These generic requirements were used by Pacaux-Lemoine *et al.* (2017) to implement an intelligent manufacturing system case study. An assistant system was developed to aid the human operator in a flexible manufacturing system. The case study evaluated the changes caused by a human-centred design on the system performance using metrics, such as energy used and production time. Furthermore, human factors such as workload and ease of use were also assessed. Pacaux-Lemoine *et al.* (2017) motivated that the manufacturing system performed better when the human is in the loop and can cooperate.

Table 1: Cooperative agent principles by Pacaux-Lemoine *et al.* (2017).

	Know-How	**Know-How-to-Cooperate**
Internal Ability	Mode knowledge, rules, skills/experience, expertise; Process workload, fatigue, distraction.	Build a model of other agents; Deduce the other agents' intentions; Analyse the task and identify the cooperative organisation; Produce a common plan regarding tasks and coordination.
External Ability	Get information (from the process and the environment); Act (on the process).	Understanding other agents; Providing other agents with information.

2.4 Human Cyber-Physical Systems in Healthcare

2.4.1 Overview

HCPSs and CPSs have been introduced in healthcare operations. Medical Cyber-Physical Systems (MCPSs) are a specialised form of CPSs involving the integration of medical devices. Such systems are becoming more prevalent in hospitals to maintain efficiency and high-quality care. An overview of MCPS are highlighted in this section to show the two key fields of MCPS applications.

Dey, Ashour, Shi, *et al.* (2018) conclude that MCPSs can be classified into either *assisted* or *controlled* applications. *Assisted* applications involve the use of biosensors to capture physiological data from a patient to aid the health of patients. *Controlled* applications are typically found in intensive care or hospital environments to improve medical workflows. Controlled MCPS applications need to capture and process information from several sources, including bedside monitors, physicians' observations and biosensors, to aid decision making about interventions and treatment for patients.

2.4.2 Applications

Le *et al.* (2010) proposed a solution called *Secured wireless sensor network-integrated Cloud computing for u-Life care* (SC^3). The WSN application uses various sensor technologies and cloud computing services to monitor patient activities and enable information sharing between doctors, caregivers, clinics and pharmacies. The application was applied to a case study involving Alzheimer's disease patients. The patients would be monitored in their home environment. If the patient falls or performs certain activities, the system should recognise and relay such information to the appropriate HP. Figure 6 illustrates the applied architecture in which doctors could receive alerts and access patient information and analysis through web applications.

Lee *et al.* (2012) developed a closed-loop physiological control MCPS to create context-aware smart alarms for caregivers, seen in Figure 7. Various medical devices captured patient information on the electronic health record (EHR) to be used in the decision support system to detect ailments and generate effective alarms for caregivers. The administrative support system, decision support system and caregivers could utilise various treatment devices either manually or through automation. Lee *et al.* (2012) highlighted the necessity for interoperability between medical devices and ensuring the safety and security of such integration.

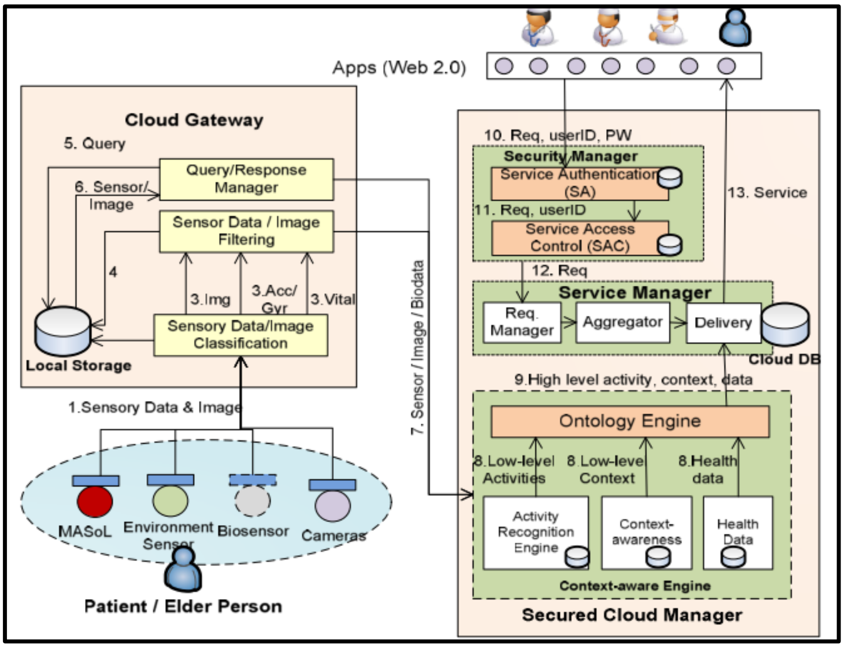


Figure 6: Functional architecture of SC^3 (Le *et al*, 2010).

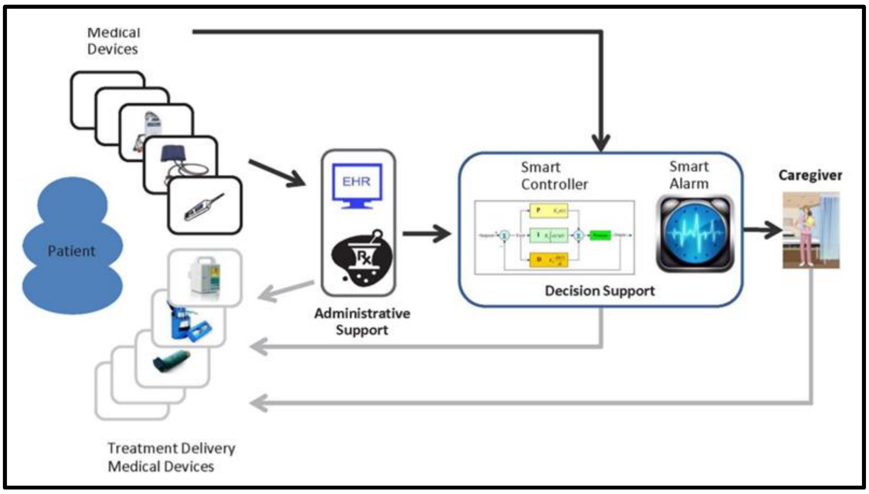


Figure 7: Smart alarm MCPS architecture (Lee *et al.*, 2012).

2.5 The BASE Architecture for Human Integration

This section presents the drivers and principles that led to the development of the BASE architecture. Furthermore, it provides an overview of the architecture and related applications.

2.5.1 Administrative Logistics

Sparrow (2021) discussed *administrative logistics* as presented in *Administrative Behaviour: A study of Decision-Making Processes in Administrative Organisations* (Simon, 1997). The concept defines general activities and interactions observed in the operation process, such as manufacturing, specifically involving humans. Administrative Logistics (ALs) are any mechanisms that allow for entities to perform Administrative Cognition (AC) functions. AC functions include *managing responsibility, training, advice, decision-making, analysis,* and *activity execution*. AL functions include *data handling, data transport, data formatting* and *data storage*.

2.5.2 Three-Stage Activity Life Cycle Model

Sparrow, Kruger & Basson (2020) formalised a common definition and structure of an activity, concerning time, to improve the communication protocol between physical and digital executors. Hence, the three-stage activity life cycle (3SAL) model provided this definition and standardisation. The model defines an activity as a "structured sequence of actions, performed in the physical world by humans or machines, or in the cyber world by software".

An activity can be split into three stages, as seen in Figure 8. The *origo* is the point in time where the activity changes from stage 1 to stage 2. Stage 1 data contains the information necessary for the execution of the activity, increasing with detail as time approaches the *origo*. Stage 2 data contains the schedule and execution data that is captured from various sensors and inputs, during the execution of the activity. Stage 3 data contains the scheduled and execution data, as well as the post-execution data which is gathered once the activity is complete (e.g. from a quality inspection).

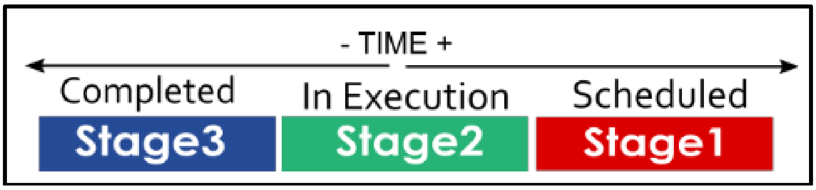


Figure 8: Three-stage activity life cycle (3SAL) model (Sparrow, Kruger & Basson, 2020).

2.5.3 Architecture Design

Sparrow (2021) created an administration shell, guided by HMS principles, to improve the AL functions for humans in a production environment. Each resource in a production environment would have an administration shell that represents the physical resource as a cyber component.

The BASE architecture, as seen in Figure 9, consists of core and plugin components. Core components reflect the properties and state of the BASE instance, while plugins encapsulate the logic for the BASE instance's functionality. The communication manager (CM) component is at the centre of the BASE core and manages all communication and data exchange between holons. The *Schedule* component maintains a digital record of planned activities, according to the 3SAL model (Sparrow, *et al.*, 2020). The *Execution* component would handle any activities that are being executed, storing the data related to these activities. The *Biography* component handles the tasks which are finished executing and stores the related historic data. The *Attributes* component stores all properties of the BASE instance. Plugins can be developed for specific applications to provide case-specific functionality to each holon. These plugins can augment AL functions, to improve the integration of resources in the HCPS.

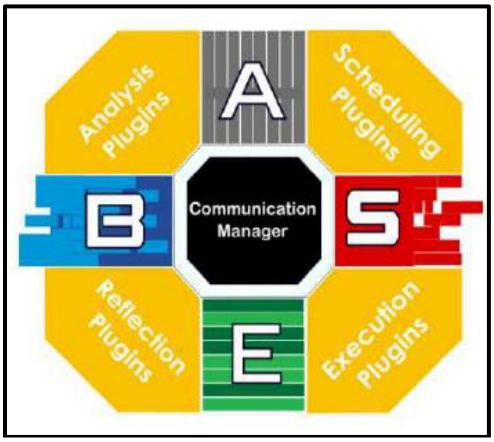


Figure 9: BASE architecture (Sparrow, 2020).

The schedule plugins, encapsulate the functionality to schedule various activities and add them to the BASE instance's schedule. Once at the *origo* of the activity stage, the execution plugin receives the activity and handles all decision-making processes to fulfil the execution target of the activity. The current activity and related data are stored in the *execution* component of the holon as stage 2 data. Completed activities and data are then stored in the *biography* via the reflection plug-in, which can be used to update the *attributes* of the instance using analysis plugins.

The BASE architecture allows for the realisation of DTs and HDTs in HCPSs by augmenting certain cognitive and cooperative functions. The various holon instances can communicate via the CM component to form a holarchy of interactions.

2.5.4 Applications

Sparrow (2020) used the BASE architecture to improve the AL functions of human workers in a production environment. Workers were tasked with manual tasks to develop components for the aerospace industry, which prioritises traceability and quality assurance. The BASE architecture was shown to aid in improving the AL of workers and enabling traceability of worker tasks.

Van Niekerk (2021) extended the BASE architecture for the scalability and robustness required in CPS. The extended architecture was developed for monitoring the feeding rates of sheep on a farm. The extended architecture improved the development productivity by generalising BASE core components.

The BASE architecture has also been mapped to the ARTI architecture. Wasserman (2022) developed a Holonic Manufacturing Execution System (HMES) for a Fischertechnik Industry 4.0 Training Factory. The evaluation of the HMES case study proved that the BASE architecture can successfully manage the production schedule, resource allocation and processes according to the ARTI architecture. It was concluded that the BASE architecture can be used as the foundation for all ARTI holons in the system.

Taylor, Kruger & Bekker, (2020) used the BASE architecture to develop a system for improving the comfort experienced by seafarers during a voyage. The developed HCPS enabled the real-time data acquisition of comfort ratings from seafarers which could be analysed along with acquired data from the ship's sensors. This study showed the benefit of using the BASE architecture for the integration of humans in real-time.

2.5.5 Erlang

The implementation software for the BASE architecture applications was developed using the Erlang programming language, which utilises the Open Telecom Platform (OTP) collection of libraries. The Erlang Runtime System (ERTS) compiles Erlang source code into bytecode, which is executed by the BEAM virtual machine part of the Erlang OTP.

Erlang is a functional programming language developed by Ericsson and has since improved to support scalable, distributed and concurrent systems with fault tolerance. Erlang’s actor model supports modularity and robustness, thereby making it a note-worthy programming language for implementing holonic systems. Erlang does lack negotiation standards (i.e. Foundation for Intelligent Physical Agents) and development tools compared to other MAS programs (e.g. JADE). Nevertheless, Erlang has computational, integrability, distributed, and fault tolerance advantages. (Kruger & Basson, 2019)

2.6 Discussion

Key I4.0 architectures and concepts attempt to enable interoperability across various functional layers in operational processes. These concepts generally prioritise modularity and encapsulation, standardised communication and mechanisms of information exchange between physical and virtual components. The reviews of I4.0 applications are often focused on only manufacturing contexts, but in other domains, I4.0 concepts were also seen to be adopted for guiding developments. The I4.0 goal for interconnectedness is the main driver for CPSs and MCPSs, which utilises various technique to integrate data across various business layers. There is a strong argument for cyber representation of each system component through heterarchical or hierarchical structures to promote interconnectedness. Developments in HCAs, along with MASs, have shown effectiveness to of decentralised control in HMSs by ensuring system components retain their autonomy and are cooperative.

The review of HCPS applications generally expresses the need for each system component, including humans, to be modelled and elevated to the cyber layer. This is needed not only to ensure interconnectivity, but to enable cooperation between all system components. No applications were found that significantly elevated the human’s cooperability and intelligence to the cyber layer. In holonic systems, HCAs often do not consider humans and only have vague descriptions of their roles. HCPSs in healthcare display even less consideration for humans in assisted MCPS and remains focused on only modelling the patient virtually. MCPS applications remain focused on robotics, body sensor networks and electronic health records.

The BASE architecture was initially developed to aid human integration. Applications of the BASE architecture have since aimed to improve the architecture to address I4.0 needs such as scalability and robustness. The BASE architecture has not yet been applied to complex operational environments that are heavily dependent on many human actors to perform tasks. The potential of the BASE architecture to overcome HSI challenges motivates its selection as implementation architecture for this study.

A more thorough definition of HSI and related design principles are required to realise HCPS which enhance human intelligence, operational roles and cooperation. Furthermore, a technical framework is required to map out the various developmental aspects that must ensure that humans are integrated effectively into HCPS.

3 Human-System Integration

This chapter defines the HSI terminology used in this thesis and presents the goals and requirements associated with HSI. Furthermore, it presents the Representation-Communication-Interfacing (RCI) framework for the analysis and improvement of the maturity level of HSI developments in the context of HCPSs. After introducing these concepts, the chapter concludes with a discussion of the interrelationships between the concepts and how the result can be used in developing HCPSs.

3.1 Definition

HSI is often described as a system design and development method for improving the consideration of *human factors* and *human-centred design* principles (Boy, 2020). HSI research has often remained ill-defined and varies in definition depending on the domain and application.

This thesis specifically focuses on HSI within the context of technological developments, on the component and integration layer, involving software and hardware components, and humans to enhance operational activities in CPS environments (i.e. HCPSs). HCPSs include physical systems (e.g. machines, equipment, materials, humans) and virtual systems (e.g. databases, software services etc.). In this thesis, HSI does not consider the influence of humans and technology on social behaviour and organisational structures.

3.2 Goals

Within operational systems, *entities* encompass any autonomous component that performs an actor role in the overall system (e.g. humans and machines). The maturity of HSI developments is considered by the level of detail of an entity's *knowledge* of itself and its surrounding environment, and an entity's understanding of its *interactions*, both over time (i.e. past, present, future). Knowledge and interaction are two fundamental aspects to promote enhanced intelligence for entity. Furthermore, HSI develops should enhance the intelligence of humans and their cooperation with other system entities. These goals can be achieved by ensuring that all system *entities* are reflected in the cyber layer in terms of CPSs, where all entities can communicate and exchange information. Information exchange in the cyber layer promotes information integrity, quality and traceability, and improves the rate of information exchange.

3.2.1 Knowledge

An entity must have knowledge of its own goals, properties and functionality. This entity should know its past, current and future state, and individual attributes. Similarly, an entity should know its surrounding entities. Varying levels of knowledge affect the integration and cooperation of entities. Therefore, humans need to be better integrated into HCPS to know their past, current and future states, while having similar knowledge of other humans and systems.

3.2.2 Interaction

An entity should understand its interactions (during operational processes) and the interactions of surrounding entities over time. There are two forms of interaction: coordination and cooperation. Coordination implies interactions involved when one entity directs other subordinate entities to achieve a goal. While cooperation implies interactions between entities which aim to achieve a united goal without hindering one another. Occasionally these terms are used interchangeably to describe interactions between multiple entities during processes.

Communication is an essential mechanism for coordination and cooperation within any interaction. Without communication, there can be no exchange or transfer of information to establish the relevant interactions successfully. Communication can exist through human-human interactions, human-machine interactions and machine-machine interactions.

Machine-machine communication has been well developed in the I4.0 domain, with standardised communication protocols, IoT infrastructure and ontologies, and exists predominantly in the cyber layer (i.e. within software and over networks). Human-human communication occurs through physical mediums, such as verbal communication, documentation and signboards, and digital mechanisms, such as WhatsApp or email. Human-machine interaction (HMI) has also emerged as a well-developed research domain, with many research efforts focusing on the design of user interfaces (UIs) to improve the interaction. These various communication mechanisms enable interactions between entities to coordinate and cooperate on activities.

3.2.3 Enhanced Intelligence

HSI developments should aim to support the intelligent characteristics of humans during activities. The characteristics of intelligent agents from Romero, *et al.* (2016) have been adapted with a human-centric perspective and are presented below with examples:

- **Purposeful:** A human has goal-seeking behaviour.
  - Manufacturing example: An artisan aiming to lathe a rod to a specific dimension.
  - Healthcare example: A doctor aims to examine all waiting patients during their session.
- **Perceptive:** A human can observe information about their environment and the system and filter the information according to that which is relevant.
  - Manufacturing example: An artisan looking through different physical drawing packs to find the relevant drawing for the rod which they need to lathe.
  - Healthcare example: A doctor accessing the healthcare database for a specific patient’s x-ray images.

- **Aware:** A human has situational awareness, relevant to their purpose.
    - Manufacturing example: An artisan can observe that all lathe machines are in use on the factory floor and thus will not be able to lathe a rod at this current moment.
    - Healthcare example: A doctor might notice that the patient queue is growing and thus realises they will not be able to see all the patients at the current rate.
- **Autonomous:** Humans can decide which activities and resources will be required to meet their goals.
    - Manufacturing example: An artisan decides that a lathe should be used to machine the specified surface finish to the rod.
    - Healthcare example: A doctor determines that a patient should have an x-ray scan.
- **Able to act:** Humans must be able to mobilise resources (physically or virtually) for activities, information gathering or cognition.
    - Manufacturing example: An artisan uses an available lathe to machine a rod.
    - Healthcare example: A doctor directs the patient to a local radiology department for an x-ray scan.
- **Reflective:** Humans must present their goals, properties and reasons to surrounding entities within their system boundary.
    - Manufacturing example: An artisan can inform their supervisor of their current progress for manufacturing the rod using the lathe.
    - Healthcare example: A doctor should inform other HPs that he will be unavailable to assist other patients until he finishes consulting the current patient.
- **Adaptability and learning:** Humans can modify activity processes and plans, or change previously defined goals when they encounter unforeseen events.
    - Manufacturing example: An artisan decides to use another machining method once he discovers the lathe cannot produce the specified tolerance.
    - Healthcare example: A doctor selects a different medication treatment plan when realising the patient is not recovering as expected.
- **Conversational and cooperative:** Humans can negotiate with other agents to formulate coordinated and cooperative goals.
    - Manufacturing example: An artisan asking another worker to help clean the lathe after its use.
    - Healthcare example: A doctor asking a nurse to monitor the patient using a heart rate machine.

It is apparent that humans inherently possess these characteristics to varying extents. Therefore, the goal of HSI developments is to improve the above-mentioned intelligence of humans and systems within HCPS, which rely on knowledge of each system element and their interactions.

3.2.4 Enhanced Cooperation

HSI developments should enhance the ability of humans to cooperate with other humans or systems in the HCPS environment. According to Pacaux-Lemoine *et al.* (2017), cooperating agents require the internal and external ability to *know-how-to-cooperate*.

Internal ability: Entities (humans and machines) must be able to understand the *model* (properties and behaviours) of other systems or humans in their environment function, especially those that they might interact with. Furthermore, each entity should be able to deduce from available information what other entities' *intentions* will be – where *intention* describes an entity's future state (planned or not). Entities should also be able to analyse tasks given to them or activities they are performing within the cooperative system, and then be able to make decisions (plan) and *coordinate* with other entities to achieve tasks.

External ability: Communication is a key mechanism for an entity to understand other entities. Each entity should be able to *exchange information* bi-directionally with other entities and should be able to access information regarding other entities.

3.3 Framework for development and evaluation

This section presents a technical classification of HSI developments in an operational HCPS environment, to better understand the requirements and evaluation for such developments to achieve the previously highlighted goals of HSI. The presented framework considers three dimensions: *representation*, *communication* and *interfacing*, as seen in Figure 10.

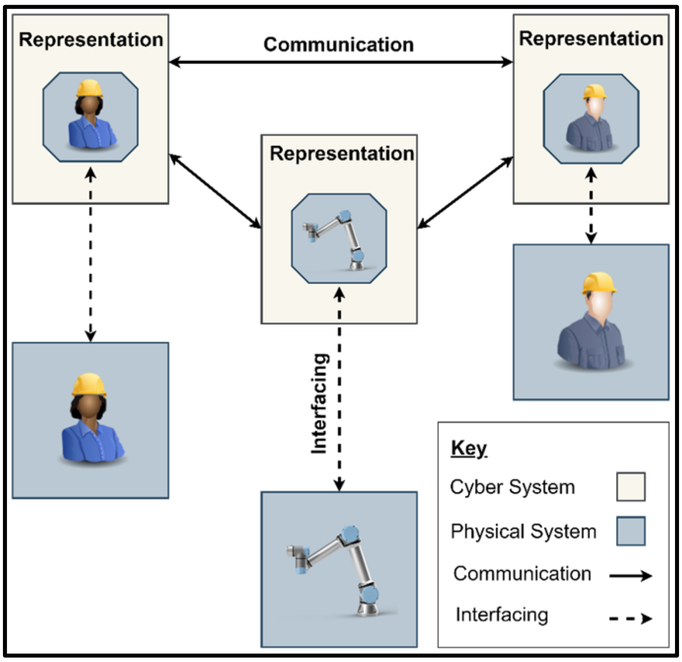


Figure 10: Diagram of the RCI framework.

3.3.1 Representation

Elevating each physical entity to the cyber layer supports the development of entity models and improves the accessibility, transparency and interoperability of information between these models. In part, this has led to the development of concepts such as DT and HDT. *Representation* should reflect and encapsulate the properties and behaviours of the physical entity truthfully and accurately, as described by Valckenaers & Van Brussel (2015). Methods of encapsulation, such as object-orientated principles in software design or MAS principles, should be employed to ensure *modularity* and support autonomy. These representations should be *reconfigurable*, allowing properties and behaviours to be added or removed as required.

3.3.2 Interfacing

Any connection which enables the transfer of data between a physical entity and its virtual representation is described by *interfacing*. This *interfacing* may include the use of technologies such as information displays, input capturing devices or sensors positioned within the physical environment. These *interfacing* technologies can be classified as either environmental or personal interfacing devices, according to Sparrow, Taylor, Kruger, *et al.* (2021). Personal interfacing devices are used by an individual at a given time, such as wearables, tablets or headset devices. Environmental interfacing devices are positioned in the physical environment and can be used by various humans or systems within the bounds of that environment. Environmental devices could include technologies such as displays, cameras or scanners.

3.3.3 Communication

Communication is a broad term encompassing information exchange between entities, such as machine to machine, machine to human and human to human entities. Within this framework, *communication* is limited to the function of data/information exchange between representations that exist within the cyber layer of the HCPS. Communication, therefore, does not relate to exchanges between system elements within the physical layer. Communication is important for enabling cooperability between different entities, since each entity must gather information regarding the models and intentions of other entities for decision making. This improves the traceability and control of interactions between entities.

3.4 Discussion

3.4.1 Framework Design

The RCI framework was developed considering the architectures and concepts presented in HMS, HCPS and HSI research. Key concepts which contributed to the design of the framework were the I4.0 component model (Zezulka *et al.*, 2016), the internal structure of a resource holon considered in the ADACOR architecture

(Leitão & Restivo, 2006), the BASE architecture (Sparrow, Kruger & Basson, 2021) and the ARTI architecture (Valckenaers, 2019).

The HCPS concepts describe the type of operational environment in which to study HSI. HCPSs require the consideration of both humans and systems, which must coordinate their activities to meet the operational goals. This led to the understanding that both system elements and humans are actors in the operational activities and, as such, should be represented accordingly in the cyber layer. Due to the lack of representation of humans in the cyber layer, the RCI framework aims to improve the intelligent characteristics of humans, as is the focus of HSI.

HDT and HMS concepts and applications emphasise the value of the virtual encapsulation of the functional properties and behaviours of different physical components in the HCPS. Within HMS, HCAs such as ARTI guide the partitioning of the system into activity and resource holons, which encapsulate the related system functionality. Additionally, the resource holon in the ADACOR architecture provides insight into how the virtual encapsulation of the physical components could be structured. Similarly, RAMI 4.0 and the I4.0 component model presented a generic administration shell to virtually represent each physical component in the system. Furthermore, DT research aims to better the virtual representation of physical components for decision making and control. Similarly, HDT research focuses specifically on the application of DT architectures for humans to model their behaviour and properties.

The resource holon structure in ADACOR, HDT architectures and the I4.0 component model highlighted the importance of mechanisms for virtual representations to exchange information with their physical counterparts – which led to the concept of *interfacing*. ADACOR's "physical interface component" of the logic control device (LCD) acts as the mechanism for data transfer. This was required for all LCDs and allowed resource holons to communicate in a standardised manner with other devices. The I4.0 component model is shown to "attach" to its physical counterpart; although only a limited explanation is provided on how this could be done. DT and HDT architectures also present a link between the virtual and physical components to ensure there is an accurate reflection of reality. These concepts and architectures highlight the noticeable link between physical and virtual, and support the need for the interfacing dimension of the RCI framework.

HMS architectures and the I4.0 component model argue for the value of enabling communication and the dynamic interactions between virtual representations to facilitate coordination. To ensure interoperability, these communications need to be standardised so that all representations can understand and participate in the communication. The elevation of communication to the cyber layer promises improvement in traceability, latency and data integrability.

3.4.2 Relevance to Human-System Integration Goals

The RCI framework guides the attempts to elevate human and non-human entities, and their cooperative ability, to the cyber layer of the HCPS. In doing so, the knowledge of each entity and the knowledge to cooperate is more transferable and better understood throughout the system. Each entity's state and behaviour are modelled by its representation, which can also understand the models of other entities to ensure cooperability. The selected interfacing technologies will affect the data accuracy and latency of these representations. Communication influences the external ability of each represented entity to cooperate. Standardising communication will result in improved information exchange between models.

In considering the goals of HSI, the ability of an entity to reflect its state and behaviour is supported by the interfacing and representation dimensions of the RCI framework. As better interfacing mechanisms are selected, the reflection capability becomes more accurate and seamless. Through representation, the autonomy of each entity is maintained in the HCPS, ensuring all related functionality is encapsulated in one cyber entity. This creates a separation of concerns between representations, and thus between physical resources. The communication between representations allows a physical resource to act by utilising other representations; as such, resources can perform activities. Physical resources will also have an improved awareness of their surroundings when there are effective methods of communication between representations and interfacing between the resource and its representation. Representations can gather and filter information according to the purpose of the physical resource. Representation also allows for learning and adaptability for physical resources, by processing information gained from other representations and aiding decision making.

3.4.3 Evaluation of HSI Maturity

The RCI framework provides a tool to evaluate the maturity of HSI developments in an operational HCPS, which can aid the design of iterative HSI improvements. Various operational HCPS environments (e.g. manufacturing, agriculture, healthcare, etc.) may have different maturity levels of HSI technological development, as described in Schumacher, Erol & Sihn (2016). Using the RCI framework, the maturity level of each aspect can be plotted on a coordinate system as shown in Figure 11, having three dimensions: representation, communication and interfacing.

Although no standardised quantitative metric is available to measure each dimension, a more qualitative and subjective approach can be applied to visualising the maturity level. The subjective approach requires intuition and sound judgement to be applied to the evaluation of the maturity level of each dimension. This subjective evaluation is typically easier to apply comparatively to two scenarios with different HSI maturity.

The aim of HSI developments is to ensure that the overall solution results in a holistic HSI improvement. A holistic HSI improvement approach should aim to balance the maturity level of all three aspects of the RCI framework. The overall benefit and maturity of the HSI developments will be limited by the lowest "scoring" aspect for an imbalanced approach (i.e. representation in Figure 11).

As an example, if representation remains immature compared to communication and interfacing, then the HSI goals will be hindered by the representation dimension. Having many sensors and data acquisition devices attached to a system component but failing to aggregate and model the data from the machine, will reduce the ability of the machine to reflect itself, adapt and learn, and cooperate with other system components. Furthermore, other system components will have a reduced awareness of the machine. This example shows how a deficiency in one or more RCI dimension can hinder the overall HSI maturity of the system.

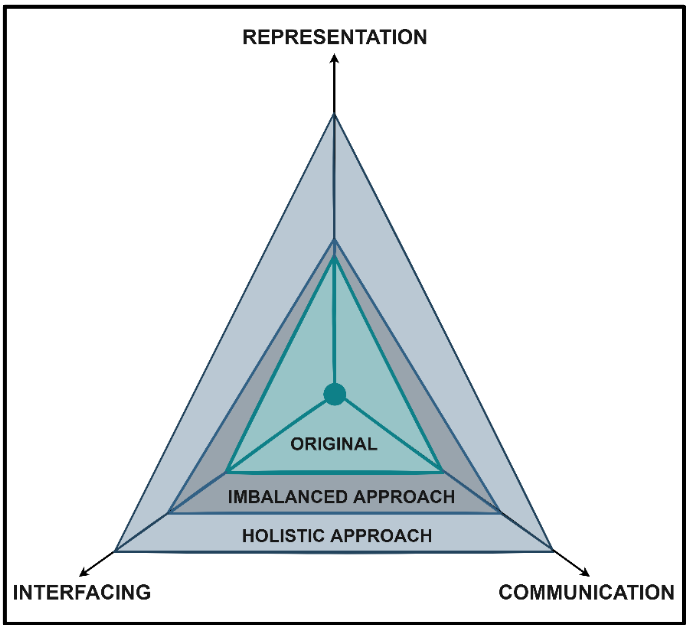


Figure 11: RCI maturity level coordinate system.

Knowing which aspects of the RCI framework are deficient in the system, will allow the development team to tailor efforts to improving those areas. Specific technologies and concepts can then be employed to overcome the barriers related to specific aspects to realise a holistic HSI solution.

3.4.4 Realising Holistic HSI Solutions

The RCI framework provides a tool for classifying the contribution of various technologies to HSI developments. A holistic HSI solution will require the application of various technologies and concepts. These technologies can be mapped relative to the RCI dimensions, as seen in Figure 12. The Venn diagram shows that some technologies could lie at the intersection of two RCI dimensions. The use of the RCI framework can aid system designers in overcoming shortfalls in their system designs. The RCI framework guides designers by ensuring all aspects of HSI are considered in a balanced manner.

In an attempt to develop an effective HCPS, the application of holonic concepts have great potential to meet the requirements and realise the goals of HSI. HCAs support the representation and communication requirements of HSI developments by enabling autonomy and cooperability of system elements. HCAs were developed to support decentralised control systems for manufacturing processes. HMSs, which employ HCAs, are composed of many holons which have autonomy and can cooperate. Distinguishing the autonomy of each holon, the system ensures that each system element is represented appropriately in the cyber layer. While allowing comparability between agents, the system ensures that each system element can communicate with other holons, in the cyber layer, toward individual and collective goals.

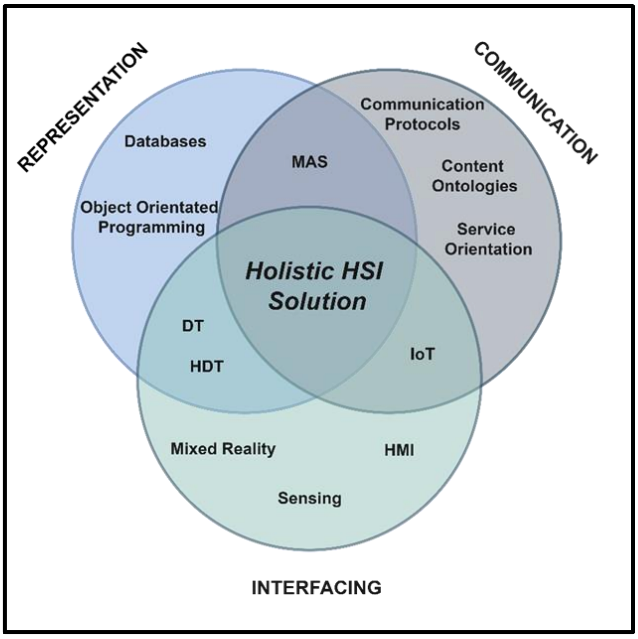


Figure 12: Venn diagram mapping current technology to the RCI framework.

One such HCA is the ARTI architecture, an extension of PROSA, which applies holonic principles by partitioning the function of system elements and processes between activity and resource holons, and types and instances. ARTI allows for cooperability between different holons. Additionally, the BASE architecture design was guided by holonic principles and provides an architecture to design the internal structure of holons such that they represent their physical counterparts. Reflecting the biography, attribute, schedule and execution data of each holon in the cyber layer enables a holistic representation of each holon. Therefore, the ARTI and

BASE architectures will guide the development of a holonic system which supports the requirements of HSI.

4 Design Process for Holonic Systems

This chapter details the formulation of a holonic system design process used to develop the HSI software for the case study discussed in Chapter 6. This design process includes the use of holonic software architectures – specifically, the ARTI and BASE architectures.

4.1 Purpose

HCAs such as ARTI (Valckenaers, 2019), ADACOR (Leitão & Restivo, 2006), and PROSA (Van Brussel *et al.*, 1998) were developed to support HMS. An incremental HMS engineering process, seen in Figure 13, was proposed by Christensen (1994) in the early stages of the developing HMS paradigm. This engineering process aimed to support incremental HMS developments to existing operational processes as the opportunity arose.

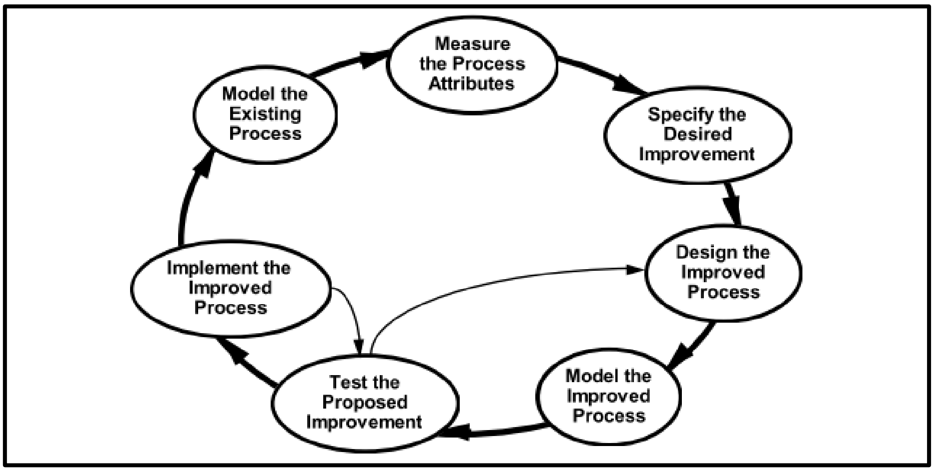


Figure 13: Incremental HMS engineering process (Christensen, 1994).

Although HCAs are well documented conceptually, they often lack the implementation details to support the “Design the Improved Process” stage of the HMS engineering process. Therefore, a gap exists between the conceptual understanding of HCAs and the method by which to design and apply them. The design process presented in this chapter aims to refine this stage of the process.

While the presented design process is expected to be compatible with different holonic architectures, this thesis specifically considers the ARTI and BASE architectures. The ARTI architecture is a generic extension of PROSA to support its application in non-manufacturing domains. ARTI was presented only conceptually in Valckenaers (2019), with little discussion on its application. Since then, ARTI applications have been published, but still lack in design approach details. Similarly, the BASE architecture has been applied to manufacturing, maritime and agricultural domains, but the resulting publications do not offer much guidance on how to conduct a software design process either.

There are multiple benefits to documenting the design process for the implementation of software architectures. The process can guide future implementations, which would reduce the time and barriers to development. Furthermore, feedback from researchers applying these design processes would result in iterations to streamline and improve the design process.

4.2 Overview

The overall design process is described using an *input-process-output* flow diagram. Each process step requires certain input information (i.e. list of requirements or descriptions), which must be considered for the following process steps, which then output new information. Iteration loops exist throughout the design process to cater for design refinements and additions/eliminations. The overall design process is presented in an upper and lower abstraction layer to aid the discussion.

The upper abstraction level of the design process is presented in Figure 14. The process begins with the input of the user requirements for the desired holonic system. User requirements may be presented through verbal or written communication of a qualitative (e.g. “improve throughput”) or quantitative (e.g. latency below 200 *ms*) nature.

Step 1 encapsulates the “Model existing process”, “Measure Process Attributes” and “Specify the Desired Improved” steps from the HMS engineering process in Figure 13. Step 1 entails capturing information – whether real, such as the physical layout of the facility, or conceptual, such as designs and plans. The information can be captured through various mechanisms (e.g. video recordings, interviews, testing, simulations and CAD designs).

The information captured in step 1 will serve as basis for the development of non-functional and functional requirements for the system. Functional requirements are specifications that influence the design of system components, while non-functional requirements describe the quality attributes of the system.

During step 2, the various requirements must be considered to develop a flow diagram to map out the various activities of, and relations between, machines and/or workers. The flow diagram should indicate any:

- Workers involved
- Machines/equipment involved
- Tasks/processes to be performed (by one or more machines/workers)
- Interactions between machines/workers

The level of detail for this process diagram will be dependent on the complexity of activities and interactions within the operational environment. Once developed, process descriptions should be written that support the process diagram and provide enough information to formulate a holistic understanding of the processes and their respective interactions.

These process descriptions are used in step 3, which considers the ARTI architecture to identify and develop activity and resource holon types and instances. During this step, additional activities and resources might be realised, resulting in changes to the original process diagram and respective process descriptions.

Within step 4, the BASE architecture provides a means of structuring the internal and external holon interactions and functions and provides an implementation platform for the previous developments. The output of this process should result in well-defined holons, each with specific BASE plugins to develop, state variables and attributes.

Steps 3 and 4 have a strong coupling because changes in either have a strong influence on the other. As such, an iterative approach should be taken to refine the developed system. After these steps, changes to the conceptual operational environment might warrant an iteration of the design process to align more closely with the user requirements.

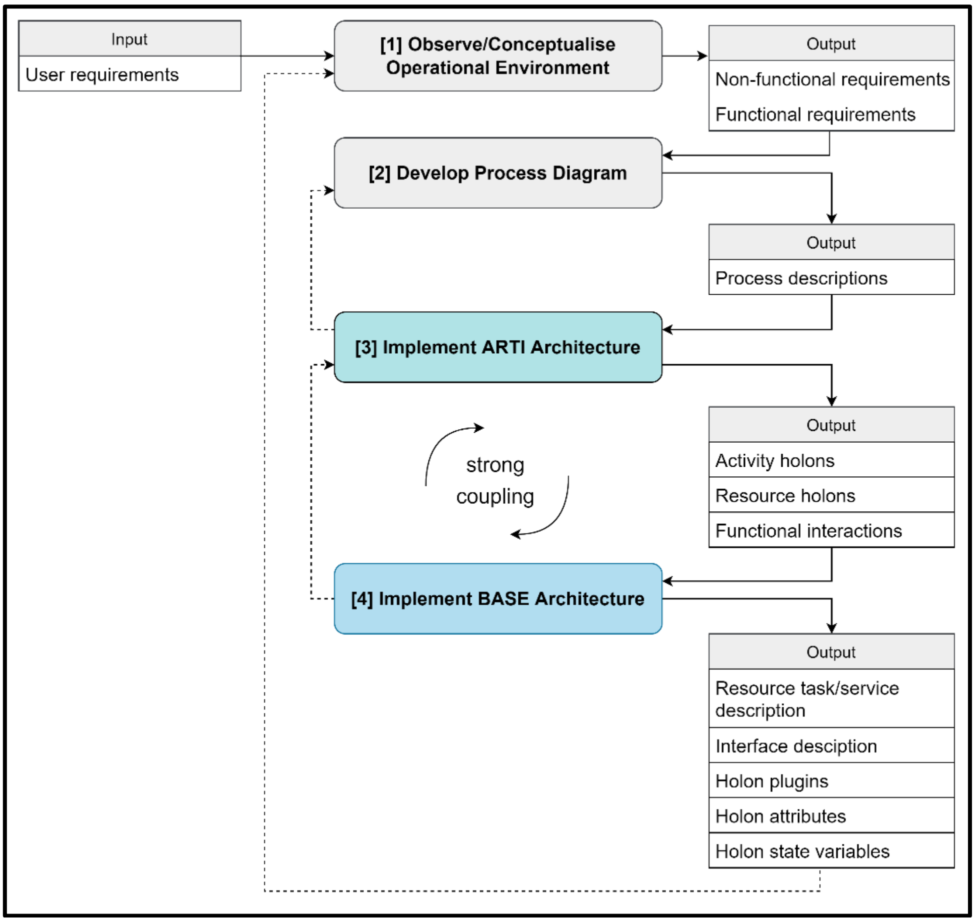


Figure 14: Upper abstraction flow diagram layer of the holonic system design process.

4.3 ARTI Architecture Design Process

The ARTI architecture guides the partitioning of the process descriptions into activity and resource types and instances. Valckenaers (2019) presented the ARTI architecture conceptually and vaguely, to create an architecture that could be adopted for a wide range of applications and domains. The application of the ARTI architecture requires clarity on three fundamental distinctions:

1. How does the distinction between *Agent* and *Being* influence the design of the system?
2. How should functionality be distributed between *Activity* and *Resource* holons?
3. How should functionality be distributed between *Type* and *Instance* holons?

The first distinction within this design process is not important to specify beyond what Valckenaers (2019) has already presented. The important design principle noted by this distinction is that functional elements of the holon should be generalised as best possible and de-coupled from case-specific implementation functionality.

The second distinction is arguably the easier distinction to make and is explained in Valckenaers (2019). An activity holon has the knowledge and ability to control a specific process, requiring coordination with other activities and resources. Resource holons should correspond to physical or software components that provide a service in the operational process. Initially, in HMSs, humans were considered as resources (Christensen, 1994), rather than the vaguely-defined *activity performers* suggested by Valckenaers (2019). The higher intelligence of humans compared to machines does support this notion of activity performers, representing entities which have cognitive intelligence. However, classifying humans as resources holons does not present any clear limitation.

The third distinction is between a type and an instance. A type holon should coordinate the creation, maintenance, termination and disposal of instance holons. Any functionality and control optimisation requiring the consideration of other instances' states and attributes should be contained within the type holon. Essentially, the type holon acts as an instance manager and container for aggregated information from all the instances. Instance holons contain the control logic and functionality to coordinate other activities or resources for the specific purpose it was created.

The lower abstraction layer flow diagram of step 3 is presented in Figure 15. The application of the ARTI architecture begins with step 3.1 – identifying the activity holons and the related processes they would need to control in the holonic system. The process descriptions, diagram and requirements must be partitioned into activity holons, which should encompass the various functions required by the respective processes. The activity type holon, like a product holon in the PROSA architecture, must encapsulate the process steps and logic. The activity instance should encapsulate the functionality related to the control of coordinating other holons for a single process instance. All the processes identified in step 2 must be encapsulated within the identified activity holons.

In step 3.2, resource holons must be identified from the process descriptions and flow diagram. Resources are various actors in the system including machines, humans and software components that can perform tasks to aid the processes in the system. Each resource holon should encapsulate the functionality of the physical or software resource, which would be utilised during the system's operations. As an example, these functions might entail pick and place actions that a robot resource can perform.

An activity holon must be "free to search and utilise suitable resources in an appropriate sequence", while a resource holon should be able "to service any activity that may desire it"(Valckenaers, 2019). Step 3.3 deals with this design requirement for the ARTI architecture. Step 3.3 requires that the functional interactions between holons be established and described. These functional interactions might result in different types of relationships, some being:

- Nested: A nested holon is created and dies during the lifetime of the primary holon.
- Parent-child: One holon is a child to only one parent holon. The child holon is born (created) within the parent's lifetime but may live beyond the parent holon.

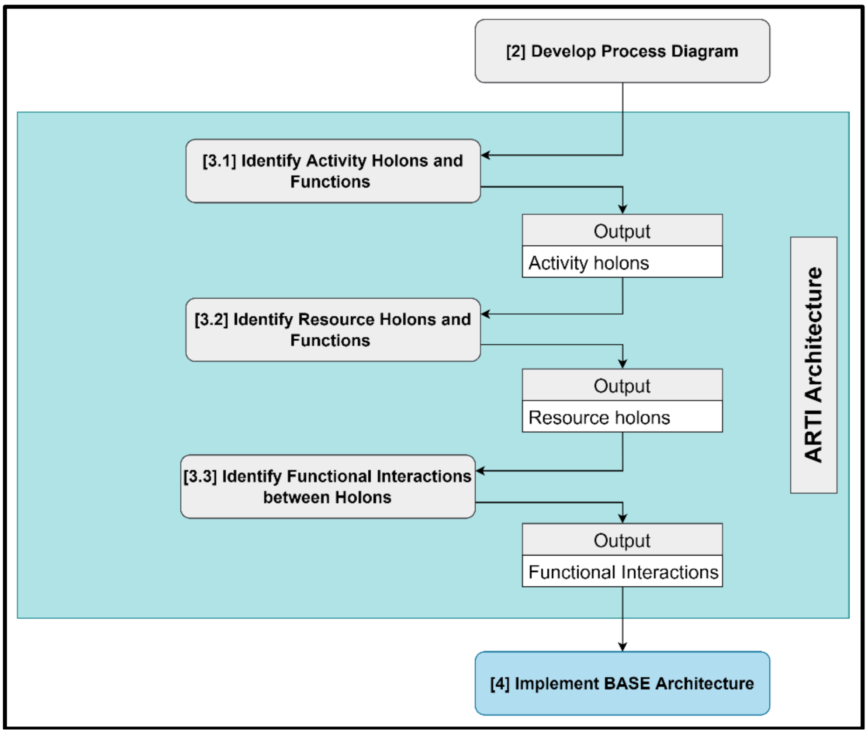


Figure 15: ARTI architecture design process flow diagram.

4.4 BASE Architecture Design Process

The ARTI architecture aids in identifying the holons required in the system, whereas the BASE architecture provides structure to the internal and external functionality of the holons. The implementation design process flow diagram is shown in Figure 16. The BASE architecture describes four different types of plugins for each holon, namely: scheduling, execution, reflection and analysis plugins. The BASE architecture guides the development of the ARTI holons to well defined BASE instances.

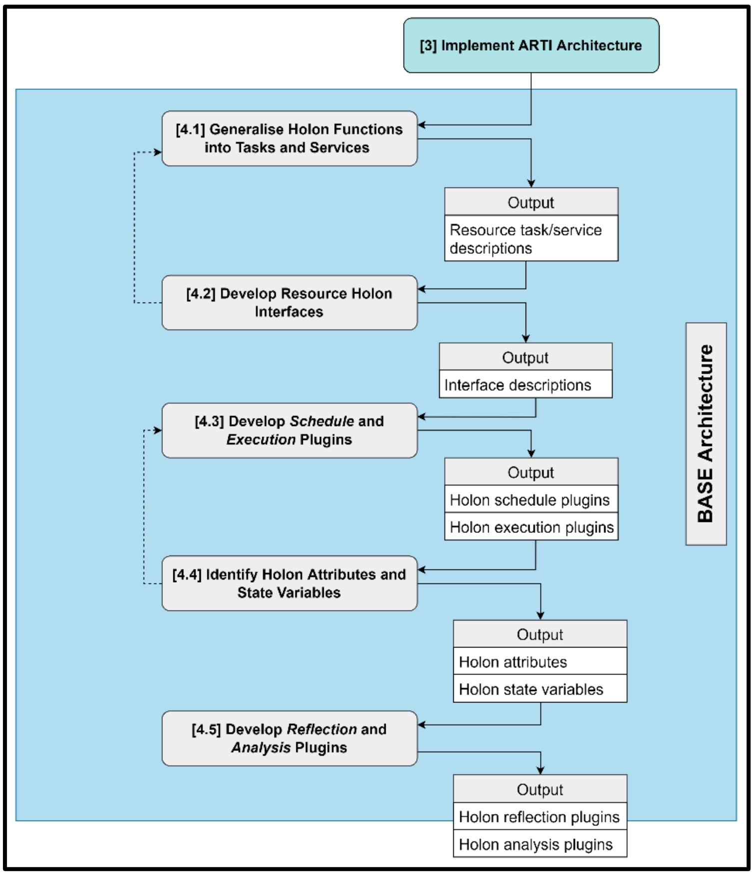


Figure 16: BASE architecture design process flow diagram.

There is a distinction between *services* and *tasks* within the BASE architecture. *Services* describe any functional interactions between two or more holons. Services normally require negotiation of some sort between the client and the service provider holons. A *task* is a function or process internal to the holon for its management, execution and analysis. Tasks are not scheduled through negotiation or some contract, but rather through internal triggers within a holon (e.g. a periodic time-out to trigger the backup of internal data). The holon functions and their functional interactions, identified in step 3.3, must be generalised into services and tasks in step 4.1. The classification between services and tasks will affect the development of plugins related to each holon.

In step 4.2, the different interfaces (or mechanisms) of data transfer between the resource holons and their respective physical counterparts must be developed. This step is motivated by the RCI framework presented in Section 3.3. An example of an interfacing mechanism might be a web user interface to allow a worker to provide and read information to/from their associated resource holon. When developing interfaces, further tasks and services might be identified, which will require an iteration of step 4.1. After this iteration, a list of interface descriptions for respective resource holons must be presented.

The first set of plugins that should be developed are the scheduling and execution plugins for each BASE holon, in step 4.3. Scheduling plugins for tasks must encapsulate the functionality and decision-making for determining the appropriate time to begin the task. The execution plugin for tasks must encapsulate the functionality and logic related to determining whether the tasks should begin at the scheduled time and how to perform specific tasks.

The scheduling plugins for service providers must handle all negotiations with (potential) clients and handle the outcomes of such negotiations (e.g. booking the services, rejecting or accepting offers). Similarly, scheduling plugins for clients should encapsulate the functionality to initiate negotiations with service providers and the decision making for selecting providers based on proposals. The execution plugins for both service providers and clients should encapsulate the decision-making functions for beginning, executing and terminating a specific activity.

BASE instance plugins might require the consideration of different holon attributes and state variables for decision making. Therefore, the required holon attributes and state variables must be identified in step 4.4. According to Valckenaers & Van Brussel (2015), the ideal holonic software system should best reflect the reality of its physical counterpart. However, such a venture might result in redundant data and be limited by computational resources. Step 4.4 attempts to mitigate that problem by ensuring that attributes and variables are only selected in considering the plugin functionality which requires them.

Lastly, the reflection and analysis plugins are developed in step 4.5. These plugins can be developed for both tasks and services, and should encapsulate the functionality and logic related to deciding which information to store for the executed service or task in the biography of the holon. This development of the reflection plugins is strongly coupled with the development of the analysis plugins, which should encapsulate the functionality related to the analysis of historic data of the holon to update the holon attributes.

After the BASE architecture design process, each BASE instance holon should have well defined interfacing mechanisms, plugins, state variables and attributes. These clear definitions will ease the development complexity for implementing BASE holon due to well established functional requirements for each plugin and their interaction with the core components.

5 Human-System Integration in Healthcare

This chapter explains the opportunities for better HSI in healthcare, decision-making behind selecting a healthcare environment, and thereby motivates it as context for HSI research. Additionally, this chapter provides an overview of ambulatory care workflows, information management and administrative logistics, which are used to formulate the needs analysis for the holonic HCPS developments. Lastly, the selection criteria for the case study, along with the description and requirements of the case study, are discussed.

5.1 Ambulatory Patient Care

5.1.1 Overview

Ambulatory clinics, also known as *outpatient* clinics, are healthcare facilities that provide medical care services to patients during the day, but do not offer overnight hospitalisation. These facilities contrast with *inpatient* care, where patients typically stay one or more nights at a healthcare facility to be monitored or receive extended treatment. Ambulatory patient care services can be performed at a variety of facilities, including private consultation rooms, laboratory testing facilities and hospitals.

Tygerberg Hospital in South Africa provides both outpatient and inpatient care services and is an academic hospital for Stellenbosch University. Tygerberg Hospital is managed by the provincial Western Cape Department of Health and is the second-largest hospital in South Africa (Tygerberg Hospital, n.d.). Information released by the Western Cape Department of Health in 2016 stated that 492 670 outpatients visit Tygerberg Hospital annually, while 107 215 inpatients are admitted annually (Tygerberg Hospital, n.d.). Tygerberg hospital is a tertiary healthcare institution, where patients must be referred by appointment to receive care. There are a vast array of ambulatory clinics and departments within Tygerberg hospital, where these clinics specialise in the care of different conditions and diseases.

South Africa’s healthcare system has been affected by diseases such as HIV and AIDS, tuberculosis and non-communicable diseases, such as chronic respiratory conditions (Maphumulo & Bhengu, 2019). The respiratory ambulatory clinic at Tygerberg Hospital mostly treats outpatients who suffer from post-tuberculosis side effects, asthma, and other respiratory-related conditions. As such, the respiratory ambulatory clinic is an important clinic to study for workflow improvements, which might be realised through HSI developments.

Ambulatory patient care at the respiratory clinic begins with an analysis of patient *history* and then an *examination*, requesting and reviewing *investigation* results and performing an *intervention*, as seen in Figure 17. This workflow is apparent in many other ambulatory clinics at Tygerberg Hospital and other healthcare facilities.

The ambulatory care service begins with the arrival of the patient at the healthcare facility and ends with the patient departure. Patients must be referred to tertiary healthcare institutions either by a primary or secondary care facility, an internal referral or a scheduled visit. Upon arrival, the patient must report to the hospital administration staff to receive their patient file, which they will carry round during the day between clinics and investigation facilities.

Patients who intend on visiting the respiratory clinic must first perform a lung function test after registering at the hospital administration and before arriving at the clinic reception. This investigation is performed by an aspiratory technical team within the hospital. Once complete, the patient may then present themselves to the clerk at the respiratory clinic and wait for an available doctor to begin the consultation.

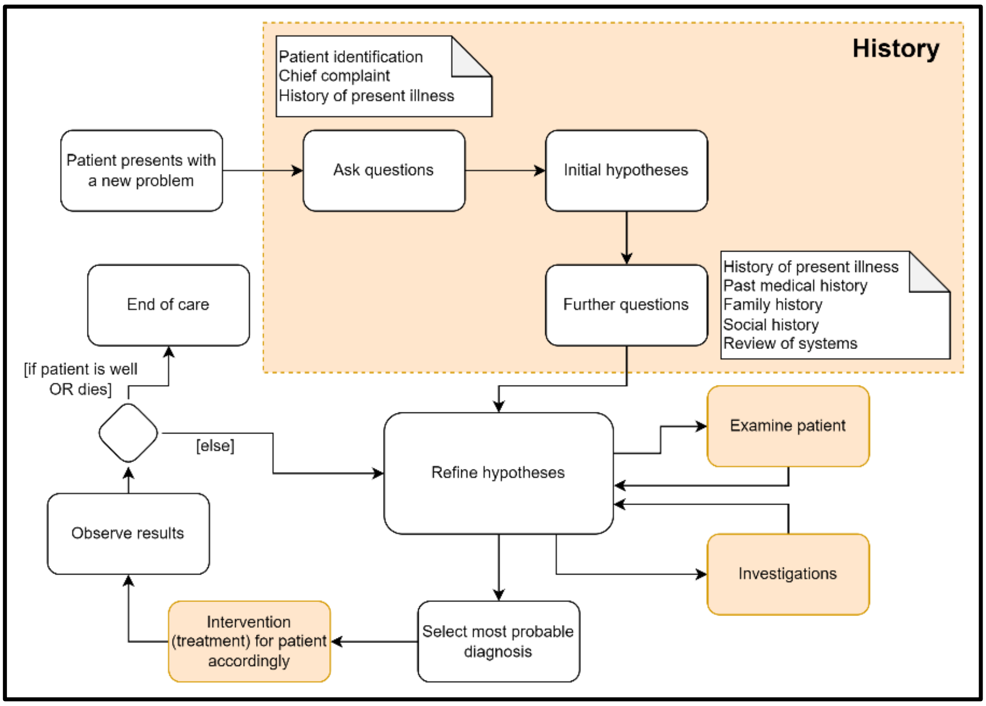


Figure 17: Overview of ambulatory care consultation adapted from (Shortliffe & Cimino, 2014).

The patient waits to be called in, by queuing outside the clinic. Once called in, the doctor takes a history of the patient. The doctors query and use an array of databases for historical patient information and ask the patient detailed questions to gain a broader understanding of the patient's health-related context. After sufficient history information is captured, the doctor proceeds to examine the patient and relevant information (i.e. patient data, medical history, current patient readings, etc.). The doctor may decide to coordinate a further investigation of the patient, which could be performed through various tests depending on the doctor's observations and investigation. These investigations and tests typically require the use of machines to analyse the condition of the patient. These machines provide health-related information and may require a caregiver to record this information in the patient's file. Once all investigations have been completed and the results are available, the doctor will analyse the examination and investigation information and decide on an appropriate intervention (or care program). The intervention may entail prescribed medication or surgery. A patient may need to return to the hospital at future appointments to monitor the effectiveness of the intervention.

5.1.2 Information Systems

HPs and staff (e.g. hospital administration, doctors, nurses, etc.) make use of various data sources and storage whilst providing ambulatory care services, represented by the physical data flow diagram (DFD) in Figure 18. Most of the data handling is done with physical repositories of documents containing a record of patient, examination, investigation and diagnosis information.

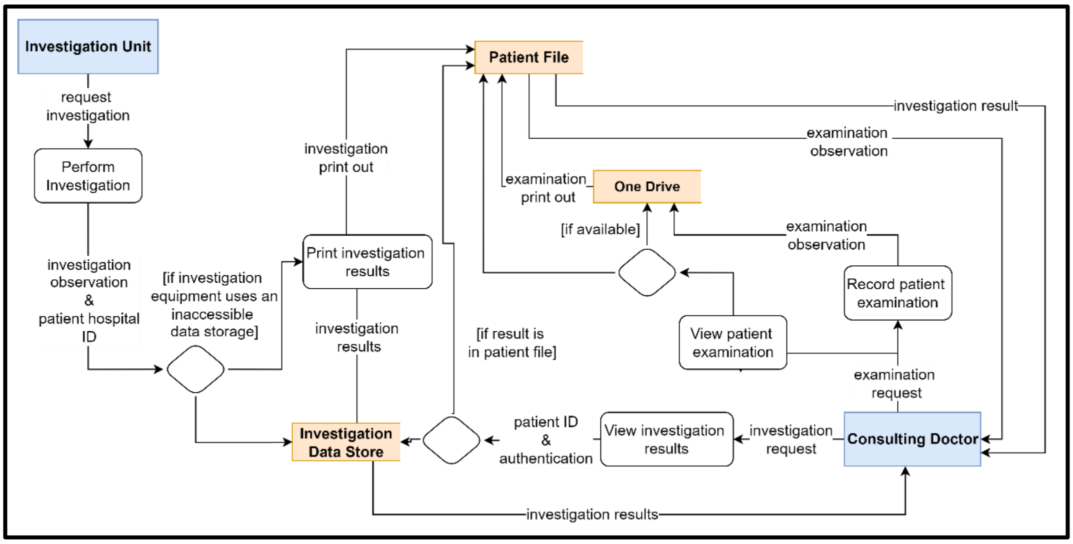


Figure 18: Outpatient consultation physical DFD.

Upon arrival, the patient receives a physical patient folder, either existing or new, in which they transport all physical documents (e.g. investigation test results, clinical notes, prescription notes, etc. During a consultation, an HP may view and record observations made during an examination onto a cloud repository, which may also be physically printed and inserted into the patient file. The HP may also view previous investigations, depending on the accessibility of the data, through a website or as a printout in the patient file. Certain proprietary data sources limit the remote accessibility of patient data due to software license constraints. Various data sources exist in data *silos*, with limited accessibility and integration, and HPs are often required to log into various website portals to access investigation and examination results.

Table 2 presents some of the data stores used in the ambulatory clinic at Tygerberg Hospital. The data stores may be local physical or digital repositories or cloud-based digital repositories. Due to the sensitive nature of the data, the accessibility to these data stores is limited and requires strict verification processes. Companies that provide certain medical testing equipment often limit access to their databases entirely, making such data proprietary. This ensures that the clinic will continue to utilise its services, but it is detrimental to the workflow efficiencies of the clinic.

Table 2: Data stores for an ambulatory clinic.

Data Stores	**Data Description**
Enterprise Content Management (ECM)	Local digital repository of electronic documents
Patient File	Physical folder containing documents (clinical notes, investigation results, examination results, etc)
One Drive	Clinic’s cloud repository of information including clinical notes and patient test results
National Health Laboratory Service (NHLS)	Website interface to a national digital repository of patient laboratory results (e.g. blood tests, culture tests, etc.)
Picture archiving and communication system (PACS)	Website interface to a national digital repository of medical images from patient investigations.
Jaeger Database	Local digital repository of spirometry test results. This data is proprietary and inaccessible to patients and HPs.
Echo (echocardiography) PACS	Local proprietary database for cardiography scans.
HERMES (nuclear medicine)	Local proprietary database for nuclear medicine.
ECCR (records)	Provincial database for patient discharge, script records and EHR.
Clinicom	Tygerberg Hospital administration system for booking admissions, hospital number generation, and recording ICD-10 codes.

5.1.3 Administrative Logistics Challenges

ALs such as data handling, transport, formatting and storage is required for cognitive functions, including managing responsibility, training, advice, decision-making, analysis and activity execution (Sparrow, 2020). The efficiency and quality of ALs can be improved through automation and digitisation. ALs in Tygerberg Hospital are currently constrained by manual processes and physical data handling, transport, formatting and storage.

5.1.3.1 Data Handling

Data handling entails functions related to the capture, analysis and representation of data and information. In healthcare, the clinical notes of HPs are captured on paper. The legibility of hand-written doctor notes is a major concern to the quality of continued care. HPs often struggle to read recorded clinic observations, decision-making and intervention notes. Furthermore, there is a lack of data analysis due to physical information not being stored in a standardised and digitised format. Patient information often remains in the patient file, where doctors and medical researchers are unable to use such data to research the effectiveness of intervention decisions or to detect medical trends in population groups. Data from proprietary databases must often be printed and inserted into the patient folder. The benefit of this data is limited to HP's ability to observe and analyse the printed information, which can in some cases mean having to read through many pages.

Much data handling is also performed through verbal or written communication. Such as when HPs exchange patient information verbally to coordinate healthcare activities. The temporal, untraceable and unstructured nature of this information exchange can result in malpractice, inefficiencies and unethical data handling without proper privacy controls.

5.1.3.2 Data Transport

The hospital administrative staff, upon patient arrival, must retrieve existing patient files from a storage room. Patients transport most of their medical information via their patient folders, which they carry with them between various clinics. Clinic clerks might temporarily take possession of these folders and put them in a queue of other patient files – a form of scheduling management for their specific clinic. HPs might also take possession of the folder during a consultation to view and record investigation and examination results. Before the patient leaves the facility, the patient file must be returned to the hospital administration, who transport it to their staff who are responsible for scanning the physical documents. This folder can get lost, or information may be removed from it accidentally, which risks the quality and continuation of care. Many investigation and specialist care clinics duplicate patient folders for private future use, due to a lack of access to patient information.

5.1.3.3 Data Formatting

Data formatting entails the standards used to structure data. Currently, clinical notes are recorded onto paper with an ill-defined structure. Furthermore, data from investigation results are typically well-structured in independent data silos but lack standardisation to promote interoperability for data integration. There is a tension between rigid and flexible formatting requirements for clinical notes and electronic health records in computer-based documentation systems (Rosenbloom, Denny, Xu, *et al.*, 2011). In conclusion, Rosenbloom *et al.* (2011) explains that data formatting should balance the doctors' expressivity, efficiency, flexibility and adaptability to the typical workflow.

5.1.3.4 Data Storage

The various mechanisms of data storage in ambulatory clinics create unique AL challenges. Local physical repositories provide only limited accessibility to all HPs in and outside the facility. Furthermore, the physical repositories are at risk of potential loss due to physical damage or misplacement. Local digital repositories also have limited accessibility and although they are secured better than physical repositories, they still may be deleted without authentication. Both local physical and digital repositories may cause time delays for retrieving and updating information for HPs to maintain a single source of truth.

The website database systems require authentication and verification processes, which causes frustration and delays for HPs. These website servers may lack the computational resources to handle high network traffic or may be disconnected, resulting in the inaccessibility of information for HPs. The lack of available patient information may result in malpractice or delayed treatment.

5.1.4 Human-System Integration Challenges

Defty *et al.* (2022) discuss the technical challenges for HSI, which aids the identification of barriers to mature HSI in ambulatory clinic environments. These challenges are structured according to the RCI framework.

The representation of HPs, patients, facilities and equipment is greatly limited and immature in the ambulatory clinic at Tygerberg Hospital. The encapsulation of personnel and equipment information is limited to physical repositories and proprietary databases, which are inaccessible to most. Therefore, it is challenging to determine the future states of personnel quickly and accurately, as future appointments are recorded in physical repositories and facility schedules are not easily accessible. Physical interactions between HPs, patients and equipment are not often represented, which limits the traceability of ambulatory clinic processes.

Due to limited representation, the communication of HP, patient, facility and equipment knowledge and interactions is also constrained. Automated data exchange between different entities does not exist due to a lack of accessibility and interoperability between representations. Furthermore, to utilise the services of others, HPs and personnel often must physically locate and assess the availability of other personnel or equipment.

Physical interfacing mechanisms, such as patient files, are not multi-modal and reduce the comprehensibility of the information. For digital interfacing, connection latency and robustness are prominent challenges in healthcare facilities

5.1.5 Generic Needs for Ambulatory Care Services

The user needs analysis was performed considering the information from supported literature and observations of the ambulatory patient care activities and challenges for Tygerberg Hospital. These needs focus on the overall improvement of workflow efficiency and resource utilisation and address common challenges expressed by HPs and the goals of South Africa's National Development Plan. The needs for the HSI developments are presented more in-depth in Appendix A.3.

5.2 Case Study

As stated in the methodology, this thesis uses a case study for the development and evaluation of the HSI developments. The case study must fairly reflect a subset of the workflow processes, systems, equipment and personnel required in typical ambulatory care services. While the evaluation of the holonic HCPS developed for the case study is limited in its applicability to all ambulatory care services, it still provides an effective method for evaluating the value of integrating humans in the HCPS. The specific details of the case study were selected to best satisfy the thesis objectives by considering the HSI, healthcare and HCPS needs and requirements.

5.2.1 Selection Criteria

The importance and benefit of the maturity of HSI developments are argued by the author in the context of this thesis for complex HCPS environments. Therefore, the criteria in Table 3 were developed to ensure that the selection of a case study fairly represents the complexity of ambulatory patient care activities.

Table 3: Case study selection criteria.

Criteria	Description
Flexibility	Multiple system element configurations and interactions exist
Throughput	Importance of reducing waiting times and process delays
Quality assurance	Importance of ensuring quality standards during activities
Resource Utilisation	Importance of ensuring facility resources and personnel are performing value-added tasks
Cooperation	Frequent interactions between machines and humans
Cost sensitivity	Importance to reduce operational costs
Context switching	Humans must frequently change tasks
Multi-tasking	Humans must perform tasks in parallel
Information handling	A high volume and variety of information must be observed, recorded and analysed

5.2.2 Description

The selected case study for the development of the holonic HCPS is described in this section. The case study included a selection of ambulatory care services to satisfy the selection criteria and showcase the full range of the ambulatory care cycle. Specifically, the developed holonic HCPS for the case study will improve the HSI maturity of the system to coordinate ambulatory care activities at a respiratory clinic, which includes a spirometry investigation for lung function testing.

5.2.2.1 Overview

The respiratory ambulatory care activities in this case study follow a similar pattern to the typical ambulatory care process. The case study was designed to follow the entire care process, presented in Figure 17, and to evaluate the HSI in the holonic HCPS. The processes within the case study, along with the physical system component interactions in Table 4, are summarised as follows:

1. The outpatient is scheduled by appointment to consult with an HP at the respiratory clinic.
2. The HP will examine the patient to record the history and physiological and health-related information to make an informed clinical decision.
3. The patient will be required to have a spirometry test taken at the spirometry clinic to analyse their lung function.
4. The spirometry technician will perform a spirometry testing considering patient information and utilising spirometry prediction data and the spirometry test device
5. The HP will analyse the health condition metrics of the patient including:
    a. Body mass index analysis
    b. Spirometry test analysis

Table 4: Case study system component involvement.

Physical System Component	**Process**				
	1	**2**	**3**	**4**	**5**
Respiratory Clinic	x	x			x
Spirometry Test Clinic			x	x	
Spirometry Test Device				x	
Technician				x	
Patient	x	x	x	x	x
Respiratory Specialist	x	x	x		x

5.2.2.2 Respiratory Ambulatory Care Services

Respiratory ambulatory care services include the capturing of patient history and examining the patient whilst considering various patient health information and vitals. The HP might request additional investigations to gather further patient data for their diagnosis of possible conditions or diseases that the patient might be experiencing. These HPs focus specifically on respiratory conditions such as TB, asthma and chronic obstructive pulmonary disease.

In this case study, the HP’s role is to capture the patient history and examine the patient to gather height, weight, gender and race data to aid the diagnosis analysis. The HP must also send the patient for a spirometry investigation to gather information on the patient's pulmonary and respiratory condition. These results, along with the initial history and examinations, are utilised during the diagnosis of the patient for any health-related conditions.

5.2.2.3 Spirometry Procedures

The spirometry investigation tests the lung function of patients, using a forced expiratory manoeuvre. Essentially, a spirometry test machine is used to measure the volume of air and the rate by which a patient can exhale from their lungs. This measured volume of air exhaled over time is measured against standards considering the age, height, sex and race of the patient. Many standards and research findings have been released. The specific standards adhered to for the case study are presented in Hankinson, Odencrantz & Fedan (1999). These standards provide the predicted results of various spirometry tests depending on the patient data.

The measured data include the Forced Vital Capacity (FVC) and Forced Expiratory Volume in one second (FEV1). FVC is the maximum volume of air exhaled forcefully after maximal inspiration. FEV1 is the volume of air exhaled during the first second. A typical FVC test duration is roughly 6 seconds, but patients with respiratory conditions might take longer for the FVC test. Typical FEV1 test results should be between 70% and 80% of the FVC result for a healthy person. The FEV1/FVC ratio is thus important to consider. FEV1/FVC ratio less than the lower limit of normal for the patient standards is indicative of lung disease. (Centers for Disease Control and Prevention (CDC), in press)

5.2.3 Requirements

The requirements for the HSI system for this case study are summarised in Table 5 and Table 6. These requirements were developed from the needs and observations for ambulatory care services, HMS architecture requirements and HSI goals. These requirements are separated into functional and non-functional requirements according to a system engineering approach.

The functional requirements (FRs) are specific to care-related activities in the case study and answer the question of *what* the HSI system should do. These FRs aim to improve the administrative functions, information handling and supporting doctors through automated analysis. While guiding the developments in this case study to improve representation, communication and interfacing for HSI.

The non-functional requirements (NFRs) constrain the design of the HSI system and answer the question of *how* the HSI system should be developed. NRF1-5 are adaptions of the requirements for HMS systems presented in Christensen (1994). These requirements guide developments for systems that can handle disturbances, improve human integration, improve reliability and maintainability of the system, as well as support the flexibility of operations. NRF5 is further broken down following the intelligent agent characteristics defined within HSI.

Table 5: Functional system requirements.

ID	Requirement
FR1	Automate and digitise the transport, handling, formatting, and storage of patient information.
FR2	Provide access and mechanisms of data exchange for humans and equipment to retrieve and store data.
FR3	Provide context-aware data handling for personnel.
FR4	Automate the analysis of patient health to aid clinical decision-making.
FR5	Provide a mechanism of quantifying the data format for qualitative records.
FR6	Automate the scheduling of workflow activities.

Table 6: Non-functional system requirements.

ID	Requirement	Description
NFR 1	Availability	Ensure reliability and maintainability independent of the scale and complexity of the system.
NFR 2	Flexibility	Support changes in patient conditions, testing, treatments and workflows.
NFR 3	Robustness	Ensure system operation amidst planned and unplanned disturbances.
NFR 4	Disturbance Handling	Handle the unpredictable behaviour of humans and related errors.
NFR 5	Human Integration	Support human intelligence in the system.
NFR 5.1	Purposeful	Allow and aid each human to fulfil goals.
NFR 5.2	Perceptive	Enable each human to receive and filter information from the surrounding environment.
NFR 5.3	Aware	Enable each human to receive and request information from other agents to aid in achieving the agent's purpose.
NFR 5.4	Autonomous	Allow each human to decide the course of action to achieve a goal.
NFR 5.5	Able to act	Enable each human to select resources or other agents to achieve the plan.
NFR 5.6	Reflective	Enable each human to represent and reason about its abilities and goals.
NFR 5.7	Adaptability and learning	Enable each human to change.

6 Case Study Implementation

This chapter describes the case study implementation details for the developed holonic HCPS system. Furthermore, it provides insight into the application of the RCI framework and holonic system design process in the healthcare case study. This chapter shows the complexity of enhancing knowledge sharing and interactions between system components. The chapter starts with an explanation of the higher level details and ends with lower level details, following the holonic system design process presented in Chapter 5.

6.1 Partitioning System Functions using the ARTI Architecture

Partitioning the observed processes and system components from the case study into activity and resource holons required a flexible and iterative approach due to the various permutations that could exist. The aim is to partition the system into "natural holons" (SEBoK, 2021b). Partitioning the system in such a manner supports the natural physical or functional decomposition of system components in the real world, to minimize the coupling between holons and maintain a loose coupling throughout the holarchy.

6.1.1 Activity Holons and Functions

The processes in the case study, mentioned in Section 5.2, were considered when identifying the activity holons required for the holonic HCPS. When grouping various process steps, the priority was on reducing the complexity of the development, ensuring modularity and allowing for scalability and flexibility. The complexity of the development would be influenced by identifying too many activity holons (i.e. one for every process) when it could be argued that certain steps have a clear and fixed interaction and order. On the other hand, attempting to group all processes into one activity would threaten the modularity of the system – preventing flexible permutations of processes.

The processes were partitioned into two activity types, namely: consult activity type (*Con_AT*) and spirometry activity type (*Spir_AT*). Therefore, the AIs are the consult activity instance (*Con_AI*) and spirometry activity instance (*Spir_AI*), respectively.

The functions of the AT holons in the system are:

- Supervision: handling the creation, start, maintenance and termination of activity instances.
- Gateway: provide access to the aggregated instance information (e.g. state, services, etc.).
- Optimisation: analyse aggregated instance state and coordinate instances in response to optimisation.

The functions of AI holons in the system:

- Control: encapsulate control logic to execute all processes correctly.
- Coordination: communicate with other holons to achieve their intended processes.
- Representation: store state information specific to the activity instance.

The *Con_AIs* encapsulate processes related to the coordination of personnel and facilities for respiratory ambulatory care services. The processes include the registration of the patient upon appointment and handling of their history, examination and diagnosis care steps. The *Con_ATs* provide the functionality for the creation of a new *Con_AI* depending on specific information, such as the scheduling details, priority and personnel.

The *Spir_AIs* encapsulate processes related to the coordination of personnel, equipment and facilities for the spirometry investigation. Processes include the acceptance of scheduled patients, retrieval of patient information, handling of the spirometry test and related equipment, and processing of test results. The *Spir_AT*, like the *Con_AT*, provides the functionality for the creation of new *Spir_AI*s depending on specific requirements.

6.1.2 Resource Holons and Functions

The resource holons and related functions identified from the case study processes are presented in Table 7. Unlike activities, RTs are fairly intuitive to identify in the process steps. Resource holons can be identified through a physical partitioning of the system components involved in the various process steps. The choice of resources and aggregation (if required) also aimed to reduce system complexity, maintain modularity, and enable scalability and flexibility in the system.

As an example, aggregating the Spirometry Device (*SpirDev*) and the Spirometry Clinic (*SpirC*) into one RT would hinder the flexibility of the system – since a spirometry test device could be moved or used by another spirometry clinic, as testing equipment is often transported through the ambulatory facility due to lower resource availability and high user demand. Retaining the autonomy of the *SpirDev* allows other clinics to utilise the device in a flexible manner. Decomposing the *SpirDev* further (i.e. air flow pump, flow rate sensor, etc.) would increase the complexity of the holonic HCPS with little benefit in terms of flexibility and modularity for the system.

The RTs, like ATs, have the same functions as listed in Section 6.1.1, while the RI functions are:

- *Coordination* - communicate with other holons to execute processes.
- *Representation* – store state and attribute information specific to the resource instance’s physical counterpart.

The interfacing devices selected for the case study had to support the exchange of information between the physical devices and their RI counterpart. Interfacing requirements for doctors and clinic personnel were based on portability (lightweight and small) and ease of use. Interfacing requirements for patients required devices to be mounted throughout the facility to prevent damage or theft by limiting patient handling. The spirometry device was emulated; therefore, the interfacing requirements were dependent on the emulated software program. These interfacing RIs were separate from other RIs and not a part of them for many reasons. One such reason is to maintain the reconfigurable nature of the holonic system and support the modular decomposition of the physical system.

Table 7: Case study resource instances.

Resource Instance	**Function**
Patient (*Pat_RI)*	Represent the patient's state Analyse the BMI and spirometry conditions of the patient
Doctor (*Doc_RI)*	Perform examination • Capture patient history • Perform examination procedure • Determine required investigations Provide the diagnosis
Respiratory clinic (*RespC_RI)*	Facilitate examination procedure • Provide a physical room for the doctor's consultation • Coordinate the provision of required resources
Spirometry clinic (*SpirC_RI)*	Facilitate spirometry testing procedure • Provide a physical room for the spirometry testing • Coordinate the provision of required resources
Spirometry test device (*SpirDev_RI)*	Capture the readings of the spirometry procedure Apply the related spirometry standards
Interfacing Device (*Interfacing_RI*)	Exchange data between the website server and RI holon

6.1.3 Functional Relationships between Holons

The coordination between different holons requires functional relationships to achieve the purposes of each holon. In some cases, AI holons utilise and coordinate with resource holons to perform specific tasks and services related to the AI. RIs offer various functions which can be utilised in the system to act upon physical resources. The AI holons do not directly interact with any physical resources and thus require the functionality of RI to execute their process steps. Diagrams and further explanations of these functional relationships are presented in Appendix B.1 and B.2.

6.1.3.1 Consultation Activity and Resource Relationships

The *Con_AI* must be created (spawned) to coordinate other holons to fulfil the consultation process. In the case study, a *Doc_RI* can initiate the creation of a new *Con_AI* by retrieving the information from the *Con_AT* on how to create a new *Con_AI*, as is the *Con_AT's* role. Subsequently the *Con*_AT then spawns (creates) the new *Con_AI*. The *Doc_RI* must specify the details which establish the requirements for a new *Con_AI*. These requirements include the date of the scheduled examination, the *Pat_RI* involved and the priority of the examination. Additional requirements could relate to a specific *Doc_RI* or medical condition for optimised scheduling.

Once the *Con_AI* is created according to the requirements, it must coordinate the discovery of resource holons and inform them of the planned activities. The *Con_AI* uses these requirement details to negotiate with the *RespC_RI* to schedule an examination for the *Pat_RI*. The respiratory clinic is only operational during allocated examination shifts throughout the week because the physical clinic facility might be required for other processes in the hospital. Therefore, the *RespC_RI* must consider which times it will be available for the requested examination. Furthermore, the *RespC_RI* will require a *Doc_RI* for the examination – the *RespC_RI* must thus negotiate with *Doc_RIs* for availability. The priority setting of the *Con_AI* will influence the decision making when the scheduled time cannot be fulfilled by the *RespC_RI*. A high priority requires that the *RespC_RI* must schedule the examination on the earliest available date even if it exceeds the clinic's capacity. A low priority means that *RespC_RIs* may search for the earliest available date with the capacity for an additional patient. The *Con_AI* may decide on which *RespC_RI* to accept.

Upon the arrival of the patient, the *Pat_RI* will inform the *Con_AI* of their arrival, which will be relayed to the *RespC_RI*. The *RespC_RI* will handle the arrival of patients using the First-In-First-Out (FIFO) approach, by maintaining a record of the arrival time and order of the *Con_AI* requests. When a contracted *Doc_RI* begins their shift at the clinic, they will be allowed to request a *Con_AI* from the *RespC_RI* to begin examining the respective *Pat_RI*. For traceability, control and awareness, any recorded examination data and exchange will occur through the *Con_AI*.

6.1.3.2 Spirometry Activity and Resource Relationships

Similar to the *Con_AI*, a *Spir_AI* is also created through coordination with the *Spir_AT* to establish the requirements. Once created, the *Spir_AI* must search and coordinate with the *Pat_RI* and *SpirC_RI* to establish the scheduled time for the spirometry test. Since a technician is assumed to be in every spirometry clinic during its active shift, no technician RI is required for this case study.

The *Pat_RI* will send a message to the *Spir_AI* upon the patient's arrival to inform the *SpirC_RI* that the patient is waiting. At the time of the spirometry appointment, the *SpirC_RI* handles patient arrivals using the FIFO method and records their arrival time. The *SpirC_RI* then selects a *Pat_RI* according to one of the *Spir_AIs* in the queue. The *SpirC_RI* must search for available *SpirDev_RI* in the clinic for the *Spir_AI* to use and coordinate the exchange of information between the *Pat_RI*, *SpirC_RI* and *SpirDev_RI*.

The *SpirDev_RI* retrieves the age, height, gender and ethnicity of the *Pat_RI* to calculate the predicted spirometry results for comparison to the actual results. Once the test has been completed, the results and comparison are sent back to the *Pat_RI* through the *Spir_AI* for traceability. The *Spir_AI* is terminated once it has served its intended purpose.

6.1.3.3 Consultation Activity and Investigation Activity Relationships

Functional relationships exist between the *Con_AI* and *Spir_AI* as well. The case study process diagram shows that any ambulatory investigations are typically part of the decision-making process during a consultation process – in this case, at the respiratory clinic. Therefore, the *Spir_AI* should only be created when necessary to gather patient information for decision making for a *Con_AI*. A *nested* functional relationship thus exists between the *Con_AI* and *Spir_AI*. A nested relationship implies that the *Spir_AI* can only be created within, and last and exist during, the lifetime of the respective *Con_AI*. This relationship prevents other resource instances from creating a *Spir_AI* without an appropriate *Con_AI*.

The *Doc_RI* can only initiate the creation of a *Spir_AI* when interacting with a *Con_AI* and a respective *Pat_RI* during an examination process. The *Con_AI* searches for the *Spir_AT* to retrieve the creation requirements and, subsequently, initiates the creation of the *Spir_AI* through the *Spir_AT*.

6.2 Holon Implementation using the BASE Architecture

This section presents the details for the holon implementation using the BASE architecture. It offers a lower-level view of the ARTI holon functions and interactions. The presented implementation details are not exhaustive, but rather aim to highlight key generalised and case-specific functionality. The details are presented loosely following the holonic design process, described in Chapter 4, to condense the discussion and key developments.

6.2.1 Resource Interfacing Mechanism

The RCI framework prescribes *interfacing* mechanisms to enable the exchange of information between the physical resources and their cyber *representations* for HSI. Designing interfacing mechanisms for humans is challenging compared to designing for machines, due to the added complexity for data translation and comprehension required for humans. Furthermore, developing multi-modal interfacing devices with low latency and robustness proves to be an important barrier to overcome to ensure effect data exchange. (Defty *et al.*, 2022)

In all cases, web applications were developed to provide the information exchange mechanism between the representation and the physical counterpart. A web application offered a simple, scalable, multi-modal interfacing mechanism that could be used on various devices. Web applications are well supported with many frameworks (Django, Svelte, React, Angular, etc.) and resources to guide developments for various hardware devices. Web applications can be implemented through a client-server model to support the scalability enabled by the holonic system. Furthermore, web applications can be interacted with through various hardware devices such as tablets, laptops and augmented reality devices, as shown in Appendix B.4.

6.2.2 Generic Tasks, Services and Plugins

Tasks in the BASE architecture describe the internal holon functions, whereas services describe the functions requiring two or more interacting holons. Services result in individual holon tasks contractually bonded together through the service negotiation details. Task and services follow the 3SAL model, which adheres to the BASE architecture. The *activity* in the 3SAL is not to be confused with the ARTI-defined activity holons. Many task and service functions were generalised to reduce code repetition and support modular developments, of which some are discussed in this chapter.

6.2.2.1 Activity and Resource Type Holon

AT and RT holons have similar generic functionality and thus can be discussed jointly. The Type holons adhere to the BASE architecture, having all four sectors, a communication manager and plugins to enable their functionality. Type holons utilise the same mechanism of communication as instance holons; therefore, architecturally look no different.

Supervision is achieved through various plugins for the Type holon. The Type holon supervises the creation of instances during system startup. Upon starting up the system, a configuration file containing information on the instances and attributes is read and stored as a startup attribute for the Type. Each Type holon has *instance management* plugins, which look at the instances which are to be created, and their corresponding startup data, and schedules a task to spawn them. This task begins immediately and is handled by an instance manager EP, which encapsulates the functionality and decision making for preventing duplicate instance creation (in the case of a system restart). Each instance, once started, becomes an executing task for the Type holon. The instance management EP handles all messaging between the type and instances during its lifetime. When an instance is terminated, the instance management EP handles the termination, while an RP stores the instance data as stage 3 data in the biography.

Additionally, the Type holon supervises the dynamic creation of new instances during the system operation. The instance management EP contains the requirements and details necessary for a holon requesting the creation of a new instance. These requirements and details are structured by an ontology to improve interoperability between holons. Such requirements could include an instance identifier, the scheduled start of an instance, initial instance attributes, etc.

The Type holon acts as a *gateway* for other holons (types or instances) to gather information on the aggregated state of instances for a Type. Active Instances are stored as executing tasks in the Type, which allows the Type to collect information on all its instances to make available to requesting holons. Holons might request the information of a certain instance, such as its state of existence, and thus would pass this request to the type holon. The Type holon can then gather the information available to reply to the request. One such example is when an interfacing holon for a resource requests the existence of a certain RI. If the instance exists, the RT passes the information of the RI to the requesting interface holon.

6.2.2.2 Activity Instance Control

The decision-making functionality for the control of an AI's tasks and services is executed through a Finite State Machine (FSM). An FSM, presented in Figure 19, offers a suitable control mechanism for an AI's internal workflow in the case study implementation, due to their sequential and known process pathways. The Erlang OTP provides a generic FSM (gen_fsm) behaviour, which can be imported into an Erlang module to support the implementation of activity FSMs. This behaviour reduces the implementation time and complexity.

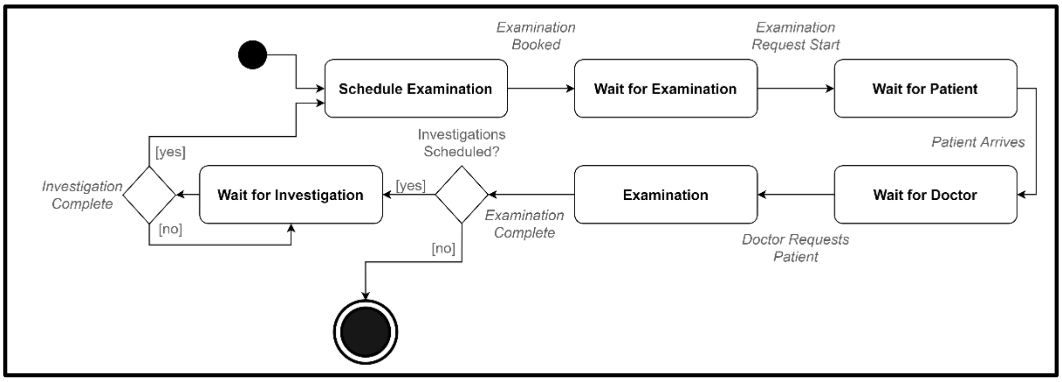


Figure 19: Consultation activity instance FSM.

An activity_fsm SP and EP were developed to encapsulate the FSM functionality in the activity instances. Upon creation of the activity instance, the SP schedules the FSM task to begin immediately. The EP behaves according to the gen_fsm behaviour. Events can trigger state transitions in the FSM depending on the event data and state logic. Unknown events, depending on the state, are ignored.

The FSM for the consultation activity instance is shown in Figure 19. Each block in the FSM diagram indicates a state, while the arrows indicate the transitions in response to *events*. Upon execution, the AI enters a "schedule examination" state, which coordinates the plugins and events to schedule the consultation activity with the necessary resources. When the examination state receives the “Examination Complete” event, logic in the FSM considers whether any investigations have been scheduled. If true, then it waits to receive a “Examination Complete” event from the appropriate plugin before doing another check. If there was an investigation scheduled, the activity instance will reschedule an examination for the patients' investigation results to be considered.

6.2.2.3 Service Negotiations

Services are made up of individual holon tasks contractually bonded. Each task adheres to the 3SAL, but due to their contract can be considered as one service that adheres to the 3SAL. The contract is negotiated and established through a contract net protocol (CNP) (Foundation For Intelligent Physical Agents, 2002). The CNPs offer a simple negotiation strategy while still catering for system disturbances and allowing holons to exist in the negotiation process at any stage. Services might require the establishment of other services and so on. The necessity for multiple service providers can result in nested negotiations, as shown in Figure 20.

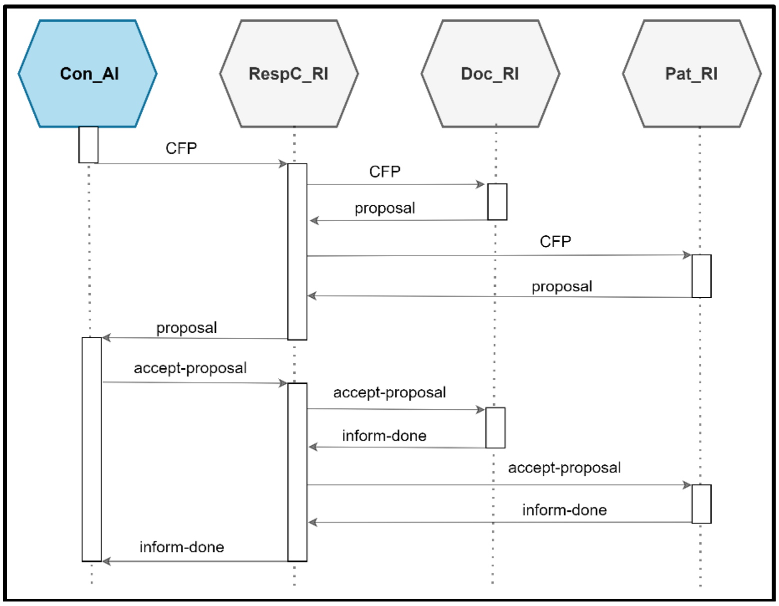


Figure 20: Consultation activity instance scheduling examination service.

An example of a *Con_AI* scheduling an examination service is used to explain the nested CNP scenario. When a consultation AI schedules an examination service, a *call for proposal* (CFP) is sent to all the examination service providers (in this case respiratory clinics), which consider the requirements of the CFP (i.e. date, priority, clinic shift times, number of doctors on shift already, number of patients already scheduled for the shift). If clinics cannot satisfy the desired date requirement, they can propose a postponed date depending on the priority requirement. If a clinic shift time is acceptable, but the clinic lacks enough doctors to partially fulfil the service, a nested negotiation is started, as seen in Figure 20. A CFP is sent to all the specialist service providers, in this case, the *Doc_RI*. *Pat_RIs* must also remain in the loop during the proposal stage. The CNP *reject* and *refuse* stages are not included in Figure 20 for simplicity but were implemented.

6.2.2.4 Web Application Resource Interfaces

The interface RIs and RTs were developed to exchange data with the web application server to the appropriate RI holon depending on the web application user. The web applications were developed using the SvelteKit framework. This framework supports modular developments and offers libraries and functional components to improve UI and user experience (UX) development. This framework also compiles all the functionality of the web application before run time to allow for seamless navigation to prevent interfacing delays. Svelte allows for reactive UI, which enables the interfacing mechanism employed to reflect the reality of representation in near-real time. Svelte is built on nodeJS and integrates JavaScript for functionality and HTML and CSS for web display and styling, which are well-documented and supported software languages. The web applications were hosted on a local server to reduce development complexity, since cloud functionality was not a requirement for the system.

The interface RT adhered to the previously defined roles of the RT. In the case study, the Type created instances using configuration data which includes the web server port numbers, Transmission Control Protocol (TCP) port numbers and identification. An interface RI was created for each respective RT.

A user accesses the web interface by connecting to a specific web server address. When the user is connected, their device acts as a client to the web server, which loads the Svelte application. The webserver also hosts a relay NodeJS server, which acts as a client to connect to the predefined TCP connection hosted by the EP of the respective interface RI, as seen in Figure 21. The interfacing procedure is as follows:

1. A human worker connects to the website server, through a device, and a new client to the web server is created.
2. The website server provides a login page for the client on their device.
3. The client uses an ID to log in, which is sent to a relay server.
4. The client login information is relayed through TCP socket (gateway), in the Erlang runtime created by an interface holon.
    a. Interface RI sends the ID to the RT that corresponds to the Type of client.
    b. The RT searches for an RI with the provided ID and passes the CM information of the RI to the interface RI holon.
    c. Interface RI holon requests a negotiation with resource instance with the respective ID to start an interface service.
    d. RI initiates the service negotiation through an interface SP. This service starts immediately, where all interface functionality is handled by an interface control EP. A new service for the interface RI holon also starts executing, which is handled by an interface session EP.

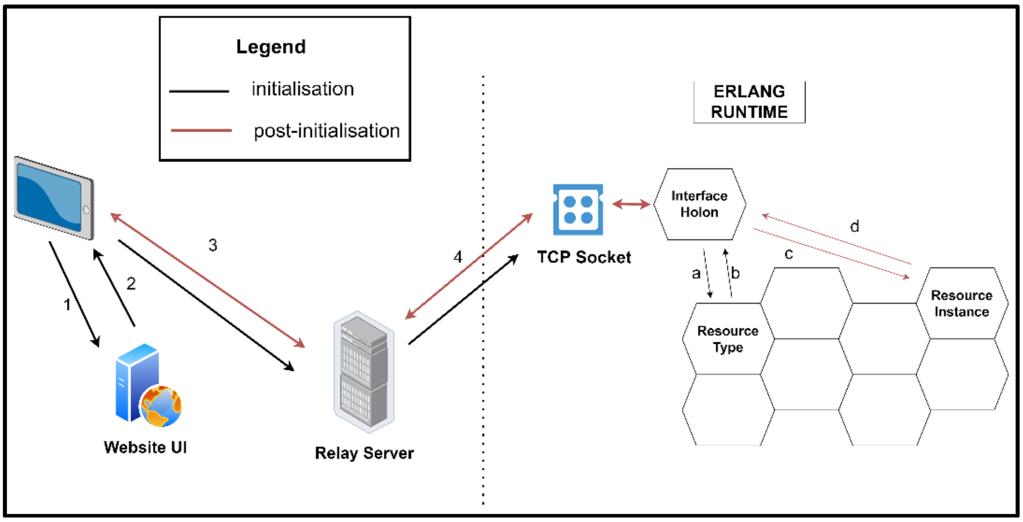


Figure 21: Interface mechanism for web applications.

Once the resource instance holon is connected to the specific website client, the information is exchanged across a clear data flow pathway as indicated by the red arrows in Figure 21.

6.2.3 Case-Specific Tasks, Services and Plugins

This section describes a portion of holonic HCPS developments for the case study. These details provide insight into how some of the complex dynamic interactions within the holonic HCPS were handled. Furthermore, this section highlights the usefulness of the BASE architecture for the implementation of the holonic HCPS.

6.2.3.1 Queuing for Clinic Facilities

For an appointment at a clinic facility, the *Con_AI* or *Spiro_AI* initiates a CNP service negotiation with the respective clinic RI. Once a date and shift time has been established, the service is scheduled for the AI and RI, as well as the *Pat_RI*. When the shift at the clinic starts (e.g. 08h00), the scheduled service on the AI, clinic RI and the *Pat_RI* will reach the *origo* and request to begin the service. The AI will enter a state waiting for the patient to arrive at the facility.

When the patient does arrive at the facility and logs in to one of the interfacing devices, the interfacing RI will become aware of the *Pat_RI* waiting state and update the UI with a question to the patient, as seen in Appendix B.3. The patient, remaining in the loop, can update their intention to wait at the clinic or not for the respective AI. If they choose to wait, the state variable of the *Pat_RI* is updated, which will inform the AI and hence the clinic RI that the patient is waiting. The clinic RI will then record the arrival time of the *Pat_RI* and store the ID of the respective AI as a state variable. The respective task remains at the *origo* during this process.

Only once the patient is requested by clinic staff will the clinic RI search the FIFO queue of AIs and begin executing the service task with the appropriate one. The AI is then removed from the state variable queue of the clinic RI. The start time of the service and arrival time of the patient are stored as stage 2 data for the clinic RI service, which is later analysed to monitor the performance of the clinic RI.

6.2.3.2 Examination Procedure

A *Con_AI* has plugins to coordinate the examination procedure between the *Pat_RI* and the *Doc_RI*, presented in Figure 22. This service is scheduled according to the *Con_AI* FSM. During the service execution, the *Con_AI* has an examination observation data template, structured according to an ontology, which is stored as an attribute that it received from the *Con_AT*. The *Doc_RI* may request the data template, which is then communicated to the *Doc_RI*'s interfacing RI (steps 6-11 in Figure 22), allowing the doctor to see the template and data on their interfacing device. The *Con_AI* automatically gathers required examination data (e.g. patient name, patient medical record number (MRN) etc.) that it is aware of before communicating the examination data template to the *Doc_RI* (steps 1-7). This enables automated data handling, improving the ALs for examination procedures.

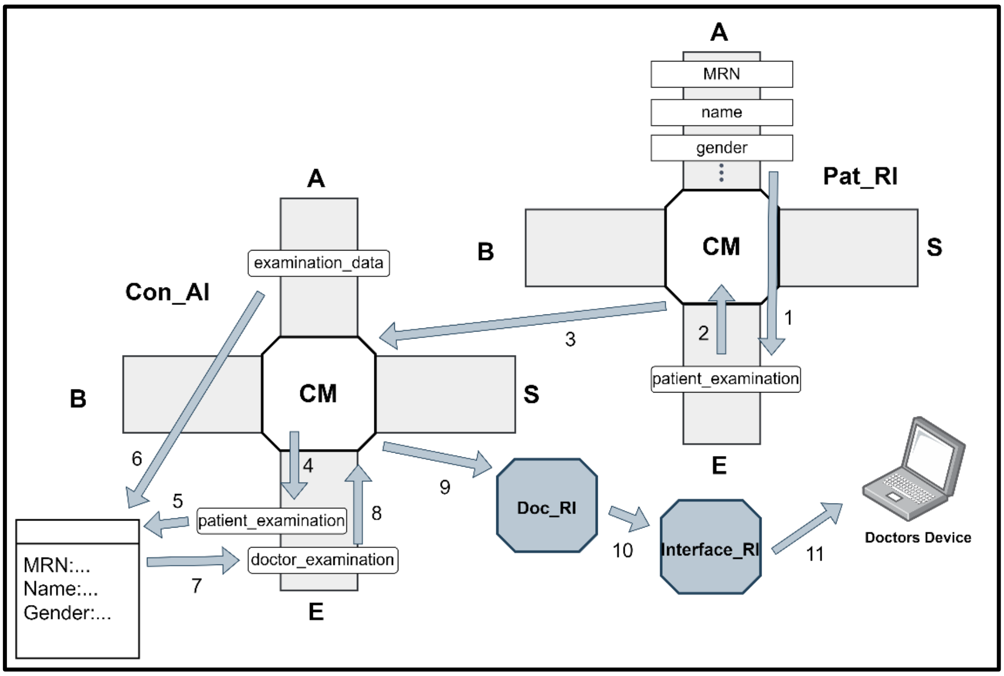


Figure 22: Holon interactions for the examination procedure.

The doctor can fill out the examination data template according to their observations of the patient, such as weight, height and history. Once completed, the doctor can submit the examination data, which is then communicated from the interface RI to the *Doc_RI* and the respective *Con_AI*, due to the examination service. This data is then updated as an attribute in the *Con_AI*.

When the examination service between the *Con_AI* and *Doc_RI* ends, it consequentially triggers the ending of the examination service between the *Con_AI* and *Pat_RI*, as orchestrated by the *Con_AI's* FSM control logic. Upon ending this service, the plugin considers any updated examination information and communicates this data to the *Pat_RI* to update their respective attributes (e.g. height, weight, gender, etc.). As long as the *Con_AI* remains active, the recorded examination data remains editable during further examination services and by different HPs. This allows the doctor to update examination data (e.g. diagnosis, medication) during a consultation process, after receiving further information from investigation results (e.g. x-ray images).

6.2.3.3 Spirometry Testing Procedure

The spirometry testing procedure, according to Figure 23, begins once the spirometry technician selects the respective patient from the *SpirC_RI* waiting queue (steps a-c-d-b in Figure 23), stored as a state variable, displayed through the *SpirC_RI* interfacing device. The technician can add a spirometry test device in which the *SpirC_RI* searches for available *SpirDev_RIs*, which are in the same location as the *SpirC_RI* and schedules an immediate service with the respective device (steps a-e-g-h-f-b). The technician can then link the *Spir_RI* to the respective *SpirDev_RI* and start the test (steps a-c-j-i).

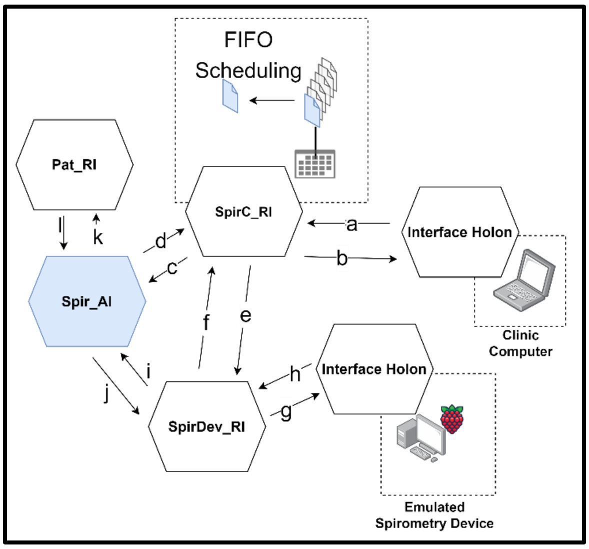


Figure 23: Holon interactions during the spirometry test procedure.

When the test is started, the *Spir_AI* for the *Pat_RI* schedules an immediate service with the respective *SpirDev_RI*. Upon execution, the *SpirDev_RI* requests specific information of the *Pat_RI* through the *Spir_AI*, such as height, gender, race and age needed for the test (steps i-k-l-j). The *SpirDev_RI* then interfaces with the device, through interface RI, to gather the measured FVC and FEV1 of the patient (steps g-h). Furthermore, the *SpirDev_RI* calculates the predicted spirometry results according to the NHANES III reference values.

Once the test is complete, the results are stored as stage 2 data for the testing service, which is communicated to the *Spir_AI* (step i). The *Spiro_AI* transfers these results to the *Pat_RI*, which are stored as data 2 (step k). The service is then biographed for the *Pat_RI*, *Spir_AI*, and *SpirDev_RI*. When the technician ends the service between the *SpirC_RI* and the *Spir_AI*, the *Spir_AI* decides, according to the FSM, which state to enter next.

6.2.3.4 Patient Health Monitoring

The patient health monitoring functionality for the patient BMI and spirometry condition is performed by an AP within each *Pat_RI*. The weight and height data, amongst other data, for the patient is captured during the doctor's examination and is stored as stage 2 data, which is later reflected as stage 3 data in the biography for the examination task, as seen in Figure 24. Similarly, the spirometry results for the *Spir_AI* are also stored as stage 2 data and later reflected as stage 3 data. The AP searches the biography of the *Pat_RI* for all examination and spirometry test tasks.

For the BMI, the AP filters through stage 3 data of the latest examination task and gathers the stored height and weight information. The AP then performs a BMI calculation and updates the BMI attribute of the *Pat_RI* and provides the categorization of the BMI value accordingly. For the spirometry condition, the AP filters through stage 3 data of the latest spirometry test task. The AP gathers the predicted and measured values of the FEV1, FVC and FEV1/FVC ratio. It then compares these values to certain measured standards to categorize the condition of the patient. It stores this result as an attribute in the *Pat_RI*. When the patient visits a doctor for an examination, the *Con_AI* can retrieve the patient condition attributes (i.e. BMI and Spirometry) from the *Pat_RI* to communicate to the *Doc_RI* through the *Con_AI*.

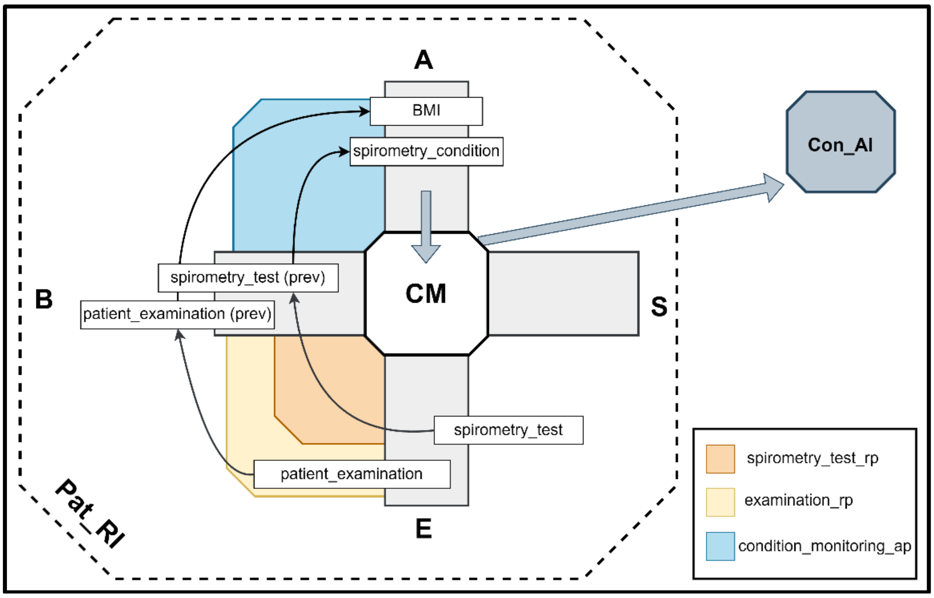


Figure 24: Intra-holon interaction for patient condition monitoring.

7 Case Study Evaluation

The holonic HCPS described in Chapter 6 was developed to meet healthcare needs and improve HSI. This chapter presents a functional validation and HSI evaluation of the developed system through an experimental procedure. The validation methodology considers the system's ability to meet the user needs, expressed in Section 5.2, and the functional and non-functional requirements. The evaluation aimed to measure the performance of the system to satisfy the HSI requirements and goals, as developed in Chapter 3. Both the validation and evaluation results aid in the discussion of the usefulness of the holonic HCPS and how it satisfies the case study objectives and contributes to the overall objective of this thesis.

7.1 Evaluation Objective

The evaluation aims to validate the ability of the system to meet the established requirements and assess the system's ability to improve human integration in a healthcare context. The validation considers different functional and non-functional aspects of the developed system showing that the system meets the user needs within healthcare and satisfies requirements developed from I4.0 concepts. The evaluation of human integration in the system provides insight into the ability of the system to improve human intelligence and cooperation. Both the system validation and HSI evaluation results provide insights into how the system meets the objectives of this thesis.

7.2 Evaluation Methodology

The evaluation method made use of an experimental procedure involving two scenarios, as described in Section 7.3, to validate the system and evaluate its HSI capability. The baseline scenario simulated the case study processes as they are currently performed in standard ambulatory clinic workflows. The second scenario, the HCPS scenario, simulated the case study processes utilising the developed holonic HCPS to aid the HSI. The experimental procedure provided a means of simulating the case study processes in a controlled environment. The system validation was performed during the use of the holonic HCPS in the second scenario. The HSI evaluation was performed comparing various HSI metrics between the baseline and HCPS scenarios.

7.2.1 System Validation

System validation "*is a set of actions used to check the compliance of any element with its purpose and functions*" (SEBoK, 2021c). The validation action must demonstrate that the system satisfies the system requirements.

The validation strategy adhered to a system engineering approach – using the validation methodology as seen in Figure 25, which is simple, and intuitive and links the validation results back to the system requirements. The *validation action* is applied to different elements of the case study system. Each *validation action* has an expected result based on the original system requirement and needs analysis. The obtained results from the validation actions are to be compared to the expected results, which may be subjective, to validate the *conformance* and *acceptability* of the system element and function. The objectivity of these results is limited by the lack of available quantitative metrics to support the validation actions. Certain requirements such as usability are instinctively subjective and dependant on the opinion of the validator. Similarly, modularity is challenging to measure objectively, but metrics such as code reuse can aid in quantifying these validations.

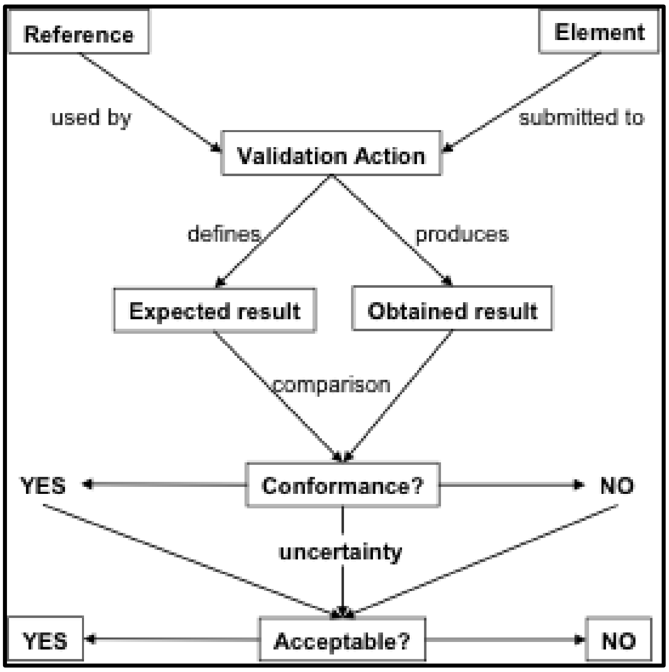


Figure 25: Usage of validation action (SEBoK, 2021c).

The validation actions, presented in Table 8, were selected to validate the requirements specified for the case study system in Chapter 5. Some of these validation actions result in only a simple yes or no, while others require quantitative measures. These validation actions provide a holistic consideration of most of the system's functional and non-functional requirements.

These actions were applied to the holonic HCPS while performing the experimental procedure. Performing the validation actions during the system operation ensured that the validation was applied fairly and with little bias.

Table 8: Validation metrics.

Ref	Element	Validation Action
FR1	Data exchange during the consultation	Does the *Pat_RI* store the information recorded during the examination?
	Data exchange during the investigation	Can the spirometry device gain patient information without user input?
		Does the *Pat_RI* store the test results calculated by the spirometry device?
	FIFO queuing data	Do the clinics automate the FIFO queueing process?
FR2	Patient data	Is patient data interpretable and accessible from different users and interfaces?
FR3	Doctors' interface during the consultation	Does the doctor receive context-aware information during the examination procedure?
FR4	Patient condition monitoring	Is the patient BMI and condition status automatically updated when new patient information is gathered?
		Is the patient spirometry condition automatically determined when new spirometry test information is gathered?
FR5	Doctor interface during the examination	Does the interface provide a mechanism for capturing qualitative health data?
FR6	Consultation activity Instance	Does the activity holon handle all scheduling conflicts or must a doctor intervene?
NFR 1	Resource scalability	RAM usage
	Activity scalability	RAM usage
NFR 2	Add a new activity type	Development time New lines of code Reused lines of code Code reuse rate
	Add a new resource type	
	Change activity flow	
	Develop new resource instance	
NFR 3	*not considered in this study	
NFR 4	Doctor interface during the examination	Does the system element prevent incomplete data during process steps?
NFR 5	Overall System	Evaluation (Next Section)

7.2.2 HSI Evaluation

The evaluation of HSI maturity in the developed holonic HCPS required the consideration of the various HSI aspects. The evaluation of HSI proved to be challenging, as the goals of HSI presented in Section 3.2 can only be considered subjectively. In an attempt to better evaluate these goals, two key performance areas were considered, namely: the time taken for humans to perform certain activities and the workload experienced by humans in the system. The time taken to perform certain activities is a useful measure of efficiency, which can be enhanced or hindered by the HCPS. In conjunction with the time taken, the perceived workload is an important consideration as it measures the cooperability of the HCPS with the human component. Considering either area in isolation would lead to a misunderstanding of the holistic maturity level of HSI.

Both the measured time and workload are only found to be insightful when considered between two scenarios. In isolation, the measured time or workload of the HCPS scenario does not provide insight into the HSI maturity of the system; therefore, it is necessary to compare the measurements with the measurements obtained from the baseline scenario.

To measure time, the time motion study (TMS) methodology selected was performed through *external continuous observations* making use of *multimedia recordings* to perform a *workflow time study* – a common methodology adopted for TMSs (Lopetegui, Yen, Lai, *et al.*, 2014). TMS annotation tags were selected, as seen in Table 9 and Table 10 to code certain activities performed by personnel. Table 9 presents the annotation tags to measure the HSI characteristics, such as time to filter, request, input and analyse information during the experiment, to evaluate the HSI goals within the system. Furthermore, annotations tags are presented to record interrupts and errors during the experiment. Table 10 presents the coding of general activities performed in the experiment to evaluate overall process efficiencies.

Table 9: Time motion study HSI metrics.

Metric	**Definition**
Time to filter (perceive) information	Time elapsed for clinic personnel, when they are searching, accessing, analysing or gathering any information necessary to perform a task.
Time to input (reflect) information	Time elapsed for clinic personnel, when they are recording, inputting or communicating any information necessary to perform a task.
Number of interruptions to workflow	Time elapsed for clinic personnel, when they are interrupted or forced to stop during a task.
Number of errors	Time elapsed for clinic personnel to handle an error to proceed with the task.

Table 10: Time motion study process metrics.

Metric	Definition
(pre-investigation) Examination time	Time elapsed for an examination process between a doctor and patient beginning, from when the patient is requested to when the doctor informs the patient that the process is complete.
Investigation time	Time elapsed for a spirometry test procedure between a technician and a patient, beginning from when the patient is requested to when the technician informs the patient that the process is complete.
(post-investigation) Examination time	Time elapsed for analysis of patient information between a doctor and a patient, beginning from when the patient is requested to when the technician informs the patient that the process is complete.
Patient waiting time	Time elapsed for a patient waiting in a queue at a clinic facility, from when the patient announces themself to the clinic personnel to when they are requested for an appointment.

The multi-media recordings were captured using three web cameras to capture video footage for processing and analysis. The position, angle and capture area of the video cameras ensured full coverage of all personnel activities during the experiment. The live footage was captured and recorded through Open Broadcaster Software, which is open source and allows for multi-feed recording.

To measure workload, a subjective workload rating questionnaire with absolute metrics was formulated, which was filled out by each personnel after each experiment scenario. The objective of the questionnaire was to gain insight into the perceived workload effort for users of the system compared to the baseline scenario. The questionnaire was developed considering the NASA TLX workload rating (Hart & Staveland, 1988) and the Bedford scale (Roscoe & Ellis, 1990).

The NASA TLX workload rating asks certain questions to determine the worker's workload demand in six categories. The mental demand relates to the amount of thinking, deciding or calculating required during the activity. Physical demand relates to the intensity and volume of physical activity. The temporal demand relates to the amount of pressure experienced by the worker. The effort relates to how hard the worker must work to achieve the expected level of performance. The performance is the level of success in completing the tasks. Finally, the frustration level is a measure of how discouraged or insecure the worker felt during the activity. This workload rating has been utilised in various applications. The typical usage of the scale uses a weighted average of the ratings across the six dimensions. In this evaluation, the NASA TLX was adjusted to incorporate a Likert Scale (Likert, 1932), to better quantify the results for comparison.

The Bedford scale was originally a workload rating used for pilots to measure their perceived effort during aviation activities. This workload rating specifically rates the workers perceived ability to complete the task given to them. For this evaluation, the scale was kept as a 10-point scale for the general range of activities that users had to perform during the experiment.

7.3 Experiment Design

The evaluation experiment was performed in a laboratory environment, to ensure controllable conditions and maintain ethical experimentation by avoiding the usage of patient data and the disruption to clinical operations. For the experiment, both scenarios comprised:

- One respiratory clinic with two consulting rooms
- One spirometry test clinic with one spirometry testing device
- Two respiratory doctors
- One technician
- Six patients

The ambulatory process flow, as depicted in Figure 26, began with each patient being scheduled to visit the respiratory clinic for an initial consultation with a doctor. The doctor's goal was to report on the BMI and spirometry condition of each patient. Therefore, after the initial consultation, patients were scheduled to the spirometry test clinic for a spirometry test. After the spirometry test, patients must go back to the respiratory clinic for an analysis and diagnosis consultation with the specialist.

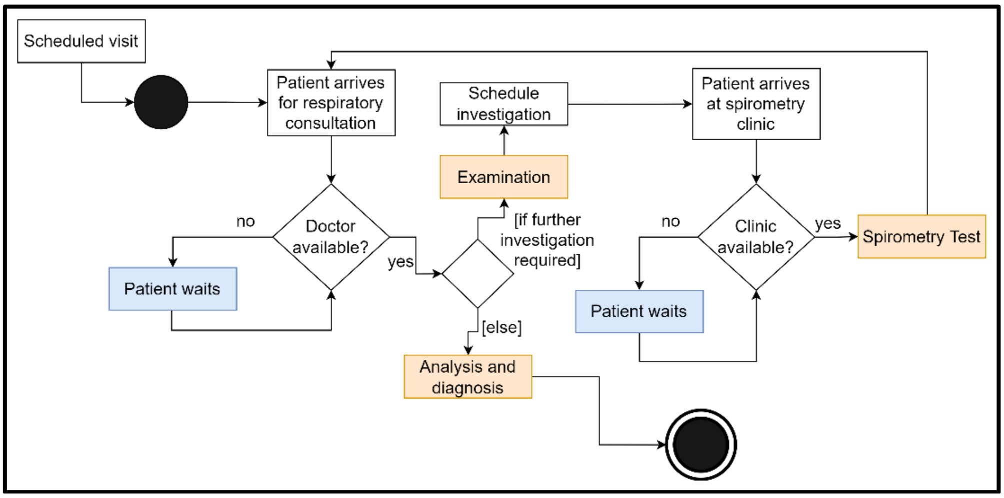


Figure 26: Process flow diagram of the evaluation experiment.

During the history process, the doctor was required to ask six standardised questions related to family health conditions and personal habits related to the patient's health. During the examination process, the doctor recorded the date of birth (DOB), age, gender, race, height and weight of the patient, which was needed for subsequent analysis and testing.

The spirometry testing required the patient's gender, race, height and age. The spirometry testing device specifically captured the patient's FVC and FEV1 and calculated the predicted and lower limit of normal for the FVC, FEV1 and FEV1/FVC ratio metrics.

During the analysis and diagnosis, the doctor considered the patients' attributes and equations in Table 11 to determine the appropriate classification for their diagnosis. Thereafter, the respective patient then finishes the experiment process. The experiment ended only once all patients had been examined, analysed and diagnosed.

Table 11: Analysis and diagnosis classification methods.

Equation	Range	Classification
$BMI = \frac{weight\ (kg)}{height(m)^2}$	<16	Severe Thinness
	16-17	Moderate Thinness
	17-18.5	Mild Thinness
	18.5-25	Normal
	25-30	Overweight
	30<	Obese
$\frac{FEV1\ (l)}{FVC\ (l)}$	<70%	Abnormal lung function
	<LLN	Abnormal lung function
$\frac{FEV1\ (l)}{predicted\ FEV1\ (l)}$	<80%	Abnormal lung function

7.3.1 Patient Profiles

To ensure consistency throughout the experimental procedures, fictitious patient profiles were established for each patient. Fictitious profiles were used to mitigate ethical risks related to sensitive patient information and to provide controls to the experiment. These profiles specified the patient information such as gender, race, height, weight, and DOB, as well as the set of answers to the standardised history questions to be asked by the specialist. Furthermore, the “measured” FVC and FEV1 values were prescribed before the experiment and displayed on the patient profile, to simulate the actual spirometry measurement. The “measured” spirometry values were prescribed because spirometry devices are expensive and offer little additional research value compared to simulating the measured value. Furthermore, simulating the spirometry device enabled simulating patient information, to predefine health abnormalities.

The six patient profiles and respective information were simulated intentionally to result in two patients suffering from BMI issues and two patients suffering from spirometry abnormalities, according to the analysis metrics in Table 11. These diagnosis results should ideally reflect in the specialists' analysis of patient information during the evaluation experiment scenarios. Research group members acted as the "patients", doctors and technicians in the experiment.

7.3.2 Baseline scenario

The baseline ambulatory scenario was aimed at reflecting the actual operations performed and tools used during an ambulatory process. The layout of the facility is shown in Figure 27 (the place markers A-C are discussed in Section 7.3.3). In effect, all patient data (i.e. examination, testing, analysis) were transported and stored using physical repositories (e.g. patient folders, notes, etc). Patients were instructed verbally by the clinical personnel. Patients had to verbally inform clinic personnel of their arrival when visiting each clinic for the first time to hand in their patient folder. This folder was then ordered in a FIFO manner at the clinic.

Specialists were required to filter through the stacked patient folders in the FIFO queueing order and call the respective patient from the clinic queue. All examination, analysis, and diagnosis notes were dictated through pen and paper onto examination sheets stored in the patient folder. The analysis of patient information had to be manually searched for and read from the patient folder. Specialists used calculators for determining certain analysis results.

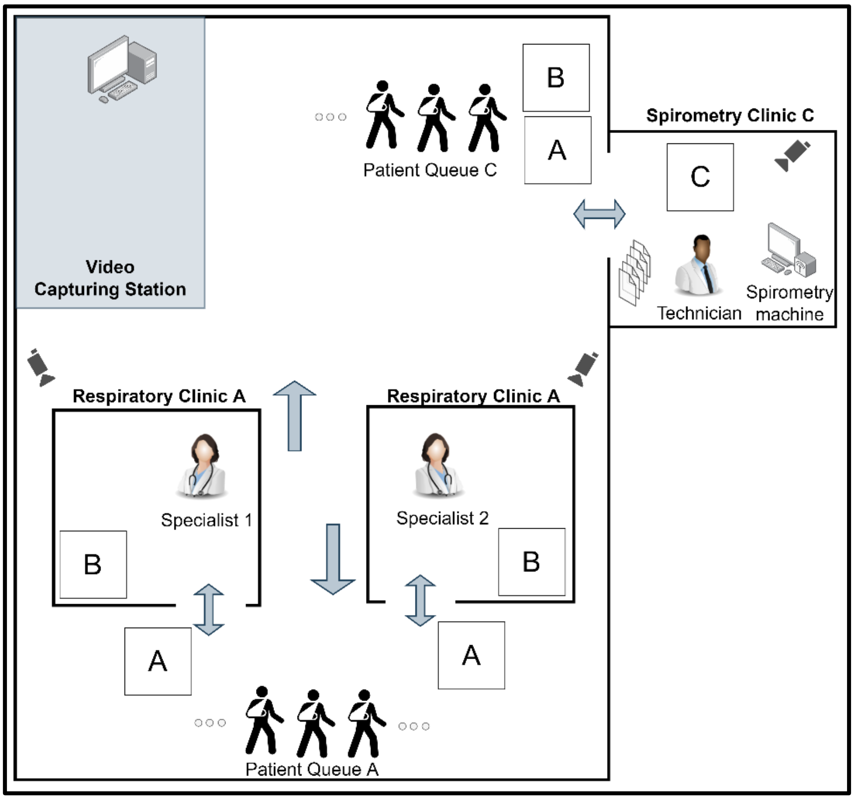


Figure 27: Experiment layout.

The spirometry technician had to search through the patient folder for examination data (i.e. height, age, gender, race) and input it on an online spirometry calculator, which incorporated the NHANES III reference standard. The patient informed the technician of their “measured” FEV1 and FVC values when requested. The spirometry test results were then printed, using the printer placed at marker “C”, and placed in the patient folder.

7.3.3 HCPS Scenario

The HCPS scenario utilised the developed holonic HCPS for the experiment scenario. An overview of the experimental clinic scenario layout is similar to that shown in Figure 27, but which makes use of different technologies compared to the baseline scenario.

No physical repositories were required for the scenario, as all data transport, storage and handling were digitised and automated. Instead of folders, patients were given QR codes to be utilised by themselves and specialists for interfacing and identification using the patients' Medical Record Number (MRN). Two electronic tablet devices were placed at the entrance of the respiratory clinic and one device at the entrance of the spirometry clinic, as shown by the place marker “A” in Figure 27. These devices allowed patients to interface with their RI in the cyber layer to perform certain functions (e.g. to be added to the waiting queue) and view information.

When queuing, patients first interfaced with their RI, through a tablet, to “inform” their arrival for the scheduled service as part of the *Con_AI*. Patients could view their planned, current and historic information according to their RI representation.

Both clinics were equipped with RI interface devices (i.e. clinic terminal), which interfaced with the respective clinic RI, to display queueing order, arrival time and average patient activity duration (i.e. waiting time, consultation time, testing time), shown by place marker “B” in Figure 27. This information was displayed to all personnel and patients, retaining human-in-the-loop principles.

Doctors used laptops to interface with their respective RI during the experiment. The doctors had to begin their clinic shifts, accept patients, record examination data and schedule patients through the interface device during the examination. The interface provided contextual information to the doctor about the patients' attributes (e.g. gender, race, etc) and historic records. To schedule the patient for a spirometry investigation, the doctor could use the QR code or manually enter the patient MRN, along with scheduling details.

For analysis, the APs of the *Pat_RI* monitored and reflected the BMI and spirometry health condition of the patient, according to Table 11, and made this information available to the consulting doctor. The analysis was automated to aid the decision making of the doctor.

The spirometry technician used a laptop, shown by place marker “C” in Figure 27, to interface with the spirometry *SpirD_RI* and the *SpirC_RI* to handle the spirometry testing activities. The location settings (attribute) of the spirometry device was set so that the *SpirC_RI* could contract the *SpirD_RI*. The technician entered the “measured” FVC and FEV1 into the spirometry device UI. Furthermore, the technician selected the patient and the spirometry device to run the test through the spirometry clinic interface. The functional interaction between the *Spir_AI SpirC_RI*, Pat_RI and *SpirDev_RI* automated the exchange of the patient attributes and testing results.

7.4 System Validation

7.4.1 Results and Analysis

The automated data handling and exchange (FR1) is validated by considering the data exchange during the consultation, investigation and patient queue processes. In a typical ambulatory clinic scenario, these processes are manual (requiring patient file handling) and thus offer comparative validation actions to validate the requirement.

The developed system is expected to reflect all examination data captured by the doctor during a consultation, as well as any data related to the spirometry test; automatically and digitally. The result of these actions validate that the system does transport, format and handle this data automatically. Furthermore, the system does order and represent the FIFO queueing with timestamps and patient information digitally when the patient arrives at the clinic. Therefore, FR1 is shown to be fulfilled through these systems functions, although there are still opportunities for improved data automation and digitisation.

The accessibility and interpretability of data (FR2) were validated by considering the patient data in the system. According to HSI goals and holonic requirements, information in the HCPS should be accessible to all entities or agents to improve cooperability and agent intelligence. The system successfully reflects all patient information digitally (within their RI holon), which is accessible and interpretable to doctors, clinics and testing devices. Specifically, doctors have access to the historic data, attributes and analysis data of patient information. This information can be seen when the doctor consults with the patient. Furthermore, the spirometry test device, during the spirometry testing procedure, could access and exchange patient information necessary to execute the test and calculate the predicted results. This demonstration of data encapsulation, accessibility and interoperability for digital representation satisfies FR2.

The context awareness functionality (FR3) of the system was validated considering the information displayed on the doctors' interfacing device. When consulting, the doctor only received the information of the assigned patient displayed on the interfacing device, without needing to manually search and filter through other patient data. This was enabled by the coordination between the *RespC_RI*’s knowledge of the patient queue, the requesting *Doc_RI*, the selected *Pat_RI* and the respective *Con_AI*. Furthermore, the developed system “understood” which patient was assigned to the doctor, via the coordination of the *Con_AI*, and could thus automatically populate the examination information (name, age, gender) on the doctors' behalf, while allowing the doctor to still edit this information if required. This functionality demonstrates the ability of the system to satisfy FR3.

The automated clinical analysis (FR4) of the system was validated considering the system's ability to aid the decision making regarding the health status of the patient's BMI and spirometry performance. The *Pat_RI* analysis plugins continuously and automatically analysed all historic data relevant to report on the health conditions (e.g. BMI) and stored these results – making them available and interpretable to other holons (i.e. doctors). The reflection and analysis plugins were able to classify the BMI bracket of patients and determine the spirometry health status of each patient according to three different analysis criteria. This improved decision-making time, as discussed in Section 7.5.2.

The quantification of qualitative examination data (FR5) was validated considering the function of the doctor's UI during the examination process. The interfacing mechanisms for recording examination data provided scales for qualitative metrics to be recorded by HPs. Quantifying such health data, rather than recording unstructured qualitative records typically practised on paper-based records, improved the interpretability of such data.

The automated scheduling (FR6) was validated by considering the functionality and ability of activity holons to coordinate scheduling on behalf of patients, clinics and personnel. Doctors were able to inform the system of certain scheduling requirements and details for both consultation and spirometry investigation procedures. The respective activity holons (e.g. *Con_AI*) handled all scheduling tasks and service negotiations with other holons without manual user input or assistance, while considering multiple factors such as resource availability and scheduling priority, and retaining *human-in-the-loop* principles. The ability of the AI to handle coordination for scheduling in the system satisfies FR6.

The scalability of the system (NFR1) was validated considering the computational requirements of the system when creating additional resource and activity instances. As seen in Figure 28, the systems' usage of random-access memory (RAM) increased with additional holon instances. Adding holon instances placed a greater memory usage demand on the computing system, as can be seen by the gradual increase in RAM usage. The change in CPU demand was minimal and, therefore, RAM remained an important validation metric to analyse the scalability of the system. RAM usage increased due to a greater demand for volatile memory for variable storage and computation with the addition of plugins and Erlang processes. These results showcase how additional holons could be supported by the system.

The flexibility of the system (NRF2) was validated by considering a developer's perspective on reconfiguring the system for new resource and activity instances and types. The developer-centric validation actions were chosen to provide a holistic viewpoint. The results, presented in Table 12, report on the reconfigurability of the system for the addition and changes to existing system elements.

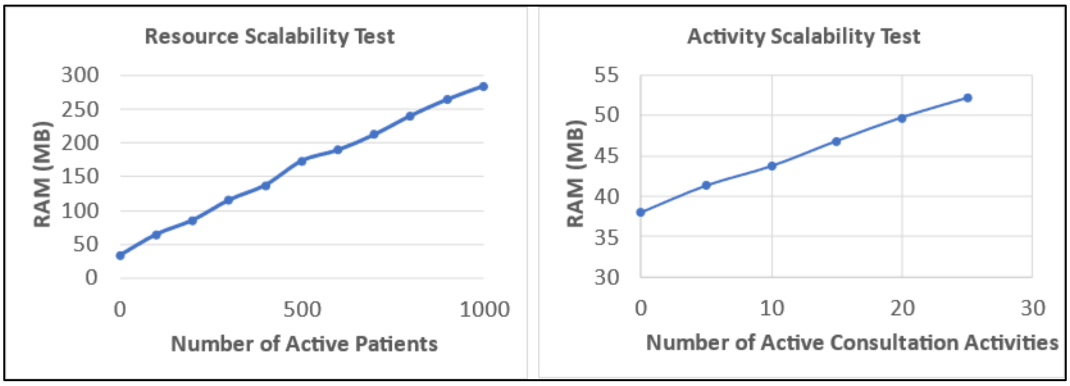


Figure 28: Scalability validation result.

A new personnel RT was created, which was a technician RT for the spirometry clinic. This technician required an SP and EP to be contracted by the *SpirC_RI* when a patient was scheduled by a doctor for a spirometry test. As seen, the code reuse rate was high due to the similar functionality in the *Doc_RI's* SP for when the respiratory clinic contracts a doctor for examination tasks. A blood test clinic RT was also added to the system, which had a high code reuse rate from the spirometry clinic, since being another investigation clinic. Developing a new AT type to handle blood test investigations required more development time and new lines of code. Changing the flow of an existing AT demanded little development effort due to the single point of control provided by the FSM in a plugin. Changing the configuration file for a new RI required the least effort - the development time is only affected by the number of unique attributes required for the RI startup.

Table 12: Flexibility validation action result.

Validation Action	System Element				
	New (Personnel) RT	New (Facility) RT	New AT	AT Process Flow	New RI Upon Startup
Development Time (min)	25	10	38	2	<1
New lines of code	521	599	766	3	6
Reused lines of code	481	574	656	0	0
Code reuse rate (%)	92	96	85	0	0

The disturbance handling (NFR4) of the system was validated considering the ability of the system to prevent processes due to human error. Incomplete process steps during the consultation are typical in ordinary ambulatory clinics are most commonly due to missing patient or examination information required for investigation processes. The system prevents doctors from submitting incomplete examination records for certain patient data necessary for spirometry testing. This functionality is supported by setting certain parameters of the AI to ensure certain data is entered before being processed.

7.4.2 Discussion

The validation of the system was performed by considering the various system elements and functions. The validation methodology was useful by providing sufficient insight into the system's functionality. The results of the mentioned validation actions were acceptable for the system performance and show satisfactory compliance with the system requirements. Furthermore, the results validate the system's adherence to the customer needs from which the requirements were derived.

7.5 HSI Evaluation

7.5.1 Results and Analysis

The results and analysis of the TMS and workload rating are used as evidence to aid the discussion of the evaluation of the system's ability to support the goals of HSI, as described in Chapter 3 and 5.

7.5.1.1 Time Motion Study

The multimedia recordings of both scenarios were processed using an open-source annotation tool called ELAN. ELAN allows for coding the video recordings according to relational hierarchies (i.e. Tiers) between annotations. An example of such, seen in Appendix D.2, is that the HSI elements, such as *filter_info*, can only be coded within a general activity, such as *exam_pre*. Once coded, the data was exported as a comma-separated value (CSV) file.

MATLAB scripts were developed to extract and aggregate the time coding from the CSV files for analysis. The total and average duration of the various time codings were computed for comparison. The overall time for each code, when comparing the two scenarios, is presented in Figure 29.

The HCPS scenario resulted in a reduction in overall time to filter information (76%), reflect information (31%), examine (28%), investigate (52%) and analyse (67%) for the clinic personnel, and a reduction in patient waiting time (46%).

No errors or interruptions were recorded in the HSI scenario. An error due to incorrectly recorded patient information in the baseline scenario was reported. This was only discovered when the patient went for the spirometry testing and thus resulted in an investigation delay. This anomaly in the data was neglected in the results and analysis.

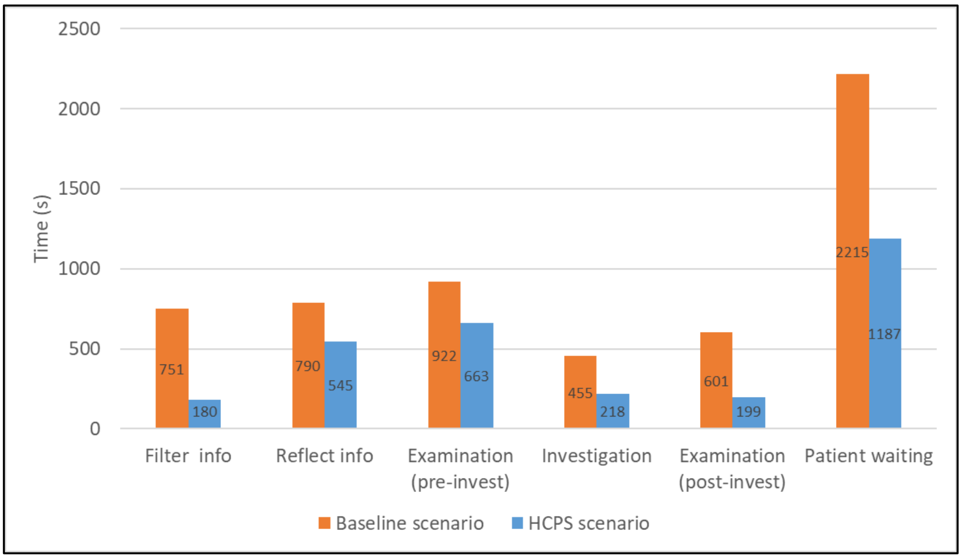


Figure 29: Time motion study analysis.

7.5.1.2 Workload

The NASA TLX and Bedford workload rating scales were captured through an electronic questionnaire and analysed using Microsoft Excel. The specialist and the technician interacted more with the developed system compared to patients. Therefore, results from clinic personnel and patients were analysed separately, due to their differing levels of involvement in the experiment activities.

The NASA TLX rating results for the clinic personnel (i.e. doctors and technicians) are presented in Figure 30, and for patients in Figure 31. The Bedford workload rating results, presented in Figure 32, show both the patients and clinic personnel.

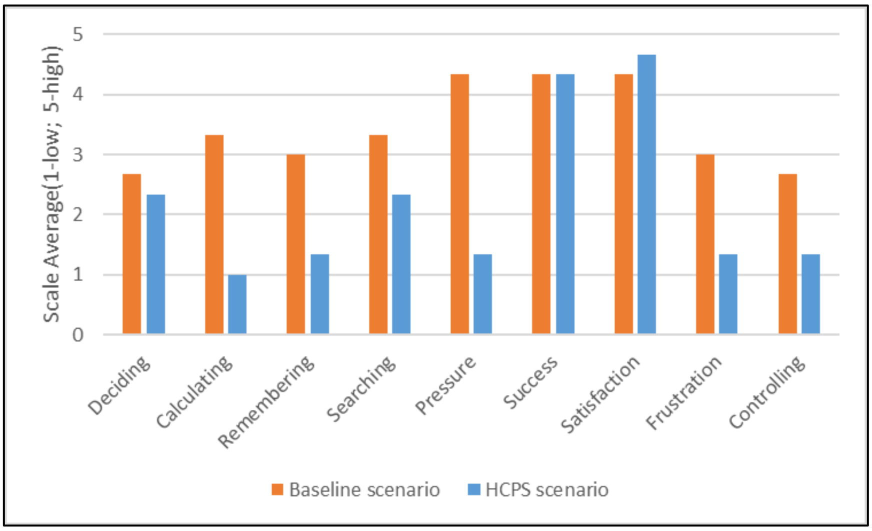


Figure 30: Clinic personnel NASA TLX rating.

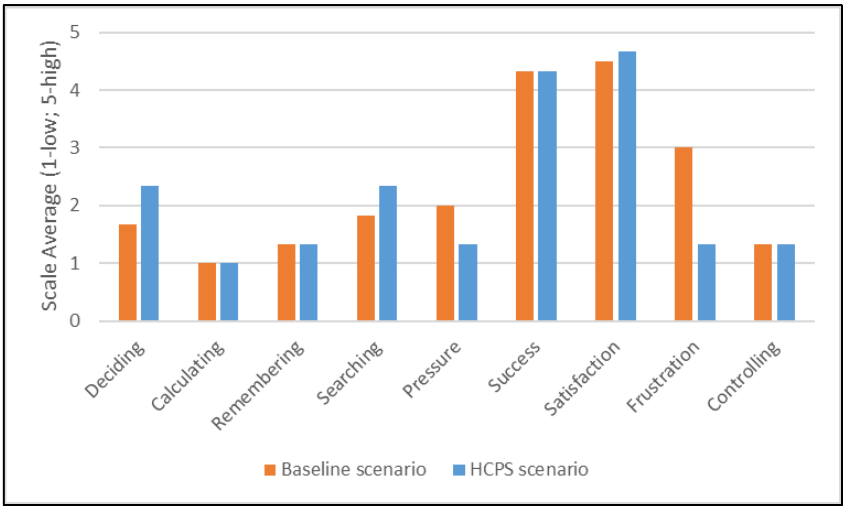


Figure 31: Patient NASA TLX rating.

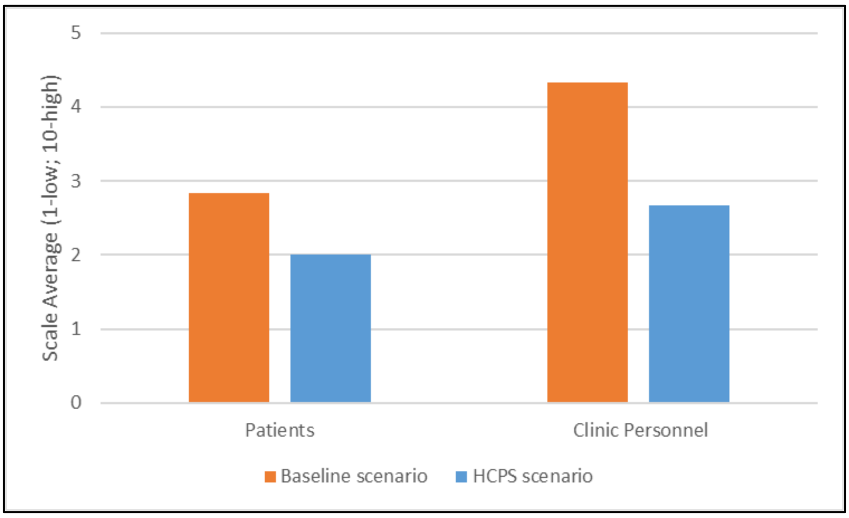


Figure 32: Bedford workload ratings.

7.5.2 Discussion

The objective of the experiment was to evaluate the HSI maturity and ability of the system to satisfy NFR5. Therefore, this section addresses the subjective evaluation of HSI goals and NFRs by considering the experimental results and analysis.

Humans are more *purposeful* in the HSI scenario, considering the overall reduction in time to complete the experiment scenario and the improved Bedford scale rating average. The time spent on each activity (i.e. examination, investigation, etc.), including patient waiting time, was significantly reduced. One of the goals for ambulatory clinics is that they achieve high throughput and maintain efficiency, which was enabled by the developed system in the HSI scenario. The reduction in procedure time is attributed to the automated data handling and transport performed by the system, reducing time spent having to filter through information to reflect process information. The context-aware information led to improved efficiencies, reducing the time spent filtering through unimportant historic patient information. The interfacing device selection, according to the RCI framework, had a significant influence on the experiment time. Furthermore, the Bedford workload rating reduction for both clinic personnel and patients indicates an improved perception by humans of completing their tasks, hence achieving their purpose in the system.

Humans in the system appeared to have greater *perception* and *awareness* of the operational environment during the HSI scenario. The reduced time to filter information by clinic personnel was due to context-aware interfacing and automated data handling, transport and patient analysis functionality. This is further evident when considering the reduced *searching* workload metric in the HSI scenario. The increase for patients in the HSI scenario is due to the patients needing to look for scheduling information on the interfacing devices, rather than being ordered in the baseline scenario. The improved perception for clinic personnel is evident with the reduction in the workload rating for the *remembering* and *calculating* metrics. Clinic personnel experienced a greater demand for remembering and calculating, as evident in the results, requiring greater consideration in this discussion. The significant reduction in calculating was due to the automated patient health status analysis performed by the *Pat_RI*'s analysis plugins. Only a slight change in these metrics is evident for patients, since they didn't have to calculate or remember much information beyond the simple orders and questions from the clinic personnel.

The developed system also aided humans in their *ability to act*, while retaining their *autonomy*. The NASA TLX rating indicated improvements in feelings of frustration, satisfaction and success during activities for both patients and personnel. The feeling of satisfaction and success with low frustration is indicative of minimal barriers hindering the ability of a person to perform tasks they desire to, as well as indicative of their ability to act freely (i.e. autonomy). Interestingly, while the clinic personnel's perceived workload rating for *deciding* decreased, it increased for patients in the HSI scenario. Reasoning suggests that because the system aids doctors and technicians in clinic decision making, the strain on decision making by the clinic personnel should be alleviated. However, since the developed system promotes patient autonomy, patients are given greater responsibility for decision making (hence autonomy) through their interfacing devices, resulting in a higher decision making workload compared to the baseline scenario.

Humans were able to better reflect their information in the HSI scenario. This can be noted in the reduced time to reflect information and the lower workload rating for *remembering* for the clinic personnel. This developed system aids data handling, transport and analysis, while reflecting the planned, current and historic state of different patients.

7.6 Discussion

The implications of the results of the case study are considered here. Because the case study was representative of many healthcare contexts, it is expected that the implications will be more widely applicable than just the case study's context.

7.6.1 Implications of the Application of the RCI Framework

The RCI framework elevated the human component, including clinic personnel and patients, to the cyber layer of the HCPS. By ensuring that all humans were represented digitally and were able to communicate with each other, the holonic HCPS shifted the main mechanisms of cooperation and interaction from the physical layer to the cyber layer. The coordination of system components during activities could be controlled and traced due to the communication between holons. This improved the integration of humans in the system by augmenting their situational awareness and ability to act.

The RCI framework guided HSI developments to improve the autonomy of humans, especially for patients. By improving the representation of patients, their coordination could be better managed by themselves. Furthermore, patients had a greater awareness of what the system expected of them for future activities. Since the framework defines the need for interfacing mechanisms for representation, this enabled HitL principles to allow patients to provide and receive feedback based on their decisions.

The framework improved the ease of the HCPS reconfiguration. By respecting the autonomy and cooperability of representations, reconfigurations of the system through existing or new processes or system components could be done with reduced development effort.

From the above, it is clear that the framework guided the selection of architectures, in this case holonic architectures, to ensure that all processes and system components were represented and allowed to cooperate. The selection of these architectures would not have been as intuitive without the clear requirement for representation and communication. Furthermore, the framework guided the selection of appropriate interfacing technologies to ensure all components could exchange information with their representations.

7.6.2 Implications of the Holonic System Design Process

The holonic system design process, using the ARTI architecture, aided in partitioning the higher-level system processes and components into distinct activities and resources to encapsulate the various system functions. Partitioning the system processes is not a trivial task when trying to respect modularity and scalability principles. The holonic system design process guides this partitioning by standardising the roles and responsibilities of different holons, by creating “natural holons”. Knowing these roles and responsibilities, the developer can more easily decide which functionality should be encapsulated by which holon.

Furthermore, the holonic system design process standardises the interactions between holons, specified by the BASE architecture, to enable cooperability between all holons. This standardisation enables the scalability and reconfigurability of the system. Additional holons can be developed accordingly without needing to understand the entire system's interactions.

Likewise, the holonic system design process standardises the internal holon interactions and data storage. The BASE architecture provides this internal structuring and thus improves the development and debugging time for developers. During the development of new holons, code can be reused, which reduces development errors and improves the understandability of the system.

In summary, the holonic system design process reduced the overall development time and complexity. Furthermore, it improved the means of collaborating with other researchers and developers for the development process and allowed for a more effective way of communicating and refining the system development amongst stakeholders.

7.6.3 Implications for HSI in Healthcare

The case study results show that a higher maturity level of HSI in healthcare facilities improves the overall workflow efficiency. The automated data handling, formatting, transport and storage led to the reduced time spent by humans performing administrative logistic tasks during clinic processes. As such, clinic personnel can spend a greater portion of their work time on valued added activities, which is a critical need in under-resourced healthcare facilities with high patient volumes. The improved HSI can also lead to reduced patient waiting times, allowing patients to receive treatment with shorter delays and experience a higher quality of care.

HSI also aids the clinical decision-making processes within healthcare by offering more context-aware information and automated analysis of health data. HPs are often required to consider large volumes of health data and perform manual analysis for clinical decision making. HSI developments can reduce this demand by automatically performing calculations and analyses of patient data, which is exchanged through the cyber layer in real-time to support HPs.

HSI developments improve the quality assurance of healthcare processes by providing mechanisms of error handling and traceability. Elevating communication to the cyber layer and implementing an interfacing mechanism for data exchange, ensure that all information can be traced digitally and inspected for errors. Decision-making with lacking evidence or information can now be prevented through improved traceability, to avoid potential malpractice. Historic information is also readily available when investigations into cases of poor-quality care arise.

8 Conclusion and Recommendations

This thesis presented a holonic HCPS that supports HSI in healthcare activities. An ambulatory healthcare clinic scenario was chosen as a case study, based on observations from Tygerberg Hospital and literature findings. The holonic HCPS aimed to improve the representation, communication and interfacing for physical components (including humans) and processes in the ambulatory case by elevating the coordination of interactions between system components to the cyber layer. The implementation used the ARTI architecture and BASE architecture.

The development of the HCPS was guided by a holonic system design process, which emphasises the necessity for practical design steps. This process was developed due to a lack of HCA implementation details available in the literature. The design process provided insight into the relevance and usefulness of the ARTI architecture for partitioning the holonic HCPS into “natural holons”, while the BASE architecture guided the design of the internal holon composition to enable the specific holon functionality. Furthermore, the holonic system design process reduced the development complexity and time for the case study implementation.

The RCI framework was presented to guide the improvement and evaluation of HSI in HCPSs. The RCI framework effectively encapsulates the considerations related to the goals of HSI developments. The RCI framework gave rise to the selection of holonic architectures and implementation technologies to satisfy the HSI goals within the HCPS. These goals aligned strongly with other architectures, applications and philosophies for CPS and HCPS, following I4.0 design principles.

The evaluation of the holonic HCPS demonstrated that the healthcare needs and requirements were satisfied in the case study. The system improved the workflow efficiencies, reduced workload and enhanced the overall human intelligence for both clinic personnel and patients. Furthermore, the evaluation provided evidence to argue the improved HSI maturity and benefit of the developed holonic HCPS. This can be attributed to the guidance of the RCI framework and the holonic design process.

While this thesis provided notable contributions, future research and development could better improve HSI in healthcare. Such future developments include:

- **Development of multi-modal interfacing mechanisms:** Interfacing, as part of the RCI framework, was not considered in-depth during the development of the holonic HCPS. Interfacing selection and development involves applying and considering a range of behavioural, social and ergonomic factors to realise benefits. Therefore, future work should consider the influence of interfacing mechanisms on these factors and improve the selection process according to the RCI framework.

- **Development of Resource-Activity Interactions**: Resource-Resource and Activity-Resource nested functional interactions were considered in this thesis. Resource-Activity nested functional interactions, where a Resource supervises an Activity holon, should be explored further. An example of such interaction for healthcare would be the spirometry device RI creating a maintenance AI to inspect the spirometry machinery over its operational lifetime.

- **Refinement of the Holonic System Design Process**: This thesis presented the first application of the presented holonic system design process. Further work should apply the process to more case study developments to gain insight into the appropriateness and usefulness of the framework.

- **Development of Data Security Mechanisms**: In healthcare, data security and protection remain a high priority due to sensitive patient data. Therefore, further work should develop a data security mechanism for the communication and representation aspects in the system.

- **Development and Evaluation of a Large-Scale Distributed Holonic HCPS**: The BASE architecture and Erlang support applications in distributed networks. Further work should explore the potential challenges when attempting to scale the holonic HCPS across multiple facilities and hospitals with different digital systems through a distributed network. This would also require testing in more realistic and dynamic environment to evaluate the potential benefit of the Holonic HCPS system.

Many HSI challenges persist for HCPS and must be overcome to realise the full potential of I4.0. The contributions in this thesis aid in overcoming some of these HSI technological barriers. Considering the work presented in this thesis and future work, improved HSI developments can truly reshape labour-intensive environments to fulfil I4.0 goals and improve human contribution and well-being.

9 References

Aceto, G., Persico, V. & Pescapé, A. 2020. Industry 4.0 and Health: Internet of Things, Big Data, and Cloud Computing for Healthcare 4.0. *Journal of Industrial Information Integration*. 18(1):100129.

Adeyeri, M.K., Mpofu, K. & Adenuga Olukorede, T. 2015. Integration of agent technology into manufacturing enterprise: A review and platform for industry 4.0. *IEOM 2015 - 5th International Conference on Industrial Engineering and Operations Management, Proceeding*. 1-10. Dubai, UAE

Becker, T. & Stern, H. 2016. Future Trends in Human Work area Design for Cyber-Physical Production Systems. *Procedia CIRP*. 57:404–409.

Beier, G., Ullrich, A., Niehoff, S., Reißig, M. & Habich, M. 2020. Industry 4.0: How it is defined from a sociotechnical perspective and how much sustainability it includes – A literature review. *Journal of Cleaner Production*. 259:120856.

Boy, G.A. 2020. *Human-systems integration: from virtual to tangible*. CRC Press.

Cañadas-De la Fuente, G.A., Vargas, C., San Luis, C., García, I., Cañadas, G.R. & De la Fuente, E.I. 2015. Risk factors and prevalence of burnout syndrome in the nursing profession. *International Journal of Nursing Studies*. 52(1):240–249.

CEN/CENELEC/ETSI Joint Working Group on Standards for Smart Grids. 2012. *CEN-CENELEC-ETSI Smart Grid Coordination Group: Smart Grid Information Security*. [Online], Available: ftp://ftp.cen.eu/EN/EuropeanStandardization/HotTopics/SmartGrids/Security.pdf [2022, July 28].

Centers for Disease Control and Prevention (CDC). (in press). Respiratory Health Bronchodilator Procedures Manual. *NHANES manual*. (January):http://www.cdc.gov/nchs/data/nhanes/nhanes_07_08/b.

Christensen, J. 1994. Holonic Manufacturing Systems: Initial Architecture and Standards Directions. *Holonic Manufacturing Systems*. (March):1–20.

Cimini, C., Pirola, F., Pinto, R. & Cavalieri, S. 2020. A human-in-the-loop manufacturing control architecture for the next generation of production systems. *Journal of Manufacturing Systems*. 54(July 2019):258–271.

Defty, T., Kruger, K. & Basson, A. 2022. Classification of Technical Challenges to Human-System Integration in Cyber-Physical Systems. *Studies in*

Computational Intelligence. 1034:175–186.

Department of Labour. 2019. Labour Market Annual Bulletin. [Online], Available: http://www.labour.gov.za/DocumentCenter/Reports/Annual Reports/Labour Market Bulletin Report/2019 -2020/Annual Labour Market Bulletin Report 2019.pdf [2022, August 04].

Derigent, W., Cardin, O. & Trentesaux, D. 2021. Industry 4.0: contributions of holonic manufacturing control architectures and future challenges. *Journal of Intelligent Manufacturing*. 32(7):1797–1818.

Dey, N., Ashour, A.S., Shi, F., Fong, S.J. & Tavares, J.M.R.S. 2018. Medical cyber-physical systems: A survey. *Journal of Medical Systems*. 42(4):1-13.

Duffie, N.A. 1990. Synthesis of Heterarchical manufacturing systems. *Computers in Industry*. 14(1–3):167–174.

Emanuele, J., & Koetter, L.E. 2007. Workflow opportunities and challenges in healthcare. *BPM & Workflow Handbook 1:157*.

Foundation For Intelligent Physical Agents. 2002. FIPA Contract Net Interaction Protocol Specification [Online], Available: http://www.fipa.org/specs/fipa00029/SC00029H.html [2022, August 10].

Gabor, T., Belzner, L., Kiermeier, M., Beck, M.T. & Neitz, A. 2016. A simulation-based architecture for smart cyber-physical systems. *Proceedings - 2016 IEEE International Conference on Autonomic Computing, ICAC 2016*. pp. 374–379.

Ghobakhloo, M. 2018. The future of manufacturing industry: a strategic roadmap toward Industry 4.0. *Journal of Manufacturing Technology Management*. 29(6):910–936.

Hankinson, J.L., Odencrantz, J.R. & Fedan, K.B. 1999. Spirometric reference values from a sample of the general U.S. Population. *American Journal of Respiratory and Critical Care Medicine*. 159(1):179–187.

Hart, S.G. & Staveland, L.E. 1988. Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research. *Advances in Psychology*. 52(C):139–183.

Kahlen, F.J., Flumerfelt, S. & Alves, A. 2016. Digital Twin: Mitigating Unpredictable, Undesirable Emergent Behavior in Complex Systems. *Transdisciplinary Perspectives on Complex Systems: New Findings and Approaches*. (August 2017):1–327.

Koestler, A. 1967. *The Ghost in the Machine*. Penguin Group.

Kruger, K. & Basson, A.H. 2019. Evaluation of JADE multi-agent system and Erlang holonic control implementations for a manufacturing cell. *International Journal of Computer Integrated Manufacturing*. 32(3):225–240.

Le, X.H., Lee, S., Truc, P.T.H., Vinh, L., Khattak, A.M., Han, M., Dang, V.H., Hassan, M.M. 2010. Secured WSN-integrated cloud computing for u-Life Care. in *2010 7th IEEE Consumer Communications and Networking Conference, CCNC 2010* IEEE.pp.1-2. Las Vegas, USA.

Lee, I., Sokolsky, O., Chen, S., Hatcliff, J., Jee, E., Kim, B., King, A., Mullen-Fortino, M. 2012. Challenges and research directions in medical cyber-physical systems. *Proceedings of the IEEE*. 100(1):75–90.

Lee, J., Bagheri, B. & Kao, H.A. 2015. A Cyber-Physical Systems architecture for Industry 4.0-based manufacturing systems. *Manufacturing Letters*. 3:18–23.

Leitão, P. & Restivo, F. 2006. ADACOR: A holonic architecture for agile and adaptive manufacturing control. *Computers in Industry*. 57(2):121–130.

Leitão, P., Casais, F. & Restivo, F. 2005. Holonic manufacturing control: A practical implementation. *IFIP Advances in Information and Communication Technology*. 159:33–44.

Likert, R. 1932. A technique for the measurement of attitudes. *Archives of Psychology*. 22(140):55.

Lopetegui, M., Yen, P.Y., Lai, A., Jeffries, J., Embi, P. & Payne, P. 2014. Time motion studies in healthcare: What are we talking about? *Journal of Biomedical Informatics*. 49:292–299.

Maphumulo, W.T. & Bhengu, B.R. 2019. Challenges of quality improvement in the healthcare of South Africa post-apartheid: A critical review. *Curationis*. 42(1).

Millot, P. & Lemoine, M.P. 1998. Attempt for generic concepts toward human-machine cooperation. *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*. 1:1044–1049.

National Department of Health. 2019. *National Digital Health Stratergy for South Africa 2019-2024*. [Online], Available: www.health.gov.za.

National planning commision. 2012. *National Development Plan 2030 | South African Government*. [Online], Available: https://www.gov.za/issues/national-development-plan-2030 [2021, November 30].

Neumann, W.P., Winkelhaus, S., Grosse, E.H. & Glock, C.H. 2021. Industry 4.0

and the human factor – A systems framework and analysis methodology for successful development. *International Journal of Production Economics*. 233(September 2020):107992.

Orth, H., Dreyer, Z.S., Makgotlho, E., Oosthuysen, W., Sinha, B., Wasserman, E., Orth, H., Makgotlho, E., et al. 2013. Tygerberg hospital. *Southern African Journal of Epidemiology and Infection*. 28(1):22–27. [Online], Available: http://www.sunorthopaedics.com/tygerberg-hospital.html [2021, November 26].

Oztemel, E. & Gursev, S. 2020. Literature review of Industry 4.0 and related technologies. *Journal of Intelligent Manufacturing*. 31(1):127–182.

Pacaux-Lemoine, M.P., Trentesaux, D., Zambrano Rey, G. & Millot, P. 2017. Designing intelligent manufacturing systems through Human-Machine Cooperation principles: A human-centered approach. *Computers and Industrial Engineering*. 111:581–595.

Pang, Z., Yang, G., Khedri, R. & Zhang, Y.T. 2018. Introduction to the Special Section: Convergence of Automation Technology, Biomedical Engineering, and Health Informatics Toward the Healthcare 4.0. *IEEE Reviews in Biomedical Engineering*. 11:249–259.

Pivoto, D.G.S., de Almeida, L.F.F., da Rosa Righi, R., Rodrigues, J.J.P.C., Lugli, A.B. & Alberti, A.M. 2021. Cyber-physical systems architectures for industrial internet of things applications in Industry 4.0: A literature review. *Journal of Manufacturing Systems*. 58:176–192.

Redelinghuys, A.J.H., Basson, A.H. & Kruger, K. 2020. A six-layer architecture for the digital twin: a manufacturing case study implementation. *Journal of Intelligent Manufacturing*. 31(6):1383–1402.

Redelinghuys, A.J.H., Kruger, K. & Basson, A. 2020. A six-layer architecture for digital twins with aggregation. *Studies in Computational Intelligence*. 853(January):171–182.

Romero, D., Bernus, P., Noran, O., Stahre, J. & Berglund, Å.F. 2016. The operator 4.0: Human cyber-physical systems & adaptive automation towards human-automation symbiosis work systems. in *IFIP Advances in Information and Communication Technology* Vol. 488.pp 677–686.

Roscoe, A.H.A. & Ellis, G.A. 1990. A subjective rating scale for assessing pilot workload in flight: A decade of practical use. *Royal Aircraft Establishment TR 90019*. (March, 1). [Online], Available: https://apps.dtic.mil/sti/citations/ADA227864 [2022, August 18].

Rosenbloom, S.T., Denny, J.C., Xu, H., Lorenzi, N., Stead, W.W. & Johnson, K.B. 2011. Data from clinical notes: A perspective on the tension between structure and flexible documentation. *Journal of the American Medical Informatics Association*. 18(2):181–186.

Ruppert, T., Jaskó, S., Holczinger, T. & Abonyi, J. 2018. Enabling technologies for operator 4.0: A survey. *Applied Sciences (Switzerland)*. 8(9):1–19.

Rüßmann, M., Lorenz, M., Gerbert, P., Waldner, M., Justus, J., Engel, P. & Harnisch, M. 2015. Industry 4.0: The Future of Productivity and Growth in Manufacturing Industries.[Online], Available: https://www.bcg.com/publications/2015/engineered_products_project_business_industry_4_future_productivity_growth_manufacturing_industries [2022, August 25].

Schluse, M. & Rossmann, J. 2016. From Simulation to Experimentable Digital Twins. *IEEE International Symposium on Systems Engineering*. pp.1–6. Edinburg, United Kingdom.

Schumacher, A., Erol, S. & Sihn, W. 2016. A Maturity Model for Assessing Industry 4.0 Readiness and Maturity of Manufacturing Enterprises. *Procedia CIRP*. 52:161–166.

SEBoK. 2021a. *Human Systems Integration*. [Online], Available: https://www.sebokwiki.org/w/index.php?title=Human_Systems_Integration&oldid=65101 [2022, August 25].

SEBoK. 2021b. *Origins of the Systems Approach*. [Online], Available: https://www.sebokwiki.org/w/index.php?title=Origins_of_the_Systems_Approach&oldid=65210 [2022, August 25].

SEBoK. 2021c. *System Validation*. [Online], Available: https://www.sebokwiki.org/w/index.php?title=System_Validation&oldid=65702 [2022, August 25].

Sendelbach, S. & Funk, M. 2013. Alarm fatigue: A patient safety concern. *AACN Advanced Critical Care*. 24(4):378–386.

Shortliffe, E.H. & Cimino, J.J. 2014. Biomedical informatics: Computer applications in health care and biomedicine: Fourth edition. *Biomedical Informatics: Computer Applications in Health Care and Biomedicine: Fourth Edition*. 1–965.

Simon, H.A. 1997. *Administrative Behavior: a Study of Decision-Making Processes in Administrative Organizations*. 4th ed. New York: New York Free Press.

Sony, M. & Naik, S. 2020. Industry 4.0 integration with socio-technical systems theory: A systematic review and proposed theoretical model. *Technology in Society*. 61(April):101248.

Sowe, S.K., Simmon, E., Zettsu, K., De Vaulx, F. & Bojanova, I. 2016. Cyber-Physical-Human Systems: Putting People in the Loop. *IT Professional*. 18(1):10–13.

Sparrow, D.E. 2020. The BASE architecture for the integration of human workers into an Industry 4.0 environment. PhD thesis, Stellenbosh University.

Sparrow, D., Kruger, K. & Basson, A. 2020. Activity lifecycle description for communication in human-integrated industry 4.0 environments. *Studies in Computational Intelligence*. 853:85–97.

Sparrow, D., Taylor, N., Kruger, K., Basson, A. & Bekker, A. 2021. Interfacing with Humans in Factories of the Future: Holonic Interface Services for Ambient Intelligence Environments. *Studies in Computational Intelligence* 952: 299–312.

Sparrow, D.E., Kruger, K. & Basson, A.H. 2021. An architecture to facilitate the integration of human workers in Industry 4.0 environments. *International Journal of Production Research*. 60(15):4778-4796

Taylor, N., Kruger, K. & Bekker, A. 2020. Mariner 4.0: Introducing Seafarers into the Maritime 4.0 Environment. in *Royal Institution of Naval Architects Ice Class Vessels Conference*.

Trentesaux, D. & Millot, P. 2016. A human-centred design to break the myth of the “magic human” in intelligent manufacturing systems. *Studies in Computational Intelligence*. 640(March):103–113.

De Ugarte, B.S., Artiba, A. & Pellerin, R. 2009. Manufacturing execution system - A literature review. *Production Planning and Control*. 20(6):525–539.

Valckenaers, P. 2019. ARTI reference architecture – PROSA revisited. in *Studies in Computational Intelligence* Vol. 803. Springer Verlag. 1–19.

Valckenaers, P. & Van Brussel, H. 2015. *Design for the unexpected: From holonic manufacturing systems towards a humane mechatronics society*. Butterworth-Heinemann.

Van Brussel, H., Wyns, J., Valckenaers, P., Bongaerts, L. & Peeters, P. 1998. Reference architecture for holonic manufacturing systems: PROSA. *Computers in Industry*. 37(3):255–274.

Van Niekerk, D.J., 2021. Extending the BASE architecture for complex and reconfigurable Cyber-Physical Systems using holonic principles. Masters Thesis. Stellenbosch University.

Wang, S., Wan, J., Li, D. & Zhang, C. 2016. Implementing Smart Factory of Industrie 4.0: An Outlook. *International Journal of Distributed Sensor Networks*. 2016.

Wasserman, A., 2022. ARTI-based Holonic Control Implementation for a Manufacturing System Using the BASE Architecture.Masters Thesis. Stellenbosch University

Western Cape Government. 2017. *Tygerberg Hospital: Overview | Western Cape Government*. [Online], Available: https://www.westerncape.gov.za/your_gov/153 [2021, November 15].

Yilma, B.A., Panetto, H. & Naudet, Y. 2021. Systemic formalisation of Cyber-Physical-Social System (CPSS): A systematic literature review. *Computers in Industry*. 129:103458.

Zezulka, F., Marcon, P., Vesely, I. & Sajdl, O. 2016. Industry 4.0 – An Introduction in the phenomenon. *IFAC-PapersOnLine*. 49(25):8–12.

Zhou, J., Zhou, Y., Wang, B. & Zang, J. 2019. Human–Cyber–Physical Systems (HCPSs) in the Context of New-Generation Intelligent Manufacturing. *Engineering*. 5(4):624–636.

Appendix A Ambulatory Clinic Observations

This section presents additional information regarding the observations and needs analysis for the respiratory ambulatory clinic at Tygerberg hospital.

A.1 Patient Process Flow Diagram

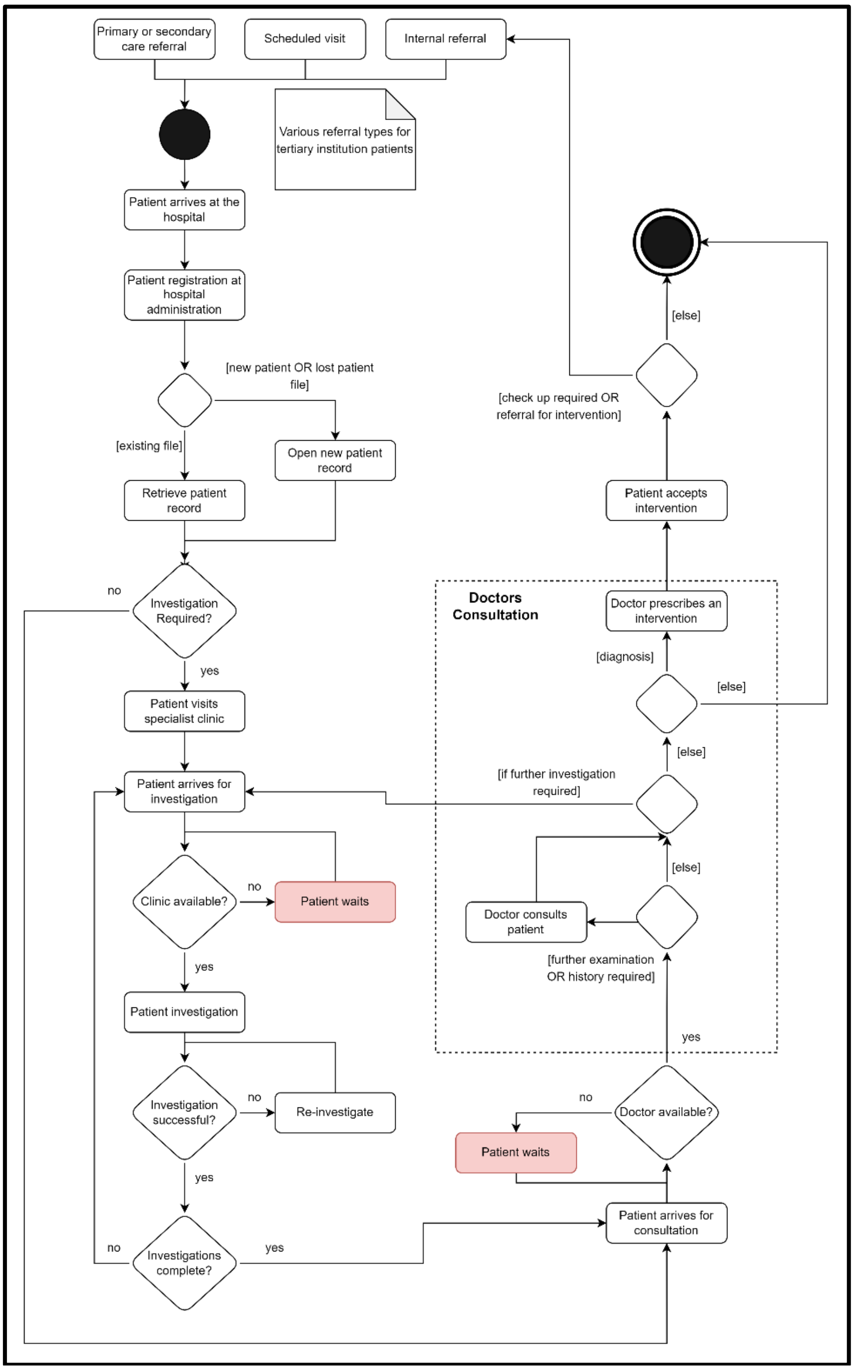


Figure 33: Ambulatory clinic process flow diagram.

A.2 Physical Data Flow Diagrams

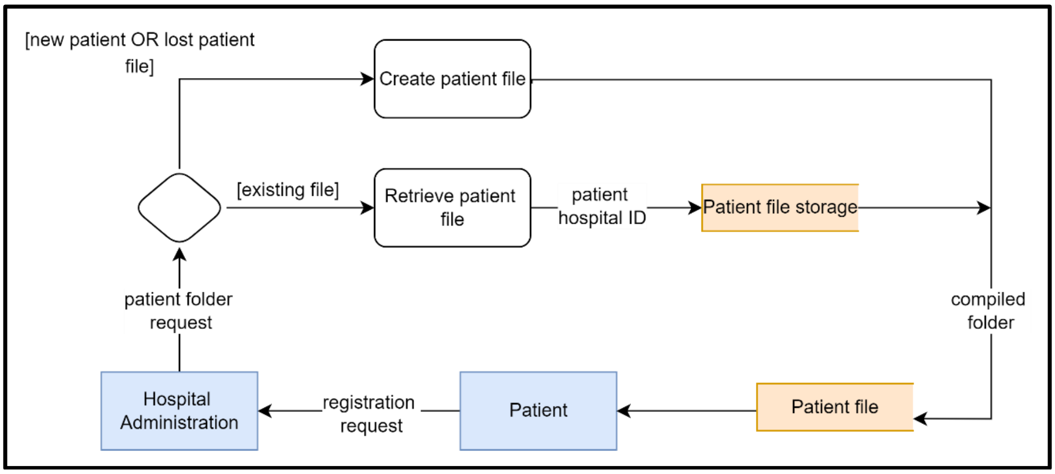


Figure 34: Data flow diagram of the patient registration process.

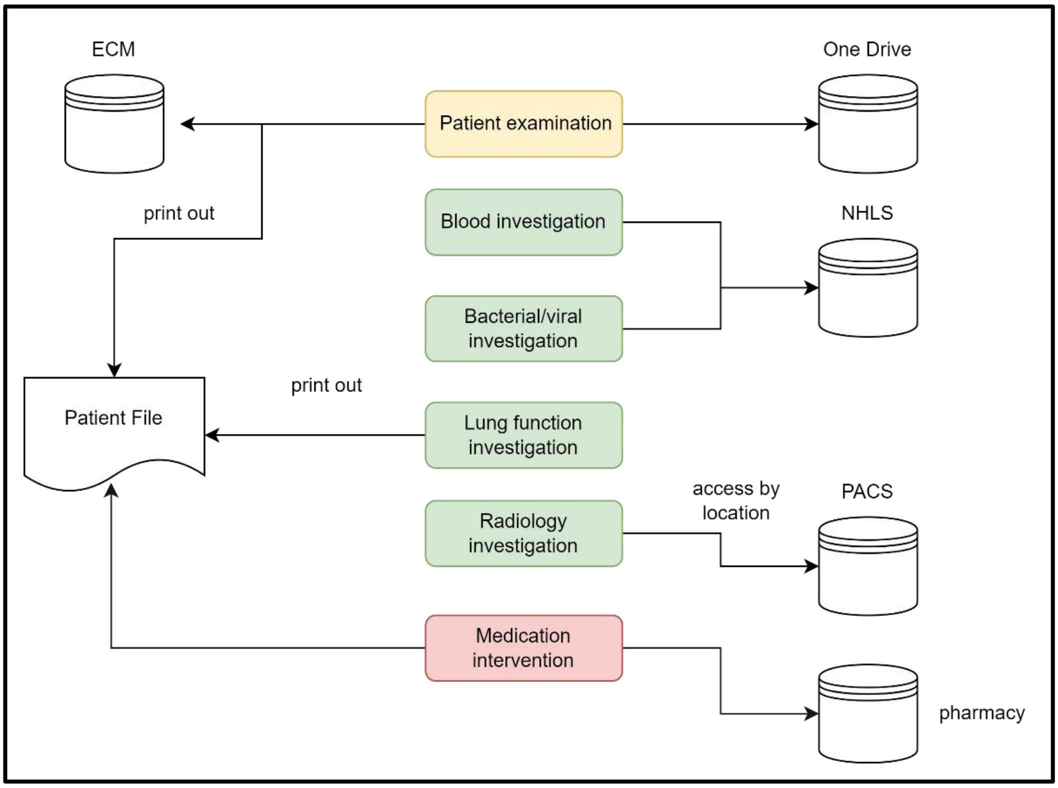


Figure 35: Ambulatory data storage types.

A.3 Generic Needs for Ambulatory Care Services

The generalised ambulatory clinic needs are presented in Table 13. These needs were expressed by users and gathered from the reviewed literature of healthcare facilities and practices.

Table 13: Needs analysis for outpatient clinical activities.

ID	Need
N1	Adapt to the stochastic nature of patients
N2	Handle emergency medical events
N3	Reduce data capture and entry errors
N4	Improve the utilisation of personnel, equipment and facilities
N5	Reduce patient and personnel waiting times
N6	Reduce time to retrieve patient data from different sources
N7	Reduce time to analyse patient data for diagnosis

HPs are often burdened with administrative logistic processes caused by patients arriving to appointments late or never. HPs and facilities need to prepare for the number of patients they expect to receive for various examination and investigation procedures. Due to constraints, the facility can only cater for a specific number of patients during certain shifts. With high patient volumes and demand, HPs aim to maximise the number of patients that can be treated per shift. When patients fail to arrive for their appointments, this results in reduced resource utilisation, where other patients could be treated. Hospital systems are often unable to react to the event when patients are not present for their appointments. Manual processes of detection and rescheduling are required to handle such events. The holonic HCPS must be capable of handling such events and automating the detection and rescheduling of patient appointments.

Emergency events are unpredictable, and HPs are required to respond immediately in most cases. The scheduling systems and manual processes HPs use during their care services often do not cater for this need. Therefore, the holonic HCPS should be able to handle such emergency events, in this case, an HP must shift to another task. Any other patients who might be affected should be handled accordingly. This might require patients to be re-scheduled for appointments while the doctor temporarily handles another emergency. Furthermore, all health-related processes and data captured before the emergency should be stored stored for when a doctor resumes the healthcare process.

HPs are required to process patient appointments swiftly to maintain a high throughput to handle the high volume of patients at healthcare facilities. HPs typically capture health-related data of the patient manually and on paper-based records. This mechanism of data capture coupled with the time constraints leads to data capture errors and the recording of misinformation. These can be detrimental to the care services provided by a facility. To avoid these errors, medical systems should improve the truthfulness and accuracy of data capturing. Such systems should have mechanisms and functions in place to check and prevent data errors. In doing so, the quality of healthcare services could be maintained.

The lack of financing and resource availability within healthcare facilities creates a demand for improved resource utilisation. Healthcare facilities, especially those in developing countries, often receive little funding for equipment and personnel to handle the high volume of patients and a variety of health conditions. The holonic HCPS must better coordinate personnel, equipment and patients to improve the utilisation rate of each during care services.

Similarly to improve the utilisation rate, the patient waiting time must be reduced especially in ambulatory clinic environments. The low availability of HPs to treat the high volume of patients leaves many patients waiting for long hours before being treated. The holonic HCPS must coordinate patients and scheduling with priority to ensure patients with severe conditions are treated in the shortest time possible.

Many data silos, holding various patient data, need to remain accessible to HPs. These data silos are typically accessed through websites and require unique credentials for access control. The accessibility of these websites is often hindered by network errors or when HPs forget their credentials. These issues delay the care services provided to patients. In some cases when certain websites are inaccessible, HPs are unable to continue with the care service without the proper information. Therefore, the holonic HCPS should reduce the delay for HPs to retrieve information necessary for the care services they provide. This entails automated access control and improved interfacing mechanisms.

Furthermore, the time spent analysing data by HPs often causes delays, since information is scattered on multiple pages in the patient file. Patient data and clinic notes are typically unstructured with little coherence between HPs. The holonic HCPS should automate data capture, formatting and analyse to improve the decision-making tasks for doctors and improve the overall efficiency of care services. This should lead to quicker and more accurate diagnoses for patients.

Appendix B Case Study Implementation

B.1 Consultation Activity Interactions

The FSM for the consultation activity is presented in Figure 36. The consultation AI handles all processes related to the patient examination and investigations, although investigations are handled more in-depth by another AI. The transition between states is triggered by *events* and logic decisions.

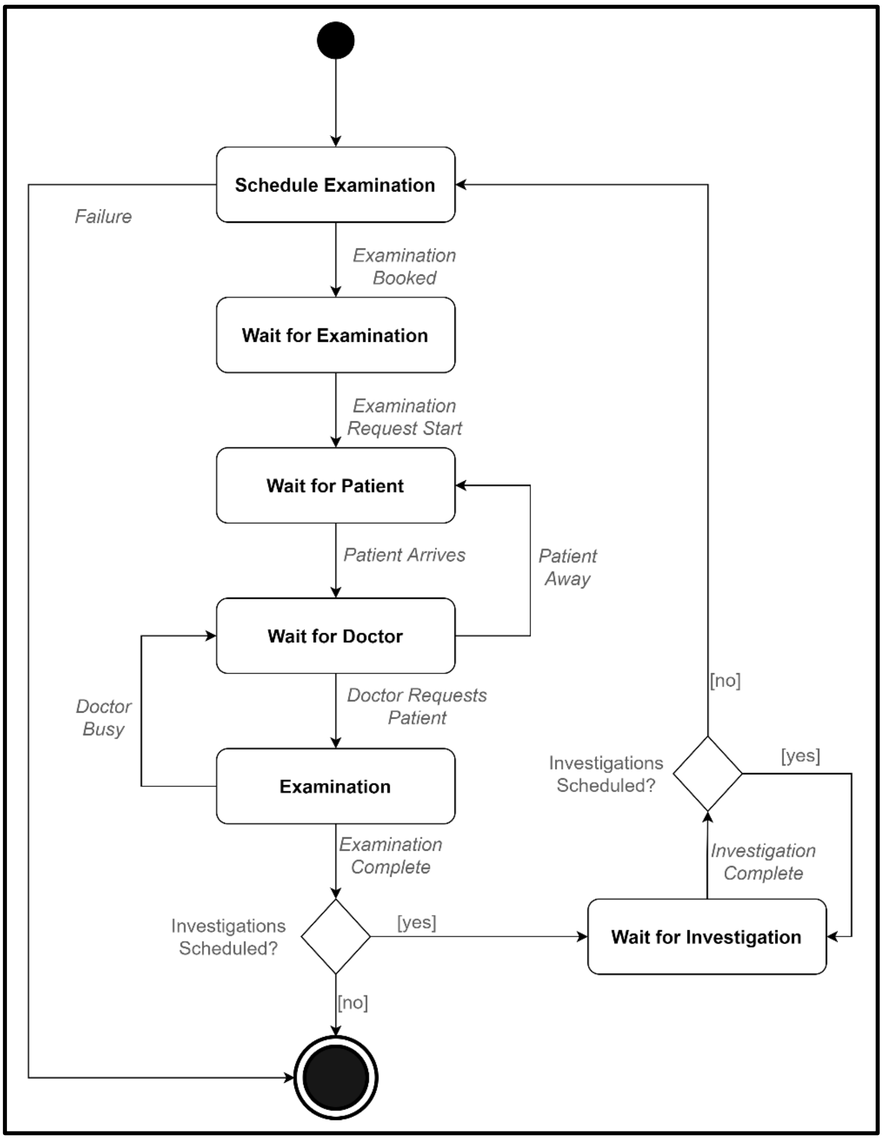


Figure 36: Consultation activity instance FSM.

Figure 37 and Figure 38 show the interactions between holons when a doctor creates a new consultation appointment resulting in a *Con_AI*. When the doctor wants to schedule a patient for a consultation appointment in the future, the doctor must first see what the scheduling requirements are for the consultation. Figure 37 shows the interactions when retrieving the requirements from the *Con_AT*. Firstly, the doctor must make use of their interfacing device to access their *Doc_RI* through the interface RI. The doctor will request (R) the scheduling requirements from the *Con_AT*, which will then reply by informing (I) the doctor. These requirements might entail the selection of the priority level, specified doctor for the appointment, the date, etc. Once these answers are specified by the doctor, these are sent back to the *Con_AT*, which will trigger the next set of interactions.

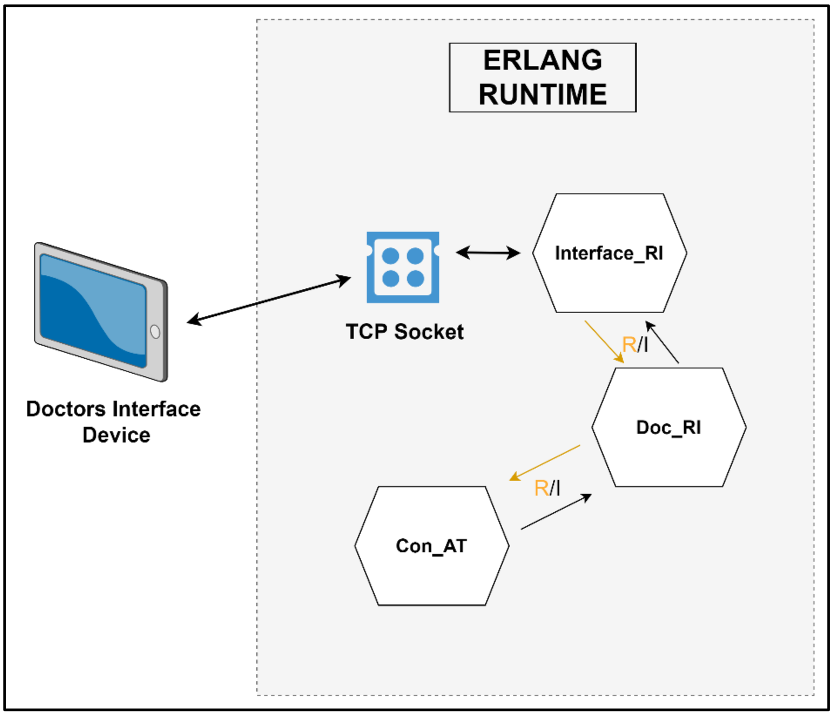


Figure 37: Consultation activity spawn requirements.

Figure 38 shows the interactions involved once the *Con_AT* receives the answered requirements for the appointment back from the doctor. The *Con_AT* will then spawn a new *Con_AI*, loading the correct attributes based on the required answers. The spawned *Con_AI* is then guided by its FSM control logic, presented in Figure 36, and the spawned attributes. The first state the *Con_AI* enters is the scheduling of the examination service between the patient, respiratory clinic and doctor. The *Con_AI* first sends out a CFP to all examination facility service providers, in this case, all the *RespC_RI*s. The *RespC_RIs* respond to the CFPs with a nested CNP by sending a CFP to all examination service providers, in this case, all *Doc_RIs*. Depending on their availability and other factors, the *Doc_RIs* provide proposals or rejection responses to the *RespC_RI* and so on. The *Con_AI* will select the best proposal from the respective *RespC_RI* and subsequently complete the nested CNP. The *Pat_RI* will always be requested for the allocated shift by the *Con_AI*.

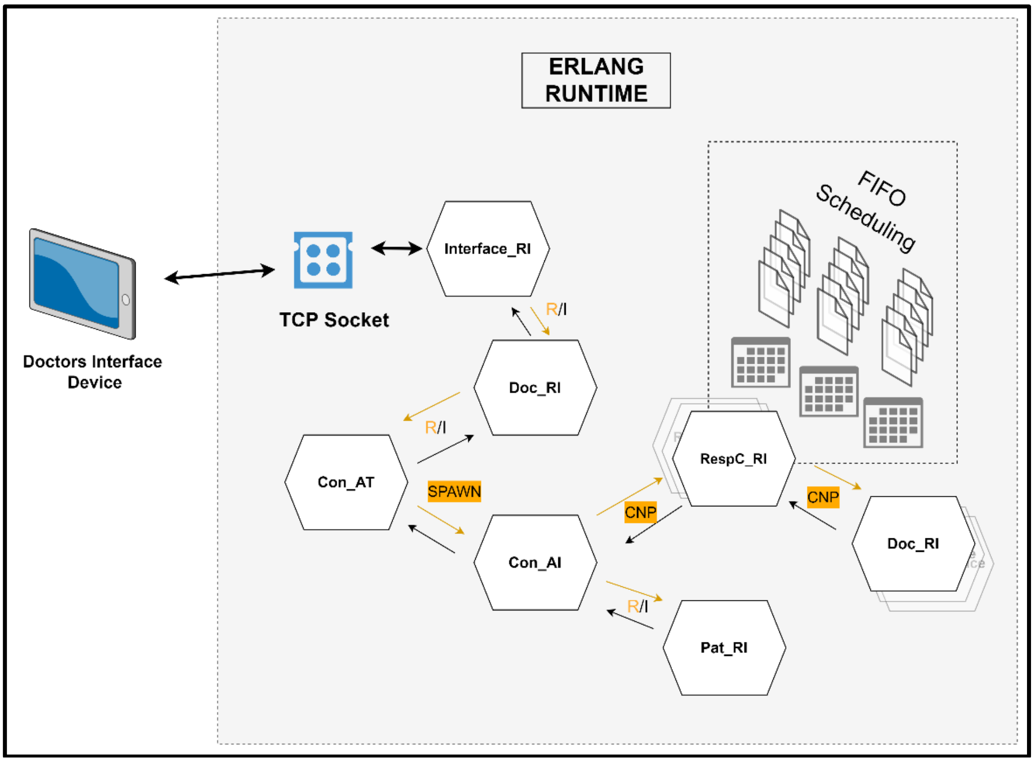


Figure 38: Consultation activity instance creation.

A snippet of the scheduling algorithm used by the *RespC_RI* for handling the scheduling with the *Con_AI* is presented below and aided with diagram in Figure 39. This is intended to show some of the complexity behind developing these interactions. This algorithm checks the schedule of the *RespC_RI* for an "open shift window" for the examination service by considering different shift times and the number of appointed doctors and patients for different shifts. Erlang is a functional programming language and does not allow for the reassignment of variables. Therefore, no *for* or *while loops* can be programmed, rather recursion needs to be used to implement "looped" functionality into the system.

The scheduling plugin takes in the service tag for the desired service, the scheduled time for the service (the centroid of the search window) and the estimated duration of the service. The search window has an incremental step interval between each recursion and a search width. The algorithm determines the day of the week that the centroid of the search window falls on and compares this to clinic's available shift window on that day of the week.

If there is available clinic shift window for that service (number 2 in Figure 39, the algorithm checks the clinics schedule for the number of doctors and patients already contracted for that shift. If there is still availability for the new patient, and sufficient doctors scheduled, then the scheduling plugin does not need to perform a nested CNP with a doctor. The service window is then successfully found and the normal scheduling service negotiation proceeds.

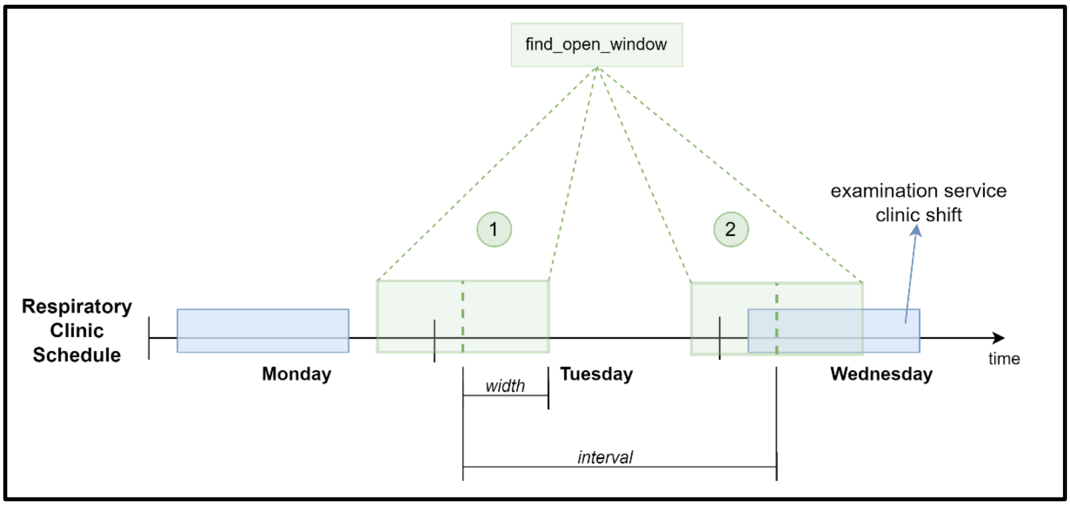


Figure 39: Scheduling algorithm for finding an available service shift.

If there is no clinic available clinic shift within the search window (number 1 in Figure 39) or in sufficient capacity for the new patient, the function recursion occurs and considers the next iteration, until it finds an available clinic shift which can offer the requested service.

The code for this algorithm is shown below:

```
find_open_window_new(ServiceTag, SchTime,Duration, CM, Cookie)->
  % establish the search window attributes
  Increment = 24*60*60*1000,    % search increment by 24 hours
  SearchInterval = 1000*60*60*12, % search window width -12 hours and +12
hours

  % convert the search time to day of the week to compare with the clinic
shift days
  {{Year,Month,Day},{Hour,Minute,Second}} =
calendar:system_time_to_universal_time(SchTime, 1000),
  DayofWeekNum = calendar:day_of_the_week(Year, Month, Day),
  ConsultDaysList = base:get_attribute(<<"CONSULT_DAYS">>,CM,Cookie), %
clinic shifts (attribute)

  % search through and compare the clinics available shift with current
no. of patients and doctors
  % booked for the shift already
  Result =  lists:foldl(fun(ConsultDay,Acc)->
    {ConsultDayofWeekNum,ConsultStartHour,ConsultEndHour} = ConsultDay,
    case (ConsultDayofWeekNum==DayofWeekNum) and (Acc==flagged) of
      true ->
        NumOfDoctors =
get_number_of_activities(SchTime,SearchInterval,<<"examine_doctor">>,CM,C
ookie),
        NumOfConsultAct =
```

```
get_number_of_activities(SchTime,SearchInterval,<<"resp_clinic">>,CM,Cook
ie) ,
        Ratio = case NumOfDoctors of
                  0-> 10000;
                  _->
                    (NumOfConsultAct)/(NumOfDoctors)
                end,

        % contract a new doctor if there are not enough doctors available
(nested CNP)
        io:format("~n [DEBUG] No of Doctors ~p No of Consults ~p
~n",[NumOfDoctors,NumOfConsultAct]),
        case (Ratio>3) of
          true->
            ShiftDetails =
{{Year,Month,Day},{ConsultStartHour,ConsultEndHour}},
            io:format("~n [INFO] A Doctor was contracted for the
Respiratory Clinic"),
            contract_doctor(ShiftDetails,ServiceTag,CM,Cookie),

            receive
              {doctor,WinChildPID,WinChildProp}->

{unflagged,{Year,Month,Day},{ConsultStartHour,ConsultEndHour},{NumOfConsu
ltAct}}
            after
              500->
                io:format("~n [INFO] Clinic didn't receive a msg from
child CNP"),
                Acc
            end;
          false ->
            io:format("~n [INFO] No doctor needs to be contracted"),

{unflagged,{Year,Month,Day},{ConsultStartHour,ConsultEndHour},{NumOfConsu
ltAct}}
        end;
      false->
        Acc
    end
              end, flagged, ConsultDaysList),

  case Result of

{unflagged,{Year1,Month1,Day1},{ConsultStartHour,ConsultEndHour},{Activit
iesNum}} ->
      SchTime1 =
(calendar:datetime_to_gregorian_seconds({{Year1,Month1,Day1},{ConsultStar
tHour,0,0}})-62167219200)*1000,
      Propose =
#{<<"ShiftStart">>=>SchTime1,<<"NumOfAct">>=>ActivitiesNum},
      Propose;
    OTHER ->
      % iterate the search window if no possibility was available
      find_open_window(ServiceTag,SchTime+Increment,Duration,CM,Cookie);
    _->
      error
  end.
```

Figure 40 shows the interactions between holons during the execution of the consultation process. At the start of the *RespC_RI* examination shift, the doctor will be required to start their examination shift with the clinic through their interface device. Once the examination shift with the clinic is started, the doctor will be able to request a patient from the respiratory clinic's queue.

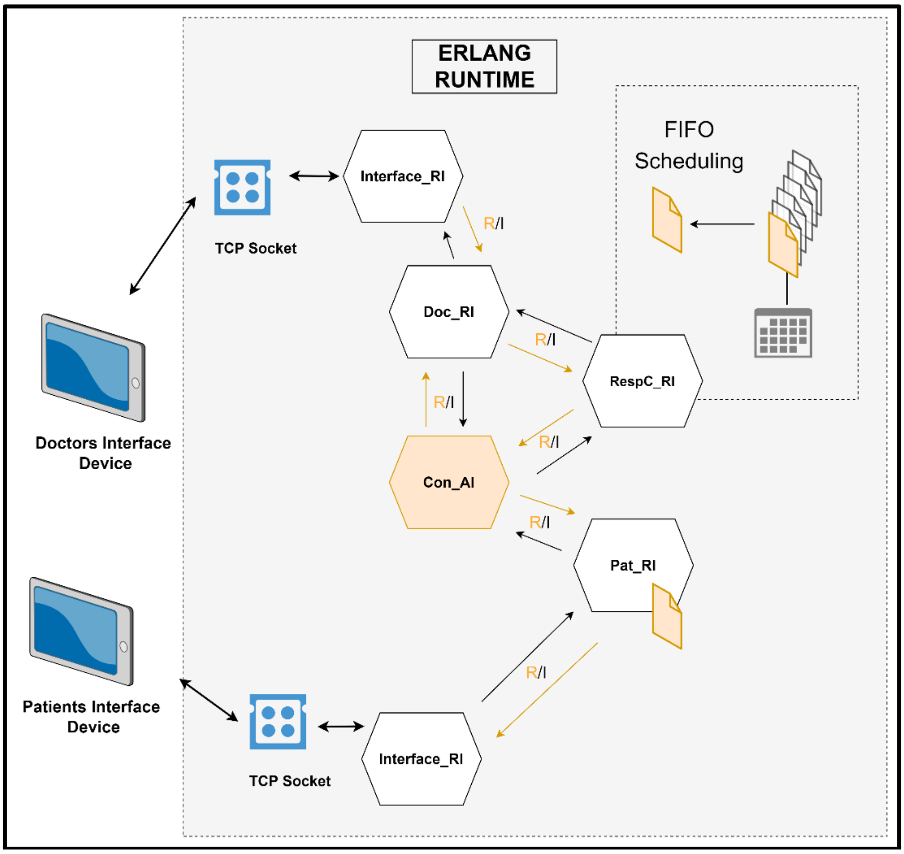


Figure 40: Consultation activity instance execution.

Patients can use the interfacing devices in the facility to log in and view the status of their *Pat_RI*. Once logged in, patients can announce themselves for various currently waiting activities or tasks. The patient will be able to see that the respiratory clinic is waiting for their arrival and be able to inform the clinic that they have arrived. Once informed the *Pat_RI* will communicate this information, through the appropriate *Con_AI* to the *RespC_RI*. The *RespC_RI* will then store the respective *Con_AI* into a state variable list to keep a FIFO record of the current queue.

The doctor can request a patient to examine from the clinic, through the interfacing device. The *Doc_RI* will then inform the *RespC_RI* of the doctor's intent. The *RespC_RI* will then select the next available *Con_AI* from the queue and inform the *Doc_RI* of this AI. The *Doc_RI* will then start an examination service with the respective *Con_AI*. After which the *Con_AI* will start an examination service with *Pat_RI*. All examination steps and data handling will be coordinated by the *Con_AI*. This promotes traceability and reduces the chance of error.

B.2 Spirometry Activity Interactions

Figure 41 presents the FSM diagram for the *Spir_AIs*.

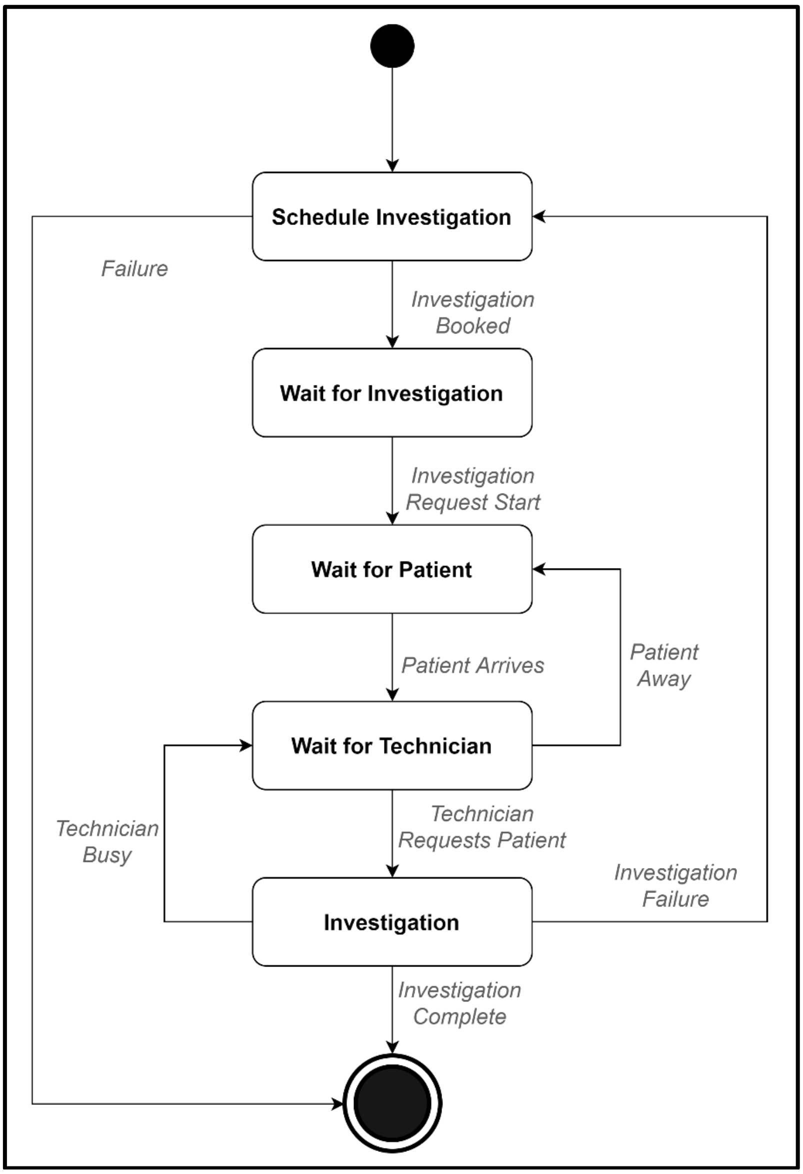


Figure 41: Spirometry activity instance FSM.

Figure 42 shows the interactions between holons for the creation of a *Spir_AI* during a consultation procedure. In the case study, the *Spir_AI* could only be created and scheduled by a doctor when involved in a consultation procedure to ensure that doctor was making an informed decision when scheduling the investigation and to ensure all decision-making was traced by the *Con_AI*.

The doctor uses their interfacing device to request the spawn requirements from the *Spir_AT*. Once answered the *Spir_AT* will spawn the *Spir_AI* with starting attributes. The FSM of the *Spir_AI* will then control the process steps of the AI, the first being to schedule a spirometry service with the *SpirC_RI*. Therefore, the *Spir_AI* will send out a CFP to all the service providers, in this case, *SpirC_RIs*. Once the CNP process is completed, the scheduled service will wait for the planned execution time and date.

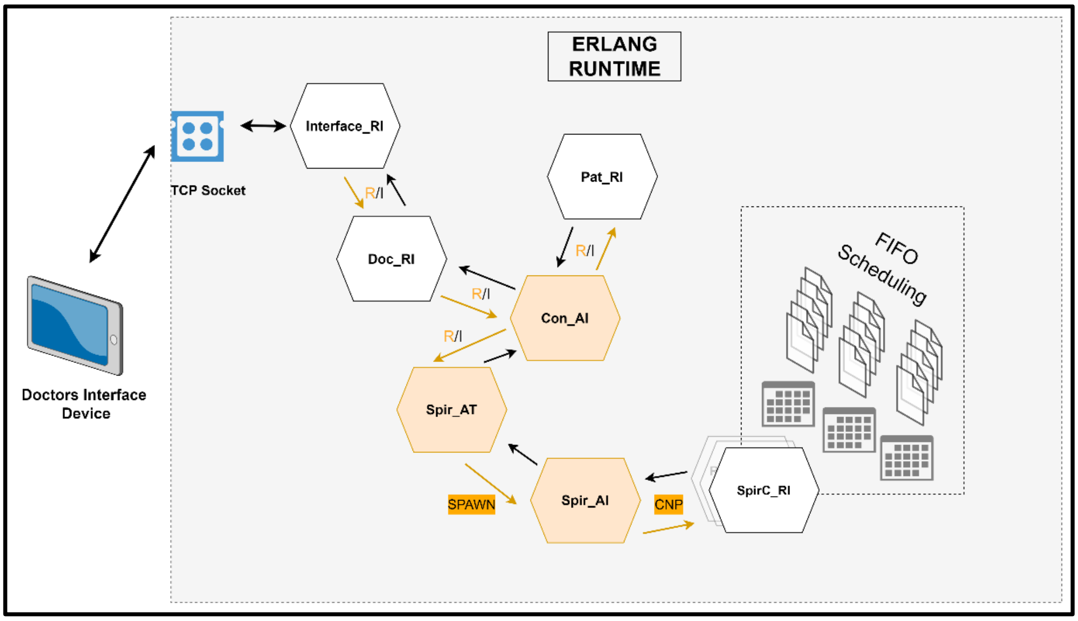


Figure 42: Spirometry activity instance creation.

Figure 43 shows the interaction between holons during the coordination of the *Spir_AI* execution of the spirometry testing service. The clinic personnel can select can request the next *Spir_AI* in the FIFO queue to begin the execution of the spirometry test.

The clinic personnel must also select a *SpirDev_RI* which has been linked with the *SpirC_RI*, through a service negotiation, meaning that it is positioned in the clinic and is available. The *Spir_AI* must be allocated to the *SpirDev_RI*, and the test can then be started, where patient information can be requested by the *SpirDev_RI* and communicated to the physical spirometry test machine through the interfacing mechanism. The results of the test can be communicated back to the *Pat_RI* through the *Spir_AI*, for traceability, and stored as data 2.

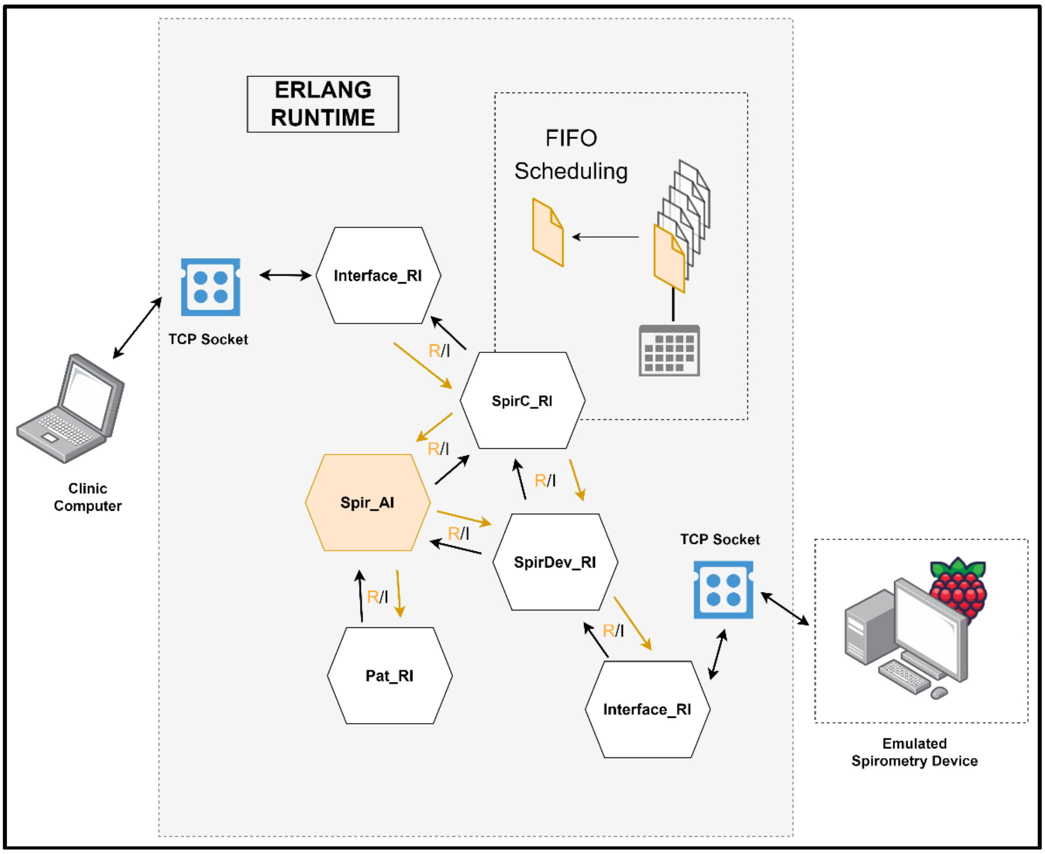


Figure 43: Spirometry activity instance execution.

B.3 Resource Interfaces

The following section shows screenshots of the various user interface displays developed for the case study.

Figure 44 shows the UI for the doctor when they want to schedule a consultation for a patient. The doctor can select which AT they want to request spawn requirements from, which then populate a list of requirements for the doctor to complete.

The requirements are structured according to an ontology which allows the interface to detect which modes of input are required for each requirement. As an example, the UI can recognise that the "PatientMRN" can be captured by scanning a QR code and thereby allows for that option. Other requirements might only require simple text or numeric inputs, or a drop-down list seen for the "Priority"

Name: TRAVIS

Status: Available

Schedule Patient

provide_consult_type

Activity Requirements

PatientMRN: BcafKJ24aCd

ICD-10 Conditions

Priority

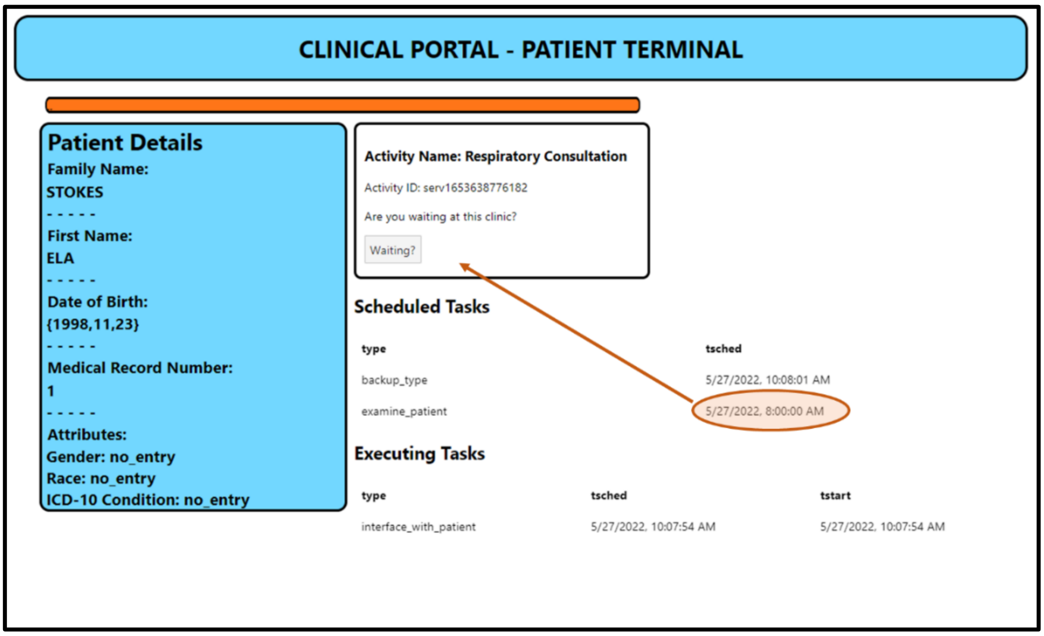
Description

Scheduled_Date

Figure 44: Screenshot of the *Doc_RI* interfacing website when scheduling a *Con_AI*.

The UI for the patient is shown in Figure 45. This patient was scheduled for an examination service with a respiratory clinic. Their respiratory clinic shift started at 8 a.m. which can be seen by the orange circle. Therefore, the UI prompts the patient asking them whether they are waiting for this shift. An interfacing service can be seen executing with the *Pat_RI*, this service is indicative of the current UI being displayed, which started at 10:07 a.m.

The specific *Pat_RI* attributes can be seen on the left, which can be reactively updated during the examination procedure or when any investigation results are communicated to the *Pat_RI*. In this case, currently, the patient is still waiting for their examination, so no gender, race or ICD-10 results have been captured



CLINICAL PORTAL - PATIENT TERMINAL

Patient Details
Family Name:
STOKES
- - - - -
First Name:
ELA
- - - - -
Date of Birth:
{1998,11,23}
- - - - -
Medical Record Number:
1
- - - - -
Attributes:
Gender: no_entry
Race: no_entry
ICD-10 Condition: no_entry

Activity Name: Respiratory Consultation
Activity ID: serv1653638776182
Are you waiting at this clinic?

Scheduled Tasks

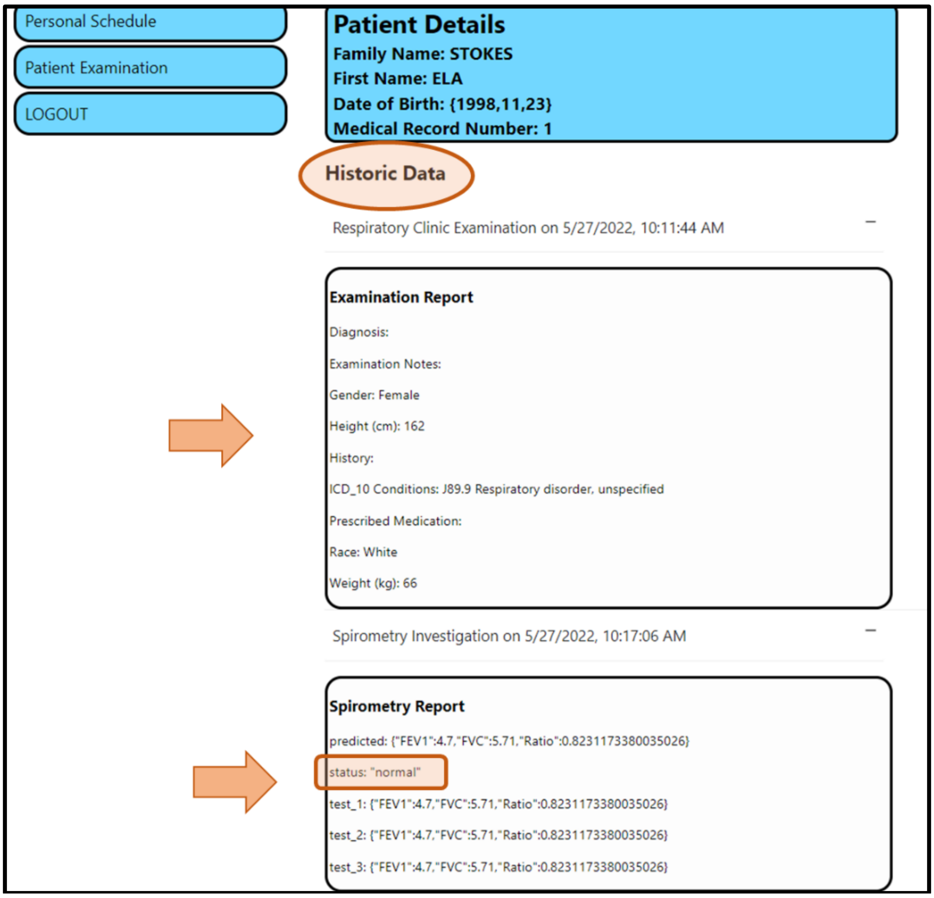
type	**tsched**
backup_type	5/27/2022, 10:08:01 AM
examine_patient	5/27/2022, 8:00:00 AM

Executing Tasks

type	**tsched**	**tstart**
interface_with_patient	5/27/2022, 10:07:54 AM	5/27/2022, 10:07:54 AM

Figure 45: Screenshot of the *Pat_RI* interfacing website when arriving at the clinic for an appointment.

Figure 46 shows a screenshot of a doctor's UI when viewing the historic data of the patient they are currently examining. The patient details are retrieved from the *Pat_RI*, through the *Con_AI* during the execution of the examination service. Furthermore, the doctor can retrieve any historic data from the *Pat_RIs* biography such as examination or investigation results.



Patient Details
Family Name: STOKES
First Name: ELA
Date of Birth: {1998,11,23}
Medical Record Number: 1

Historic Data

Respiratory Clinic Examination on 5/27/2022, 10:11:44 AM

Examination Report

Diagnosis:
Examination Notes:
Gender: Female
Height (cm): 162
History:
ICD_10 Conditions: J89.9 Respiratory disorder, unspecified
Prescribed Medication:
Race: White
Weight (kg): 66

Spirometry Investigation on 5/27/2022, 10:17:06 AM

Spirometry Report

predicted: {"FEV1":4.7,"FVC":5.71,"Ratio":0.8231173380035026}
status: "normal"
test_1: {"FEV1":4.7,"FVC":5.71,"Ratio":0.8231173380035026}
test_2: {"FEV1":4.7,"FVC":5.71,"Ratio":0.8231173380035026}
test_3: {"FEV1":4.7,"FVC":5.71,"Ratio":0.8231173380035026}

Figure 46: Screenshot of the *Doc_RI* interface website when viewing the *Pat_RI's* historic medical information.

B.4 Interface Devices

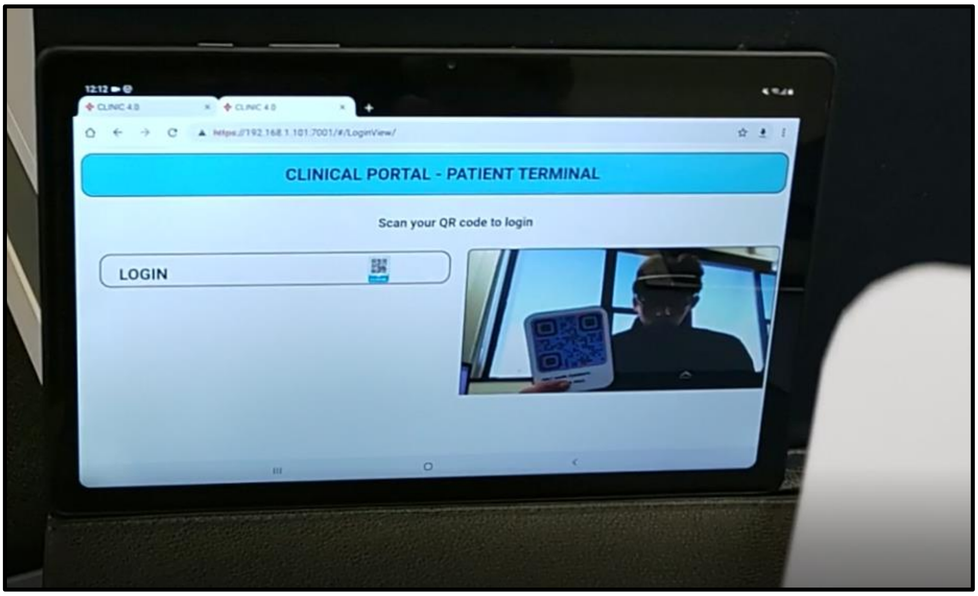
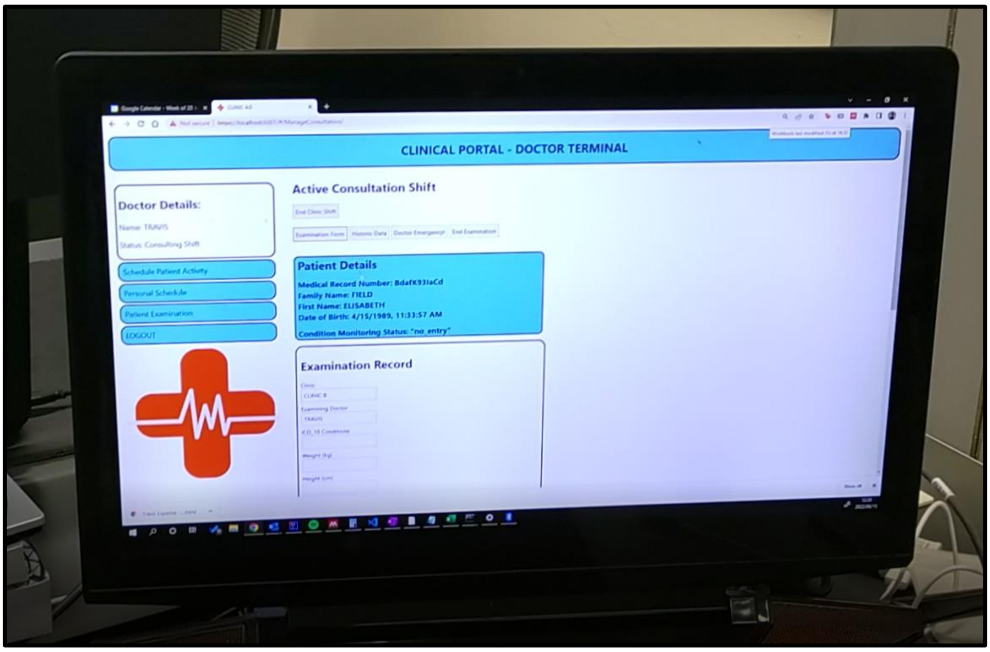


Figure 47: Image of the *Pat_RI's* interface using a tablet device.



Doctor Details:

Name: TRAVIS

Status: Consulting Shift

Patient Details

Medical Record Number: BdafK93IaCd
Family Name: FIELD
First Name: ELISABETH
Date of Birth: 4/15/1989, 11:33:57 AM

Condition Monitoring Status: "no_entry"

Clinic: CLINIC 8

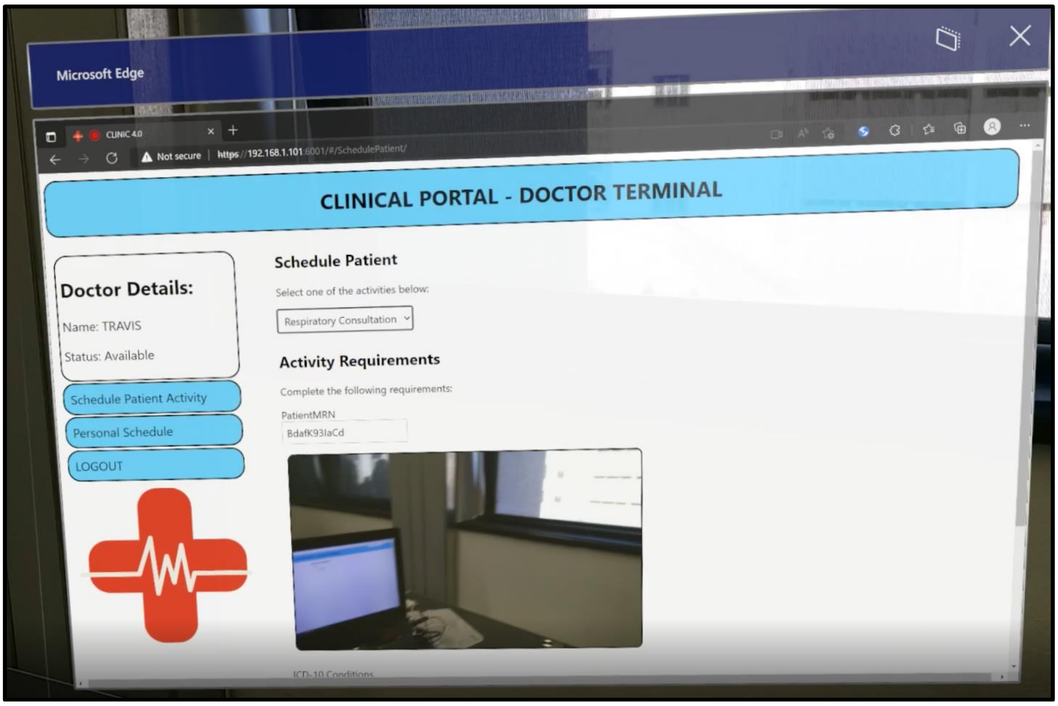
Examining Doctor: TRAVIS

ICD_10 Conditions

Weight (kg)

Height (cm)

Figure 48: Image of the *Doc_RI's* interface using a laptop device.



CLINICAL PORTAL - DOCTOR TERMINAL

Doctor Details:

Name: TRAVIS

Status: Available

Schedule Patient

Select one of the activities below:

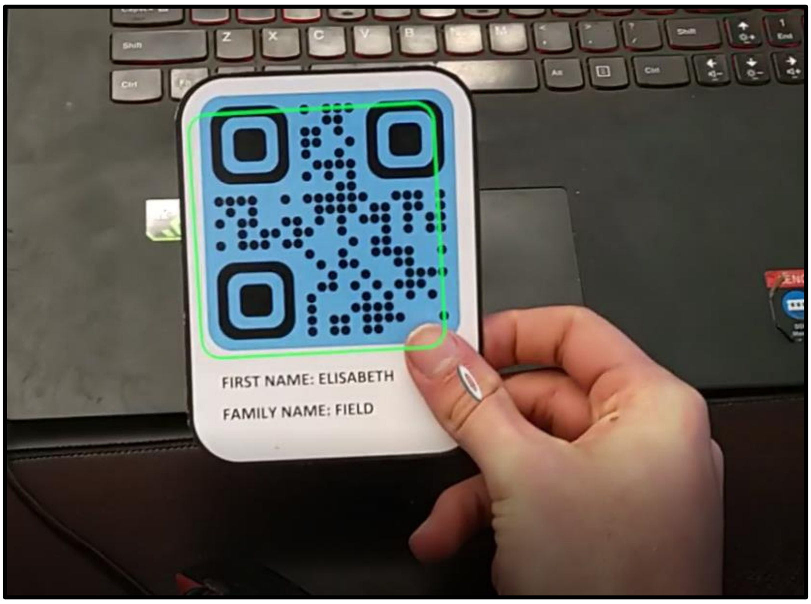
Respiratory Consultation

Activity Requirements

Complete the following requirements:

PatientMRN: BdafK93laCd

Figure 49: Screenshot of the *Doc_RI's* interface using a Microsoft HoloLens 2 augmented reality device.


FIRST NAME: ELISABETH

FAMILY NAME: FIELD

Figure 50: Screenshot of the QR scanning using the Microsoft HoloLens 2 augmented reality device.

B.5 NHANES III Spirometry Prediction Algorithm Implemented in Erlang

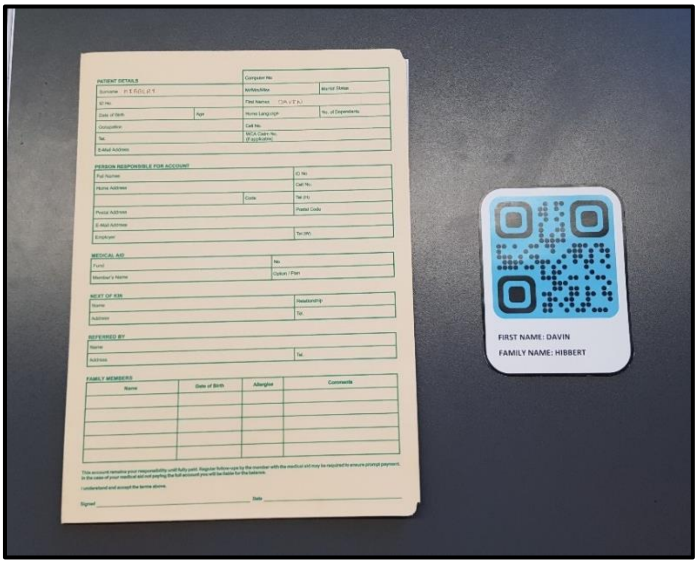
```
hankinson_ref_equation(Sex, Race, Age, Height)->
  io:format("~n [INFO] LF Device is running prediction hankinson
algorithm..."),
 Result =  case Sex of
    <<"Male">>->
      case Race of
        <<"Caucasian">>->
          case Age<20 of
            true->
              FEV1 = -0.7453 -0.04106*Age +
0.004477*(math:pow(Age,2)) + 0.00014098*(math:pow(Height,2)),
              FVC = -0.2584 -0.20415*Age +
0.010133*(math:pow(Age,2)) + 0.00018642*(math:pow(Height,2)),
              Ratio = 88.066 - 0.2066*Age ,
              FEV1_LNN = -0.7453 -0.04106*Age +
0.004477*(math:pow(Age,2)) + 0.00011607*(math:pow(Height,2)),
              FVC_LNN = -0.2584 -0.20415*Age +
0.010133*(math:pow(Age,2)) + 0.00015695*(math:pow(Height,2)),
              Ratio_LNN = 78.388 - 0.2066*Age ,

#{<<"predicted">>=>#{<<"FVC">>=>FVC,<<"FEV1">>=>FEV1,<<"Ratio">>=>
Ratio},<<"LNN_results">>=>#{<<"FVC">>=>FVC_LNN,<<"FEV1">>=>FEV1_LN
N,<<"Ratio">>=>Ratio_LNN}};
            false->
              FEV1 = 0.5536 -0.01303*Age -
0.000172*(math:pow(Age,2)) + 0.00014098*(math:pow(Height,2)),
              FVC = -0.1933 +0.00064*Age -
0.000269*(math:pow(Age,2)) + 0.00018642*(math:pow(Height,2)),
              Ratio = 88.066 - 0.2066*Age ,
              FEV1_LNN = 0.5536 -0.01303*Age -
0.000172*(math:pow(Age,2)) + 0.00011607*(math:pow(Height,2)),
              FVC_LNN = -0.1933 +0.00064*Age -
0.000269*(math:pow(Age,2)) + 0.00015695*(math:pow(Height,2)),
              Ratio_LNN = 78.388 - 0.2066*Age ,

#{<<"predicted">>=>#{<<"FVC">>=>FVC,<<"FEV1">>=>FEV1,<<"Ratio">>=>
Ratio},<<"LNN_results">>=>#{<<"FVC">>=>FVC_LNN,<<"FEV1">>=>FEV1_LN
N,<<"Ratio">>=>Ratio_LNN}};
            _->
              missing_patient_information
          end;
        <<"African-American">>->
          case Age<20 of
            true->
              FEV1 = -0.7048 -0.05711*Age +
0.004316*(math:pow(Age,2)) + 0.00013194*(math:pow(Height,2)),
              FVC = -0.4971 -0.15497*Age +
0.007701*(math:pow(Age,2)) + 0.00016643*(math:pow(Height,2)),
              Ratio = 89.239 - 0.1828*Age ,
              FEV1_LNN = -0.7048 -0.05711*Age +
0.004316*(math:pow(Age,2)) + 0.00010561*(math:pow(Height,2)),
              FVC_LNN = -0.4971 -0.15497*Age +
```

```
0.007701*(math:pow(Age,2)) + 0.00013670*(math:pow(Height,2)),
              Ratio_LNN = 78.822 - 0.1828*Age ,

#{<<"predicted">>=>#{<<"FVC">>=>FVC,<<"FEV1">>=>FEV1,<<"Ratio">>=>
Ratio},<<"LNN_results">>=>#{<<"FVC">>=>FVC_LNN,<<"FEV1">>=>FEV1_LN
N,<<"Ratio">>=>Ratio_LNN}};
            false->
              FEV1 = 0.3411 -0.02309*Age  +
0.00013194*(math:pow(Height,2)),
              FVC = -0.1517 -0.01821*Age  +
0.00016643*(math:pow(Height,2)),
              Ratio = 89.239 - 0.1828*Age ,
              FEV1_LNN = 0.3411 -0.02309*Age  +
0.00010561*(math:pow(Height,2)),
              FVC_LNN = -0.1517 -0.01821*Age  +
0.00013670*(math:pow(Height,2)),
              Ratio_LNN = 78.822 - 0.1828*Age ,

#{<<"predicted">>=>#{<<"FVC">>=>FVC,<<"FEV1">>=>FEV1,<<"Ratio">>=>
Ratio},<<"LNN_results">>=>#{<<"FVC">>=>FVC_LNN,<<"FEV1">>=>FEV1_LN
N,<<"Ratio">>=>Ratio_LNN}};
            _->
              missing_patient_information
          end;
        _->
          missing_patient_information
      end;
    <<"Female">>->
      case Race of
        <<"Caucasian">>->
          case Age<18 of
            true->
              FEV1 = -0.8710 +0.06537*Age +
0.00011496*(math:pow(Height,2)),
              FVC = -1.2082 +0.05916*Age  +
0.00014815*(math:pow(Height,2)),
              Ratio = 90.809 - 0.2125*Age ,
              FEV1_LNN = -0.8710 +0.06537*Age +
0.00009283*(math:pow(Height,2)),
              FVC_LNN = -1.2082 +0.05916*Age  +
0.00012198*(math:pow(Height,2)),
              Ratio_LNN = 91.015 - 0.2125*Age ,

#{<<"predicted">>=>#{<<"FVC">>=>FVC,<<"FEV1">>=>FEV1,<<"Ratio">>=>
Ratio},<<"LNN_results">>=>#{<<"FVC">>=>FVC_LNN,<<"FEV1">>=>FEV1_LN
N,<<"Ratio">>=>Ratio_LNN}};
            false->
              FEV1 = 0.4333 -0.00361*Age -
0.000194*(math:pow(Age,2)) + 0.00011496*(math:pow(Height,2)),
              FVC = -0.3560 +0.01870*Age -
0.000382*(math:pow(Age,2)) + 0.00014815*(math:pow(Height,2)),
              Ratio = 90.809 - 0.2125*Age ,
              FEV1_LNN = 0.4333 -0.00361*Age -
0.000194*(math:pow(Age,2)) + 0.00009283*(math:pow(Height,2)),
              FVC_LNN = -0.3560 +0.01870*Age -
0.000382*(math:pow(Age,2)) + 0.00012198*(math:pow(Height,2)),
```

```
              Ratio_LNN = 91.015 - 0.2125*Age ,

#{<<"predicted">>=>#{<<"FVC">>=>FVC,<<"FEV1">>=>FEV1,<<"Ratio">>=>
Ratio},<<"LNN_results">>=>#{<<"FVC">>=>FVC_LNN,<<"FEV1">>=>FEV1_LN
N,<<"Ratio">>=>Ratio_LNN}};
            _->
              missing_patient_information
          end;
        <<"African-American">>->
          case Age<18 of
            true->
              FEV1 = -0.9630 +0.05799*Age  +
0.00010846*(math:pow(Height,2)),
              FVC = -0.6166 -0.04687*Age +
0.003602*(math:pow(Age,2)) + 0.00013606*(math:pow(Height,2)),
              Ratio = 91.655 - 0.2039*Age ,
              FEV1_LNN = -0.9630 +0.05799*Age  +
0.00008546*(math:pow(Height,2)),
              FVC_LNN = -0.6166 -0.04687*Age +
0.003602*(math:pow(Age,2)) + 0.00010916*(math:pow(Height,2)),
              Ratio_LNN = 80.978 - 0.2039*Age ,

#{<<"predicted">>=>#{<<"FVC">>=>FVC,<<"FEV1">>=>FEV1,<<"Ratio">>=>
Ratio},<<"LNN_results">>=>#{<<"FVC">>=>FVC_LNN,<<"FEV1">>=>FEV1_LN
N,<<"Ratio">>=>Ratio_LNN}};
            false->
              FEV1 = 0.3433 -0.01283*Age -0.000097*Age +
0.00010846*(math:pow(Height,2)),
              FVC = -0.3039 +0.00536*Age -0.000265*Age +
0.00013606*(math:pow(Height,2)),
              Ratio = 91.655 - 0.2039*Age ,
              FEV1_LNN = 0.3433 -0.01283*Age -0.000097*Age +
0.00008546*(math:pow(Height,2)),
              FVC_LNN = -0.3039 +0.00536*Age -0.000265*Age +
0.00010916*(math:pow(Height,2)),
              Ratio_LNN = 80.978 - 0.2039*Age ,

#{<<"predicted">>=>#{<<"FVC">>=>FVC,<<"FEV1">>=>FEV1,<<"Ratio">>=>
Ratio},<<"LNN_results">>=>#{<<"FVC">>=>FVC_LNN,<<"FEV1">>=>FEV1_LN
N,<<"Ratio">>=>Ratio_LNN}};
            _->
              missing_patient_information
          end;
        _->
          missing_patient_information
      end;
    _->
      missing_patient_information
  end,
  io:format("~n [INFO] LF Device is finished running the
prediction hankinson algorithm..."),
  Result.
```

Appendix C Experimental Evaluation

C.1 Method



PATIENT DETAILS

Surname HIBBERT

ID No.

Date of Birth

Age

Occupation

Tel.

E-Mail Address

Computer No.

Mr/Mrs/Miss

Marital Status

First Names DAVIN

Home Language

No. of Dependants

Cell No.

WCA Claim No. (if applicable)

PERSON RESPONSIBLE FOR ACCOUNT

Full Names

Home Address

Code

Postal Address

E-Mail Address

Employer

ID No.

Cell No.

Tel (H)

Postal Code

Tel (W)

MEDICAL AID

Fund

Member's Name

No.

Option / Plan

NEXT OF KIN

Name

Address

Relationship

Tel.

REFERRED BY

Name

Address

Tel.

FAMILY MEMBERS

Name	Date of Birth	Allergies	Comments

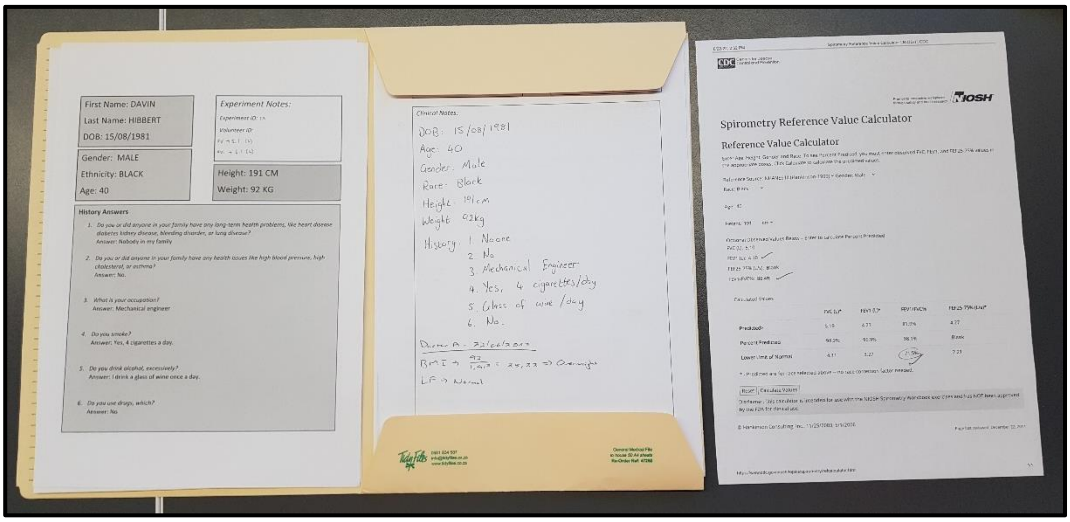
I understand and accept the terms above.

Signed ____________________ Date ____________________

FIRST NAME: DAVIN

FAMILY NAME: HIBBERT

Figure 51: Physical patient file and QR code for the experimental evaluation.



First Name: DAVIN

Last Name: HIBBERT

DOB: 15/08/1981

Gender: MALE

Ethnicity: BLACK

Age: 40

Experiment Notes:

Experiment ID: 13

Volunteer ID:

[illegible]

Height: 191 CM

Weight: 92 KG

History Answers

1. *Do you or did anyone in your family have any long-term health problems, like heart disease diabetes kidney disease, bleeding disorder, or lung disease?*
   Answer: Nobody in my family
2. *Do you or did anyone in your family have any health issues like high blood pressure, high cholesterol, or asthma?*
   Answer: No.
3. *What is your occupation?*
   Answer: Mechanical engineer
4. *Do you smoke?*
   Answer: Yes, 4 cigarettes a day.
5. *Do you drink alcohol, excessively?*
   Answer: I drink a glass of wine once a day.
6. *Do you use drugs, which?*
   Answer: No

Clinical Notes.

DOB: 15/08/1981

Age: 40

Gender: Male

Race: Black

Height: 191cm

Weight: 92kg

History: 1. Noone

2. No

3. Mechanical Engineer

4. Yes, 4 cigarettes/day

5. Glass of wine /day

6. No.

Doctor A - 22/06/2022

BMI → $\frac{92}{1,91^2}$ = 25,22 ⇒ Overweight

LF → Normal

Spirometry Reference Value Calculator

Reference Value Calculator

Calculated Values

Reset | Calculate Values

Figure 52: Patient profile containing examination and investigation results for the baseline scenario.

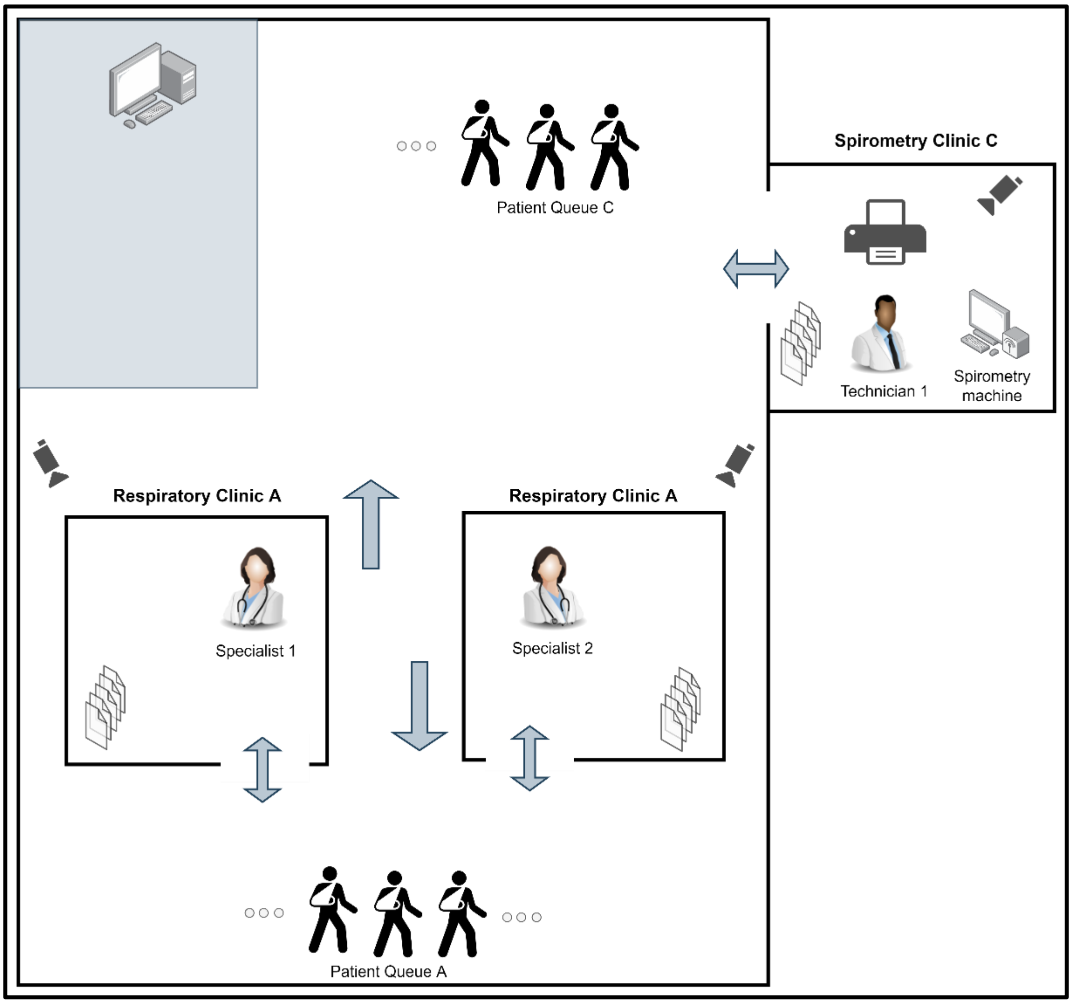


Figure 53: Physical layout of the baseline scenario.

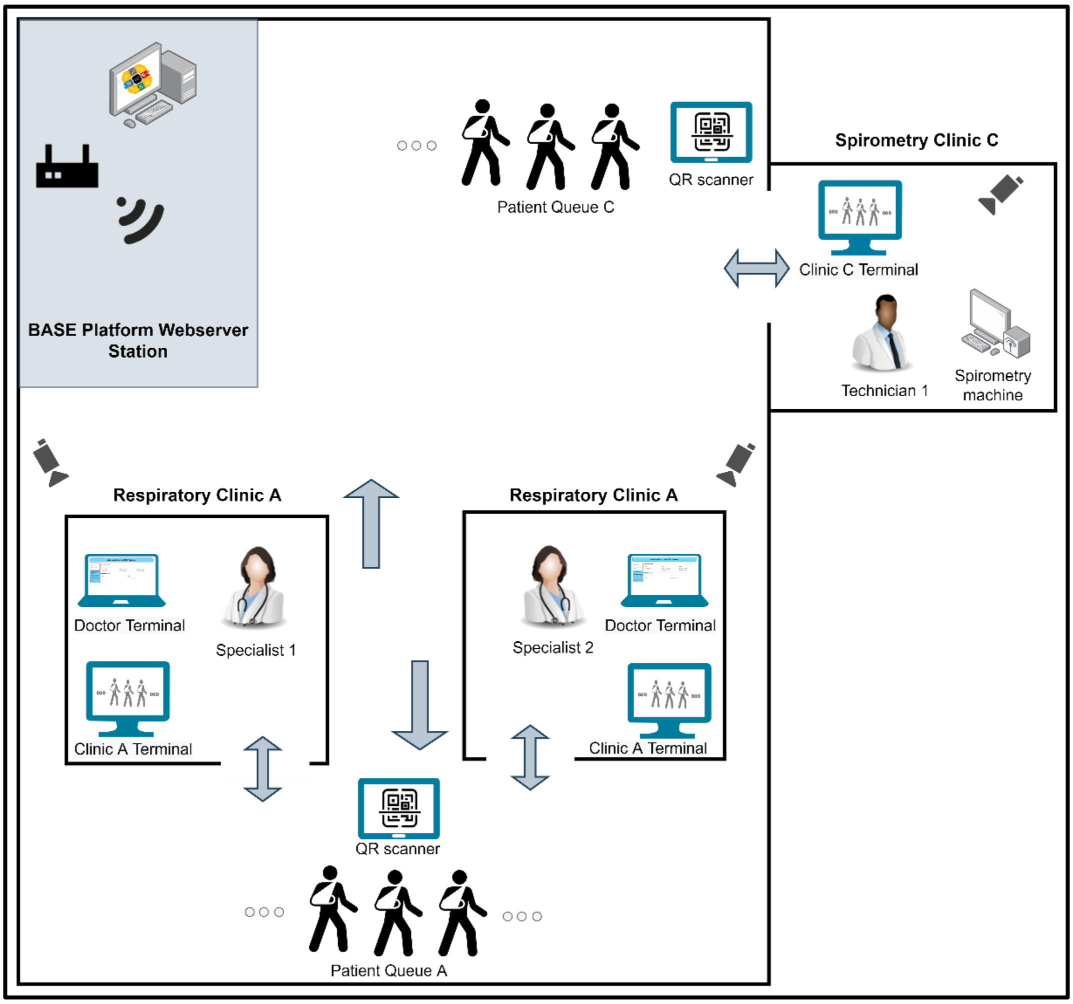


Figure 54: Physical layout of the HSI scenario.

Figure 55: Spirometry clinic waiting area.

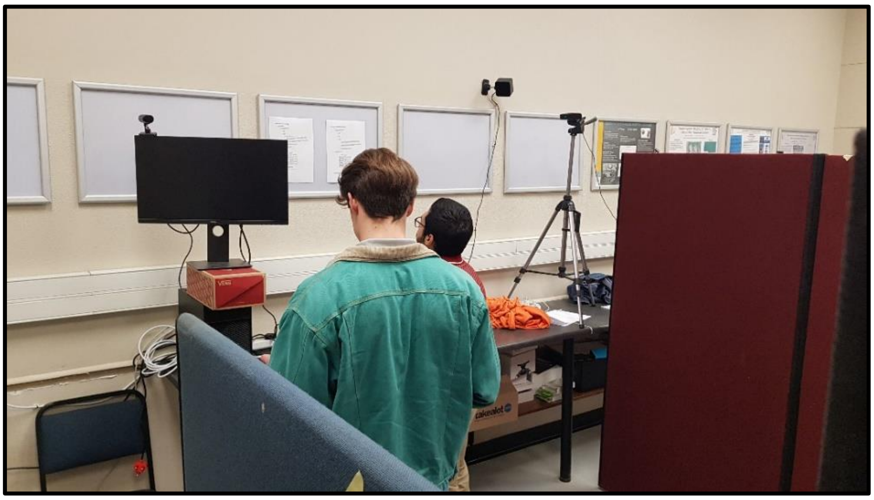


Figure 56: Doctor consultation room.

C.2 ELAN Software

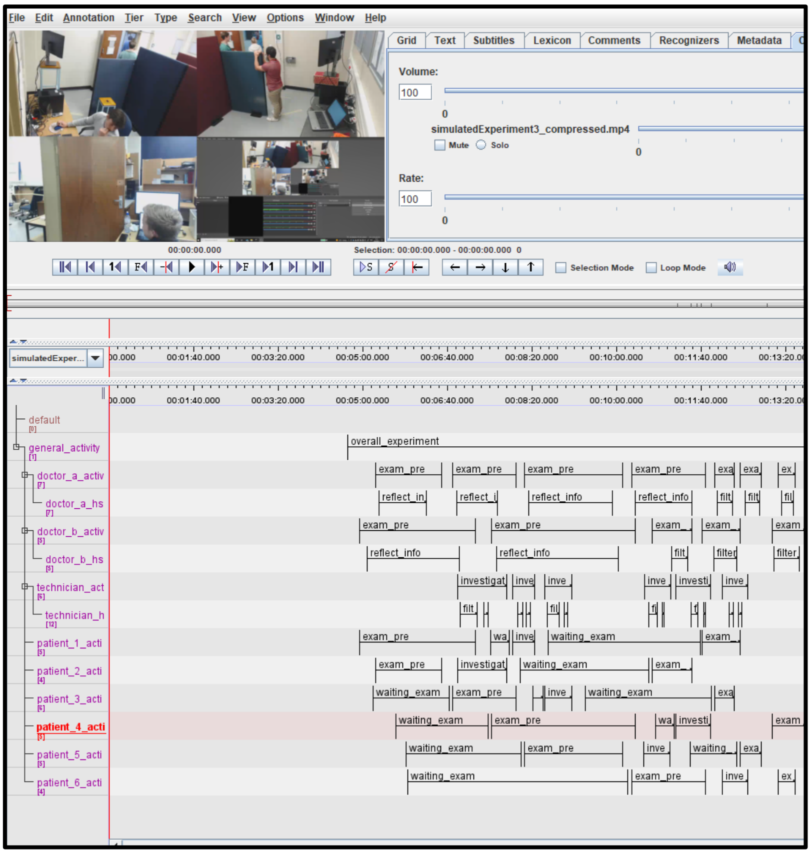
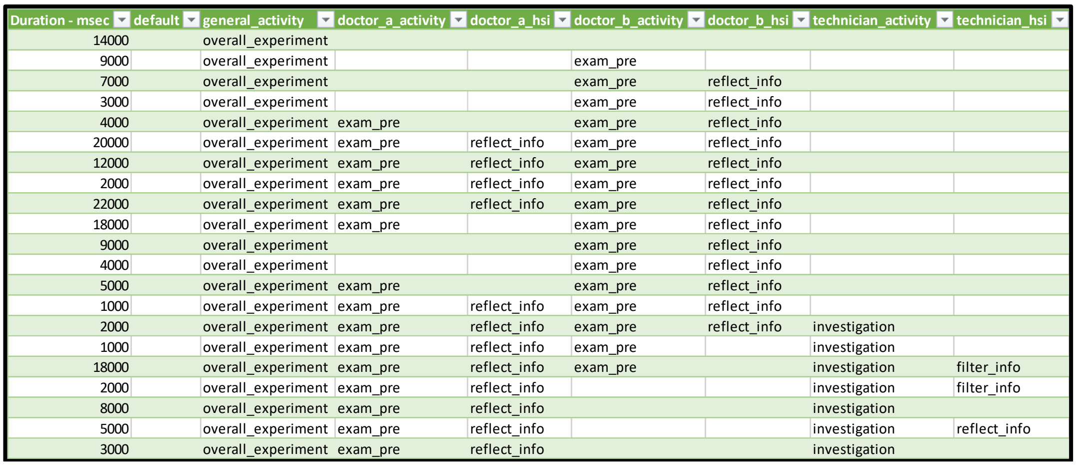


Figure 57: Screenshot of the ELAN software for coding an experimental scenario.



Duration - msec	default	general_activity	doctor_a_activity	doctor_a_hsi	doctor_b_activity	doctor_b_hsi	technician_activity	technician_hsi
14000		overall_experiment						
9000		overall_experiment			exam_pre			
7000		overall_experiment			exam_pre	reflect_info		
3000		overall_experiment			exam_pre	reflect_info		
4000		overall_experiment	exam_pre		exam_pre	reflect_info		
20000		overall_experiment	exam_pre	reflect_info	exam_pre	reflect_info		
12000		overall_experiment	exam_pre	reflect_info	exam_pre	reflect_info		
2000		overall_experiment	exam_pre	reflect_info	exam_pre	reflect_info		
22000		overall_experiment	exam_pre	reflect_info	exam_pre	reflect_info		
18000		overall_experiment	exam_pre		exam_pre	reflect_info		
9000		overall_experiment			exam_pre	reflect_info		
4000		overall_experiment			exam_pre	reflect_info		
5000		overall_experiment	exam_pre		exam_pre	reflect_info		
1000		overall_experiment	exam_pre	reflect_info	exam_pre	reflect_info		
2000		overall_experiment	exam_pre	reflect_info	exam_pre	reflect_info	investigation	
1000		overall_experiment	exam_pre	reflect_info	exam_pre		investigation	
18000		overall_experiment	exam_pre	reflect_info	exam_pre		investigation	filter_info
2000		overall_experiment	exam_pre	reflect_info			investigation	filter_info
8000		overall_experiment	exam_pre	reflect_info			investigation	
5000		overall_experiment	exam_pre	reflect_info			investigation	reflect_info
3000		overall_experiment	exam_pre	reflect_info			investigation	

Figure 58: ELAN software exported data.

C.3 Results and Analysis

Table 14: Baseline scenario overall results.

Overall Metrics	Annotation	Result	Units
HSI Characteristic Metrics			
Time to filter info	filter_info	751	(s)
Time to request information	request_info	-	(s)
Time to reflect info	reflect_info	790	(s)
Number of interruptions to workflow	interrupt_code	-	No.
Number of errors	error_code	-	No.
Element time study			
Examination (pre-investigation)	Exam_pre	922	(s)
Investigation	investigation	455	(s)
Examination (post-investigation)	Exam_post	601	(s)
Patient waiting	wait	2215	(s)
Other	idle	-	

Table 15: Baseline scenario doctor A results.

Doctor	Annotation	Result	Units
HSI Characteristic Metrics			
Time to filter (perceive) information	filter_info	556	(s)
Time to request information	request_info	-	(s)
Time to input (reflect) information	reflect_info	790	(s)
Number of interruptions to workflow	interrupt_code	-	No.
Number of errors	error_code	-	No.

Table 16: Baseline scenario doctor B results.

Doctor B	Annotation	Result	Units
HSI Characteristic Metrics			
Time to filter (perceive) information	filter_info	92	(s)
Time to request information	request_info	-	(s)
Time to input (reflect) information	reflect_info	107	(s)
Number of interruptions to workflow	interrupt_code	-	No.
Number of errors	error_code	-	No.

Table 17: Baseline scenario technician results.

Technician	Annotation	Result	Units
HSI Characteristic Metrics			
Time to filter (perceive) information	filter_info	195	(s)
Time to request information	request_info	-	(s)
Time to input (reflect) information	reflect_info	126	(s)
Number of interruptions to workflow	interrupt_code	-	No.
Number of errors	error_code	-	No.

Table 18: Baseline scenario patient results.

Patients	Result	Units
Overall waiting time	2215	(s)
Average waiting time	369	(s)
Average examination waiting time	361	(s)
Average investigation waiting time	8	(s)

Table 19: HSI scenario overall results.

Overall Metrics	Annotation	Result	Units
HSI Characteristic Metrics			
Time to filter (perceive) information	filter_info	180	(s)
Time to request information	request_info	-	(s)
Time to input (reflect) information	reflect_info	545	(s)
Number of interruptions to workflow	interrupt_code	-	No.
Number of errors	error_code	-	No.
Element time study			
Examination (pre-investigation) average	Exam_pre	663	(s)
Investigation average	investigation	218	(s)
Examination (post-investigation) average	Exam_post	199	(s)
Patient waiting	wait	1187	(s)
Other	idle	-	

Table 20: HSI scenario doctor A results.

Doctor A	Annotation	Result	Units
HSI Characteristic Metrics			
Time to filter (perceive) information	filter_info	125	(s)
Time to request information	request_info	-	(s)
Time to input (reflect) information	reflect_info	523	(s)
Number of interruptions to workflow	interrupt_code	-	No.
Number of errors	error_code	-	No.

Table 21: HSI scenario doctor B results.

Doctor B	Annotation	Result	Units
HSI Characteristic Metrics			
Time to filter (perceive) information	filter_info	92	(s)
Time to request information	request_info	-	(s)
Time to input (reflect) information	reflect_info	107	(s)
Number of interruptions to workflow	interrupt_code	-	No.
Number of errors	error_code	-	No.

Table 22: HSI scenario technician results.

Technician	**Annotation**	**Result**	**Units**
HSI Characteristic Metrics			
Time to filter (perceive) information	filter_info	55	(s)
Time to request information	request_info	-	(s)
Time to input (reflect) information	reflect_info	22	(s)
Number of interruptions to workflow	interrupt_code	-	No.
Number of errors	error_code	-	No.

Table 23: HSI scenario patient results.

Patients	**Annotation**	**Result**	**Units**
Overall waiting time		1187	(s)
Average waiting time		198	(s)
Average examination waiting time		188	(s)
Average investigation waiting time		9	(s)

Table 24: Experimental analysis results.

Analysis	**Result**	**Unit**
Reduction % in filter	76	%
Reduction % in reflect	31	%
Reduction % in Examination (pre-investigation)	28	%
Reduction % in investigation	52	%
Reduction % in Examination (post-investigation)	67	%
Reduction % patient waiting	46	%