A Holonic Multi-Agent System for the Support of the Differential Diagnosis Process in Medicine

Dissertation

zur Erlangung des Doktorgrades

„Dr. rer. nat.“

der Fakultät für Wirtschaftswissenschaften

der Universität Duisburg-Essen

vorgelegt

von

Zohreh Akbari

aus

Teheran, Iran

Betreuer:

Prof. Dr. Rainer Unland

Datenverwaltungssysteme und Wissensrepräsentation (DAWIS)

Institut für Informatik und Wirtschaftsinformatik (ICB)

Essen, 18.01.2021

Erstgutachter: Prof. Dr. Rainer Unland

Zweitgutachter: Prof. Dr. Bogdan Franczyk

Tag der mündlichen Prüfung: 18.01.2021

DOI: 10.17185/duepublico/73939
URN: urn:nbn:de:hbz:464-20210216-102251-4

Abstract

The primary concern of the available Medical Diagnosis Systems (MDSs), including the state of the art, is to find the perfect link between the patient's medical record and their health knowledge. As a result, regardless of how powerful or ground-breaking they are in performing this action, it is always possible that their final strong deduction is based on some incomplete input, which may very likely lead to a misdiagnosis. To date, no computer-aided Decision Support System (DSS) has been introduced to address this issue and this is the reason why the available MDSs cannot be well integrated into the workflows of the healthcare institutions. Prior to using these systems, a physician should complete the patient's medical record by performing a complaint-directed History and Physical examination (H&P) and prepare the required thorough input. The H&P is steered by Differential Diagnosis (DDx), which is the process of differentiating between two or more conditions which share similar signs or symptoms. When a physician performs this examination to provide an MDS with the right input, if no complications occur, at the end of the H&P s(he) will reach the diagnosis too, and as a result the later use of the MDS would be of less value. Consequently, MDSs are used very seldom in practice and are exclusively used in complicated medical cases, where the H&P performed by the physician has not led to a definitive diagnosis.

This study aims to introduce a Diagnostic Decision Support System (DDSS) that guides the user in performing DDx directed H&Ps. Of course, this system can be used by the experts who are originally meant to perform the H&P, i.e., physicians and if allowed by the healthcare system of the country Nurse Practitioners (NPs) and/or Physician Assistances (PAs), in order to have some reminders while completing the H&P report. However, as the shortage of medical doctors is worsening in the recent years, this system intends to guide simple nurses in performing the H&P, helping the doctors to be able to see more patients in a certain period of time, as they would just need to review and asses the already prepared H&P report. In third world countries with medical treatment being far away, the user of the system can even be a person with some basic medical knowledge or the patient him- or herself, who may use the system in order to become aware of the possibilities and receive suggestions on finding the right experts to be contacted. When used by experts, as might be expected, in case of any complications it is again possible to use the output of this system as the input of available MDSs that are designed to address such cases.

The differential diagnostic problem can be recursively broken down into sub-problems by weighting the likelihood of the presence of possible diseases. These subproblems may induce different abstraction levels and can be of different granularities. Moreover, according to the nature of DDx, the problem solvers should be collaborative and those dealing with similar diseases need to have more communications, which are to be conducted in a timely manner. These features clearly show that the DDx domain meets the characteristics of the holonic domains, which involve agents that can simultaneously be a whole and a part. Consequently, a Holonic Multi-Agent System (HMAS) composed of agents that are either expert in diagnosing a disease or a group of related diseases is proposed in this study to address the DDx problem.

From the complexity of the medical data the system must deal with and the fact that medical knowledge demonstrates a steady upward growth it can be inferred that the proposed system should be capable of learning and adaptation to the new findings in this field. Moreover, since diagnosis is very much affected by the geographical regions, the system should also be capable of learning and adaptation to the local patterns too. This means that the proposed system should be empowered with appropriate Machine Learning (ML) techniques. As the holonic approach is a new trend in computer science there are very few studies on the learning techniques that can be applied to HMASs. Accordingly, this research also includes a study on determining the right ML techniques for the objectives of the system. Therefore, even though the development of the Holonic Medical Diagnosis System (HMDS), which is capable of performing DDx, is the practical contribution of this work, the introduction of the ML techniques that are used to adapt its functionality can be considered as the conceptual/theoretical contribution of this research, as the proposed techniques can be applied to other HMASs that adopt a similar approach for problem solving to the one followed in this study.

This work also includes assessment simulations of the proposed system that monitor the system's general behavior in performing the H&P and examine the learning abilities of the system by providing the system with appropriate inputs and evaluating the corresponding outputs. The results of these assessments show that the proposed system is a promising tool for addressing the DDx problem and eventually helping the MDSs to gain acceptance from the health service providers in practice.

Acknowledgement

I would like to express my sincere gratitude to my advisor Prof. Dr. Rainer Unland for the continuous support of my PhD study and related research, for his patience, immense knowledge and insightful comments that prompted me to broaden my research horizons. I deeply thank him for his valuable moral support throughout my studies, which has significantly helped me to be able to carry out my research. I am also very grateful to my second advisor Prof. Dr. Bogdan Franczyk, the Chair of Information Management, the Institute of Business Information Systems at the University of Leipzig, for accepting to support my research with his great knowledge and experience in the field.

I gratefully acknowledge the informative and encouraging discussions on the medical aspects of the project with my brother-in-law, Dr. Farzad Fakouri. I would like to express appreciation and gratitude for his knowledgeable insight and expertise, that greatly assisted the research.

Furthermore, I thank my former colleagues from the research group of Data Management Systems and Knowledge Representation (DAWIS), Institute for Computer Science and Business Information Systems (ICB), who were always willing to share their knowledge and experience. Their input and feedback on our routine presentation days have been helpful and constructive.

My deep appreciation goes to my dear parents, and my beloved brother and sister, for their unconditional love and care, and for encouraging and supporting me spiritually throughout writing this thesis. I would not have made it this far without them and I will be forever grateful for their presence in my life. And last but not the least, I would like to thank my loving husband, who has always been there for me, and believed in me. This dissertation would not have been possible without his warm love, continued patience, and endless support. I would also like to dedicate this work to my dear daughter, Celina, who joined us when I was writing this dissertation, and brought us boundless joy and happiness.

Table of Contents

Table of Figures

Table of Tables

List of Abbreviations

Abbreviation	Expansion/Explanation
A&P	Assessment and Plan
ABM	Agent-Based Modeling
AI	Artificial Intelligence
All/RXNs	Allergies/Reactions
ANN	Artificial Neural Network
APA	American Psychological Association
APSO	Assessment, Plan, Subjective, Objective (approach)
AUML	Agent Unified Modeling Language
CAPS	Concern, Assessment, Plan, Supporting information (approach)
CC	Chief Complaint or Chief Concern
CDSS	Clinical Decision Support Systems
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
DDI	Drug-Drug Interaction
DDP	Disease Description Pattern
DDSS	Diagnostic Decision Support System
DDx	Differential Diagnosis
DM	Decision Maker
DMI	Digital Medical Imaging
DP	Dynamic Programming
Dx	Diagnosis
ECG or EKG	Electrocardiography
EHR	Electronic Health Record
EMR	Electronic Medical Record
ER	Emergency Room
ES	Expert System
FH or FHx	Family History
GA	Genetic Algorithm
H&P	History and Physical examination
HetS	Heterogeneous Swarm
HetSRL	Heterogeneous Swarm RL
HIS	Health Information Systems
HMAS	Holonic Multi-Agent System
HMDS	Holonic Medical Diagnosis System
HPI	History of Present Illness
HQL	Holonic-Q-Learning
ICD	International Statistical Classification of Diseases and Related Health Problems

M.D.	Doctor of Medicine
MARL	Multi-Agent Reinforcement Learning
MAS	Multi-Agent System
MDP	Markov Decision Process
MDS	Medical Diagnosis System
MEDS	Medications
ML	Machine Learning
MOMDP	Multi-Objective MDP
NLP	Natural Language Processing
NN	Neural Network
NP	Nurse Practitioner
PA	Physician Assistant
PE	Physical Exam
PHY	Physician
PMH	Past Medical History
PMS	Patient monitoring systems
PSH	Past Surgical History
PSO	Particle Swarm Optimization
QL	Q-Learning
QV	Q-Value
RL	Reinforcement Learning
ROS	Review of Systems
S/S or S/Sx	Signs and Symptoms
SD	Standard Deviation
SDMP	Sequential Decision-making Problem
SH or SHx	Social History
SI	Swarm Intelligence
SOAP	Subjective, Objective, Assessment, Plan (approach)
SPP	Shortest Path Problem
SRS	Software Requirements Specification
SVM	Support Vector Machine
UML	Unified Modeling Language
WHO	World Health Organization

Glossary of Medical Terms

Term	Definition
Differential Diagnosis (DDx)	"The process of weighing the probability of one disease versus that of other diseases possibly accounting for a patient's illness. The differential diagnosis of rhinitis (a runny nose) includes allergic rhinitis (hay fever), the abuse of nasal decongestants and the common cold" [1].
Disease elimination	Disease "elimination refers to the reduction to zero (or a very low defined target rate) of new cases of the disease in a defined geographical area" [2].
Disease eradication	Disease "eradication refers to the complete and permanent worldwide reduction to zero new cases of the disease through deliberate efforts" [2].
History and Physical examination (H&P)	"A critical component of a patient encounter in which information relevant to present complaint is obtained, by asking questions about family and personal medical history and the organ systems examined in as great detail as necessary to manage the present condition or evaluate-workup-the patient" [3].
Hospitalist	"A hospital-based general physician. Hospitalists assume the care of hospitalized patients in the place of patients' primary care physicians. In the most prevalent US model of hospitalist care, several physicians practice together as a group and work full time to care for inpatients" [1].
Nurse	"A person trained, licensed, or skilled in nursing that is the profession concerned with the provision of services essential to the maintenance and restoration of health by attending the needs of sick persons" [1].
Nurse Practitioner (NP)	"A Registered Nurse (RN) who has completed an advanced training program in a medical specialty, such as pediatric care. An NP may be a primary, direct health care provider, and can prescribe medications. Some NPs work in research rather than in direct patient care" [1].
Pathognomonic	"A sign or symptom that is so characteristic of a disease that it can be used to make a diagnosis. For example, Koplik spots in the mouth opposite the first and second upper molars are pathognomonic of measles" [1].
Physician (PHY)	"A person who is trained in the art of healing. In the UK, a physician is a specialist in internal or general medicine, whereas in the US a physician is any doctor of medicine. The term generally refers to a person who has earned a Doctor of Medicine (MD), Doctor of Osteopathy (DO), or Doctor of Naturopathy (ND) degree and who is accepted as a practitioner of medicine under the laws of the state, province, and/or nation in which he or she practices" [1].
Physician Assistant (PA)	"A physician assistant (PA) is a mid-level medical practitioner who works under the supervision of a licensed doctor (an MD) or osteopathic physician (a DO). The physician assistant came about in the 1960s as a response to the need for more clinicians and better access to health care" [1].
Sign	"Any objective evidence of disease, as opposed to a symptom, which is, by nature, subjective. For example, gross blood in the stool is a sign of disease; it is evidence that can be recognized by the patient, physician, nurse, or someone else. Abdominal pain is a symptom; it is something only the patient can perceive" [1].
Sine qua non	"Sine qua non is a phrase used in radiology, and more widely in clinical medicine, to refer to a symptom, sign, radiology finding, etc., which is absolutely necessary for a diagnosis to be made. For example, if one is querying a thoracic aortic dissection then the presence of a visible dissection flap is a sine qua non for the diagnosis to be true, i.e., the presence of a flap is absolutely necessary to make the diagnosis" [4].
Symptom	"Any subjective evidence of disease. In contrast, a sign is objective. Blood coming out a nostril is a sign; it is apparent to the patient, physician, and others. Anxiety, low back pain, and fatigue are all symptoms; only the patient can perceive them" [1].

Chapter 1
INTRODUCTION

1.1 A Brief Introduction to the Research Domain

Since the 1950s computer scientists have aimed to support and improve healthcare systems. Even though early applications were very limited, with the introduction of Artificial Intelligence (AI) big improvements were made in this area and the digitalization of healthcare became inevitable. As today the trend towards an increasingly digitalized healthcare industry plays a significant role in the exponential growth of medical knowledge, information technology itself has proven to be a solution to this problem. In fact, this technology supports the healthcare providers with the right applications to assimilate and apply the expanding medical knowledge effectively. One of the major application areas of computer systems in medicine is medical diagnosis (abbreviated Dx). A Diagnostic Decision Support System (DDSS) or Medical Diagnosis System (MDS) is a specific type of Clinical Decision Support System (CDSS) that is developed to provide an ordered list of potential diagnoses for given signs and symptoms. It is to be noted that these systems do not aim to replace the physicians but are best used to remind them about the critical possible diagnoses that might have simply been ignored. Accordingly, such systems basically intend to improve the diagnosis quality.

Chapter 2 and chapter 3 provide a deep understanding of the MDSs, and chapter 3 studies the state of the art of these systems. This review shows that available MDSs mainly focus on finding the perfect link between the given input and their health knowledge. This implies that these systems assume that the input already contains all the required information for a flawless diagnosis. Accordingly, before using these systems the physician still needs to gather all the relevant information, which basically includes information about the absence or presence of signs and symptoms of any possible diagnosis. In fact, one of the big challenges of using the MDSs is the preparation of the input data, which again requires the physician to perform the routine examination.

In patient encounter the physician would carefully listen to the signs and/or symptoms explained by the patient, i.e., the Chief Complaint (CC), considers some potential diagnoses and then tries to collect enough evidence and supporting information to make sure that the rest of the candidates have lower probabilities. This means (s)he would collect the relevant signs and symptoms, i.e., would question the signs and symptoms that might have been simply ignored by the patient, or would request the patient to undertake some medical examinations. This diagnosis method is called the Differential Diagnosis (DDx). As defined in [5], DDx is "the distinguishing of a disease or condition from others presenting with similar signs and symptoms", which according to [6, p. 107] is "the key to reducing diagnosis error". This is very critical, as misdiagnosis may lead to delay in treatment, as well as exposure to inappropriate medication that can lead to serious irrecoverable effects.

In a patient encounter, this method is applied in a process called the History and Physical examination (H&P). Accordingly, H&P is "a critical component of a patient encounter in which information relevant to present complaint is obtained, by asking questions about family and personal medical history and the organ systems examined in as great detail as necessary to manage the present condition or evaluate-workup-the patient" [3]. Consequently, the MDSs in fact take the output of H&P as their input. It is to be noted that if a physician has already conducted the H&P, in most of the cases (s)he has already reached the final diagnosis. This is exactly one of the constrains of these systems, which hinders their thorough integration into the clinical workflow of the hospitals. As a result, MDSs are solely being used in complicated cases, where a definitive diagnosis cannot be determined, or diagnosis is clear but patient-specific treatment is needed.

In fact, a system capable of guiding the H&P could be well integrated into the clinical workflow. Such systems could save the physician time or even guide less experienced nurses in performing this step. Undertaking the H&P, they would help the doctors to be able to see more patients in a certain period of time, as they would just need to review and asses the already prepared H&P report (see section 1.3.1). Moreover, as mentioned, in case of a complex medical case this system could provide the all-encompassing input for the cutting-edge medical diagnosing systems for reassessments or creating patient-specific treatment plans (see chapter 2). As a result, this study aims to analyze the DDx domain and identify the right approach in order to implement a system capable of guiding the H&P process.

1.2 Research Question

Previous subsection indicated a gap in MDSs, i.e., the lack of support for H&P. This shortcoming is basically one of the key reasons for the lack of motivation of health service providers to use MDSs. In order to empower the healthcare providers with an MDS capable of guiding the H&P, its key component, i.e., the DDx should be implemented. The abstract Software Requirements Specification (SRS) provided in Table 1 gives a detailed description of the intended software system (For further information on system features and requirements please refer to section 3.1).

In order to develop a software system that implements the DDx, this process should be studied carefully. This research shows that the DDx domain meets the characteristics of Holonic Multi-Agent Systems (HMASs)[1] (see chapter 4) and as a result proves that using this approach an MDS capable of guiding the H&P can be designed. Accordingly, the research question of this study is whether the holonic multi-agent architecture can be used to implement the DDx method in H&P and consequently promote the wider use of medical diagnosis systems.

Medical knowledge continuously improves, and diagnosis is also very much affected by the geographical regions. As a result, any realistic MDS should be capable of learning and adaptation to the new findings and the local patterns. This means that the proposed system should be

[1] HMASs involve agents that can simultaneously be a whole and a part, and the organizational structure of a holonic society is called a holarchy (HMASs are introduced in section 4.2).

empowered with appropriate Machine Learning (ML) techniques. As the holonic approach (see section 4.2) is a new trend in computer science there are very few studies on the learning techniques that can be applied to HMASs. Accordingly, this research includes a study on determining the right ML techniques for the system's objectives and suggests an adaptation of one of the available techniques to the holonic systems (see chapter 6). Therefore, even though the development of the Holonic Medical Diagnosis System (HMDS), which is capable of performing DDx, is the practical contribution of this work, the introduction of ML techniques that support the functionality of this system can be considered as the conceptual/theoretical contribution of this research (see section 1.3). In fact, the proposed techniques can be applied to any HMAS that adopts a similar problem-solving approach to the one followed in this study.

Table 1. The Software Requirements Specification

The Software Requirements Specification (SRS)	
1. Introduction	**1.1. Purpose:** to provide a detailed overview of the diagnostic decision support system that supports the DDx process in medicine **1.2. Intended Audience:** The research team, healthcare professionals **1.3. Intended Use:** Medical diagnosis – H&P (The implementation of DDx) **1.4. Scope:** see included and excluded functions in Table 2. **1.4.1.** Complete implementation of the H&P, empowering the user to conduct the process **1.4.2.** Delivering the H&P report to the physician for final diagnosis and feedback **1.4.3.** Updating the DDPs (Disease Description Patterns) according to new instances (Learning from feedback) **1.4.4.** Updating the regional distribution of diseases (Learning from Feedback) (For the included and excluded functions please refer to Table 2.) **1.5. Definitions and Acronyms:** see section 1.1, chapter 2, and the table of abbreviations
2. Overall Description	**2.1. User Needs:** Two types of end users may use the system: **2.1.1.** User: A user can be a physician, a nurse, or simply the patient, who uses the system to enter a diagnosis request (signs and symptoms) and receive a medical diagnosis. **2.1.2.** Expert: An expert is a physician who monitors the diagnosis result and provide the system with feedback. An Expert can also modify the medical data of the system. **2.2. Assumptions and Dependencies:** The system assumes that the feedback is given by qualified physicians and uses this input to learn and adapt itself to the new findings.
3. System Features and Requirements	**3.1. Functional Requirements:** **3.1.1.** Differential Diagnosis **3.1.1.1.** The system allows the user to enter its chief complaint. **3.1.1.2.** The system questions the relevant signs and symptoms. **3.1.1.3.** The system allows the user to add signs and symptoms to its diagnosis request. **3.1.1.4.** The system displays the diagnosis. **3.1.2.** Data Modification **3.1.2.1.** The system allows the expert to modify available medical information **3.1.2.2.** The system allows the expert to delete eradicated diseases **3.1.2.3.** The system displays the diagnosis result and allows the expert to enter feedback (see Figure 1) **3.2. Nonfunctional Requirements (Quality Requirements):** Adaptability (ML abilities) **3.2.1.** Updating the DDPs according to new instances **3.2.2.** Updating the regional distribution of diseases

1.3 Research Contributions

According to the research questions, this study has a number of practical and theoretical contributions. Some results of the general work have already been published in [7, 8, 9, 10, 11].

1.3.1 The Practical Contribution

As mentioned in the previous subsection, the introduction of an MDS capable of guiding the H&P is the key contribution of this research and this system intends to implement DDx using the holonic approach (see chapter 4). It should be noted that this study basically concentrates on the implementation of DDx and does not aim to cover all the capabilities that an MDS may have. It is rather suggested that the proposed system should be integrated into the available MDSs in order to provide them with a means to generate the required all-encompassing input. Figure 1 and Table 2 show the system scope, i.e., the functions included in or excluded from the system.

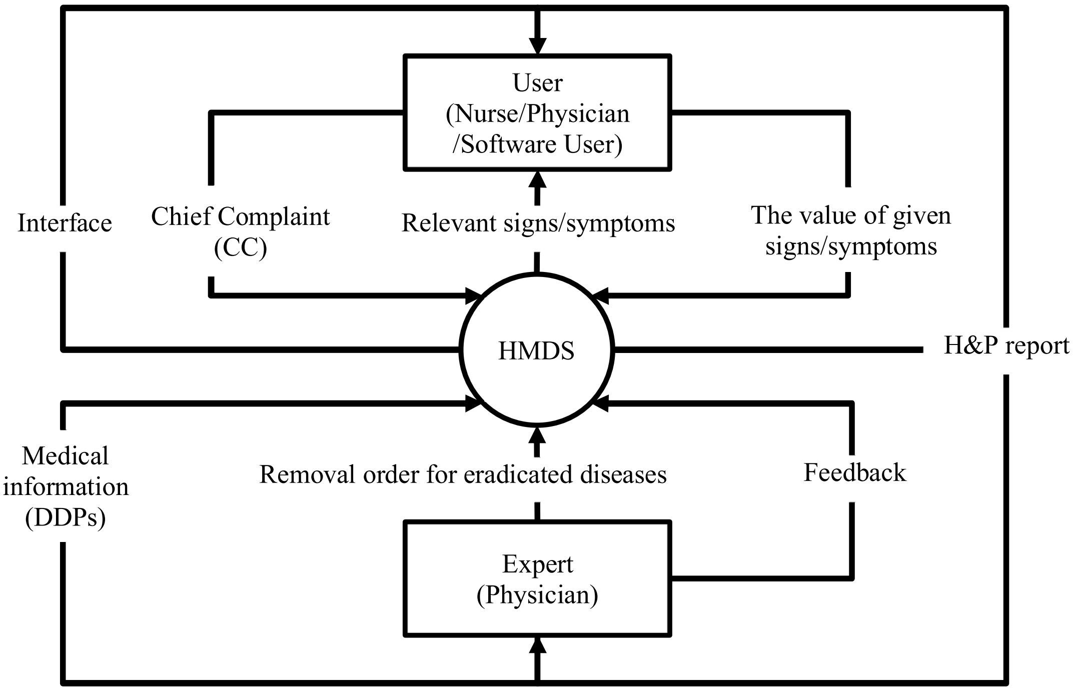


Figure 1. System Scope - Context Diagram

Figure 1 illustrates the context diagram of the holonic medical diagnosis system. The circle represents the system and the rectangles show the external entities that send data to or receive data from the system. Two types of users are foreseen for the system: the users, who utilize the system in order to perform H&P, and the experts, who are responsible for the medical knowledge of the system.

As mentioned, the HMDS aims to guide the users throughout the different stages of H&P. Accordingly, this system can be used by physicians, who would like to have some reminders while completing the H&P report. However, the users of the system may in fact be simple nurses, who would be able to perform the H&P under the guidance of the system. Basically, depending on the healthcare system of the country the user of the system may be different. In Germany, for example, a nurse is not meant to examine patients and that is the task of a doctor. In US, however, NPs and PAs are able to perform the H&P. As mentioned, henceforward this system may even allow simple nurses to perform this examination. In third world countries with medical treatment being far away, the user can even be a person with some basic medical knowledge or the patient him- or herself, who may use the system in order to find the right expert to be contacted. In all cases, the physicians as experts may monitor the performance of the system, provide the system with the relevant feedback, and control the medical information of the system.

The system provides the users with an interface, which allows them to enter the chief complaint declared by the patient. At this stage the system will determine the relevant signs and symptoms based on DDx and will then list them so that the user could question or examine them (For more information on the determination of the relevant signs and symptoms please see section 5.2). This list could be updated instantly and regularly based on the new entries. Having the value of the relevant signs and symptoms, the system can then prepare the H&P report. This report will be given to the physician for final control and at this stage (s)he will be able to provide the system with feedback on the diagnosis result. The system uses this information as learning data in order to improve its performance (For more information on the learning abilities of the system please refer to chapter 6).

It is to be noted that the physician as the expert is able to control the medical knowledge given to the system at the initialization stage and is also responsible for any necessary updates. Additionally, the system allows the expert (physician) to remove possible eradicated diseases from the system. However, the system does not conclude the eradication or elimination of diseases[2] in order to suggest removals, even if a disease has not been occurred in a long time.

Table 2. The system scope (included/excluded functions)

Included Functions	**Excluded Functions**
1. Complete implementation of the H&P, empowering a simple nurse to conduct the process	1. Mapping among the different biomedical vocabularies (Support for NLP – Natural Language Processing)
2. Delivering the H&P report to the physician for final diagnosis	2. Suggesting possible new diseases / outbreaks or reasoning on the eradication or elimination of diseases
3. Updating the DDPs (Disease Description Patterns) according to new cases of the diseases (Learning from feedback)	3. Suggesting patient-specific treatments, e.g. avoiding Drug-Drug Interactions (DDIs), …
4. Updating the regional distribution of diseases (Learning from Feedback)	4. Knowledge extraction from natural language sources (Support for NLP)

Table 2 shows the functions that are included in or excluded from the system scope. In order to have a better understanding of these functions a diagnosis case can be considered here. As mentioned, during the H&P the physician receives the chief complaint, performs the DDx process, and completes the H&P report. Together with the diagnosis, the physician will not only suggest the treatments but (s)he is also able to provide a patient-specific treatment to avoid DDIs, prevent allergic reactions, etc. (e.g., a patient who is diagnosed with rheumatic fever and is already taking ibuprofen for minor injuries cannot take aspirin for his or her rheumatic fever and should try another medication that can be used to treat this disease). It should also be noted that regardless of which words and terms are used by the patient, the physician is able to understand him or her and eventually adapt his or her word choice for a better communication. Experience allows the physicians to have a better understanding of the signs and symptoms of the different diseases and their distributions. Physicians can also update their knowledge by studying the up-to-date medical papers and apply their knowledge to their diagnoses.

[2] According to the World Health Organization (WHO) “elimination refers to the reduction to zero (or a very low defined target rate) of new cases in a defined geographical area”, however, “eradication refers to the complete and permanent worldwide reduction to zero new cases of the disease through deliberate efforts”. As stated in [239], “to date, only one infectious disease that affects humans has been eradicated”. Smallpox was declared eradicated by the World Health Assembly in 1980, after decades of efforts by the World Health Organization. On the other hand, polio can be mentioned as an example of the eliminated diseases, as it was eliminated in the United States by 1979 after widespread vaccination efforts.

As mentioned, the system aims to guide the H&P and therefore, the most important deliverable of it will be the H&P report. To this end, the system receives the chief complaint and determines the relevant signs and symptoms to be checked. Upon receiving the value of these signs and symptoms the system is then able to generate the ordered DDx list. Providing the diagnoses, the system will also be capable of suggesting the relevant drugs and treatments. However, within the scope of the current project the system does not intend to suggest patient specific treatments or avoid DDIs, especially in case of multiple diseases. Within the scope of the current project the system does not intend to support natural language processing too. It is assumed that the user of the system is capable of covering this need, matching the patient's words with the equivalent words provided and recognizable by the system. For example, if the patient complains about shortness of breath and the term saved in the system for this condition is dyspnea, it is assumed that the user is capable of finding this link. Adding the capability of understanding the natural language is itself a separate project which may be considered as future work.

The system is, however, intended to be capable of learning, that is, it can update the Disease Description Patterns (DDPs) and keep track of regional distribution of the diseases by saving the frequency of different diseases. This information helps the system to improve its results while calculating the probability of different diseases for a given case. However, as mentioned, it is not suggested to use this data to conclude the elimination or eradication of diseases. As the system collects statistical data about the disease occurrences, it may suggest the removal of eliminated diseases and ask for expert's approval. However, the idea was rejected in the early stages of the project. In fact, even if a specific disease has not been spotted for a long time, this doesn't necessarily mean that the disease has been completely eradicated. Of course, since the system saves the frequency of the diseases it will suggest the presence of such disease with lower probability. Indeed, one of the main advantages of the holonic medical diagnosis system is the ability to remind the critical rare possible diagnoses that have been ignored. The removal of diseases that occur seldom prevents this goal from being achieved.

The system receives its medical data from the available disease/symptom databases (see section 1.5) and is capable of updating this data through learning (see chapter 6). However, this project does not include knowledge extraction from natural language sources and considers this as a feature that may be added to the system in future works. It should also be mentioned that extra functions may be added to the system to make it able to use its data in order to suggest the possibility of new diseases or outbreaks, and then provide supplementary information on the matter. However, this function is also not covered in the current project.

1.3.2 The Theoretical Contributions

1.3.2.1 The Holonic-Q-Learning (HQL)

One of the attractive characteristics of HMASs is the support for self-organization, which can be defined as the mechanism or the process enabling a system to change its organization without explicit external command during its execution time [12]. As the organizational structure of the HMDS plays a key role in problem solving and has direct influence on the result quality, it is of eminent importance to support this process with appropriate ML techniques (For more information see chapter 6). This study proposes the Holonic-Q-Learning (HQL) as a specific Reinforcement Learning (RL) method for the purpose of self-organization in the HMDS. This method can also be applied to any HMAS that follows a similar approach to tackle its problem. In order to introduce an appropriate technique for the system at hand this work studies the available ML techniques that could be applied to the problem and accordingly explains the need for a novel RL technique. HQL combines concepts from RL and swarm intelligence. As will be discussed in chapter 6 a more general representation of this technique can also be used by any Heterogeneous Swarm (HetS), which aims to solve a Sequential Decision-making Problem (SDMP) that can be modeled as Markov Decision Process (MDP) (see chapter 6).

1.3.2.2 The Automated Determination of the Input Parameter of DBSCAN

The proposed system uses clustering in order to find its initial structure, i.e. holarchy (see section 6.1.1). For this reason, the DBSCAN (Density-Based Spatial Clustering of Applications with Noise) [13] has been used, which is one of the most common clustering algorithms that is also highly cited in the scientific literature. Despite its strengths, DBSCAN has a shortcoming in automatically determining its input parameters, which involves interaction with the user, providing some graphical representation of the data. As the determination of the parameters gets complicated when dealing with complex data such as medical data, during this study a simple and effective method for automatically determining the input parameter of DBSCAN has been introduced (see chapter 6).

1.4 Research Methods

According to the research questions a combination of research methods is used in this study. Considering the terminology used in [14] for research methods in computer science, this research basically focuses on formal and model methods for different cases.

Formal methods are most frequently used in theoretical computer science and intend to prove facts about algorithms and system [14]. One of the interesting applications of such methods is when dealing with the time or space complexity of an algorithm, or with the correctness or the quality of the solutions generated by the algorithm. As a result, mathematical proof techniques fall into this category of research methods. In this study mathematical techniques are used to develop a method for automatically determining the input parameter of DBSCAN (see section 6.1.1.2). Moreover, a mathematical proof technique has been used to prove the convergence of HQL. This assures that the system structure can reach an optimal organizational structure applying this method (see section 6.1.3.9).

On the other hand, a model method concentrates on defining an abstract model for a real system. This model will be much less complex than the system under study, allowing the researcher to have a better understanding of the system. Moreover, the researcher can use this model to perform experiments that could not be performed using the system itself because of cost or accessibility reasons. Experiments based on a model are called simulations [14]. This study creates a model of the HMDS that deals with a limited number of diseases in order to provide a less complex environment to study the functionality and learning abilities of the real system (see chapter 7). Of course, the implementation of the real system which could diagnose thousands of diseases[3] requires industrial investments. Using the system model, the structure, interactions and behavior of the system is demonstrated, and a number of simulations were also run to show the learning abilities of the system (see chapter 8). To put it in a nutshell, formal methods have been used in this research to develop and prove the theoretical contributions of this project. However, the practical contribution of this study has been demonstrated using a model method.

[3] According to WHO [2] the total number of diseases is about 30.000.

1.5 Medical Data Sources Used in this Research

In order to develop the system model and accordingly run the simulations, real medical data is required. As explained in chapter 5 the system needs to save the signs and symptoms of the diseases in order to be able to match them with the signs and symptoms provided in the diagnosis request. In the simulations provided in this study (see chapter 7) the required data has been gathered from the available medical sources and as mentioned already, the system does not intend to extract this knowledge using NLP. Accordingly, it is assumed that during the initialization stage of the system an expert enters the signs and symptoms of each disease into the system. This data can henceforth be updated based on the feedback received from the physicians or by considering new instances of the diseases (see chapter 6). The following medical data sources are used in this project:

(1) Official site of MAYO CLINIC[4] [15]: This source includes a comprehensive guide to hundreds of conditions. Choosing each disease this reference provides its signs, symptoms, causes, related medical test, and treatments.

(2) WebMD[5] [16]: This source is a health information services website, which publishes content around health and health care topics, including diseases.

(3) Medscape[6] [17]: This source provides a comprehensive overview of the diseases including the DDx list to be considered for the diagnosis of each of the diseases.

(4) Disease Database[7] [18]: The Diseases Database is a medical textbook-like index and search portal that provides a cross-referenced index of human disease, medications, symptoms, signs, abnormal investigation findings, etc.

(5) eH&P[8] [19]: eH&P™ (custom History & Physical Exam™) is an innovative medical application software program for the H&P, progress notes, and clinical checklists.

[4] The Mayo Clinic is a nonprofit academic medical center based in Rochester, Minnesota, focused on integrated clinical practice, education, and research.

[5] WebMD is an American corporation known primarily as an online publisher of news and information pertaining to human health and well-being.

[6] Medscape is a website providing access to medical information for clinicians; the organization also provides continuing education for physicians and health professionals (Owner: WebMD).

[7] Provided by Medical Object-Oriented Software Enterprises Ltd

[8] Provided by ScyMed that is a medical information company developing medical decision systems™ and clinical application software programs (Apps) for Physicians and healthcare professionals.

Choosing a disease, it provides the user with an H&P form, which shows the signs, symptoms, and in general all the relevant items. This form can be best used for educational purposes and it should be noted that this system performs the opposite action comparing to the HMDS, since the HMDS actually starts with the symptoms in order to reach the diagnosis.

(6) ICD-10 [20]: ICD-10 is the 10th revision of the International Statistical Classification of Diseases and Related Health Problems (ICD), which is a medical classification list provided by the World Health Organization (WHO). This source assigns codes to diseases, signs and symptoms, abnormal findings, complaints, social circumstances, and external causes of injury or diseases.

1.6 Thesis Layout

Chapter 2 presents the research domain and describes the concepts related to the MDSs and the H&P. Chapter 3 explains the research motivation by concentrating on the software requirements, which is followed by a literature review with a focus on the state of the art of MDSs. Chapter 4 studies the concept of DDx in greater detail and introduces the HMASs as a means to implement this method. This chapter basically covers the analysis of the system and therefore describes the problem-solving approach. Chapter 5 includes the system design and accordingly illustrates the overall structure of the proposed system. Details of the functionality and the self-organization of the system are also presented in this chapter. Chapter 6 concentrates on the empowering of the system with the appropriate ML techniques, which allow the system to update its medical data and improve its performance. Next, chapter 7 presents a model of the system and describes its implementation. This chapter includes the description of the system behavior and demonstrates how the system actually performs the DDx. Assessment simulations are conducted in chapter 8, where the system is tested by providing appropriate real medical cases as input, verifying the output and comparing it with the actual results. This includes the evaluation of the functionality and the learning abilities of the system. Chapter 9 then concludes with a review on the research contributions and some suggestions for future works. The following table presents in more detail the content of each chapter (Table 3).

Table 3. Thesis Layout

Chapter	Content
Chapter 1: **Introduction**	This section provides a brief introduction into the research domain by presenting the research question and then the research contributions. This chapter also describes the research methods that are used in this project in order to tackle the research question. Also, the thesis layout is given here.
Chapter 2: **Research Domain**	This chapter describes the concepts related to the medical diagnosis systems and the history and physical examination.
Chapter 3: **Research Motivation**	This section consists of two major subsections: The software requirements and the related work. The software requirements subsection provides a description of the intended MDS, concentrating on its distinctive characteristics. The related work subsection is the literature review with a special focus on the state of the art, which shows the significance of the research contribution.
Chapter 4: **System Analysis: The Software Engineering Approach**	This chapter covers the system analysis process for the HMDS. For this purpose, the DDx domain, is studied in greater detail and proven to meet the characteristics of HMASs. Accordingly, the HMASs are also introduced in this chapter.
Chapter 5: **System Design: The Holonic Medical Diagnosis System (HMDS)**	This chapter covers the system design process for the HMDS. Details of the functionality and the self-organization of the system are also presented in this chapter.
Chapter 6: **System Design: Learning in the HMDS**	Three different ML techniques are applied to different aspects of the HMDS. Clustering is used to determine the initial organizational structure of the system, exponential Smoothing is used to update the medical data, and reinforcement learning has been applied in order to support the self-organization of this system. This chapter shows how these techniques were used and if necessary were adapted to the HMASs. As a result, the theoretical contributions of this study are presented in this chapter.
Chapter 7: **System Simulation**	This chapter implements a simulation model of the suggested system using the GAMA platform [21], which is a modeling and simulation development environment for building spatially explicit agent-based simulations. This chapter thus provides a brief introduction to GAML and shows how the HMDS has been simulated using this platform.
Chapter 8: **System Assessment**	This chapter includes the assessment simulations of the system. The tests monitor the system's general behavior in performing the H&P. Additionally, learning abilities of the system are examined by providing the system with appropriate inputs and evaluating the corresponding outputs.
Chapter 9: **Conclusion and Future Works**	This chapters reviews the research objectives and contributions, and eventually suggests some ideas for future works. Functionalities that were excluded from the system are also described here in greater detail as potential future work topics.

Chapter 2
RESEARCH DOMAIN[9]

"Most medical errors come not from bad reasoning based on well observed facts, but from good reasoning based on inaccurate information."

- Blaise Pascal (1623 - 1662)

As the aim of this project is to implement a medical diagnosis system, which is capable of guiding the history and physical examination, this chapter mainly focuses on the concepts related to these two subjects. The brief introduction to MDSs mainly aims to identify the main missing ability of these systems, i.e., the ability to guide the history and physical examination. Describing the history and physical examination this chapter also introduces some of the available systems that have considered this examination. This brief review shows that these systems either solely provide a digital form for this process or are teaching tools that do not guide the user in performing a focused examination.

[9] The author gratefully acknowledges the informative and encouraging discussions with Dr. Farzad Fakouri on the medical aspects of the project, and would like to express appreciation and gratitude for his knowledgeable insight and expertise, that greatly assisted the research.

2.1 Medical Diagnosis Systems (MDSs)

In previous decades, medical knowledge has been expanding exponentially. According to [22], the projected doubling time of medical knowledge, which was once 50 years in 1950, shortened to 7 years in 1980, and again to 3.5 years in 2010, is estimated to be 73 days by 2020. As the trend towards an increasingly digitalized healthcare industry plays a significant role in this expansion, information technology itself can be a solution to this exponential knowledge growth. In fact, this technology can help the healthcare providers to assimilate and apply this knowledge effectively.

As already mentioned, since the 1950s computer scientists have aimed to support and improve healthcare systems. Some uses of computer in medicine are Digital Medical Imaging (DMI), Patient monitoring systems (PMSs), Health Information Systems (HISs), and Clinical Decision Support Systems (CDSSs). This study exclusively concentrates on the latter systems that aim to assist the physicians and other health professionals in clinical decision-making tasks. As proposed by Robert Hayward of the Centre for Health Evidence, “Clinical decision support systems link health observations with health knowledge to influence health choices by clinicians for improved health care”. Some comprehensive overviews on the CDSSs are presented in [23, 24, 25].

A Diagnostic Decision Support System (DDSS) or Medical Diagnosis System (MDS) is a specific type of Clinical Decision Support Systems that is developed to provide an ordered list of potential diagnoses for given signs and symptoms. The physician then takes the suggested diagnoses together with the supportive information and determines which diagnoses might be relevant and which are not, and, if necessary, orders further tests to narrow down this list [23]. In fact, these systems are best used to remind the physicians about critical possible diagnoses that might have simply been ignored.

Available MDSs, including the state of the art (see section 3.2.2), mainly focus on finding the perfect link between the given input and their health knowledge. As a result, regardless of how powerful or ground-breaking they are in performing this action, it is always possible that their final strong deduction is based on some incomplete input, which may very likely lead to a misdiagnosis. To date, no computer-aided Decision Support System (DSS) has been introduced to address this issue and this is the reason why the available MDSs cannot be well integrated into the workflows of the healthcare institutions and hence are not widely used. In fact, in order to utilize these systems a physician should first provide such systems with the right input. When a physician performs the required examination (see section 2.2) to provide an MDS with the right input, if no complications occur, s(he) will reach the diagnosis too, and as a result the later use of the MDS would be of less value. Consequently, MDSs are used very seldom in practice and are exclusively used in complicated medical cases, where the information gathered by the physician has not led to a definitive diagnosis, or diagnosis is clear but patient-specific treatment is needed.

As a result, in order to use MDSs in practice, prior to a diagnosis process they should guide the user in providing the correct, all-encompassing input, so that a user with less medical knowledge such as a nurse can perform the examination and just pass the results to the physician for final approval. To this end, upon receiving the patient's complaint the system should be able to determine the missing information that is crucial for the final diagnosis and eventually demand it from the user. This is similar to what a physician does when listening to a patient. In patient encounter, (s)he would carefully listen to the symptoms explained by the patient, considers some potential diagnoses and then tries to gather enough evidence and supporting information to ensure the considerably lower probability of the other candidates by questioning the signs and symptoms that might have been simply ignored by the patient, or requesting the patient to undertake some medical examinations. During this process the physician is actually applying the DDx and hence the implementation of this method is in fact what the MDSs need in order to be able to guide the user in preparing the all-encompassing input.

As already mentioned, in medicine, a DDx is the distinguishing of a particular disease or condition from others exposing similar symptoms [5], which leads to more accurate diagnoses [6]. This systematic method is used to identify the presence of a disease entity where multiple alternatives are possible and intends to gather enough evidence and supporting information to shrink the probability of the other candidates. This is very critical, since a misdiagnosis leads to a delay in identifying the correct diagnosis, as well as an exposure to inappropriate medication that can lead to serious irrecoverable effects. In order to gain better insight into the use of differential diagnosis in medicine, the process, which implements this method, is described in next section.

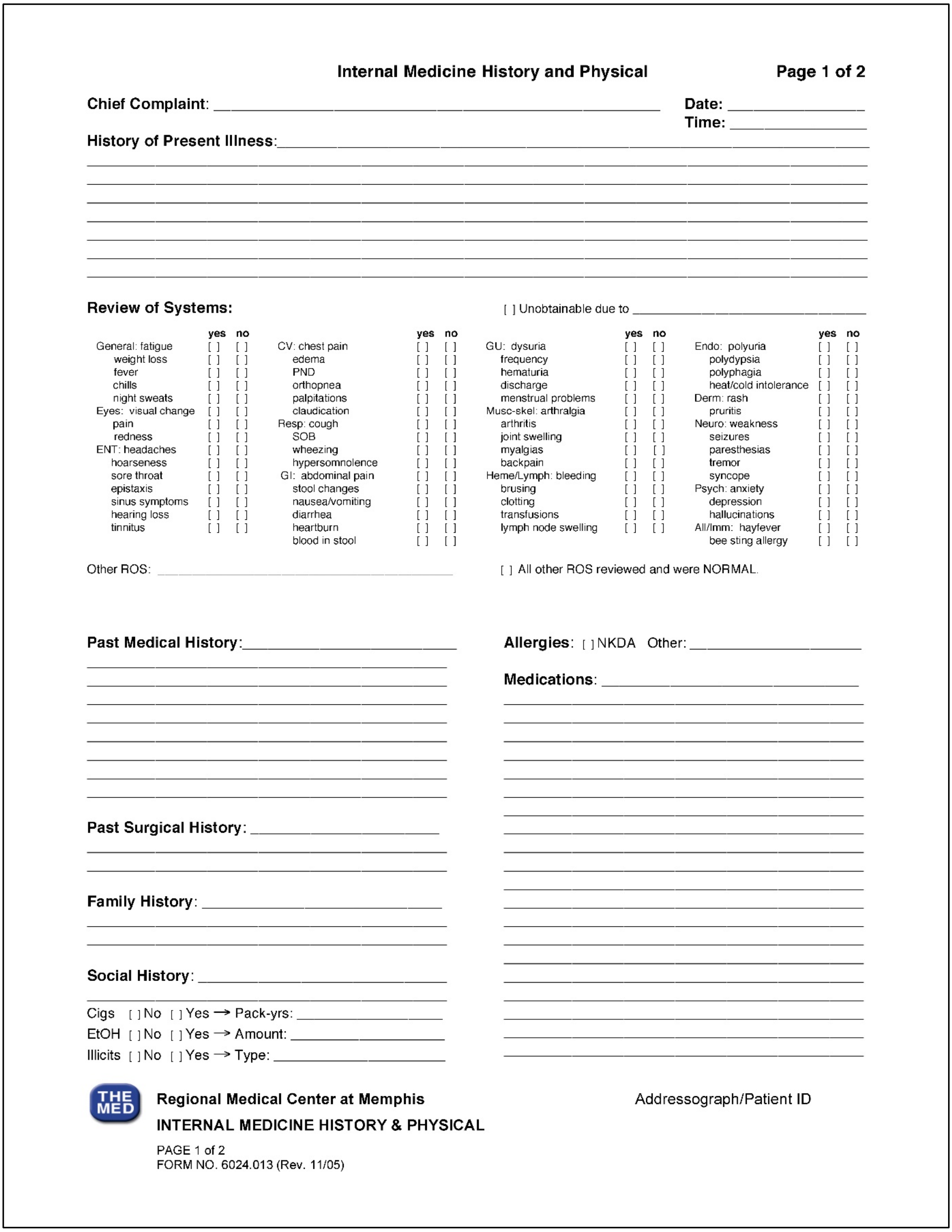
2.2 The History and Physical examination (H&P)

In a patient encounter, the DDx is used in a process called the History and Physical examination (H&P). The H&P is “a critical component of a patient encounter in which information relevant to present complaint is obtained, by asking questions about family and personal medical history and the organ systems examined in as great detail as necessary to manage the present condition or evaluate-workup-the patient” [3]. As the first step in the encounter with a patient, this examination organizes the patient data and allows the physician to narrow down the DDx list to a few possibilities. The physician can then narrow down this list even more using the results of executive physicals, such as laboratory and radiographic findings.

Adherence to the regulations related to H&P, and its documentation not only guarantee the execution of DDx, but also provide a reference document that may also be used as an important medical-legal document. As stated in [26, p. 772], “The major goal of the medical record is to serve as a repository of the clinician’s observations and analysis of the patient”. H&P typically initiates any clinician’s recorded interactions with a patient [26]. As stated by Charlie Goldberg, M.D., UCSD School of Medicine and VA Medical Center, San Diego, California, the written H&P serves several purposes:

1. It is an important reference document that provides concise information about a patient's history and exam findings at the time of admission.
2. It outlines a plan for addressing the issues which prompted the hospitalization. This information should be presented in a logical fashion that prominently features all of the data that’s immediately relevant to the patient's condition.
3. It is a means of communicating information to all providers who are involved in the care of a particular patient.
4. It allows students and house staff an opportunity to demonstrate their ability to accumulate historical and examination-based information, make use of their medical fund of knowledge, and derive a logical plan of attack. [27]

Figure 2 to Figure 5 illustrate two examples of H&P form.



Internal Medicine History and Physical **Page 1 of 2**

Chief Complaint: ______________________________ **Date**: ____________

Time: ____________

History of Present Illness: ______________________________

____

____

____

____

____

____

____

Review of Systems: [ ] Unobtainable due to ____________________

	yes	no		yes	no		yes	no		yes	no
General: fatigue	[ ]	[ ]	CV: chest pain	[ ]	[ ]	GU: dysuria	[ ]	[ ]	Endo: polyuria	[ ]	[ ]
weight loss	[ ]	[ ]	edema	[ ]	[ ]	frequency	[ ]	[ ]	polydypsia	[ ]	[ ]
fever	[ ]	[ ]	PND	[ ]	[ ]	hematuria	[ ]	[ ]	polyphagia	[ ]	[ ]
chills	[ ]	[ ]	orthopnea	[ ]	[ ]	discharge	[ ]	[ ]	heat/cold intolerance	[ ]	[ ]
night sweats	[ ]	[ ]	palpitations	[ ]	[ ]	menstrual problems	[ ]	[ ]	Derm: rash	[ ]	[ ]
Eyes: visual change	[ ]	[ ]	claudication	[ ]	[ ]	Musc-skel: arthralgia	[ ]	[ ]	pruritis	[ ]	[ ]
pain	[ ]	[ ]	Resp: cough	[ ]	[ ]	arthritis	[ ]	[ ]	Neuro: weakness	[ ]	[ ]
redness	[ ]	[ ]	SOB	[ ]	[ ]	joint swelling	[ ]	[ ]	seizures	[ ]	[ ]
ENT: headaches	[ ]	[ ]	wheezing	[ ]	[ ]	myalgias	[ ]	[ ]	paresthesias	[ ]	[ ]
hoarseness	[ ]	[ ]	hypersomnolence	[ ]	[ ]	backpain	[ ]	[ ]	tremor	[ ]	[ ]
sore throat	[ ]	[ ]	GI: abdominal pain	[ ]	[ ]	Heme/Lymph: bleeding	[ ]	[ ]	syncope	[ ]	[ ]
epistaxis	[ ]	[ ]	stool changes	[ ]	[ ]	brusing	[ ]	[ ]	Psych: anxiety	[ ]	[ ]
sinus symptoms	[ ]	[ ]	nausea/vomiting	[ ]	[ ]	clotting	[ ]	[ ]	depression	[ ]	[ ]
hearing loss	[ ]	[ ]	diarrhea	[ ]	[ ]	transfusions	[ ]	[ ]	hallucinations	[ ]	[ ]
tinnitus	[ ]	[ ]	heartburn	[ ]	[ ]	lymph node swelling	[ ]	[ ]	All/Imm: hayfever	[ ]	[ ]
			blood in stool	[ ]	[ ]				bee sting allergy	[ ]	[ ]

Other ROS: ____________________ [ ] All other ROS reviewed and were NORMAL.

Past Medical History: ____________________

____

____

____

____

____

____

____

____

Past Surgical History: ____________________

____

____

Family History: ____________________

____

____

Social History: ____________________

____

Cigs [ ] No [ ] Yes → Pack-yrs: ____________

EtOH [ ] No [ ] Yes → Amount: ____________

Illicits [ ] No [ ] Yes → Type: ____________

Allergies: [ ] NKDA Other: ____________

Medications: ____________________

____

____

____

____

____

____

____

____

____

____

____

____

____

____

____

____

____

____

____

THE MED

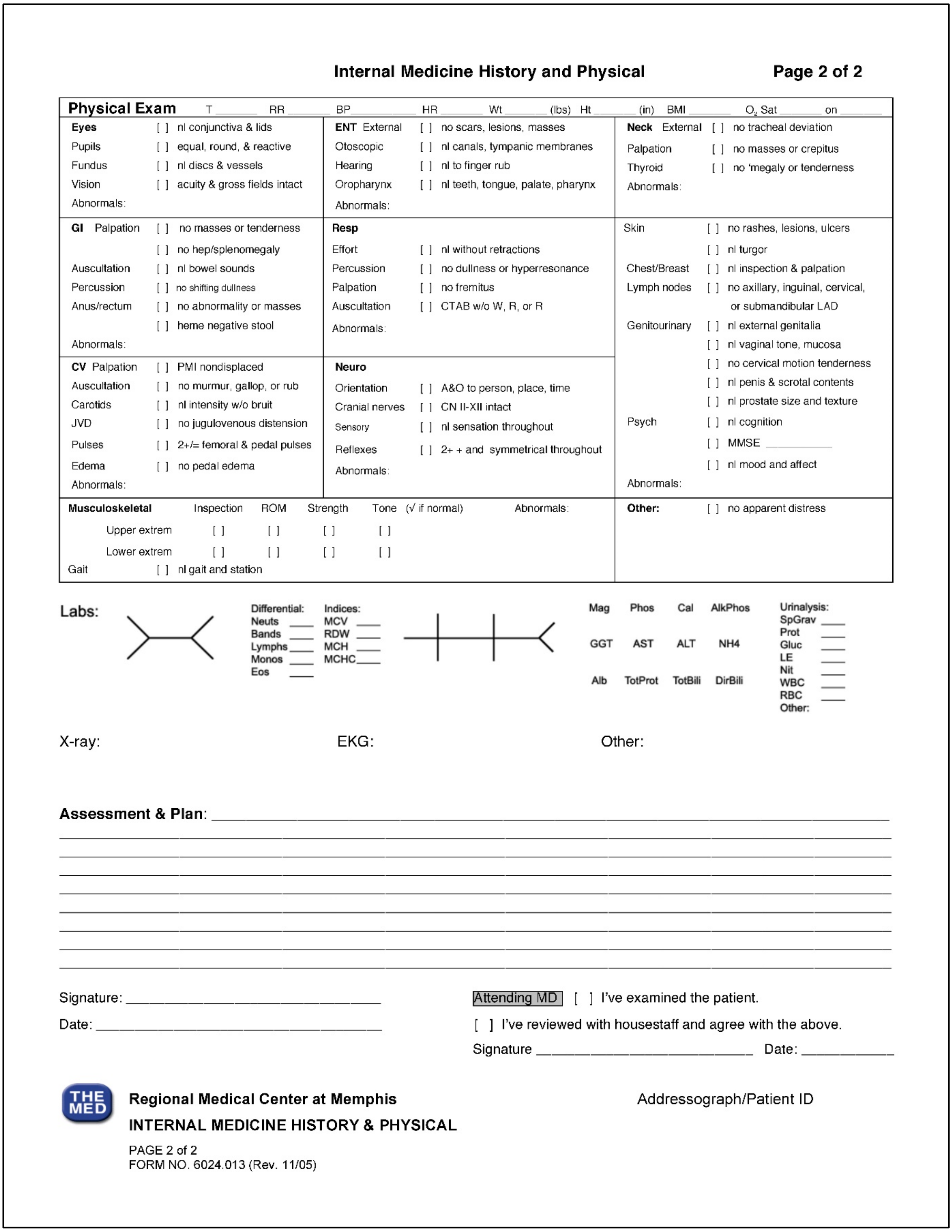
Regional Medical Center at Memphis

INTERNAL MEDICINE HISTORY & PHYSICAL

PAGE 1 of 2
FORM NO. 6024.013 (Rev. 11/05)

Addressograph/Patient ID

Figure 2. H&P form (example 1) – page 1 of 2



Internal Medicine History and Physical **Page 2 of 2**

Physical Exam T ______ RR ______ BP ________ HR ______ Wt ______ (lbs) Ht ______ (in) BMI ______ O_2 Sat ______ on ______

Eyes [ ] nl conjunctiva & lids
Pupils [ ] equal, round, & reactive
Fundus [ ] nl discs & vessels
Vision [ ] acuity & gross fields intact
Abnormals:

ENT External [ ] no scars, lesions, masses
Otoscopic [ ] nl canals, tympanic membranes
Hearing [ ] nl to finger rub
Oropharynx [ ] nl teeth, tongue, palate, pharynx
Abnormals:

Neck External [ ] no tracheal deviation
Palpation [ ] no masses or crepitus
Thyroid [ ] no 'megaly or tenderness
Abnormals:

GI Palpation [ ] no masses or tenderness
[ ] no hep/splenomegaly
Auscultation [ ] nl bowel sounds
Percussion [ ] no shifting dullness
Anus/rectum [ ] no abnormality or masses
[ ] heme negative stool
Abnormals:

Resp
Effort [ ] nl without retractions
Percussion [ ] no dullness or hyperresonance
Palpation [ ] no fremitus
Auscultation [ ] CTAB w/o W, R, or R
Abnormals:

Skin [ ] no rashes, lesions, ulcers
[ ] nl turgor
Chest/Breast [ ] nl inspection & palpation
Lymph nodes [ ] no axillary, inguinal, cervical, or submandibular LAD
Genitourinary [ ] nl external genitalia
[ ] nl vaginal tone, mucosa
[ ] no cervical motion tenderness
[ ] nl penis & scrotal contents
[ ] nl prostate size and texture
Psych [ ] nl cognition
[ ] MMSE ________
[ ] nl mood and affect
Abnormals:

CV Palpation [ ] PMI nondisplaced
Auscultation [ ] no murmur, gallop, or rub
Carotids [ ] nl intensity w/o bruit
JVD [ ] no jugulovenous distension
Pulses [ ] 2+/= femoral & pedal pulses
Edema [ ] no pedal edema
Abnormals:

Neuro
Orientation [ ] A&O to person, place, time
Cranial nerves [ ] CN II-XII intact
Sensory [ ] nl sensation throughout
Reflexes [ ] 2+ + and symmetrical throughout
Abnormals:

Musculoskeletal	Inspection	ROM	Strength	Tone (√ if normal)	Abnormals:
Upper extrem	[ ]	[ ]	[ ]	[ ]	
Lower extrem	[ ]	[ ]	[ ]	[ ]	

Gait [ ] nl gait and station

Other: [ ] no apparent distress

Labs:

Differential: Neuts ___ Bands ___ Lymphs ___ Monos ___ Eos ___

Indices: MCV ___ RDW ___ MCH ___ MCHC ___

Mag Phos Cal AlkPhos
GGT AST ALT NH4
Alb TotProt TotBili DirBili

Urinalysis: SpGrav ___ Prot ___ Gluc ___ LE ___ Nit ___ WBC ___ RBC ___ Other:

X-ray: EKG: Other:

Assessment & Plan: ____

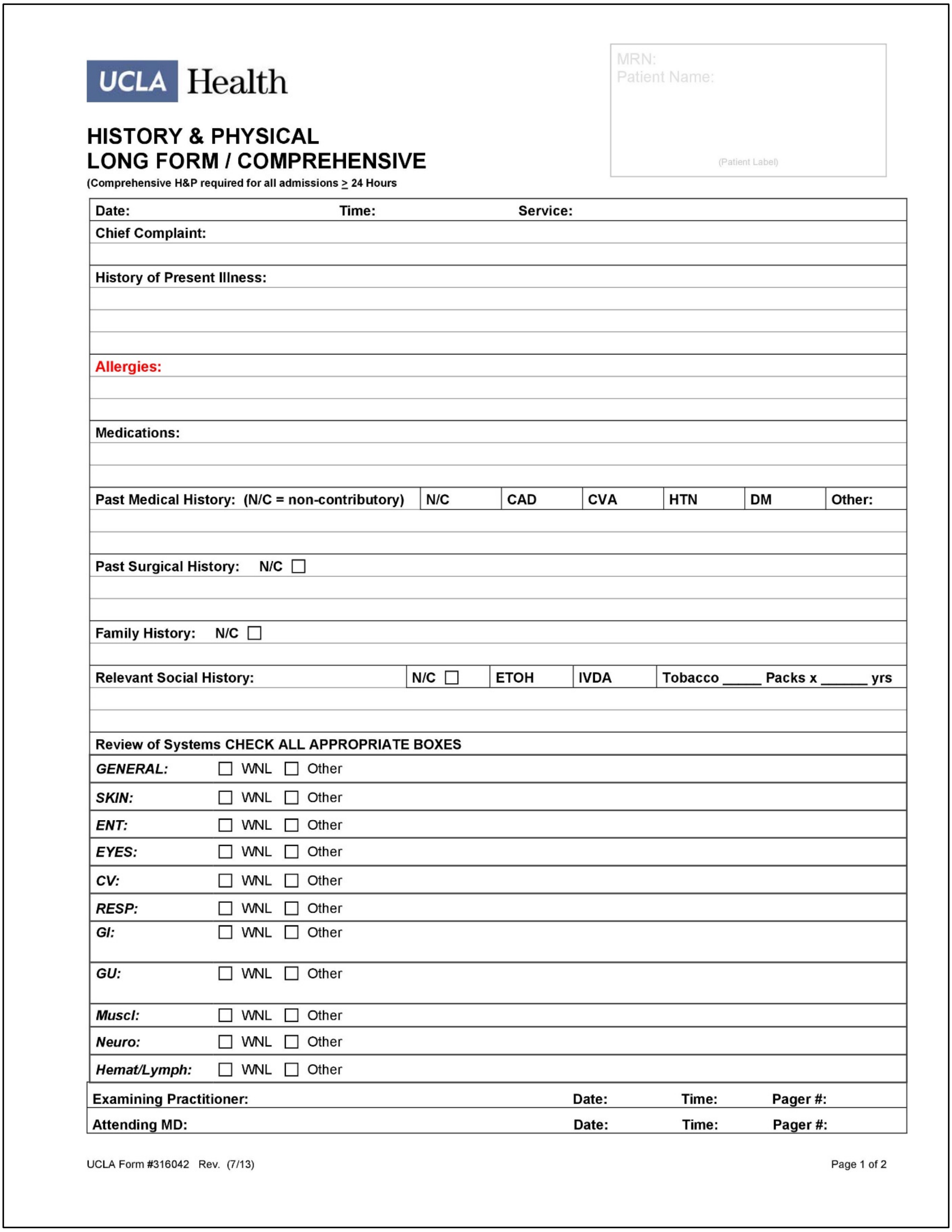
Signature: ______________________________
Date: __________________________________

Attending MD [ ] I've examined the patient.
[ ] I've reviewed with housestaff and agree with the above.
Signature ________________________ Date: __________

Regional Medical Center at Memphis
INTERNAL MEDICINE HISTORY & PHYSICAL
PAGE 2 of 2
FORM NO. 6024.013 (Rev. 11/05)

Addressograph/Patient ID

Figure 3. H&P form (example 1) – page 2 of 2



UCLA Health

MRN:
Patient Name:

(Patient Label)

HISTORY & PHYSICAL
LONG FORM / COMPREHENSIVE
(Comprehensive H&P required for all admissions ≥ 24 Hours)

Date: **Time:** **Service:**

Chief Complaint:

History of Present Illness:

Allergies:

Medications:

Past Medical History: (N/C = non-contributory)	**N/C**	**CAD**	**CVA**	**HTN**	**DM**	**Other:**

Past Surgical History: **N/C** □

Family History: **N/C** □

Relevant Social History:	**N/C** □	**ETOH**	**IVDA**	**Tobacco _____ Packs x ______ yrs**

Review of Systems CHECK ALL APPROPRIATE BOXES

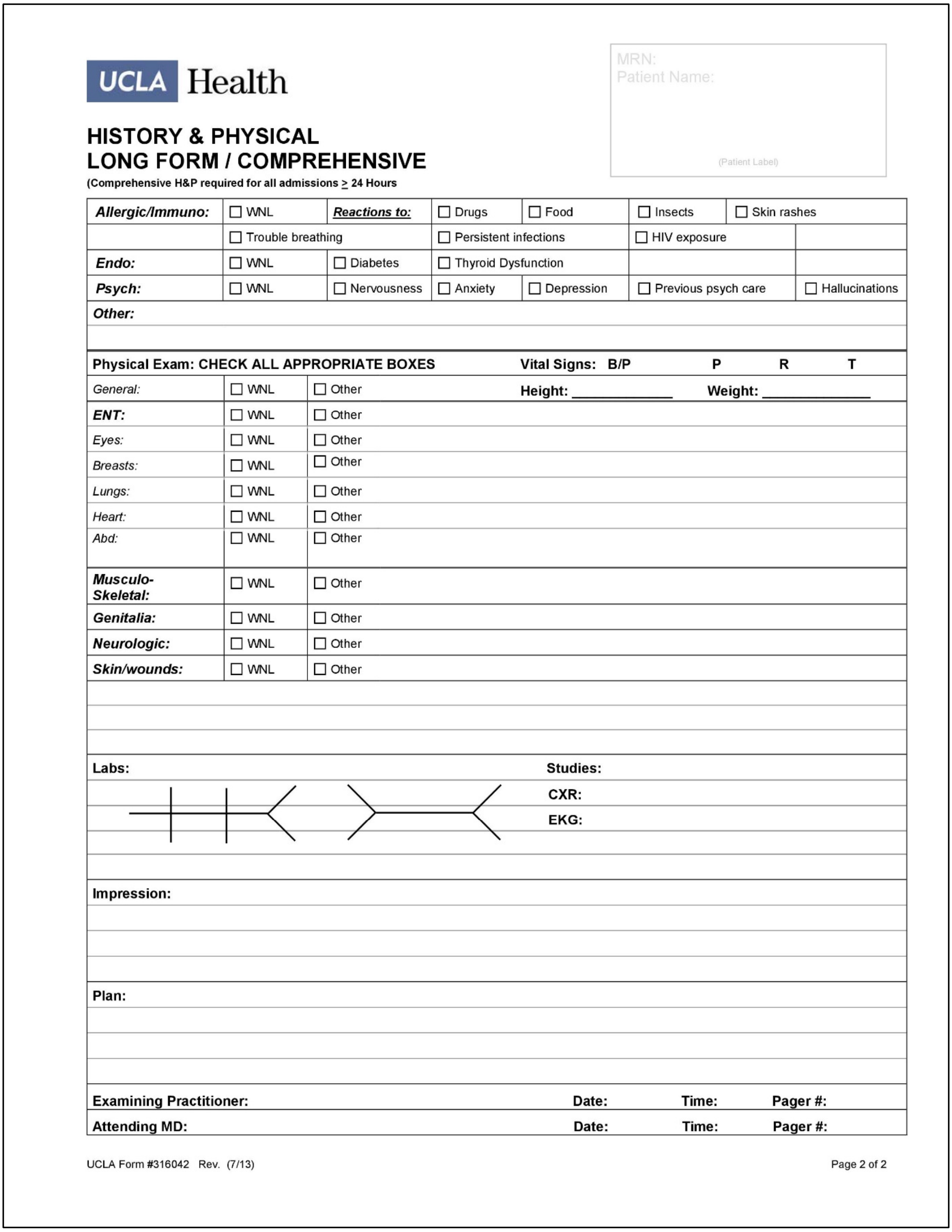
GENERAL:	□ WNL	□ Other
SKIN:	□ WNL	□ Other
ENT:	□ WNL	□ Other
EYES:	□ WNL	□ Other
CV:	□ WNL	□ Other
RESP:	□ WNL	□ Other
GI:	□ WNL	□ Other
GU:	□ WNL	□ Other
Muscl:	□ WNL	□ Other
Neuro:	□ WNL	□ Other
Hemat/Lymph:	□ WNL	□ Other

Examining Practitioner:	**Date:**	**Time:**	**Pager #:**
Attending MD:	**Date:**	**Time:**	**Pager #:**

UCLA Form #316042 Rev. (7/13)

Page 1 of 2

Figure 4. H&P form (example 2) – page 1 of 2



UCLA Health

MRN:
Patient Name:

(Patient Label)

HISTORY & PHYSICAL
LONG FORM / COMPREHENSIVE
(Comprehensive H&P required for all admissions ≥ 24 Hours

Allergic/Immuno:	□ WNL	***Reactions to:***	□ Drugs	□ Food	□ Insects	□ Skin rashes
	□ Trouble breathing		□ Persistent infections		□ HIV exposure	
Endo:	□ WNL	□ Diabetes	□ Thyroid Dysfunction			
Psych:	□ WNL	□ Nervousness	□ Anxiety	□ Depression	□ Previous psych care	□ Hallucinations
Other:						

Physical Exam: CHECK ALL APPROPRIATE BOXES			**Vital Signs: B/P P R T**
General:	□ WNL	□ Other	**Height:** ____________ **Weight:** ____________
ENT:	□ WNL	□ Other	
Eyes:	□ WNL	□ Other	
Breasts:	□ WNL	□ Other	
Lungs:	□ WNL	□ Other	
Heart:	□ WNL	□ Other	
Abd:	□ WNL	□ Other	
Musculo-Skeletal:	□ WNL	□ Other	
Genitalia:	□ WNL	□ Other	
Neurologic:	□ WNL	□ Other	
Skin/wounds:	□ WNL	□ Other	

Labs:

Studies:

CXR:

EKG:

Impression:

Plan:

Examining Practitioner: **Date:** **Time:** **Pager #:**

Attending MD: **Date:** **Time:** **Pager #:**

UCLA Form #316042 Rev. (7/13)

Page 2 of 2

Figure 5. H&P form (example 2) – page 2 of 2

A comprehensive description of the different sections of an H&P report are provided by Charlie Goldberg, M.D., UCSD School of Medicine and VA Medical Center, San Diego, California:

- **Chief Complaint or Chief Concern (CC):** One sentence that covers the dominant reason(s) for hospitalization.
- **History of Present Illness (HPI):** This section covers all the events leading to the patient's arrival in the Emergency Room (ER)[10] (or the floor[11], if admission was arranged without an ER visit). The HPI should provide enough information without being too inclusive.
- **Past Medical History (PMH):** This should include any illness (past or present) that the patient is known to have, ideally supported by objective data.
- **Past Surgical History (PSH):** All past surgeries should be listed, along with the rough date when they occurred.
- **Medications (MEDS):** Includes all currently prescribed medications as well as over the counter and non-traditional therapies. Dosage and frequency should be noted.
- **Allergies/Reactions (All/RXNs):** Identify the specific reaction that occurred with each medication.
- **Social History (SH or SHx):** This is a broad category which includes: Alcohol Intake, Cigarette smoking, Other Drug Use, Marital Status, Sexual History, Work History, and others (e.g. travel, pets, hobbies).
- **Family History (FH or FHx):** This includes history of illnesses within the patient's immediate family.
- **Review of Systems (ROS):** The review of systems is a list of questions, arranged by organ system, designed to uncover dysfunction and disease, i.e., the patient's positives and negatives related to the chief complaint. The ROS is generally noted at the end of the HPI.
- **Physical Exam (PE):** is the process by which a medical professional investigates the body of a patient for signs of disease.

[10] The emergency room also referred to as the emergency department, is where a patient is first seen after a sudden and serious illness or injury.

[11] A floor unit in a hospital is where a patient is cared for when (s)he doesn't require especially close monitoring.

- **Lab Results, Radiologic Studies, EKG[12] Interpretation, Etc.**
- **Assessment and Plan (A&P):** Assessment includes a discussion of the Differential Diagnosis (DDx) and supporting history and exam findings. The plan addresses treatments of each problem. [27]

In this study, for simplicity reasons, the H&P sections are categorized and referred to as follows[13]:

1. Chief Complaint (CC)
2. Medical History (Hx): HPI, PMH, PSH, MEDS, ALL/RXNs, SHx, FHx, ROS
3. Physical Exam (PE): PE, Lab Results
4. Assessment and Plan (A&P)

As it can be seen on the sample H&P form (see Figure 2 and Figure 3), this document covers a wide range of signs and symptoms. As stated in [27], “Knowing what to include and what to leave out of the H&P report is largely dependent on experiences and one’s understanding of illness and pathophysiology”. Accordingly, special knowledge and experience is needed to perform the H&P successfully, and here again DDx concerns could keep the whole process focused. “In fact, irrelevant questions and tests should be ignored and every single piece of information should be used in order to narrow down the possibilities” [8, p. 274]. According to the World Health Organization (WHO) [2] the total number of diseases is about 30.000 and the International Statistical Classification of Diseases and Related Health Problems - 10th Revision (ICD-10) [20] includes roughly 1.000 signs and symptoms. These numbers indicate that performing a focused H&P is of imminent importance.

If, for example, a medical student is unaware that chest pain is commonly associated with coronary artery disease, (s)he would be unlikely to mention other coronary risk-factors when writing the history, and until enough experience is gained, the write-ups will be somewhat poorly focused [27]. Moreover, not knowing the right questions to be asked the H&P may take a long

[12] Electrocardiography (ECG or EKG)

[13] It should be noted that this is not an attempt to define a new medical note format, and that this classification solely aims to simplify the reference to the H&P stages. A comprehensive description of medical note formats is presented in [28] (see Table 4).

time. A physician, however, is capable of keeping this process as focused as possible and identifies the relevant questions and examinations based on the chief complaint, and also during the examination when new evidences may occur.

Basically, upon receiving the CC, the physician will have the initial DDx list to begin with, and this will lead to a focused H&P. After obtaining the relevant information the physician will be able to narrow down this list and report the revised DDx list (Figure 6).

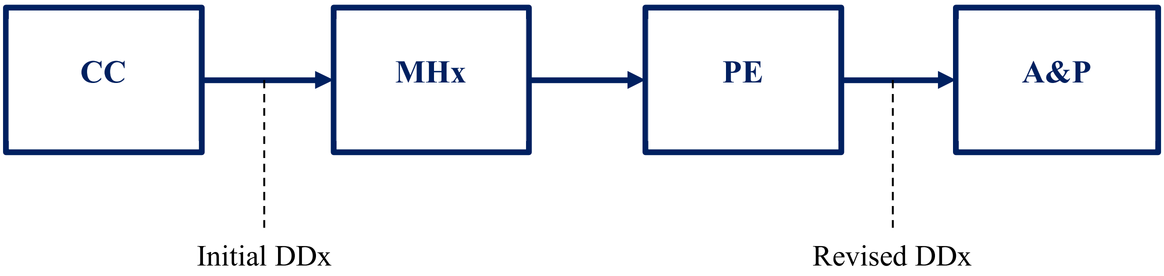


Figure 6. Initial DDx list and revised DDx list

Different methods of documentation have been defined and deployed by health care providers to write the H&P report. A comprehensive description of medical note formats is presented in Table 4 [28, p. 544]. According to [28, p. 542] the SOAP (Subjective, Objective, Assessment, Plan) approach [14] to notes, first recommended in the 1960s, “did a remarkable job of conveying the physician’s thought process, supporting data, and conclusions”. However, recently the difficulty of finding the most relevant information in the records, i.e., the assessment and the plan, has led to a proposal for a rearrangement of the traditional SOAP note. As stated in [28, p. 543], “The APSO (Assessment, Plan, Subjective, Objective) note [29, 30] was created for inpatient daily progress notes, a situation in which the patient’s concern is unlikely to change dramatically on a daily basis and was not intended for use in outpatient clinics”. In the age of digitalized healthcare, large quantities of data are included in the patient notes that have no connection to or do not clearly convey the physician’s thought process. The CAPS (Concern, Assessment, Plan, Supporting information) approach [28] introduced in 2016 transfers this data to the bottom of the note, and allows more efficient communication about the true purpose for the patient’s visit, the diagnosis, and the physician’s approach to resolving the patient’s concern. The CAPS note aims to “retains the patient-centered, problem-oriented spirit of the SOAP format, while moving potentially supportive yet distracting data fields to later in the note. Thus, it is applicable to both inpatient and outpatient settings" [28, p. 543].

Table 4. Comparison of medical note formats

Format	Structure
SOAP [31]	**S**ubjective information Chief concern History of present Illness Patient's medical, surgical, family, social history **O**bjective information Physical examination Laboratory and test data **A**ssessment **P**lan
APSO [29, 30]	**A**ssessment **P**lan **S**ubjective information (as above) **O**bjective information (as above)
CAPS [28]	**C**oncern Patient's chief concern History of present illness, injury **A**ssessment Diagnosis with clinical reasoning **P**lan Itemized list of actions to address patient's concern and condition **S**upporting information (objective and subjective) Vital signs and physical examination Results of laboratory, radiographic, other tests Comprehensive review of systems Patient's medical, surgical, family, social history Current medications Allergies

Regardless of which note format is used, healthcare providers are now saving the data in digital format. As stated in [32], "the Electronic Health Record (EHR) is an evolving concept defined as a longitudinal collection of electronic health information about individual patients and populations. Primarily, it will be a mechanism for integrating health care information currently collected in both paper and Electronic Medical Records (EMRs) for the purpose of improving quality of care". Based on a report on a series of studies conducted at four institutions that provide ambulatory care, the use of EHRs was viewed as the most promising tool for improving the overall quality, safety, and efficiency of the health delivery system [33].

According to [26, p. 774] there are many advantages to the use of the EMR:

1. Unlike the paper record, the EMR can potentially be used by anyone who needs it at any time. It can also be accessed easily from remote sites, such as a clinic across the town or even across the country.
2. It is unlikely that data will be lost or misplaced. With an appropriate back-up mechanism, the EMR should serve as a permanent record of an individual's interaction with the health care system.
3. With the availability of all the patient's data, new views and other summaries can be generated instantaneously.
4. With the potential for the incorporation of reminders and decision support, the likelihood of mistakes and omissions should decrease.
5. In addition to benefiting the individual patient, the EMR is also likely to benefit the larger population. Clinical research will likely be enhanced, as researchers have easier access to information about patients that will increase the understanding of the disease and its treatment. Screening and other preventive measures will become easier to implement as patients of various attributes (i.e., gender, age, presence of other risk factors) can be identified and contacted.

As mentioned, any clinician's recorded interactions with a patient usually begin with the H&P [26]. Electronic Health Record Systems (EHRSs) provide their users with digital H&P forms. The AmericanEHR [34] is an online community of clinicians that provides a trusted source for unbiased EHRS recommendations. This reference includes the list of certified EHRSs and their ratings. According to the reviews done by the hospitalists[14] [34], EpicCare Ambulatory EMR [35] and Cerner PowerChart Ambulatory [36] are the best rated EHRS with respect to their specialty.

Available EHRS are able to capture patient's data, analyze it, and provide insight for clinicians, however, in terms of the H&P they solely provide a digital form and do not guide the user in performing a focused examination. Therefore, it is still the user who has to decide on the right

[14] Hospital medicine in the medical specialty concerned with the care of acutely ill hospitalized patients. Physicians whose primary professional focus is caring for hospitalized patients only while they are in the hospital are called hospitalists [237].

questions and exams. A number of computer programs and mobile applications have been designed that aim to assist their users in performing the H&P, however they are mostly teaching tools and none of them are designed to guide the diagnosis process. Some notable examples of such applications are:

1. eH&P™ is a clinical application designed and developed by ScyMed®, which is "a medical information technology company developing medical decision systems and clinical applications for physicians and healthcare professionals" [19]. eH&P™ is not a diagnostic program but a unique decision support tool that is capable of creating disease-specific H&Ps that can be used in delivery of care, teaching and research activities. Therefore, this program does not intend to guide the decision process, but to create the H&P report for a specific case, marking the elements that should have been questioned or examined.
2. Smart Medical Apps - H&P is a "comprehensive guide to medical histories, physical examination and documentation of findings". It includes "instruction guides, which assist medical students in each element of history-taking and physical examination", and it is the "first application to translate checklist findings into grammatically correct progress note for presentation". Additionally, its "video tutorials provide detailed instruction in correctly performing physical examination components" [37].
3. Clinicals – History & Physical designed by Medical Gear "offers medical students a clear structured and systematic approach to history taking", which is "divided into three main sections: General History, Symptomatology, and Physical Examination" [38].
4. The History & Physical Exam a-pocketcards developed by Börm Bruckmeier Publishing LLC provides "a complete handy tool for the physician to use in the clinical set-up", with "comprehensive coverage of the history and examination of a patient". The tool includes "a detailed checklist for the physical examination of all system of the body", and "a detailed checklist of the review of all the system of the body" [39].

To put it in a nutshell, focused H&P is the key to a flawless diagnosis and the available EHRSs provide their users with digital H&P forms, however, they do not guide the user in performing a focused H&P and it is still the user who has to decide on the right questions and examinations. The advantages of a system that is capable of guiding a focused H&P will be discussed in the next chapter, which will then eventually lead to the introduction of the research motivation.

Chapter 3
RESEARCH MOTIVATION

This section consists of two major subsections: The software requirements and the related work. The software requirements subsection provides a description of the intended medical diagnosis system, concentrating on its distinctive characteristics. The second subsection, i.e., the related work, is the literature review with a special focus on the state of the art. This part serves as a survey of the medical diagnosis systems, covers the different approaches to the research problem from a computer scientist's perspective, and includes critical assessments of the state of the art in order to redemonstrate the significance of the research contribution.

3.1 Software Requirements

As already mentioned, MDSs are developed to provide an ordered list of potential diagnoses for given signs and symptoms. It should be noted that the output of such systems is still to be controlled by a physician, and these systems are solely meant to remind about critical possible diagnoses that might have simply been ignored. In addition, a valid diagnosis demands a valid diagnosis request and as an MDS does not aim to guide the user in providing the right input this data should be carefully provided by a physician too (see Figure 7 - a). A diagnosis request is not just the patient's complaint but the combination of the chief complaint together with all the information about the presence and/or absence of the relevant signs and symptoms, which can be specified based on the DDx method (see Figure 7 - b). Accordingly, as in patient encounter this method is applied in H&P, the original input of the MDSs is the output of H&P, and the output of it is in fact the revised A&P (see Figure 7 - c).

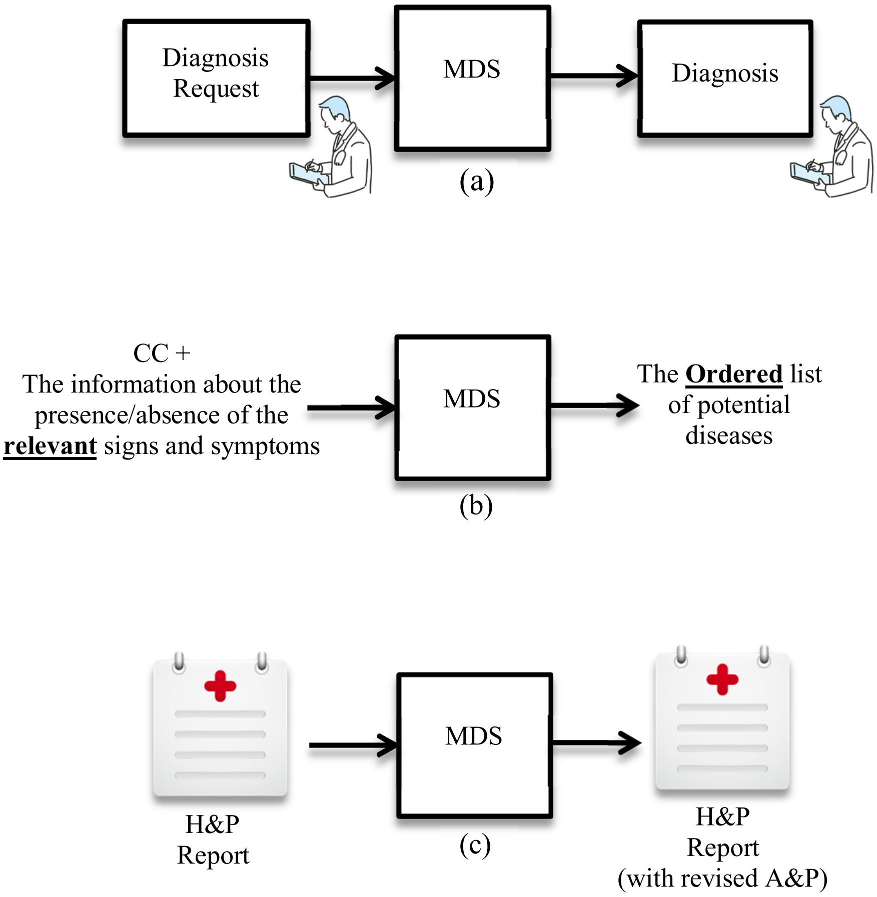

Diagnosis Request
MDS
Diagnosis
(a)
CC +
The information about the presence/absence of the **relevant** signs and symptoms
MDS
The **Ordered** list of potential diseases
(b)
H&P Report
MDS
H&P Report (with revised A&P)
(c)

Figure 7. input and output in MDSs

In other words, the available MDSs including the state of the art (see section 3.2.2) merely support the last stage of the H&P, i.e., the assessment and plan step, and use the output of the preceding stages as their input. This is exactly one of the limitations of these systems, and the reason why they cannot be well integrated into the clinical workflow of the hospitals[15]. As explained in section 1.1, when a physician performs this examination to provide an MDS with

[15] For more information on the other reasons for a lack of motivation to use MDSs please confer to [236].

the right input, since s(he) is basically applying the DDx method if no complications occur, at the end of the H&P s(he) will reach the diagnosis too, and as a result the later use of the MDS would be of less value. Consequently, MDSs are used very seldom in practice and are exclusively used in complicated medical cases where the information gathered in the first three stages of H&P has not led to a definitive assessment and plan. This shortcoming, however, should not be a reason to ignore the benefits of using these systems.

The state of the art of medical diagnosis systems includes IBM Watson [40] and Isabel [41, 42]. These systems are introduced in greater detail in the next section, however, in order to introduce the research motivation and eventually the contribution of this research a brief description is provided here. IBM Watson is claimed to be the most powerful artificial intelligence-based system capable of performing medical diagnosis. This system is a cognitive computer system developed by IBM that has the capability of understanding the natural language. From the technical point of view, upon receiving a request, Watson searches a large knowledge base, generates potential hypotheses, and then initiates another search to collect evidence that supports them. Isabel on the other hand, is a CDSS that facilitates diagnostic reminders and DDx [41, 42]. A systematic review and meta-analysis of DDx generators was conducted in [43], according to which Isabel was associated with the highest rates of diagnosis retrieval[16]. However, as stated in [44], Isabel is still too slow, and its accuracy drops significantly if only limited information is available [45]. As Isabel fails to guide the user in performing a focused H&P, using this system the user may end up with a misdiagnosis that is caused by incomplete input.

One can clearly deduce that the most common shortcoming of these systems is that they do not guide the users in providing the right input. As stated already, no matter how powerful they are in connecting the patient's medical history to medical knowledge, they cannot necessarily guarantee an error-free diagnosis as it is always possible that their final strong deduction is based on some incomplete input, and hence a misdiagnosis. The H&P is the key to a flawless diagnosis and in fact, as discussed in [46]: „No matter how good you are at diagnosing and treating, unless you asked the right questions in a timely manner, all the knowledge in the world won't be helpful.“

[16] The ability to generate DDx list for given input should not be confused with the ability to guide the H&P based on the concept of DDx.

H&P is originally performed by physicians, however, as the shortage of medical doctors has been worsening in the recent years, roles such as Physician Assistant (PA) and Nurse Practitioner (NP) have been defined to ease the load. PAs and NPs are qualified to perform the H&P steps, diagnose medical problems and carry out necessary treatments mainly under the supervision of physicians. Undertaking the H&P, they help the doctors to see more patients in a certain period of time, as they would then just need to review and asses the already prepared H&P report. In case of a complex medical case available MDSs can be used for reassessment or creating patient-specific treatment plans. As shown in Table 5 and Figure 8, the last stage of H&P, i.e. the A&P, is supported by IBM Watson and Isabel. These systems receive the results of the three first stages of H&P as their input and provide the user with the respective assessment and plan, i.e. the diagnosis and treatment. Of course, the results are then given to a physician for approval and if changes are needed the H&P report will be revised.

Table 5. The H&P sections that can be supported by IBM Watson/Isabel

		Hospital		Hospital using IBM Watson		Hospital using Isabel		
						Pro/Active Intelligence		Symptom Checker
H&P	CC	PHY	PA/NP	PHY	PA/NP	PHY	PA/NP	User
	Hx	PHY	PA/NP	PHY	PA/NP	PHY	PA/NP	-
	PE	PHY	PA/NP	PHY	PA/NP	PHY	PA/NP	-
	A&P	PHY	PA/NP[1]	IBM Watson[1]		Isabel Pro/Active Intelligence[1]		Isabel Symptom Checker[2]

[1] Physician's confirmation needed [2] Results to be discussed with physician

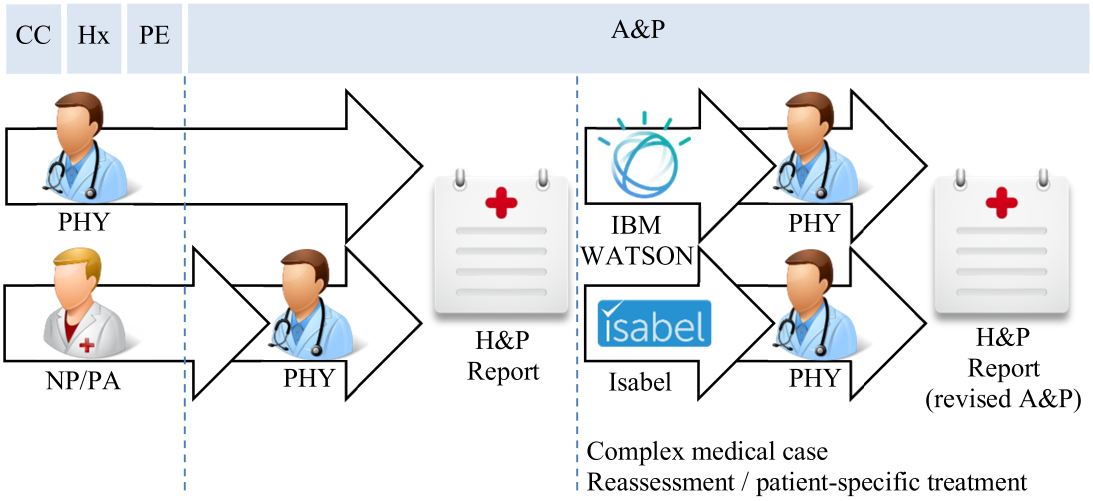


Figure 8. The input of the MDSs

Although PAs and NPs are able to compensate for the shortage of doctors to some extent, they do not solve the problem completely. Both groups require a special formal degree of education and years of experience; and there are also never enough PAs and NPs available. This is where artificial intelligence could help. In fact, a system capable of undertaking the H&P could save the physician time and guide less experienced nurses in performing this examination. Upon receiving the CC this system should be able to suggest the right signs and symptoms to be checked and remind the relevant physical exams. Gathering the patient's data, the system is then expected to suggest some possible diagnoses accordingly (see Figure 9). This study will show how a Holonic Multi-Agent System (HMAS) approach (see section 4.2) can simply enable a system to perform this process based on DDx concerns.

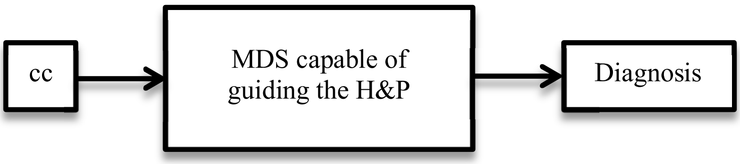


Figure 9. Input and output in an MDS capable of guiding the H&P

The research question of the general project is hence whether a holonic multi-agent architecture could support the implementation of the DDx method and consequently improve the use of medical diagnosis systems. Accordingly, the development of the HMDS, which is capable of performing DDx, is the practical contribution of this work. Moreover, as is the case with any intelligent system, the functionality of this system should be supported by appropriate machine learning techniques, and as a result, the adaptation of such techniques to the holonic approach used in this project is the conceptual/theoretical contribution of this work.

As mentioned in section 1.2 following functionalities signify the expectations from the final software product[17]:

1. Functional Requirements:
    1.1. For the purpose of DDx:
        1.1.1. The system allows the user to enter its chief complaint: The system allows the user to enter its diagnosis request in the form of a combination of signs and symptoms, which are recognizable by the system and consequently selectable

[17] For software requirements specifications please refer to section 1.2.

by the user. As mentioned already, within the scope of the current project the system does not intend to support natural language processing and it is assumed that the user of the system is capable of covering this need, matching the patient's words with the equivalent words provided and recognizable by the system.

1.1.2. The system questions relevant signs and symptoms: After receiving the chief complaint, based on DDx method the system considers some potential diagnoses and starts questioning the relevant signs and symptoms that are not mentioned by the user.

1.1.3. The system allows the user to add signs and symptoms to its diagnosis request: Questioning the relevant signs and symptoms, the system allows the user to enter the revised diagnosis request in the same format.

1.1.4. The system displays the diagnosis: The most probable diseases together with all the relevant unspecified signs, symptoms or test results are shown to the user.

1.2. For the purpose of data modification:

1.2.1. The system allows the expert to enter and modify available medical information: The expert can initialize and constantly modify the DDP and any other relevant information about the diseases directly and/or by providing the system with the right feedback.

1.2.2. The system allows the expert to delete eradicated diseases: The expert is able to manage the list of the diseases that are recognizable by the system and in case of an eradicated disease, this disease can be removed from the system.

1.2.3. The system displays the diagnosis result and allows the expert to enter feedback: Receiving the feedback, the system will be able to learn and improve its performance.

2. Nonfunctional Requirements (Quality Requirements): Adaptability (Machine learning abilities)

2.1. Updating the DDPs according to new instances: At the initialization stage the values of the signs and symptoms saved for each disease solely show the possibility of absence or presence in a single disease by assigning value 0.0 to absent signs and symptoms and value 1.0 to existing ones. However, these values can be updated based on the cases of the diseases in order to represent diseases with more realistic values. This implies that the system can also rely on the statistical information in order to improve the quality of its output.

2.2. Updating the regional distribution of diseases: The system can keep track of the distribution of the diseases by considering the new cases of the diseases. This data can also help the system to improve the quality of its output by considering the frequency of the different diseases in a DDx.

3.2 Related Work

3.2.1 Literature Review

The use of computers in medicine has always been an appealing research topic since 1950s [47, 48]. The pioneering methods (e.g. clinical algorithms and clinical databanks that include analytic functions) [49] were very limited and this was the case until early 1970s when researchers started to use artificial intelligence in this field. During 1970s the majority of the leading researches were based on Expert Systems (ESs) [50]. In 1980s and 90s, systems based on fuzzy set theory, Bayesian belief network, and Neural Networks (NN) were developed [51].

With the introduction of the agent concept in 1990 [52], medical systems started taking advantage of Multi-Agent System (MAS) technology. This paradigm, with its distributed architecture, is a promising solution to the shortcomings of the old methods, and furthermore offers much more flexibility, adaptability and scalability. Having these advantages in mind, researchers started to combine the multi-agent system technology with the earlier paradigms and design neural network agents, expert system agents, and data mining agents (e.g. [53, 54, 55]). Recently, however, with the introduction of the cognitive computing[18] healthcare industry is aiming to create systems that mimic the way humans think, and eventually improve the diagnosis quality (Figure 10).

[18] The term cognitive computing is typically used to describe AI systems that aim to simulate human thought. A number of AI technologies are required for a computer system to build cognitive models that mimic human thought processes, including machine learning, neural networks, Natural Language Processing (NLP) and sentiment analysis [238].

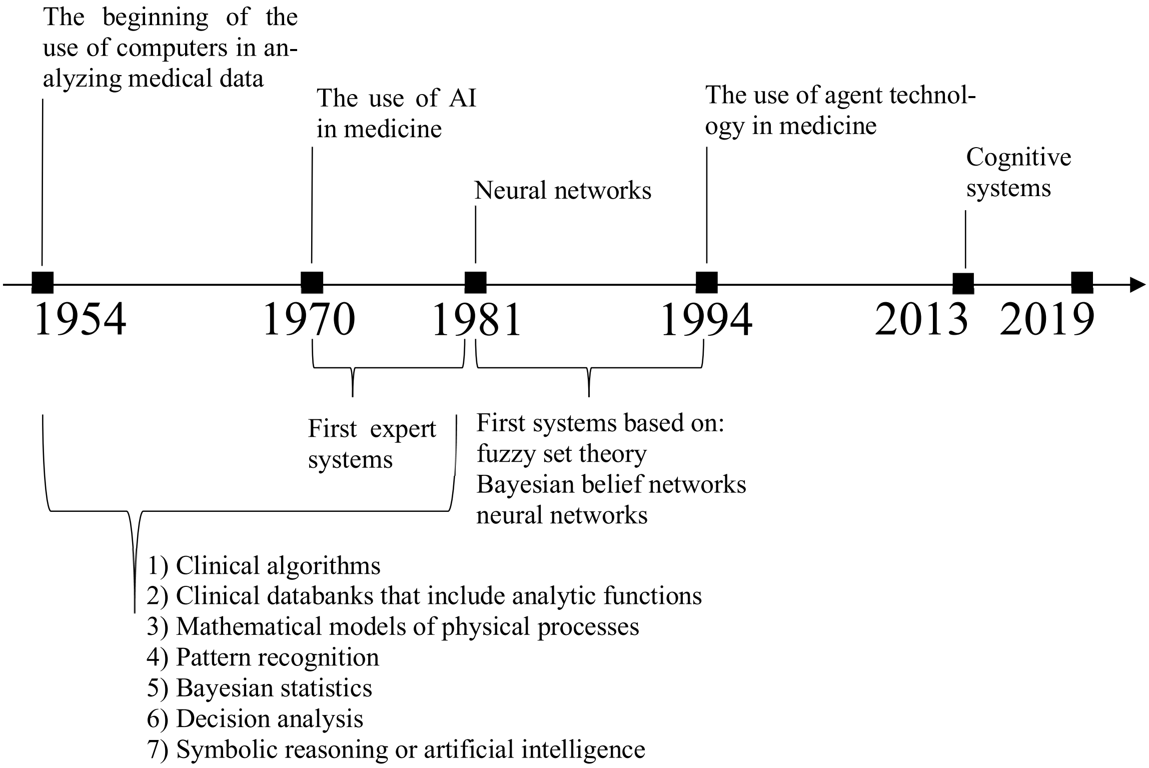

The beginning of the use of computers in analyzing medical data
The use of AI in medicine
Neural networks
The use of agent technology in medicine
Cognitive systems
1954
1970
1981
1994
2013
2019
First expert systems
First systems based on:
fuzzy set theory
Bayesian belief networks
neural networks
1) Clinical algorithms
2) Clinical databanks that include analytic functions
3) Mathematical models of physical processes
4) Pattern recognition
5) Bayesian statistics
6) Decision analysis
7) Symbolic reasoning or artificial intelligence

Figure 10: The history of the use of computers in analyzing medical data

As the main focus of this research is on MDSs, this section is mainly dedicated to providing the readers with an overview of these systems as a specific type of CDSSs. A comprehensive survey of CDSSs can be found in [23], which suggests two main types of CDSS: Knowledge-based and Non-knowledge-based CDSSs. Knowledge-based CDSSs are rooted in early expert systems. “These systems attempted to replicate the logic and reasoning of a human decision maker, reaching firm decisions based on existing knowledge" [25, p. 630]. A knowledge-based CDSS consists of three parts [23, p. 3]:

- The Knowledge Base: The knowledge base consists of complied information that is often, but not always, in the form of if-then rules.
- The Inference Engine: The inference engine contains the formulas for combining the rules or associations in the knowledge base with actual patient data.
- The communication mechanism: The communication mechanism is a way of getting the patient data into the system and getting the output of the system to the user who will make the actual decision.

Some notable examples of such systems can be found in Table 6. This table sorts the MDSs according to their release year. As can be seen, early MDSs were mainly expert systems.

Table 6. Notable examples of MDSs

Ref.	MDS	Discipline	Year
[56]	INTERNIST I: Rule-based expert system	Internal Medicine	1974
[57]	CASNET (Causal ASsociational NETworks)	Diagnosis and treatment of Glaucoma	1974
[58]	PIP (Present Illness Program): Medical expert system	Evaluation of patients with edema	1976
[59]	MYCIN: Rule-based expert system	Infectious disease	1976
[60]	PUFF: an expert system for interpretation of pulmonary function data	Pulmonology	1983
[61]	QMR (Quick Medical Reference): DDSS for internists	Internal medicine	1986
[62]	CADUCEUS: Medical expert system (could diagnose up to 1000 diseases)	Internal medicine	1986
[63]	DXplain: MDS with evidential support for each DDx, along with recommended follow-up	Internal medicine	1987
[64]	Iliad: Expert system for internal medical diagnosis (teaching tool)	Internal medicine	1991
[65]	Papnet: a commercial NN-based computer program for assisted screening of Pap (cervical) smears	Cytology	1991
[66]	An ANN for the diagnosis of myocardial infarction	Cardiology	1991
[67]	Artificial neural networks for single photon emission computed tomography (SPECT)	Radiology	1993
[68]	An ANN for diagnosis of acute pulmonary embolism	Pulmonology	1995
[69]	Evolving neural networks for detecting breast cancer	Oncology	1995
[70]	Neuroserum: An artificial neural net-based diagnostic aid tool for serum electrophoresis	Pathology	1998
[71, 72]	VisualDx	Internal medicine, Dermatology	2001
[41, 42]	Isabel: The most powerful DDx generator on the market (covering over 10,000 conditions)	Internal medicine	2002
[73]	An ANN for automated diagnosis of heart disease in patients with heart murmurs	Cardiology	2006
[74]	Intelligible support vector machines for diagnosis of diabetes mellitus	Endocrinology	2010
[75]	Tuberculosis disease diagnosis using ANN trained with genetic algorithm	Pulmonology	2011
[76]	GA based system for the diagnosis of cervical precancer	Cytology	2011
[77]	An ANN for automatic diagnosis of small bowel tumor	Gastroenterology	2012
[78]	An ANN for diagnosis of Coronary heart disease	Cardiology	2012
[79]	IBM Watson	Oncology	2013
[80]	Support vector machines for preliminary diagnosis of tuberculosis disease	Pulmonology	2017

In contrast to knowledge-based decision support systems, non-knowledge-based CDSSs "use a form of artificial intelligence, called machine learning, which allows the computers to learn from past experiences and/or to recognize patterns in the clinical data" [23, p. 5]. The most popular ML algorithms for CDSSs include Artificial Neural Networks (ANN), Genetic Algorithms (GA), Support Vector Machines (SVM), and decision trees [81]. Some examples of non-knowledge-based CDSSs are also listed in Table 6. The most remarkable non-knowledge-based CDSSs are based on neural networks and genetic algorithms [23]. Reviews on the ANNs that are introduced for medical diagnosis purposes can be found in [82], [83] and [84]. Additionally, a comprehensive survey on the applications of genetic algorithms in medicine is given in [85].

Early studies on MDS showed that the use of computers in this field can benefit the quality of the medical diagnosis. The earlier approaches were, however, just useful for narrow application areas, as their scalability has always been a challenging issue. As stated in [86, p. 88] "As long as the number of rules in the rule base is small enough, an expert system can be highly effective. However, the extension of rule bases, e.g., as part of a learning or adaption process, can cause serious problems since it may happen that the overall semantics and behavior of the rule base gets out of control". Other approaches had their own shortcomings too. According to [87], challenges that can be encountered while applying neural networks include selecting the best topology, avoiding overtraining and undertraining, and determining the training cases. The complexity of such problems also grows with the size of the neural networks. Regarding the genetic algorithm based CDSSs, dealing with large amount of data the complexity is again a big challenge. Moreover, the very low rate of convergence, along with the lack of guarantee of finding the global maxima, make such approaches less appealing. Furthermore, it should be noted that although machine learning based CDSSs may have a better performance than the average clinicians, as due to the black box nature of machine learning [88], these systems do not reveal the reasons behind their conclusions, most clinicians do not use them directly for diagnoses, for reliability and accountability reasons [23].

To tackle the scalability problem, computer scientists have started using the MAS technology. The distributed architecture of MASs offers not only the required scalability, but also flexibility and adaptability to the target system. Having these advantages in mind, researchers started to combine this technology with the earlier paradigms and design agent based CDSSs composed of neural network agents and expert system agents. A survey on multi-agent based DSSs for medical classification problems is presented in [89], and some notable multi-agent based MDSs are listed in Table 7.

Table 7. Agent-based MDSs

Ref.	Purpose	Agents	Agent type	Year
[90]	A multi-agent based decision-making tool, composed of smaller expert systems operating locally as a part of the more global system	In this approach, each functional knowledge-based module operates autonomously and communicates with the other modules (agents) via message passing using the strategy "when needed". This approach explores the idea of the blackboard architecture [91], and the idea of the inheritance principle (taxonomy hierarchy) firstly used in MYCIN [92].	ES	2001
[93]	An Agent-based Data Mining Info-structure (ADMI) that provides a suite of healthcare-oriented decision-support/strategic planning services	This system is an agent-based framework for knowledge discovery in a distributed environment that includes multiple heterogeneous healthcare data repositories. The system's agents are Interface Agent, Data Collection Agent, Data Mining Agent, and Service Generation Agent.	Data Mining	2002
[94]	A multi-agent distributed DSS, to help in the early diagnosis and prognosis of brain tumors	This system employs agents' negotiation and argumentation mechanism developed for distributed resource allocation problems. The system's agents are GUI Agent, DB Agent (local/external), Training Petitioner Agent, Classifier Petitioner Agent, Classifier Agent, and Yellow Pages Agent.	Data Mining (Classification)	2006
[95]	An MDS capable of solving problems that require the cooperation of multiple systems with different capabilities and capacities.	The MDS is a heterogeneous system with agents (human and artificial) specialized in medical diagnosis, a doctor called moderator and assistant agents (human and artificial)	ES	2006
[96]	A multi-agent hospital management framework empowered by agents that perform data mining to support the diagnosis of a patient	Hospital Manager, Facilitator Agent, Room Manager, Monitor Agent, Service Modules, Expert Mobile Agent. The expert mobile agent (EMA) classifies patient's data using a neuro-fuzzy algorithm for the consultation report (Neuro-fuzzy systems are the hybrid of ANNs and fuzzy systems).	Neuro-Fuzzy	2007
[97]	An MDS capable of solving problems that require the cooperation of multiple systems with different capabilities and capacities. (improvement of [95])	Hybrid ES agents represent ES agents endowed with hybrid components, e.g. neural networks components. Each problem-solving component of an agent can be implemented as a well-adapted problem-solving method for a class of problems. For example, an ES component can be well adapted for medical diagnosis and a NN component can be well adapted for problems like noisy image recognition.	Hybrid ES (ES + NN component + other components)	2007
[98]	A MAS that distributes the diagnosis to three agents	Each agent is a specialist (Otorhinolaryngology, digestive, cardiology) performing reasoning using their rule bases and communicating with other agents using a blackboard.	ES	2008
[54]	medical diagnosis system	Several NNs, each represented by an agent and covering a group of diseases are connected through an administrative instance that selects the right NN for diagnosis process based on the given symptoms.	NN	2009

It should be noted that even though the use of agents in MDSs can reduce the complexity problem and simultaneously increase the scalability and adaptability of these systems, finding the right distribution and, furthermore, the inevitable coordination plan that controls the collaboration between the agents, are new problems that concurrently rise. If the diagnosis problem is dealt with only a few numbers of agents, addressing the coordination of agent interactions will be simple but in this scenario each agent is again facing the old problems, i.e., complexity, scalability, and adaptability. On the other hand, if the system keeps the agents very simple, i.e., with limited knowledge and few tasks, the number of agents should consequently be relatively big. In this scenario, again, the interaction between the agents will be an issue. To put it in a nutshell, the MAS technology may be used for diagnosis problems to break down the problem into some simpler ones, but the mechanism which derives the final diagnosis from the agents' outputs should be designed very carefully.

Complicated domains that demand decomposable problem settings in different abstraction levels can best be dealt with by systems that are based on one of the well-known MAS architectures, i.e., the HMAS architecture (see chapter 4). The term holon was first introduced in [99] in order to name recursive and self-similar structures. As stated in [100, p. 7], "A holonic agent of a well-defined software architecture may join several other holonic agents to form a superholon; this group of agents now acts as if it were a single holonic agent with the same software architecture" (for further information please refer to section 4.2). In next chapter the DDx domain and the holonic domain are both studied in detail and it will be proven that the DDx domain meets the characteristics of HMASs.

Few attempts have been made to use the holonic structure for medical systems ([101], [102], and [103]). Table 8 lists these systems and describes how these systems used the holonic approach in order to solve their problem and organize the agents in their systems. Clearly, none of these systems were designed to implement the DDx or covers the H&P process, and in fact, as mentioned in chapter 1 and section 2.2, no system has been designed so far that is capable of guiding the diagnosis process in H&P. The system introduces in [101] is an MDS, however, it applies a completely different approach than that adopted in DDx. The system introduced in [102] aims to implement a remote healthcare system and uses the holonic approach in order to make the collaboration between the remote medical entities possible. Similarly, [103] proposes a holon-based architecture for the hospital information systems and uses the holonic approach in order to manage the information flow to support the administrational needs of hospitals.

Table 8. Holonic multi-agent-based medical systems

Ref.	Purpose	Holons	Year
[101]	Medical Diagnosis System	This system is composed of two types of agents: (1) Disease Specialist Agent (DSA), and (2) Disease Representative Agent (DRA). DRAs are atomic and represent a specific disease, DSAs are holons and represent groups of related diseases (DSAs and/or DRAs). The agents collaborate with each other for the purpose of diagnosis using blackboards in different layers of holarchy, i.e., the organizational structure of a holonic society. Two different approaches are suggested: static and dynamic. In static approach DSAs rather map medical disciplines. Dynamic approach, however, suggests case-based emergence allowing the DSAs and DRAs to change their roles. The quality of the collaborations between each agent and its head(s) is measured to support the self-organization, however, this approach does not reveal any information about the quality of the connections on the higher levels.	2003
[102]	Remote diagnosis, prediction and ubiquitous healthcare (telemedicine)	This approach "extends the holonic enterprise paradigm to the medical domain. A medical holarchy is a system of collaborative medical entities (patients, physicians, medical devices, etc.) that work together to provide a needed medical service for the benefit of the patient" " [102, p. 1]. As stated in [102, p. 11], the elements defining the levels of a medical Holonic Enterprise (HE) are: (1) Inter-Enterprise: Hospitals, Pharmacies, Medical Clinics/Laboratories (2) Intra-Enterprise: Sections/Units/ Departments of each medical enterprise (3) Resource Level: Machines for medical tests, medical monitoring devices, information processing resources (medical files, computers, databases, decision support systems), physicians, medical personnel (technicians, assistants, etc.)	2004
[103]	Hospital information system	This system suggests a holon-based architecture for the informational system of hospitals and is comprised of the following classes of holons: (1) Service holons, (2) Resource holons, (3) Medical specialty holons, (4) Supervisor holons. Supervisor holons (SupH) coordinate the holons' activities in a holarchy. Service holons (SerH) are responsible for the management, planning, execution and delivery of all services in the system. Resource holons (ResH) represent the primary resources of the system, such as patients' data, medical devices, the clinic's data, infrastructure, etc. and their management. Medical speciality holons (MedSpecH) are special holons dedicated to particular fields in medicine (urology, cardiology, ophthalmology, etc.).	2018

3.2.2 The State of the Art

As mentioned in research motivation section the state of the art of the MDSs includes IBM Watson and Isabel. This section studies these systems in more detail in order to demonstrate the significance of the research motivation.

3.2.2.1 IBM Watson

Watson is a cognitive computer system developed by IBM. "Built with the purpose of being a question answering machine, Watson also has the capability of understanding natural language, creating a more natural relationship between humans and computers" [104]. Several application areas have been considered for Watson, including healthcare.

According to [105], "healthcare was one of the first industries to which Watson technology was applied". In 2013, the first commercial implementation of Watson took place as the Memorial Sloan Kettering Cancer Center implemented Watson began using it to ensure that its lung cancer patients were receiving the right treatment while reducing costs [106]. Ever since, other medical centers such as Cleveland Clinic, Maine Center for Cancer Medicine, and Westmed Medical Group have also implemented Watson tools [107, 108].

However, not every implementation has gone successfully. After spending more than $62 million over the course of four years on implementing a decision support system powered by Watson technology, in 2017, hospital administrators of MD Anderson Cancer Center in Houston canceled the project, saying it had failed to meet its goals [109]. Healthcare is still, however, a primary focal point for IBM Watson technology.

IBM Watson is claimed to be the most powerful artificial intelligence-based system capable of performing medical diagnosis. This system is a cognitive computer system that has the capability of understanding the natural language. Watson uses IBM's DeepQA (a software architecture for deep content analysis and evidence-based reasoning) and the Apache UIMA (Unstructured Information Management Architecture) for analyzing unstructured data. DeepQA "represents a powerful capability that uses advanced natural language processing, semantic analysis, information retrieval, automated reasoning and machine learning. DeepQA deeply analyzes natural language input to better find, synthesize, deliver and organize relevant answers and their justifications from the wealth of knowledge available in a combination of existing natural language texts and databases" [110]. From the technical point of view, upon receiving a request, Watson searches a large knowledge base, generates potential hypotheses, and then initiates another search to collect evidence that supports them. Although Watson is very powerful in performing this action, the success is very much depending on the quality of the input, i.e., the output of an already performed H&P, and IBM Watson is best used to create patient-specific treatment plans, e.g., cancer treatments. To the best of our knowledge, Watson has never been involved in the initial core medical diagnosis process, but only in improving the diagnosis and assisting with identifying treatment options for patients who have already been diagnosed [79].

3.2.2.2 Isabel

The second MDS to be considered here is Isabel, which is a CDSS that facilitates diagnostic reminders and differential diagnosis [41], [42]. Isabel is actually an internet-delivered CDSS consisting of a knowledge base and an inference engine, implemented using a commercially available software, Autonomy [111], which utilizes Bayesian inference and Shannon's principles of information theory to generate pattern matching algorithms in order to enable sophisticated concept extraction from documents [42]. A systematic review and meta-analysis of differential diagnosis generators was conducted in [43] in order to investigate their efficacy and utility. According to this study, Isabel was associated with the highest rates of diagnosis retrieval compared to all other types of differential diagnosis tools [43].

According to [112] and [113], all the Isabel DDx systems are equipped with its cutting-edge proprietary Disease Pattern Recognition Engine Platform (DPREP), which is an engine powered by innovative, statistical natural language processing software with the ability to understand the meaning and context of unstructured free text. This engine has been applied to a medical database of every known possible presentation of over 10,000 conditions and has effectively been trained for over almost two decades. An additional set of algorithms that is also tuned over many years is then applied to the initial results from the Isabel DPREP to ensure that only those results relevant to the patient's age, gender and geographical region are displayed.

The Isabel team believes that this unique structure, which is very different from the traditional rule-based medical diagnosis systems, gives Isabel many inherent advantages:

- Easy to use yet able to rapidly handle any permutation of clinical features
- Scalability makes it easy to maintain, update and expand
- Published API makes it easy to integrate into other systems
- Platform structure enables the same core system to serve many different audiences from clinicians to patients and in multiple languages. [113]

As stated in [113], “Benefitting from close to two decades of continuous development, validation and clinician feedback, Isabel today is widely acknowledged as the most accurate and versatile system available. When clinicians say they have ‘Isabeled’ a patient you know they have thought carefully about their diagnosis and are providing high quality, cost effective care”.

According to [112], Isabel offers different products including the Isabel Pro, Isabel Symptom Checker (Isabel SC), Isabel Active Intelligence (Isabel AI), and Isabel Clinical Educator. Isabel Pro for clinicians is the most powerful DDx generator on the market. Using this product clinical features including labs, vitals and co-morbidities along with age, gender and travel history are simply entered manually in free text or received automatically from the EMR, next the DDx list together with a list of drugs that may potentially cause the symptoms are generated. The Isabel Symptom Checker (Isabel SC) on the other hand allows the patients to describe their symptoms in their own words. The system then generates results in seconds. Its integrated ‘Where now?’ feature helps patients figure out their next steps and it encourages patients to connect and engage with relevant healthcare providers. Isabel can also be integrated into the hospitals electronic systems and provide automatic diagnostic decision support from within the EMR. Isabel Active Intelligence (Isabel AI) uses advanced Natural Language Processing (NLP) software to extract key clinical features including lab test results, vital signs and social factors from the EMR documentation. The key features which have been automatically extracted can then be reviewed and sent to the Isabel diagnosis engine to produce a differential diagnosis for the clinician within the EMR workflow to work up. Together with the applications that are designed for diagnosis purposes, Isabel offers a learning tool, i.e., Isabel Clinical Educator, which is a powerful case-based learning platform for developing, measuring and honing critical diagnosis reasoning skills in clinical learners.

As mentioned already, no matter how powerful and groundbreaking an MDS is in connecting the patient’s data to its medical knowledge, this process does not necessarily guarantee an error-free diagnosis and it is always possible that the final strong deduction is actually a misdiagnosis based on some incomplete input. Focused H&P is the key to a flawless diagnosis, and this means the physicians should still provide the MDSs with the right input. The goal of this study is to design an MDS capable of guiding the H&P. Therefore, a reliable and at the same time easy scalable MDS is needed, that is able to explain its reasoning and moreover is capable of learning in order to adapt itself to the new findings. The system should be able to cover a large number of diseases, i.e., all the internal medicine diseases, and it's different from those simple systems that are solely used to determine the presence or absence of a particular disease or distinguish between diseases in some small subcategory of diseases. Working in this scale the use of MAS paradigm is highly promising, however, detailed study on the DDx domain is needed in order to find the right organizational structure for the system. The next section discusses the feasibility of using the holonic multi-agent approach to build this system.

It should be noted that the output of this research, i.e., the DSS that is capable of guiding the H&P, can also be used together with the state of the art MDSs by the healthcare providers in order to provide them with the all-encompassing input. In this case the physician reviews the H&P report and if the H&P reveals a complex medical case for which reassessment or patient-specific treatment is needed the H&P report can be given to the cutting-edge systems like IBM Watson and Isabel for further assessments or treatment plans.

Chapter 4
SYSTEM ANALYSIS: THE SOFTWARE ENGINEERING APPROACH

This chapter covers the system analysis process. To this end, the key component of H&P, i.e., the DDx, is studied in greater detail. Eventually, it is proven that the DDx domain meets the characteristics of HMASs. Accordingly, the HMASs are also introduced in this chapter.

4.1 Problem Analysis and Solution Proposal

In order to be able to recommend practical solutions for solving the DDx implementation problem, this section analyzes the problem and provides a deep understanding of this domain. As mentioned, in medicine, a DDx is “the distinguishing of a particular disease or condition from others presenting with similar signs and symptoms” [5]. This action is the key component of patient encounter. In order to perform the DDx, the physician carefully listens to the symptoms explained by the patient, considers some potential diagnoses and then tries to gather enough evidence and supporting information to shrink the probability of the other candidates by questioning some signs and symptoms that might have been simply ignored by the patient, or requesting the patient to undertake some medical examinations. This means that even if a disease is highly probable, there are always a number of diseases whose signs and symptoms overlap with those of that disease. Considering Tuberculosis (TB) for instance, asthma, bronchitis, and in general a group of pulmonary diseases should all be inspected for DDx (see Table 9), which means that in order to ensure that tuberculosis is the case the physician should be able to confirm the considerably lower probability of the other candidates. It should be noted that the values on the table solely show the possibility of absence or presence in a single disease by assigning value 0.0 to absent signs and symptoms and value 1.0 to existing ones. However, in an MDS these values can be updated based on the cases of the diseases in order to represent them with more realistic values. This implies that MDSs can also rely on the statistical information in order to improve the quality of their outputs. More information on this topic are provided in chapter 6.

Table 9. The DDx of Pulmonary Tuberculosis

		1 Anxiety	2 Chest discomfort (pain, tightness, suffocating)	3 Chills	4 Cough	5 Cyanosis	6 Diarrhea	7 Fainting	8 Fatigue and weakness	9 Fever	10 Frequent respiratory infections	11 Heart palpitations and arrhythmias	12 Hoarseness	13 Itching	14 Loss of appetite	15 Nausea and vomit	16 Night sweats	17 Phlegm (bloody/colored)	18 Shortness of breath (dyspnea)	19 Sweats	20 Swollen ankles, feet or legs (edema)	21 Swollen lymph nodes (painless)	22 Weight loss	23 wheezing
1	Asthma	0.0	1.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	1.0
2	Bronchitis	0.0	1.0	1.0	1.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0
3	COPD	0.0	1.0	0.0	1.0	1.0	0.0	0.0	1.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0	0.0	1.0	0.0	1.0	1.0
4	Lung Cancer	0.0	1.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	1.0	0.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0	0.0	1.0	1.0
5	Lymphoma	0.0	0.0	1.0	0.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0	0.0	1.0	1.0	0.0	1.0	0.0	0.0	0.0	0.0	1.0	1.0	0.0
6	Pneumonia	0.0	1.0	1.0	1.0	0.0	1.0	0.0	1.0	1.0	0.0	1.0	0.0	0.0	0.0	1.0	0.0	1.0	1.0	1.0	0.0	0.0	0.0	0.0
7	Pulmonary Edema	1.0	1.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0	0.0	0.0	1.0
8	Pulmonary Embolism	1.0	1.0	0.0	1.0	1.0	0.0	1.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0	1.0	1.0	0.0	0.0	0.0
9	Tuberculosis	0.0	1.0	1.0	1.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0	0.0	0.0	1.0	0.0	1.0	1.0	0.0	0.0	0.0	0.0	1.0	0.0
10	Sarcoidosis	0.0	1.0	0.0	1.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	1.0	1.0	1.0

Basically, when a physician receives the CC (s)he will have an initial DDx list in mind. At this stage the physician will start controlling the absence or presence of the signs and symptoms of the diseases in this initial list in order to revise the list. Of course, special knowledge and experience is needed to perform this action successfully. In fact, irrelevant questions and tests should be ignored and every single piece of information should be used in order to narrow down the possibilities, i.e., the whole process should be kept focused.

It should be noted that the differential diagnostic procedure may be simplified in presence of a pathognomonic sign or symptom or absence of a sine qua non sign or symptom. A pathognomonic sign or symptom is "A sign or symptom that is so characteristic of a disease that it can be used to make a diagnosis. For example, Koplik spots in the mouth opposite the first and second upper molars are pathognomonic of measles" [1]. In contrast with pathognomonic, the term sine qua non is often used for signs, symptoms, or findings whose absence would mean the absence of the target disease. For example, the presence of Reed-Sternberg cells and Hodgkin cells, the mononucleated variants of the former, is considered a sine qua non for the diagnosis of Hodgkin lymphoma [114].

The DDx of some common signs and symptoms can be found in [115] and [116]. The former book lists nearly 200 symptoms, physical signs, laboratory test results, and radiologic findings and their differential diagnoses, and the latter offers students, residents, and practitioners a systematic approach to DDx of signs and symptoms seen by primary care physicians. A guide to DDx of the top ten symptoms in primary care is also presented in [117]. In many DDx cases there might be different groups of diseases to be considered, and these groups may be of different granularities or abstraction levels. In the example given in Table 10 [118, p. 174] dyspnea or shortness of breath is the CC of the patient. As it is demonstrated diseases including dyspnea can be decomposed recursively into groups of different granularities and abstraction levels. As a result, the differential diagnostic problem can be recursively broken down into sub-problems by weighting the likelihood of the presence of possible diseases. These subproblems may induce different abstraction levels and can be of different granularities. To implement a system that is capable of performing DDx, a system composed of components that are expert in medical specialties and subspecialties is highly plausible. MAS technology (see next subsection) has been shown to be highly appropriate for the engineering of open, distributed or heterogeneous systems [119].

Table 10. The DDx of Chronic Dyspnea (Shortness of Breath)

System	Type	Possible Diagnosis
Pulmonary	Alveolar	Bronchoalveolar carcinoma, chronic pneumonia
	Interstitial	Drugs (e.g., methotrexate, amiodarone) or radiation therapy, lymphangitic spread of malignancy, passive congestion
	Obstructive	Asthma/bronchitis/bronchiectasis, bronchiolitis obliterans, chronic obstructive pulmonary disease, intrabronchial neoplasm, tracheomalacia
	Restrictive (extrinsic)	Kyphoscoliosis, obesity, pleural disease/effusion, pneumothorax
	Vascular	Chronic pulmonary emboli, idiopathic pulmonary hypertension
Cardiac	Arrhythmia	Atrial fibrillation, inappropriate sinus tachycardia, sick sinus syndrome/bradycardia
	Myocardial	Cardiomyopathies, coronary ischemia
	Restrictive	Constrictive pericarditis, pericardial effusion/tamponade
	Valvular	Aortic insufficiency/stenosis, congenital heart disease, mitral valve insufficiency/stenosis
Gastrointestinal	Dysmotility	Gastroesophageal reflux disease/aspiration, neoplasia
Neuromuscular	Metabolic	Acidosis
	Neurogenic	Amyotrophic lateral sclerosis, muscular dystrophies, phrenic nerve palsy, poliomyelitis
Other	Anemias	Iron deficiency, hemolysis
	Deconditioning/obesity	Sedentary lifestyle
	Pain/splinting	Pleural-based malignancy
	Psychological/functional	Anxiety/hyperventilation, depression

In [119, 120, 121] sets of criteria have been identified to establish the suitability of a MAS. According to [120, p. 183] factors that point to the appropriateness of a multi-agent based approach are:

1. The environment is open, or at least highly dynamic, uncertain, or complex;
2. Agents are a natural metaphor;
3. Distribution of data, control or expertise
4. Legacy systems.

A system capable of performing DDx should cover all the diseases included in its specialty and if a new disease is discovered[19], it should definitely recognize it. New diseases or even some instances of existing diseases lead to the introduction of new collaboration/connections demands. This indicate that the environment is open. In addition, the system input includes diverse instances of different diseases, so the environment is also dynamic. This input may be incomplete, i.e., may not cover all the signs and symptoms (see chapter 1), and/or indicate multiple diseases. This also implies that the environment is uncertain. Moreover, medical diagnosis is clearly a complex task. As a result, this system fulfills the first criterion.

As mentioned above, in order to implement a system that is capable of performing DDx, a system composed of components that are expert in medical specialties and subspecialties is highly plausible. This indicates that agents are natural metaphors in this system and that the system meets the second criterion as well. Moreover, there are many medical specialties and subspecialties, so it is clear that this system aims to implement a distribution of expertise and satisfies the third criterion too.

The MAS technology may be used for diagnosis problems to break down the problem into a number of simpler ones, but the mechanism which derives the final diagnosis from the agents' outputs should be designed very carefully. As mentioned, the differential diagnostic problem can be recursively broken down into sub-problems by weighting the likelihood of the presence of possible diseases. These subproblems may induce different abstraction levels and can be of different granularities. According to the nature of DDx, the problem solvers are collaborative and those dealing with similar diseases need to have more communications, which are to be conducted in a timely manner. These characteristics meet the characteristics of HMASs (see section 4.3.1). To put in a nutshell, a problem that can be broken down into subproblems can be implemented using MAS technology, however, since the DDx decomposes recursively into subproblems in number of levels, the HMAS approach, which is designed for agents that use the whole-part conceptual relation, is an appropriate approach to solve this problem (for more information on this topic please refer to section 4.3.2).

[19] The World Health Organization warned in its 2007 report that infectious diseases are emerging at a rate that has not been seen before. Since the 1970s, about 40 infectious diseases have been discovered, including SARS, MERS, Ebola, chikungunya, avian flu, swine flu, Zika and, most recently, COVID-19.

4.2 Holonic Multi-Agent Systems (HMASs)

As the HMASs combine the concepts of MASs and holons, this section introduces the basic concepts of HMASs starting with the definitions given for MASs and holons.

4.2.1 An Introduction to Multi-Agent Systems (MASs)

4.2.1.1 What is an Agent?

The concept of agent-oriented programming was first introduced by Yoav Shoham in 1990 [52]. "An agent is a computer system that is situated in some environment, and that is capable of autonomous action in this environment in order to meet its delegated objectives" [122, p. 4, 120, p. 21] (see Figure 11 [123, p. 16]).

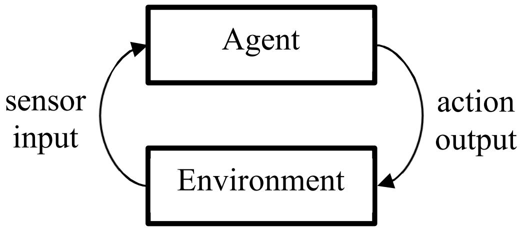


Figure 11. An agent in its environment

In [120], "Wooldridge distinguishes between an agent and an intelligent agent, which is further required to be reactive, proactive and social" [124, p. 1]. The definition used in this research is the one given in [124]. An Intelligent Agent is a piece of software that is:

1. Situated – exists in an environment
2. Autonomous – independent, not controlled externally
3. Reactive – responds (in a timely manner) to changes in its environment
4. Proactive – persistently pursues goals
5. Flexible – has multiple ways of achieving goals
6. Robust – recovers from failure
7. Social – interacts with other agents. [124, p. 3]

It is also to be mentioned that throughout this research by the term ‘agent’ an intelligent software agent is meant. A comprehensive introduction to agents and multi-agent systems is provided in [125].

4.2.1.2 The Multi-Agent Systems (MASs)

A Multi-Agent System (MAS) consists of a collection of individual agents, and its capability is an emergent functionality that surpasses some of the capabilities of each of the individual agents [100]. In [126] a literature review on the architectures of complex multi-agent systems is presented. According to this survey, MAS architectures can be categorized in numerous ways according to different criteria:

In terms of the mechanism:

1. Pipes-and-filters architectures,
2. Event-based architectures, and
3. Layered architectures.

In terms of their characteristics:

1. Hierarchical architectures,
2. Distributed architectures,
3. Open architectures,
4. Reconfigurable architectures,
5. Mobile architectures, and
6. Fault-tolerant architectures.

In a general view:

1. Information-flow oriented architectures,
2. Role-oriented architectures (generic and domain-specific), and
3. Control-oriented architectures.

The field of MAS is part of Distributed Artificial Intelligence (DAI) in the sense that a MAS lends itself naturally to distributed problem solving, where each agent has the characteristics of a distinct problem solver for a specific task. Many distributed problems exhibit a recursive structure, which implies that agents that appear as single entities to the outside world but are in fact composed of many sub-agents may be the inherent problem solvers for these problems. Solving such problems many agents may decide that it is advantageous to join into the coherent structure of a super-agent and thus act as a single entity [100]. As already mentioned, in year 1999 agents consisting of sub-agents with same inherent structure were named holonic agents [100]. Holonic architecture is a type of hierarchical architecture, which itself is a typical control-oriented architecture for MASs [126]. In this project, the holonic architecture in used to control the information flow between the agents that are specialists for different disease or groups of diseases.

4.2.2 An Introduction to Holonic Multi-Agent Systems (HMASs)

4.2.2.1 What is a Holonic Agent?

The term holon was originally introduced in 1967 by Arthur Koestler [127] in order to name recursive and self-similar structures in biological and sociological entities. According to [127], a holon is a natural or artificial structure that consists of several holons as sub-structures. “The organizational structure of a holonic society, or holarchy, offers advantages that the monolithic design of most technical artifacts lack: They are robust in the face of external and internal disturbances and damages, they are efficient in their use of resources, and they can adapt to environment changes [100, p. 6]”.

The terms holon and holonic agents are used synonymously in holonic MASs. “A holonic agent of a well-defined software architecture may join several other holonic agents to form a super-holon; this group of agents now act as if it were a single holonic agent with the same software architecture” [100, p. 7]. Accordingly, by super-holon, a composition of subordinate agents, also called sub-holons or sub-agents, are denoted. In contrast to sub-structures in Koestler’s framework, in HMASs all entities are restricted to agents, and furthermore, sub-holons should always have the same structure as the super-holons [100]. As stated in [128, p. 64], “a holon can be seen, depending on the level of observation, either as an autonomous atomic entity, or as an organization of holons. This duality is sometimes called the Janus[20] Effect, in reference to the two faces of a holon. In other words, a holon is a whole-part construct that is composed of other holons, but it is, at the same time, a component of a higher level holon". According to [129, p. 31], "holons are self-contained, self-regulating, and semi-autonomous entities. They appear to be autonomous wholes for the lower level while, at the same time, being a (conditionally) dependent part for the upper (control) level".

As already mentioned, one of the most attractive characteristics of holonic systems is the self-organization capability, which can be defined as "the mechanism or the process enabling a system to change its organization without explicit external command during its execution time" [12]. The benefits of self-organization are twofold, on the one hand this adaptation/optimization "ensures that a holarchy exhibits robust and stable behavior that allows it to automatically and efficiently deal with many kinds of disturbances and unforeseen events. On the other hand, it allows it to automatically adapt itself to changing environments and requirements from the outside" [129].

Machine learning techniques should be applied to this aspect of the HMASs in order to guarantee system improvement. A generic framework for holonic systems is presented in [128], containing a generic engine that will guide the holons in their self-organization. In this research, this framework is considered, however, its self-organization engine is adjusted in order to match the application's objectives and merging criteria.

4.2.2.2 A Generic Framework for the Modelling of Holonic Systems

This section introduces the generic framework for HMAS modelling as presented in [128]. This framework is not limited to any specific architecture or domain and attempts to cover all the aspects of a holonic multi-agent system. On the one hand, the application of this framework to the HMDS ensures the coverage of all the aspects of the HMAS in this project. On the other hand, the self-organization engine, which this framework proposes, can be used as a generic guideline to conduct the learning process in our system. It should be pointed out that this framework is not used here as a modelling framework, but rather as a checklist of all the aspects of

[20] In ancient Roman religion and myth, Janus is the god of beginnings, gates, transitions, time, duality, doorways, passages, and endings. He is usually depicted as having two faces, since he looks to the future and to the past.

HMASs. In addition, the Role-Interaction-Organization (RIO) Model [130] used by the framework will not be used in this project, and instead the Agent Unified Modeling Language (AUML) is considered as the modelling language (for information on the AUML, see appendix A).

4.2.2.2.1 The important aspects of a Holonic MAS

According to the framework the three important aspects of a Holonic MAS are:

1. Holon Structure and Management: A super-holon is an entity in its own right, but it is composed by its members. This part of the framework considers how the members organize and manage the super-holon.
2. Goal-Dependent Interactions: Super-holons are created with an objective and to fulfill certain tasks. To achieve these goals/tasks, the members must interact and coordinate their actions. The framework also offers a means to model these aspects of the super-holons' functioning.
3. Dynamics: Dynamics are inherent characteristics of MAS. The framework considers in particular two of the most attractive characteristics of Holonic MASs: Merging (creating and joining a super-holon) and Self-Organization. [128, p. 66]

In next three sections (see 4.2.2.2.1.1, 4.2.2.2.1.2, and 4.2.2.2.1.3) each of these aspects are briefly described.

4.2.2.2.1.1 Holon Structure and Management

Three different structures are proposed by [100] for holonic multi-agent systems: Federation of autonomous agents, Moderated Group, and Fusion (Figure 12 [128, p. 69]).

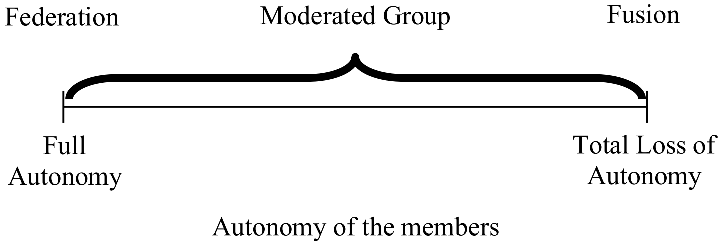


Figure 12.The structures for holons

The framework presented in [128] has decided on a moderated group. According to [128] in a moderated group, two statuses can be differentiated for the members: (1) moderator or representative, which acts as the interface with non-member holons; and (2) represented, which are masked to the outside world by their representative. The framework presented in [128] also suggests four roles to describe a moderated group as an organization: Head, Part, Multi-Part and StandAlone. “The Head role is the representative or moderator of the group” [128, p. 70]. The “Part role is played by those holons belonging to only one super-holon and Multi-Part role by those holons shared by more than one super-holon” [128, p. 71]. “The StandAlone role represents, on the other hand, how non-members are seen by existing holons” [128, p. 70].

The most common functionalities needed to manage a holon are the inclusion / exclusion of the members, the modification of goals and tasks, the destruction of the holon, and the modification of the rule. Of course, some other domain-independent functionalities may also be added to this list. These functionalities all require a decision-making process. For instance, when an external holon generates a membership request, the members of the super-holon can use a voting mechanism to take a decision. As a result, first the required functionalities are to be identified; and then, the level of authority that the head will have over these functionalities can be defined. The framework suggests conducting this by assigning a voting mechanism to each of the functionalities. “In order to parametrize a voting mechanism, three elements must be defined: Requester, Participants and Adoption Mechanism” [128, p. 76].

As stated in [128, p. 76], “the vote requester defines which members are allowed to request for a vote. Participant makes reference to who is authorized to take part in the vote, and finally the adoption mechanism defines how a proposal is accepted or rejected. For the requester and participants three possible options are available: all members, heads only, and subgroup of holons.” For the adoption mechanism also a number of options can be imagined, e.g. consensus, two-thirds, etc. [128].

Considering only the number of voters and percentage of heads and parts involved in the decision-making process, the framework distinguishes four particular configurations.

1. Monarchy: the command is centralized in the single hands of a head. However, it does not refer to the non-election of the head. The framework considers that the nomination

process is a different issue from the decision-making process and requires detailing it as well. In this configuration, only one head controls the entire decision-making process.

2. Oligarchy: A little group of heads share the command without referring to the Part members.
3. Polyarchy: A little group of heads share the command, but they have to refer to the Part for certain decisions.
4. Apanarchy: The command is completely shared between all members of the super-holon. Everyone takes part in the decision-Making process. [128, p. 77]

4.2.2.2.1.2 Goal-Dependent Interactions

The previous section introduced an aspect of the framework that allows the description of the different statuses of the members and how they manage their super-holon. However, without including the interactions of the members concerning goal-driven actions this description would be incomplete.

A super-holon will often need to accomplish a number of tasks in order to achieve its objective. Accordingly, in order to distribute sub-tasks, exchange information, etc., the members need to organize internally. "These tasks are usually application dependent and vary from holon to holon. These domain dependent organizations are called Internal Organizations" [128, p. 79]. The framework has chosen an organizational approach to describe the internal organizations. "The holon's model contains a set of organizations. One of these organizations is the Holonic Organization defining the status of the members. The others are the organizations that define the required interactions to achieve the goal of the super-holon (Internal Organizations)" [128, p. 79]. In this way, the holonic non atomic agent (instantiating the model) contains:

- A unique Holonic Group, instance of the Holonic Organization, which defines how the members are organized. All members of the (super-)holon must belong to this group.
- A set of groups, instances of the Internal Organizations, created to coordinate the interactions of the members. These groups are created based on the objectives/tasks of the members. A group may contain only a subset of the members of the super-holon [128, p. 79].

Following this approach, “the description of a holon involves a number of organizations. The only mandatory organization is the Holonic Organization that describes the member’s status. Other organizations can be added to describe additional behaviors required for the functionality of the super-holon” [128, p. 81].

4.2.2.2.1.3 Dynamics

This part of the framework deals with “the creation of new super-holons in the system and the integration of new members into existing holons. This process is called merging” [128, p. 81].

From the practical point of view, it is necessary that external holons can join existing super-holons. This, in fact, can considerably reduce the number of the levels in the resulting holarchy. “In order to support the integration of the new members, external holons should be provided with a standard interface so they can request their admission”. “When a super-holon is created, only Heads belong to the interface of the super-holon. Thus, other members (Part and Multi-Part) should not be visible by external holons”, and “StandAlone holons may interact only with the heads of the super-holon” [128, p. 83]. This organization offers a merging interaction that provides a mean for a holon to request admission as a new member.

“The merging process may also be used between holons to create a new entity (super-holon) in the system. In this case, all rules that will govern the life of the super-holon have to be defined” [128, p. 84]. From an engineering point of view, different approaches can be used:

1. Predefined: The holons were conceived so that the rules for the super-holon are predefined and known by members in advance. This approach may be useful when developing closed applications. The adaptability of these types of system will remain constrained to the anticipated cases only and will probably be proved to be impossible to use in large open environments.
2. Negotiation: The merging process foresees a mechanism to negotiate the configuration of the super-holon. This approach allows a wider range of applications and improved adaptive capabilities. But the negotiation process may induce important overheads. A mixture of this and the previous approach could help to reduce the overhead.
3. Evolutive: The super-holon is created with a minimum of engagements of the members. The members can then increase their commitment toward the super-holon when they

> consider it useful. The minimal rules set contains only one rule: Add new rules. Using this rule with a voting mechanism, any new rule or modification of it can be obtained. [128, p. 84]

As stated in [128, p. 84], “a Predefined mechanism can be useful for closed, rather small, systems. However, it seems improbable that such a mechanism can be used in an open untrusted environment. The Negotiation is what we might call a generic approach. However, other problems are to be considered; for instance, the communication language used in the negotiations. In addition, trying to define all rules of a super-holon may prove to be a consequent task, introducing an enormous overhead to the creation of the super-holon”.

4.2.2.2.2 A Satisfaction/Affinity Based Self-Organization Engine for HMASs

The second component of the framework provides a generic engine that will guide the holon in their merging process. This engine is based on the roles presented in section 4.2.2.2.1.1. Using these roles, the framework defines a set of possible transitions between the roles. These transitions represent the possible evolutions of an entity inside its super-holon. By adding conditions to these possible transitions, a guide to the evolution of the holon inside its super-holon can be provided.

The framework proposes a specialization of the generic engine based on the affinity and satisfaction between holons, which can be defined as:

Affinity: “The affinity measures, according to the application’s objectives, the compatibility of two holons to work together toward a shared objective” [128, p. 86].

Satisfaction: “The satisfaction measures the progress of the holon toward the accomplishment of its current goal” [128, p. 87].

In order to define the different conditions for the possible transitions in the self-organization engine, the framework defines different kinds of satisfaction according to the actions of other agents:

Self-Satisfaction (SS_i): Satisfaction for the holon i produced by its own work.

Collaborative Satisfaction (CS_i^H): Satisfaction produced for the holon i by its collaboration with other members of the holon H. This satisfaction can be either positive, when the other members' work helps i in its task, or negative, when the other members' work imposes barriers to the achievement of the holon's task.

Accumulative Satisfaction (AS_i): Satisfaction produced for the holon i by its collaboration with members of multiple super-holons. This satisfaction is only used when the Multi-Part role is allowed, i.e., holon i may belong to more than one super-holon. When a holon belongs to a super-holon and it is unsatisfied, two options are available: the holon may quit its current super-holon and join a new one, or it may join a second super-holon without leaving the first. This satisfaction guides the decision in this situation.

$$AS_i = \sum_p CS_i^p \quad \forall p \in superholon(i) \tag{1}$$

where the super-holon function returns the super-holons of i.

Instant Satisfaction (IS_i): Is the overall current satisfaction of holon i

$$\forall i \in HMAS \quad IS_i = \begin{cases} CS_i + SS_i & if\ R_i = Part\ \vee\ R_i = Head \\ AS_i + SS_i & if\ R_i = MultiPart \\ SS_i & if\ R_i = Stand - Alone \end{cases} \tag{2}$$

where R_i is the role played by the holon i. [128, p. 87]

Using these definitions the framework adapts its generic engine as shown in Figure 13 [128, p. 88].

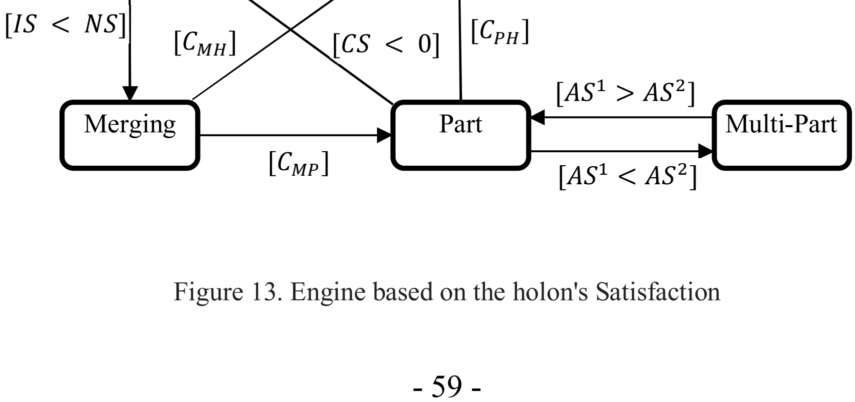


Figure 13. Engine based on the holon's Satisfaction

As described in [128], each condition is enclosed here by square brackets. Each state in this automaton represents the role the holon will play in the super-holon.

Merging state has been added to represent when the holon has started the merging interaction. The merging interaction can be an atomic and simple interaction in some systems, whereas in some cases this interaction can be a complex and elaborated acceptance procedure. In that case the holon may remain in the merging state for a certain time. The conditions that will make the holon either head or part, C_{MP} and C_{MH}, should be defined by the selected merging process.

The notation AS^1 and AS^2 are used to represent the AS when belonging to one or two super-holons. In addition, the engine defines the function Necessary Satisfaction (NS), which estimates the satisfaction required for the holon to finish its task within the constraints established for the tasks. This function should be adapted to the problem under consideration.

This engine can be useful specially when the organization of the holons into a holarchy represent the solution of the problem, and as a result no further interactions are left to be specified. However, it "is intended to guide the holons in their selection and merging, and thus it is not limited to applications where the holarchy provides the solution of the problem" [128, p. 88]. If an application that has certain self-organizing properties is desired, the automaton should be refined to match the application's objective and merging criteria.

4.3 The DDx Domain and the Holonic Domain

As mentioned in section 4.1, the domain attributes that indicate the appropriateness of a multi-agent based solution are presented in [120]. These criteria are clearly met by the DDx problem. Firstly, the environment is open, dynamic, uncertain and complex. Moreover, considering the different medical specialties and subspecialties, an MDS can be naturally modelled as societies of cooperating agents, and as a result here the agents are natural metaphors. This also implies that data, control, and expertise exhibit a distributed nature. This section shows that the DDx meets the characteristics of HMASs in particular.

4.3.1 The Characteristics of Holonic Domains

“The boundaries between domains that are suitable for holonic agents and those that are not, are blurred” [100, p. 21]. Consequently, as suggested in [100] a collection of criteria can be considered as a guide for the classification:

1. Operator abstraction: Holonic systems are well suited for domains with actions of different granularities.
2. Hierarchical structure (Abstraction levels): The application domain that induces abstraction levels can be modelled naturally in a holonic system.
3. Decomposability: Holonic agents support decentralized or decomposable problem settings and can additionally easily realize decompositions of different granularities.
4. Communication: Holons provide facilities for efficient intra-holonic communication, supporting higher frequent communication inside the holon than among different holons (inter-holonic).
5. Social Elements: If there are cooperative elements in the domain, holonic agents can be used to model the cooperative sub-domains.
6. Situatedness and real time requirement: The holonic architecture allows us to set the requirement of bounded rationality for all members of sub-holons in order to find the best possible action within a given resource allocation (a satisfactory solution rather than an optimal one).

4.3.2 The DDx Domain is a Holonic Domain

The DDx domain meets the characteristics of holonic domains:

1. Operator abstraction: A DDx is a process in which different diseases or group of diseases might be considered. As a result, actions can be of different granularity.
2. Hierarchical structure (Abstraction levels): DDx is to be conducted in different abstraction levels. Macro-level actions are carried out by holon’s head that coordinates the actions of the sub-holons. In addition, the abstraction levels help the system to be able to react correctly in case of limited information.

3. Decomposability: The differential diagnostic problem can be recursively broken down into sub-problems by weighting the likelihood of the presence of possible diseases.
4. Communication: In order to perform DDx, agents representing similar diseases usually need to interact more and therefore can be grouped in same super-holons. Holons provide facilities for efficient intra-holonic communication, supporting higher frequent communication inside the holon than among different holons (inter-holonic).
5. Social Elements: In order to perform DDx agents should cooperate. Cooperative elements in the domain can be implemented using holonic agents in order to model the cooperative sub-domain.
6. Situatedness and real time requirement: In DDx, real-time behavior is a vital issue and beside improving the preciseness of the diagnosis, the system is expected to speed up the patient encounter process. The holonic architecture allows us to set the requirement of bounded rationality for all members of sub-holons in order to find the best possible action within a given resource allocation.

In short, the differential diagnostic problem can be recursively broken down into sub-problems by weighting the likelihood of the presence of possible diseases. These subproblems may induce different abstraction levels and can be of different granularities. According to the nature of DDx, the problem solvers are collaborative and those dealing with similar diseases need to have more communications, which are to be conducted in a timely manner. As a result, the DDx domain clearly meets the characteristics of holonic domains.

Chapter 5
SYSTEM DESIGN: THE HOLONIC MEDICAL DIAGNOSIS SYSTEM (HMDS)

Systems design is the process of defining the architecture, modules, interfaces, and data for a system to satisfy specified requirements [131]. This chapter covers the system design process for the HMDS. Details of the functionality and the self-organization of the system, which are respectively designed based on the functional and nonfunctional requirements of the system, are also presented in this chapter.

It should be mentioned that the system owes its coming to be and its inspiration to the pioneering work presented in [101, 132, 86], which introduced a holonic multi-agent system for medical diagnosis. There are, however, many differences between the system introduced in this work and the old one. Unlike that previous work, the architecture and the functionality of the current system is based on the DDx approach for medical diagnosis, and this results in new holon formation strategies and in a different self-organization approach. Moreover, this system applies a number of machine learning techniques that are first introduced during the development of this system and support the self-organization of the current system with its unique approach to medical diagnosis, so clearly, they have not been used in the previous system. This section starts with the architecture of the HMDS, and then shows the functionality of the system. This is then followed by examples of system's functionality in number of simulations of the system.

5.1 The Architecture of the HMDS

5.1.1 System Structure

In general, a medical diagnosis system may either rely on highly smart deliberative agents as one extreme or on a large set of comparatively simple (reactive) agents as the other extreme. The first means that agents need to fully understand at least their area of expertise and need to have at least a basic understanding of the real world. This means that agents need to rely on a deep-going knowledge and deduction model that usually requires intensive computing power. The other extreme, which is chosen here, is to keep agents extremely simple and to get the smartness out of the smart and sophisticated interplay of extremely large amounts of simple agents as it is realized by swarm intelligence-based systems. The HMDS as an HMAS realizes an improved version of the second approach. It consists of two types of agents: comparatively simple Disease Representative Agents (DRAs) as the end nodes of the holarchy and more sophisticated Disease Specialist Agents (DSAs) as decision makers on the higher levels of the system [8]. Figure 14 illustrates the DRAs and DSAs in the HMDS.

DRAs are atomic agents, thus, are not further decomposable and form the leaves of the holarchy. Each DRA is an expert on a specific disease or even only on a different appearance of it. It maintains a pattern store that contains the Disease Description Pattern (DDP) – an array of possible signs, symptoms, and test results, i.e., the holon identifier. Thus, in order to join the diagnosis process, these agents only need to perform some kind of pattern matching (i.e., calculating their Euclidean distance to the diagnosis request description pattern).

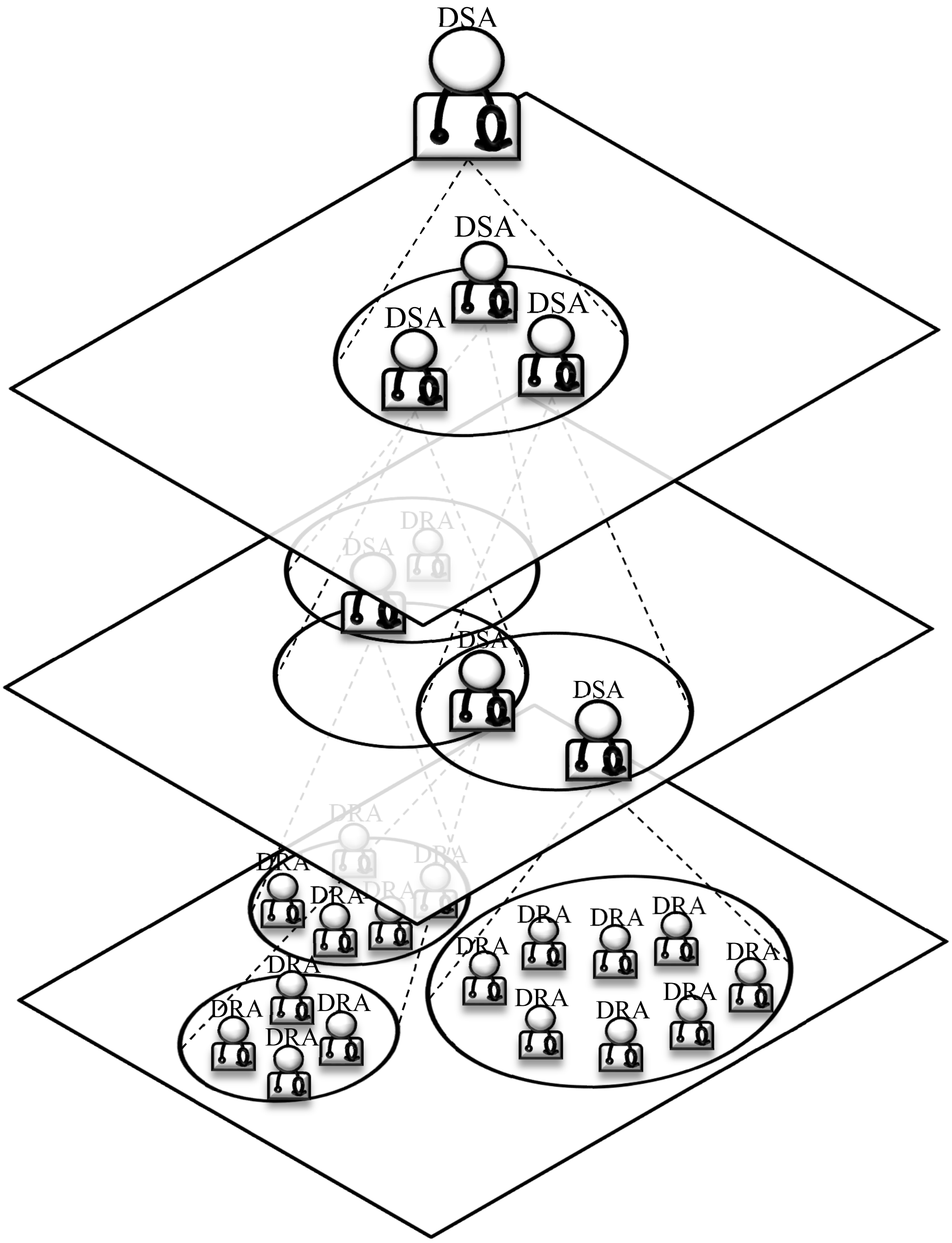


Figure 14. DRAs and DSAs in the HMDS

DSAs are holons consisting of numbers of DRAs and/or DSAs that rely on similar sets of symptoms; i.e., represent similar diseases. In this case, the holon identifier will be an average of holon identifiers of the members. This encapsulation, in fact, enables the implementation of the DDx. DSAs can deal with a more or less broad domain of instances of related diseases. The higher they are in the holarchy the more general and broader their knowledge needs to be. A DSA on a higher level is assumed to cover a superset of all sets of diseases that are represented by all its body agents on the next lower level, however, on a more abstract level. For each DSA, a head is defined for its lifetime, representing its members by providing the common interface to the outside of the holon, i.e., to the next higher level in the holarchy [8] (see Figure 15 [8]).

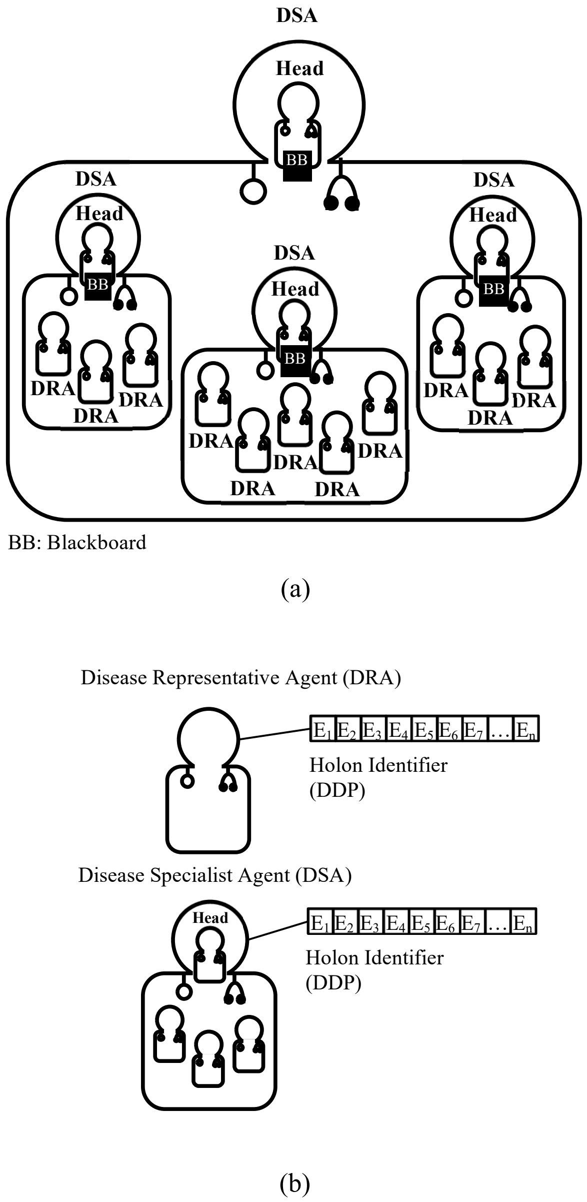


Figure 15. (a) DRAs and DSAs in the HMDS (b) Holon identifier in DRA and DSA

This head will not be chosen from the available members, but will be created for the lifetime of the holon, based on agent cloning (for more information on agent cloning please refer to [133, 134, 135]). Agent cloning is a "comprehensive approach to the problem of local agent overloads" [133, p. 58], and since the agents may leave their super-holons, it can reduce the complexity of the system. Given the fact that heads have same functionalities and their creation is merely needed when a new super-holon is being formed within an existing super-holon, indicates how agent cloning is a perfect solution to the mentioned problems. For this purpose, each head is capable of cloning, i.e., creating a copy of its code, and passing the relevant information to the new agent [8].

The holarchy has one root, in fact a DSA, which will play the role of the most common and inclusive interface to the outside world for the complete holarchy. Due to its self-organization ability the system can start with this DSA, take all the DRAs as its members, and then let the DSAs form automatically. Although this process is based on the affinity and satisfaction, and at the beginning, no information about the satisfaction factor is available, it is still possible to initially form the DSAs based on the affinity, i.e., the similarity between them. For this reason, the mentioned DSA accepts the initial description of the diseases in form of DRAs, as its members, clusters them, and defines for each of the clusters (i.e., super-holons) a head[21]. This is repeated recursively until no further clustering is necessary[22]. This step is not mandatory but can be performed once as the system is being defined and accelerate the self-organization. Later on, the system can still reorganize its architecture using the rest of its self-organization techniques [8] (For more information please refer to section 5.3 and chapter 6[23]).

The communication between agents is solely done via the blackboard of each DSA. More information about blackboard systems is presented in [136] and [137]. According to their types, agents in the HMDS also need to save a subset of the following data in their memory: Respective symptoms, Super-holons and their corresponding Q-Values (QVs) (see chapter 6), Subholons and their corresponding Q-values, Diagnosis request, Intermediate results of the diagnosis process. In addition, the members of the super-holons need to have access to some of the

[21] According to the general possibilities for modeling holonic structures presented in [100], this approach will design super-holons as moderated groups, where agents give up part of their autonomy to the super-holon, which will be achieved by the introduction of one agent as a representative or head of the holon. For this reason, either one of the members of the holon would the role of the head and as a result gains the additional functionality or a new agent would be created for the lifetime of the holon.

[22] The Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [13] is one of the best algorithms for this issue. In [7] a simple and effective method for automatically detecting the input parameter of DBSCAN is presented, which helps best to deal with complicated data such as diseases.

[23] The number of levels of the holarchy depends on the number of diseases the system covers and their signs and symptoms. At a rough estimate, the logarithm of the number of diseases, taking the average number of super-holon members as the base, can be considered for this purpose. As a simple example, supposing that the system can diagnose 10000 diseases that in average are linked to 10 diseases for the reason of differential diagnosis, and that each DSA is also associated to 10 other DSA in average, then $\log_{10} 10000 = 4$ levels are to be expected.

data kept by their super-holons, and even share some information with the other members of their super-holons. With this regard, the super-holon's functionalities can best be supported by blackboard systems [8].

The class diagram presented in Figure 16 includes the different types of agents considered in the HMDS, their operations (methods), and their relations (the AUML notation reference for the class diagrams is presented in appendix A).

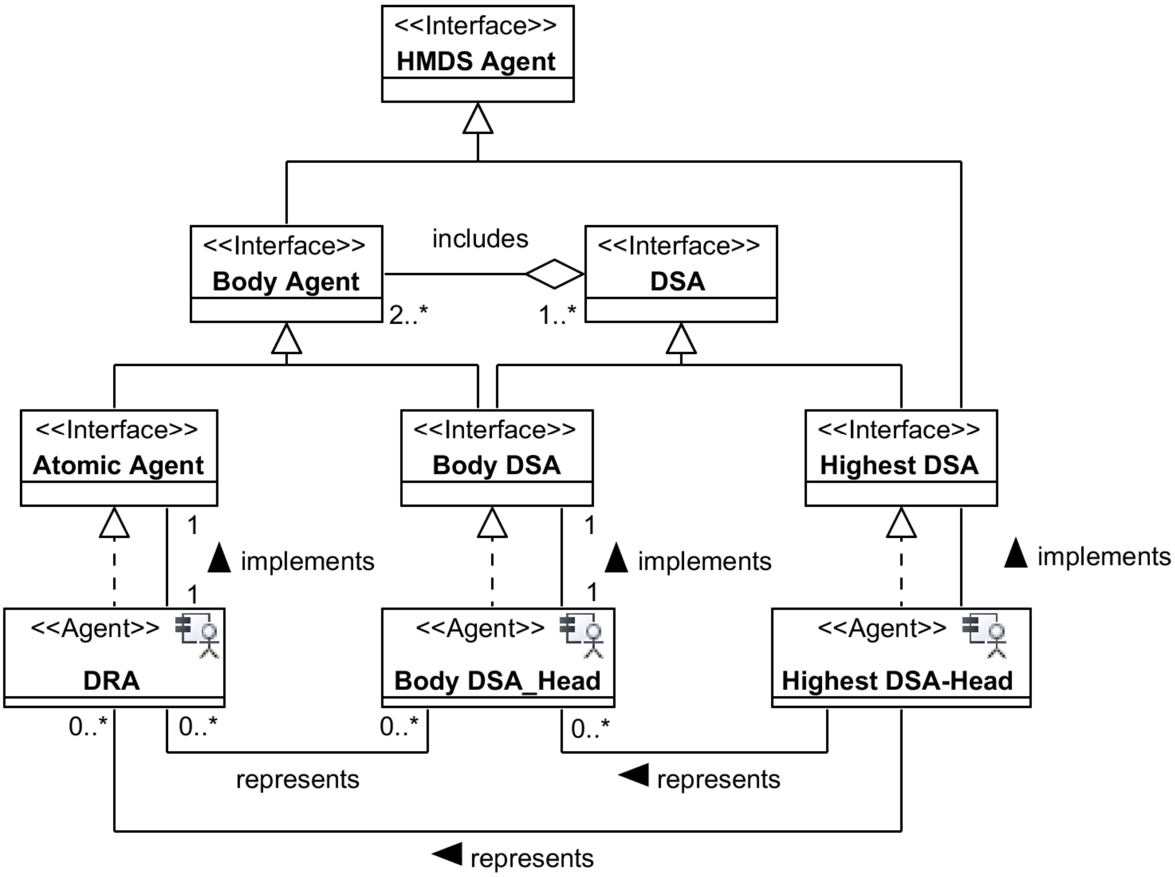


Figure 16. The AUML class diagram for the HMDS[24]

The HMDS Agent interface is the generalization of all the elements in the system. In general, the agents in the HMDS are either DSAs or DRAs. However, from another point of view one can distinguish the agent on the highest level of the system from the body agents. This classification is because of the different approaches they have to the same operations. The DSA interface in this diagram is the generalized super-interface of the highest DSA interface and the body

[24] Designed using Visual Paradigm Version 14.0

DSA interface. On the other hand, the Body DSA interface inherits together with the Atomic Agent interface all the attributes of the Body Agent super-interface. Each DSA as a super-holon is represented by a head. This head will not be chosen from the available members, but will be a completely different agent, created for the lifetime of the holon, based on agent cloning (For more information please refer to section 5.1.4). In the implementation of the system each DSA is implemented by a single head agent (depicted by the realization and association relationships in Figure 16). As a result, the system is consisted of DRAs and DSA heads.

Moreover, it should be noted that a DSA is a super-holon consisting of several holons. The Body Agent interface is defined here as the generalization of the atomic DRAs and Body DSAs. The aggregation relationship depicted in Figure 16 shows that each DSA consists of at least two body holons and that each body holon may be a member of one or several DSAs. In the implementation, however, a Body DSA may represent less than two members at the point it decides to kill itself (see section 5.3). This is depicted by the association relationships at the bottom of the diagram. The aggregation relationship, however, indicates that DSAs remain alive only if they have more than one member (see section 5.3).

5.1.2 Agent Architecture

According to [122, p. 13], based on their architectures, intelligent agents fall into one of the following classes:

1. logic based agents – in which decision-making is realized through logical deduction;
2. reactive agents – in which decision-making is implemented in some form of direct mapping from situation to action;
3. belief-desire-intention agents – in which decision-making depends upon the manipulation of data structures representing the beliefs, desires, and intentions of the agent; and finally;
4. layered architectures – in which decision-making is realized via various software layers, each of which in more-or-less explicitly reasoning about the environment at different levels of abstraction.

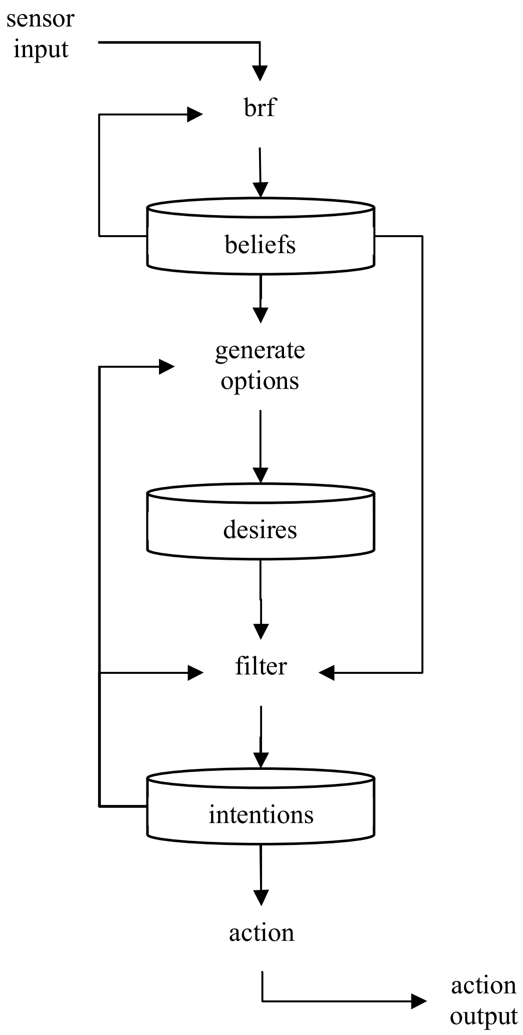


Figure 17. Schematic diagram of a generic belief-desire-intention architecture

In the HMDS, agents are designed based on the Belief-Desire-Intention (BDI) architecture. The process of practical reasoning in BDI agent is summarized in Figure 17 [122, p. 32]. As this figure illustrates, there are seven main components to a BDI agent:

1. a set of current beliefs, representing information the agent has about its current environment;
2. a belief revision function (brf), which takes a perceptual input and the agent's current beliefs, and on the basis of these, determines a new set of beliefs;
3. an option generation function (options), which determines the options available to the agent (its desires), on the basis of its current belief about its environment and its current intentions;

4. a set of current options, representing possible courses of actions available to the agent;
5. a filter function (filter), which represents the agent's deliberation process, and which determines the agent's intentions on the basis of its current beliefs, desires, and intentions;
6. a set of current intentions, representing the agent's current focus – those states of affairs that it has committed to trying to bring about;
7. an action selection function (execute), which determines an action to perform on the basis of current intentions. [122, p. 31]

These components can be mapped to the following features in the HMDS:

1. Set of current beliefs: holon identifier (DDP), current super-holon(s), corresponding Q-value(s) (see chapter 6).
2. Belief revision function: every agent is able to update its beliefs based on the new perceptual inputs. Whenever an agent reacts to a diagnosis request and joins the diagnosis process, there will be the possibility to update its relevant Q-value(s) and/or holon identifier (DDP). Moreover, if there are changes in the memberships, it can update its super-holon(s).
3. Option generation function: this component determines the desires of an agent. For each agent the options may include remaining in the current super-holon(s), moving to a specific super-holon, or starting an exploration.
4. Set of current options: the output of the option generation function, representing the courses of actions available to the agent.
5. Filter: this component determines the intentions of an agent with respect to the system's self-organization. These decisions are made based on Q-values and toward their maximization.
6. Set of current intentions: each intention represents a course of actions available to agent, such as moving vertically or horizontally in the holarchy, including the diverse number of steps possible.
7. Action selection function: determines the actions, i.e., the movement of an agent in the holarchy on the basis of current intention.

For more information on the self-organization process please refer to chapter 6.

5.1.3 Agent Environment

Agent Environment is where agents act and learn. This indicates that in order to guarantee the quality of the final diagnosis and the improvement of the system the environment of each agent should be chosen very carefully. If an agent is acting in a wrong environment, it cannot collaborate with the right agents and eventually will not meet its delegated objectives, which in the HMDS will lead to wrong diagnoses. Moreover, medical knowledge demonstrates a steady upward growth, and diagnosis is also very much affected by the geographical regions. As a result, in order to adapt and improve the behavior of the system, it is needed to: (1) update the medical knowledge based on the new instances, (2) improve the holarchy according to the experience and the feedback. In the HMDS, holon identifiers are updated applying the exponential smoothing (supervised learning), and the self-organization of the holarchy is supported by Holonic-Q-learning (reinforcement learning) (for more information please refer to chapter 6). Since both methods require feedback from the environment, it is clear that the quality of the feedback is of central importance.

It should be noted that the terms “feedback” and “reward” have different definitions here. A feedback is the final diagnosis, suggested by the physician for a given diagnosis request. However, a reward is a numerical value, which is calculated using a reward function that considers the feedback. Agents receive their rewards from their environments. According to [128] hierarchical MAS with composition structure, which do not imply a hierarchical centralized control may use three different approaches in order to interact with the environment (see Figure 18 [128, p. 50]):

- Vertical Environment: Every layer of the system may interact with the environment, each one acting at a different level of abstraction. This approach requires specific means to modify and perceive the environment at its own level”.
- Horizontal Environment: Only one layer is allowed to interact with the environment. This layer, usually the lowest one, acts as the interface with the environment interacting with it through actions and perceptions.
- Disjoint Environment: Each layer has its own environment and they do not overlap. So, they do not act in the same environment. [128, p. 50]

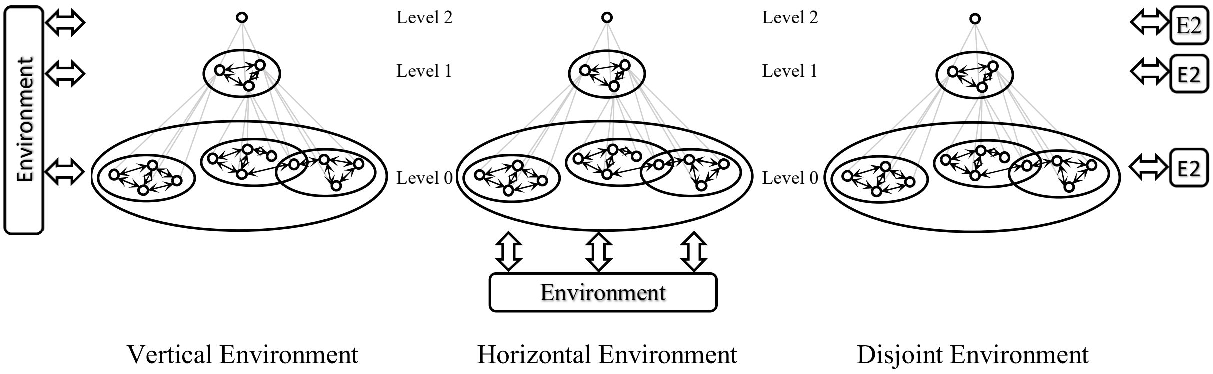


Figure 18. Environment representation

In the HMDS, the highest and most inclusive DSA will play the role of the unique interface of the system to the outside world for the complete holarchy (Figure 19). The members of a super-holon are masked to the outside world by their representative, i.e., the head of their super-holon. Joining a super-holon, these agents accept to lose a part of their autonomy (moderated group [128]) and limit the information they can receive to what their head would provide them.

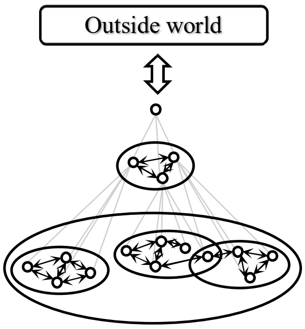


Figure 19. The unique interface of the HMDS to the outside world

The HMDS has distributed environments, i.e., the highest holon receives the feedback from the outside world, however, the environment for the rest of the holons is their own super-holon, from which they receive their rewards (Figure 20).

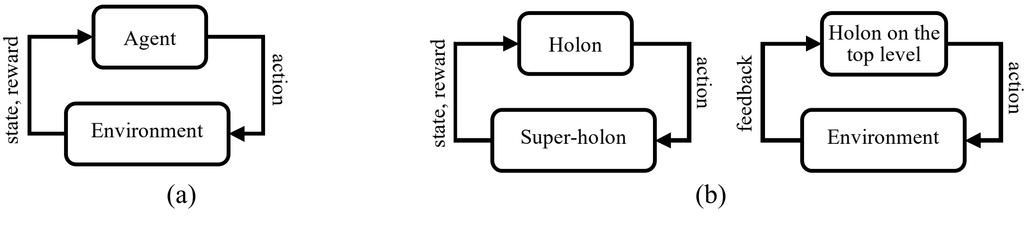


Figure 20. The agent-environment interaction in (a) Q-learning (b) Holonic-Q-learning

5.1.4 Agent Birth, Cloning, Spawning and Death

Generally, any self-organizing system should deal with the birth and the death of its agents. These two topics are very much related and can be addressed together. The birth of an agent may also be implemented through cloning and/or spawning of existing agents. As a result, this section covers the realization of these four cases in the HMDS.

Clearly, the birth of an agent is the result of the need for its existence and its death occurs when the system does not need its actions any longer. Agent cloning first proposed in [133] is a "comprehensive approach to the problem of local agent overloads" [133, p. 58]. For the purpose of cloning, an agent should create a copy of its code, which may have to undergo some modification. In addition, the agent should pass "only the relevant tasks and information which are necessary for the tasks passed to the clone" [133, p. 64]. Agent cloning has already been considered for the self-organization purpose in holonic multi-agent systems. In [138], it is discussed that a self-organizing system includes three principles: having a good address book, sharing knowledge, and recruiting new able collaborators. The latter is mentioned in [134] as cloning/spawning[25]; and it is argued that holonic multi-agent systems as one of the main self-organizing systems, should support agent cloning. In [135, p. 53], "holonification based on partial agent cloning and merging" is proposed. In this approach, "in order to solve a certain problem cooperatively, participating agents create a copy of their knowledge and functionality which are relevant for the given problem (partial agent cloning). In the next step, all partial agent copies are merged within a new agent which represents the holon" [135, p. 53].

The HMDS performs a unique process in order to initiate its self-organization. At the beginning each disease is represented by a DRA and a single DSA accepts all the present DRAs as its members. This DSA will be the highest and most inclusive holon of the holarchy. This agent then performs clustering based on similarity, during which subgroups of its members are merged in order to create new super-holons/DSAs. Next the existing DSA will clone itself and create new DSAs as the heads of the new super-holons. The new DSAs will now be the immediate members of the older holon, indicating an average of the symptoms under consideration

[25] The difference between the two methods is that for cloning, for cloning, "the cloned agents are perfect replicas of the original agents and fulfill the same roles and responsibilities as the original agents", while for spawning, "the spawned agents are specialized on a subpart of the spawning agent's task structure, which is no longer the responsibility of the spawning agent" [140, p. 545].

by each of its member, and their members are now masked to their old head through their new DSAs. In the same manner, each head should now be capable of cloning itself, in case some of its members aim to merge and create a new super-holon. This process eventually stops when no further clustering is possible and results in the initial holarchy (for more information please refer to chapter 6).

The mentioned process explains the birth of the agents during the first stage of the self-organization. However, as mentioned the system continues to perform the self-organization using ML techniques (see section 5.3 and chapter 6). During this process, the need for new collaborations may be indicated, which would eventually necessitate the birth of new DSAs. New DSAs may be introduced each time by merging the relevant agents and cloning the immediate DSA as mentioned above. It is also to be noted that, if agents realize that any collaboration, i.e., membership in a certain holon, is no longer productive, they can leave their super-holon and this memberless DSA then will decide to die.

Another situation to be considered for the birth of the new agents is when the system is unable to finish its assigned tasks on time. In the current study, this may be the case if an agent is assigned with multiple tasks requested by different agents. In the HMDS, it may happen a lot that one disease should be considered in two or more super-holons for the sake of differential diagnosis. However, this does not necessarily mean that multiple DRAs are needed for such diseases. There are two main reason not to perform cloning for this purpose. First, holonic multi-agent systems allow the agents to be a part of more than one super-holon, if the super-holons' goals do not contradict each other or the holon is indifferent to those conflicting goals. As mentioned in chapter 4 such agents are called multi-part. The second reason comes from the fact that agents perform exploration, hence, if there are same agents in some other parts of the holarchy, it is not necessary for them to explore those parts, but by creating multiple DRAs for same diseases there are no mechanism for them to know this.

As mentioned, multi-parts may often receive multiple tasks at the same time and become unable to finish them on time. Facing this problem, one solution for the agent is to create a new agent to take a part of its workload. The agent has two options, namely cloning an agent or spawning off an agent [134]. According to [139, p. 415], "while agent cloning is a possible response of an agent to overloads, agent spawning includes, in addition, consideration of the data transfer necessary for task execution, and it relaxes the requirement of creating an identical copy of the original agent. Thus, spawning further enhances efficiency of source utilization and reduction of communication and computation loads". For cloning, "the cloned agents are perfect replicas of the original agents and fulfill the same roles and responsibilities as the original agents", while for spawning, "the spawned agents are specialized on a subpart of the spawning agent's task structure, which is no longer the responsibility of the spawning agent" [140, p. 545].

As discussed in [134] spawning is triggered when and agent cannot finish the assigned tasks on time. A spawned agent is a subordinate of the original agent and cannot establish relation with other agents. When such a spawned agent finishes the assigned task, it will remain in idle status and after a pre-defined period it will destroy itself. On the other hand, cloning happens when an agent is overloaded with too many neighbors. In this case, the original agent assigns a part of its tasks/neighbors to the cloned agent. Different from the spawned agents, cloned agents will not destroy itself even if it keeps an idle status.

In the HMDS, using multi-parts will help to reduce the communication and exploration load, however, in case several heads are requesting the multi-part to perform different sub-tasks the calculation load will increase. In order to face this overload spawning is used. As a result, whenever there are multiple heads requesting sub-tasks from a multi-part, this agent will create a spawned agent for each of them, assigning the corresponding sub-tasks to them. In our case, these sub-tasks, merely include the calculation of affinity and updating the Q-value (For more information please refer to chapter 6). After the calculations, if the spawned agent is idle for a long time, it will die. However, regarding the multi-part agent, if any of its memberships happens to be ineffective the DRA will not die but will leave the super-holon and drop the connection.

To put in a nutshell:

- A new DSA can be created by an existing DSA using cloning.
- Spawning may be used by agents that are assigned with multiple tasks that cannot be finished on time.
- A DRA will not die, however, as an unsatisfied multi-part it may leave any of its super-holons.
- A DSA will die if it has no members or only one member reading its blackboard.
- A spawned agent will die if it is idle (has no tasks to undertake) for a certain period of time.

5.1.5 Agent Memory

Memory plays an important role in systems with learning abilities, and in general, any adaptive system is to implement this capability. Using memory, agents can modify their behavior based on the results of the previous actions, i.e., they can learn to adapt.

In the HMDS, agents use memory in order to estimate the state of their environment from individual observations and adjust their actions accordingly, which at the same time improves the collective performance. According to their types, agents in the HMDS need to save a subset of the following data in their memory:

- Respective signs and symptoms (DDP)
- Super-holons and their corresponding Q-values
- Sub-holons and their corresponding Q-values
- Diagnosis request
- Intermediate results of the diagnosis process

In addition, the members of the super-holons need to have access to some of the data kept by their super-holons, and even share some information with the other members of their super-holons in order to perform their functionalities. Considering these needs, the super-holon's functionalities can best be supported by blackboard systems.

"Blackboard systems were the first attempt at integrating cooperating software modules. The goal was to achieve the flexible, brainstorming style of problem solving exhibited by a group of diverse human experts working together to address problems that no single expert could solve alone" [137, p. 1]. According to [136] and [137] a blackboard system consists of three main components (see Figure 21 [137, p. 3]):

- Knowledge sources (KSs) are independent computational modules that together contain the expertise needed to solve the problem. KSs can be widely diverse in their internal representation and computational techniques and are anonymous in that they do not interact directly with one another or know what other specific KSs are present in the system.

- The blackboard is a global data repository containing input data, partial solutions, and other data that are in various problem-solving states. All interactions between the KSs are via changes made on the blackboard.
- A control component that makes runtime decisions about the course of problem solving and the expenditure of problem-solving resources. The control component is separate from the individual KSs. [137, p. 3]

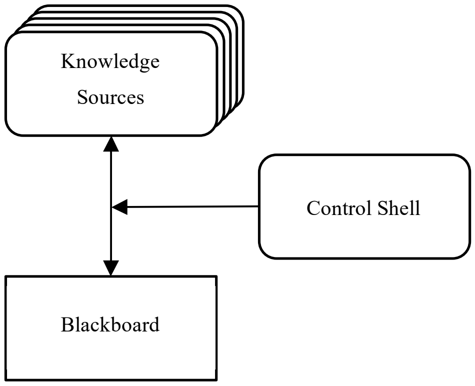


Figure 21. Blackboard system components

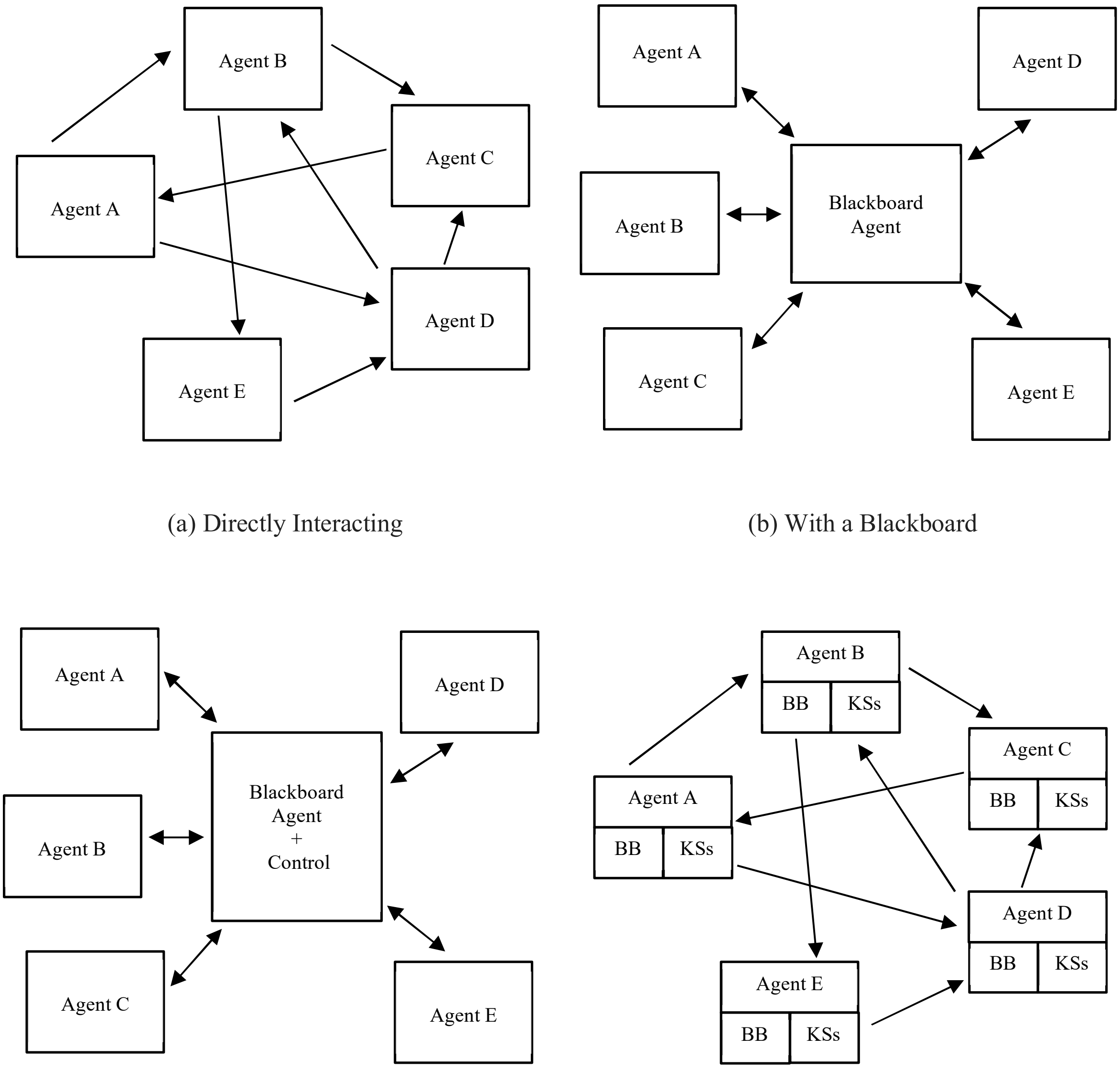
A comparison between the blackboard system and multi-agent system approach has been done in [137]. Considering each KS as an agent in a multi-agent system, this research suggests the combination of these two techniques (see Figure 22 [137, p. 10]).

According to [137], Figure 22(a) represents a MAS in which agents do not have a blackboard in their environment. As a result, each agent must not only decide what results to share, but what other agents to share them with. Additionally, each agent must keep track of the results received from other agents, as well as the ones it has produced locally. If agents need to save large amounts of data, some sort of repository will be needed for each them. Finally, each agent must decide what it should be doing, using only its local view of problem-solving activities. These tasks could be simplified for the agents by providing them with simple rules for what and who they should communicate with, however, if these become too static the desired flexibility in the interactions cannot be achieved and the system will have a limited directly connected interaction structure.

In Figure 22(b) a special blackboard agent has been added to the set of agents. In this structure the agents can communicate directly with one another or interact indirectly via the blackboard agent. Since in this combination, the blackboard agent acts as a repository of information, the need for each agent to maintain its own repository can be eliminated. As discussed in [137, p. 10], "even though such functional centralization would have been harshly criticized by early MAS researchers, similar approaches have been used in federated multi-agent architectures, where specialized agents served as matchmakers, facilitators, or brokers in order to eliminate the redundant work by centralization". To further simplify the communication decisions in this combination, all agent communications can be required to be with the blackboard agent. To ensure that agents are notified when useful information is given to the blackboard agent, the agents will also have to register their desires with the blackboard agent beforehand. When such information arrives, the blackboard agent will notify the appropriate agents of its arrival. In this approach, the agents still have to make local control decisions about what they should be doing, however, much of the information needed to make these decisions is now located at the blackboard agent.

The agents can also delegate their local control decisions to the blackboard agent, which now also becomes a manager agent that tells the other agents what to do (Figure 22(c)). While co-locating the control decision-making with the data required for these decisions makes sense, the resulting MAS has failed to implement the local autonomy of traditional MASs. In fact, a closer look at Figure 22(c) shows that this approach essentially implements the KS interface in a traditional blackboard system using MAS technology.

As discussed in [137, p. 11], "traditional multi-agent and blackboard systems can be viewed as two diverse points in the collaborating-software design space. Traditional blackboard-system research has concentrated on closely collaborating problem solving techniques with a single thread of activity operating in a centralized setting". MAS research, on the other hand, has concentrated on agents that collaborate concurrently in a distributed environment. "What will be important is using the appropriate technology in the right context" [137, p. 11]. Figure 22(d) illustrates a simple example of an architecture that provides flexibility in grouping closely interacting entities together in an agent-based environment.



(a) Directly Interacting

(b) With a Blackboard

(c) With a Blackboard and Control (Manager) Agent

(d) Full-Fledged Blackboard Agents

Figure 22. Collaborating Agents

The logic behind this architecture is very similar to the reason for using HMASs, in which closely interacting holons that work on same subset of data form super-holons and can even further be seen as KSs of a larger blackboard system (Figure 23). As a result, blackboard system is used in the implementation of the HMDS in order to provide the holons with the right memory and means of collaboration. As a result, DSAs facilitate their member's interactions by a blackboard, and the DRAs will be provided with a simple memory to solely save their DDP, superholon(s) and the corresponding Q-value(s).

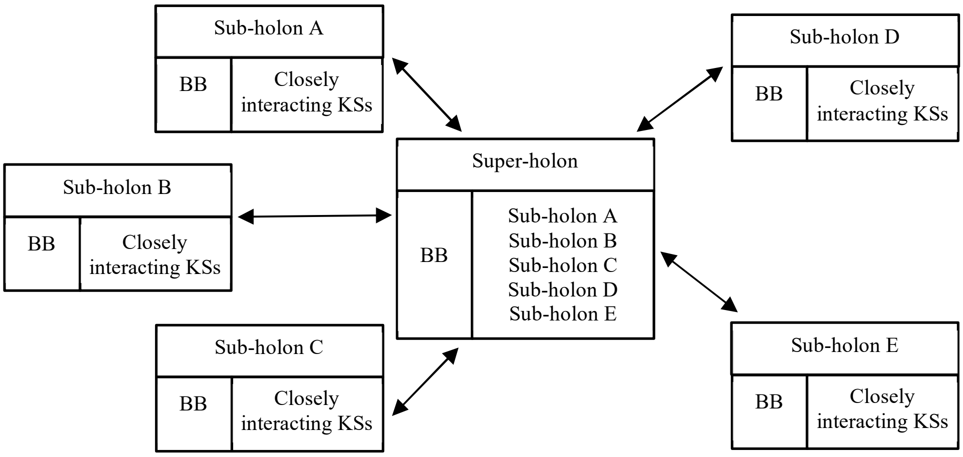


Figure 23. Holonic Blackboard System

5.2 The Functionality of the HMDS

In principle, the proposed system works as follows: When a request for a medical diagnosis is sent to the HMDS it is actually received by the head of this holarchy (as explained already above). This head receives the request as a specific combination of signs, symptoms and medical test results and places it as an array on its blackboard. Each agent of the system which has knowledge of this blackboard, i.e., any member of this super-holon, can read the messages on this blackboard. A DRA's reaction to a request message is to send back its similarity to the request. However, based on the provided information a DSA may decide that it wants to try to join the diagnosis process or not. This will actually control the data flow in the holarchy. The decision is made based on some simple statistical information about the DSA's members. The head knows its distance to each of its members. So, it just calculates its distance to the request and in case the request is not an outlier, the head will decide to join the diagnosis process. This means that it will read all the information from the blackboard of its head and will place it on its own blackboard. Then the same process starts again and repeats recursively until the request reaches the final level of the holarchy [8]. Results obtained by participating agents now flow the other way round from bottom to the top of the holarchy. On their way up the results and their corresponding questions are sorted according to their similarity. More precisely, each agent will send its final results, suggestions and questions to its super-holon including: the top diagnoses together with all the signs, symptoms or test results that are relevant from the agent's point of view. This implies that originally not provided relevant information might be requested from the user in a second step. In fact, according to the DDx, the system may suggest the user to provide more information or undertake specific medical tests to improve diagnostic accuracy [8] (see Figure 24). Section 7.3 demonstrates the functionality of the system in some simulations of the system.

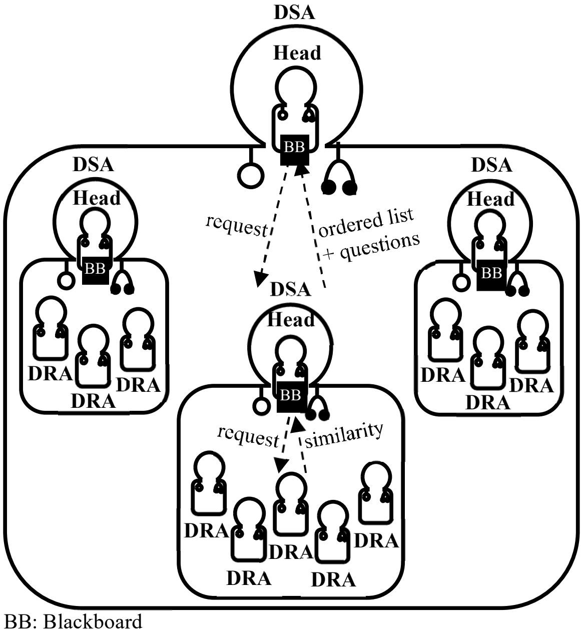


Figure 24. Diagnosis process in the HMDS

As mentioned, the system output consists of the diseases with highest similarities to current input and the suggested signs, symptoms, and test results to be controlled. It should be noted that each supplementary input will refine and update these lists as the user continues to complete its input. The system orders the diagnoses based on their similarity to the diagnosis request. The frequency of diseases will also be displayed as valuable hints for final diagnosis. These values may also be used to detect possible outbreaks faster. It should be noted that the system does not consider the frequency of diseases as a factor in ordering the top diagnoses, since as a DDSS one of the main goals of this system is to remind the physician about the critical possible diseases, which could very likely be rare diseases.

Regarding the questions, i.e., the signs, symptoms, and test results to be controlled, it should be noted that even though the system aims to speed up the H&P by suggesting well-focused questions, these questions should still be comprehensive enough to avoid missing the right diagnosis or multiple concurrent diseases. In cases where the diagnosis request is not complete enough to reach the DSAs that contain DRAs, the DSA that has joined the diagnosis process will send back all the signs and symptoms that are present in any of its members. However, if it receives replies from its member, i.e., if the diagnosis request is complete enough to activate some of the DSA's members, it will solely send the results and questions of its participating members to its super-holon.

As a result, the quickest and safest way to shorten the list of questions is to remove signs and symptoms by ruling out the conditions that are associated with them as soon as possible. To this end, in such cases, it is suggested to start with questioning or examining possible signs and symptoms that are associated with closest to one half of the total number of listed conditions. This prioritization helps the system to rule out more diseases after same number of questions in comparison with scenarios, in which questions are displayed in random order.

Regarding the questions suggested by lowest DSAs, which consist of DRAs, it is suggested to control all the relevant signs, symptoms, and tests. In fact, the diseases listed in the differential diagnosis list may belong to one or more of the following groups:

1. leading or provisional diagnosis (most similar diagnosis)
2. common diagnosis
3. urgent "don't miss" diagnosis (fatal or seriously harmful)
4. Unusual feature diagnosis (rare or concurrent disease)

The aim is to not miss any of the possible diagnoses and as a result each diagnosis should be addressed individually. With respect to the order of this procedure, according to [141] and [142], aside from the inefficient possibilistic approach, which considers all known causes equally likely, the physician may apply one of the following approaches:

1. probabilistic approach: starting with those diseases that are more likely
2. prognostic approach: starting with those diseases that are fatal or seriously harmful if undiagnosed and untreated
3. pragmatic approach: starting with those diseases that are more responsive to treatment if offered

As explained in [142], each approach has its own limitations and experienced clinicians simultaneously integrate probabilistic, prognostic, and pragmatic approaches while updating and reordering the differential diagnosis list. As the pragmatic approach mainly becomes meaningful when dealing with the final version of the differential diagnosis list and not in the preceding rounds, the HMDS continuously applies the prognostic and when not applicable the probabilistic approach in order to avoid delay in treatment. It is recommended to alternately consider the immediate DSAs that are contributing to the current DDx list during this disease selection to avoid missing multiple concurrent diseases. Considering each disease, the system then starts with questioning and / or examining possible relevant signs and symptoms that are marked as pathognomonic and / or sine qua non, followed by the most frequent ones in that disease in order to rule diseases in / out as quickly as possible.

To put in a nutshell, it is suggested to consider two different approaches with respect to the immediate DSA that is suggesting the sign or symptom to be questioned:

1. If the diagnosis request has stuck at higher level DSAs, it is suggested to start with questioning or examining possible signs and symptoms that are associated with closest to one half of the total number of listed conditions in order to rule disease categories in / out in a timely manner.
2. In case the diagnosis request is complete enough to reach the DRAs, it is recommended to alternately consider the immediate DSAs that are contributing to the current DDx list and to continuously integrate the prognostic and when not applicable the probabilistic approach (considering both similarity and frequency) for each of them. After deciding on the order of the diseases to be considered, for each disease it is recommended to begin with questioning or examining possible signs and symptoms that are marked as pathognomonic or sine qua non, followed by the most frequent ones for that disease.

It should be noted that as it is generally much easier and more efficient to control the signs and symptoms before performing laboratory tests, questions that consider laboratory tests results may always appear at the end of the list. However, the user can enter the answers in any desired order or even change the value of the elements that have not been questioned at any time, as the system solely aims to act as a reminder and suggest the questions.

The goal diagram presented in Figure 25 describes the diagnosis process in the HMDS and shows how different type of agents collaborate in order to provide the user with the final diagnosis (the AUML notation reference for the goal diagrams is presented in appendix A).

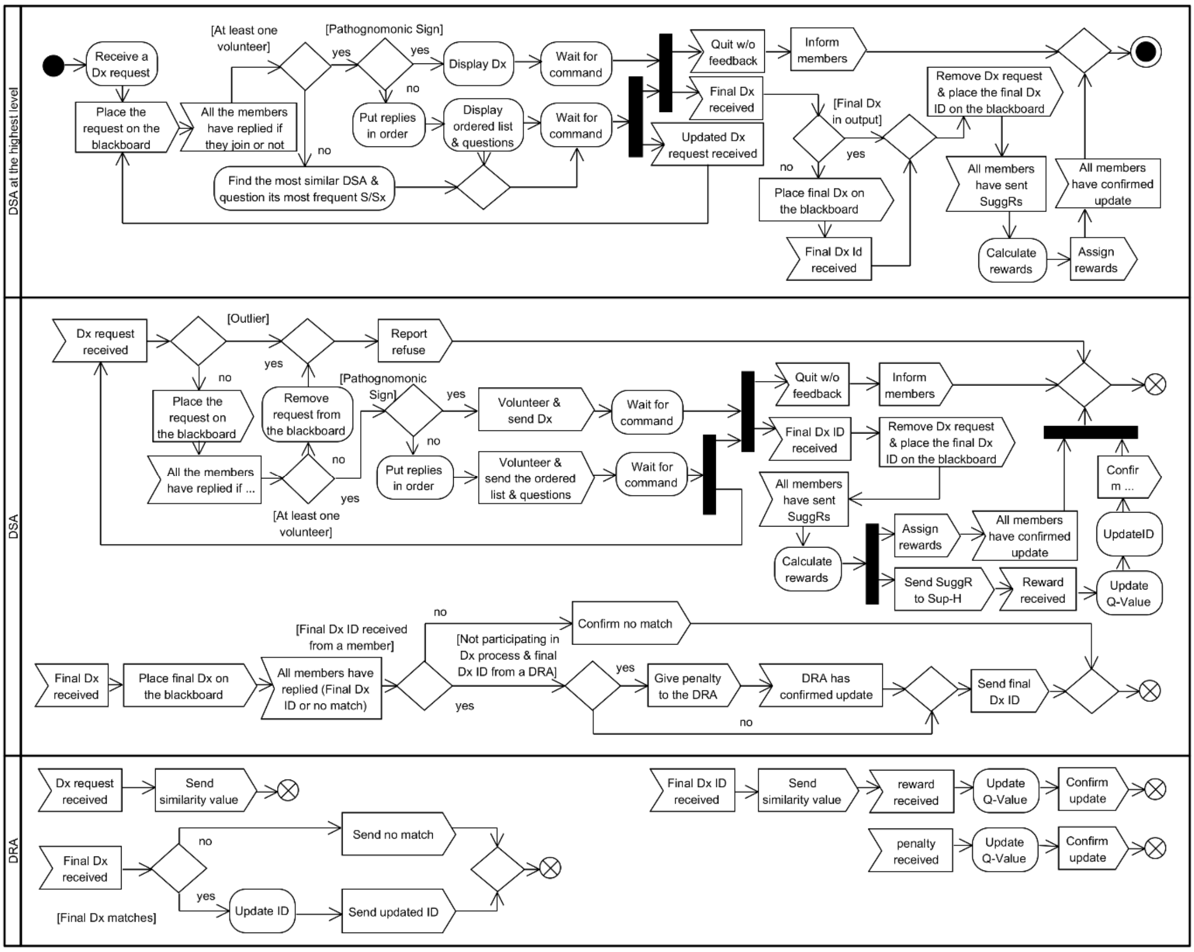


Figure 25. The goal diagram for the diagnosis process in the HMDS[26]

5.3 Self-Organization in the HMDS

As already mentioned, the HMDS performs clustering in order to initiate its self-organization and build its initial holarchy (for more information see section 6.1.1). Having this starting point the system will then continuously use RL (for more information see section 6.1.3) for the purpose of self-organization. As noted, self-organization is defined as the mechanism or the process enabling a system to change its organization without explicit external command during its

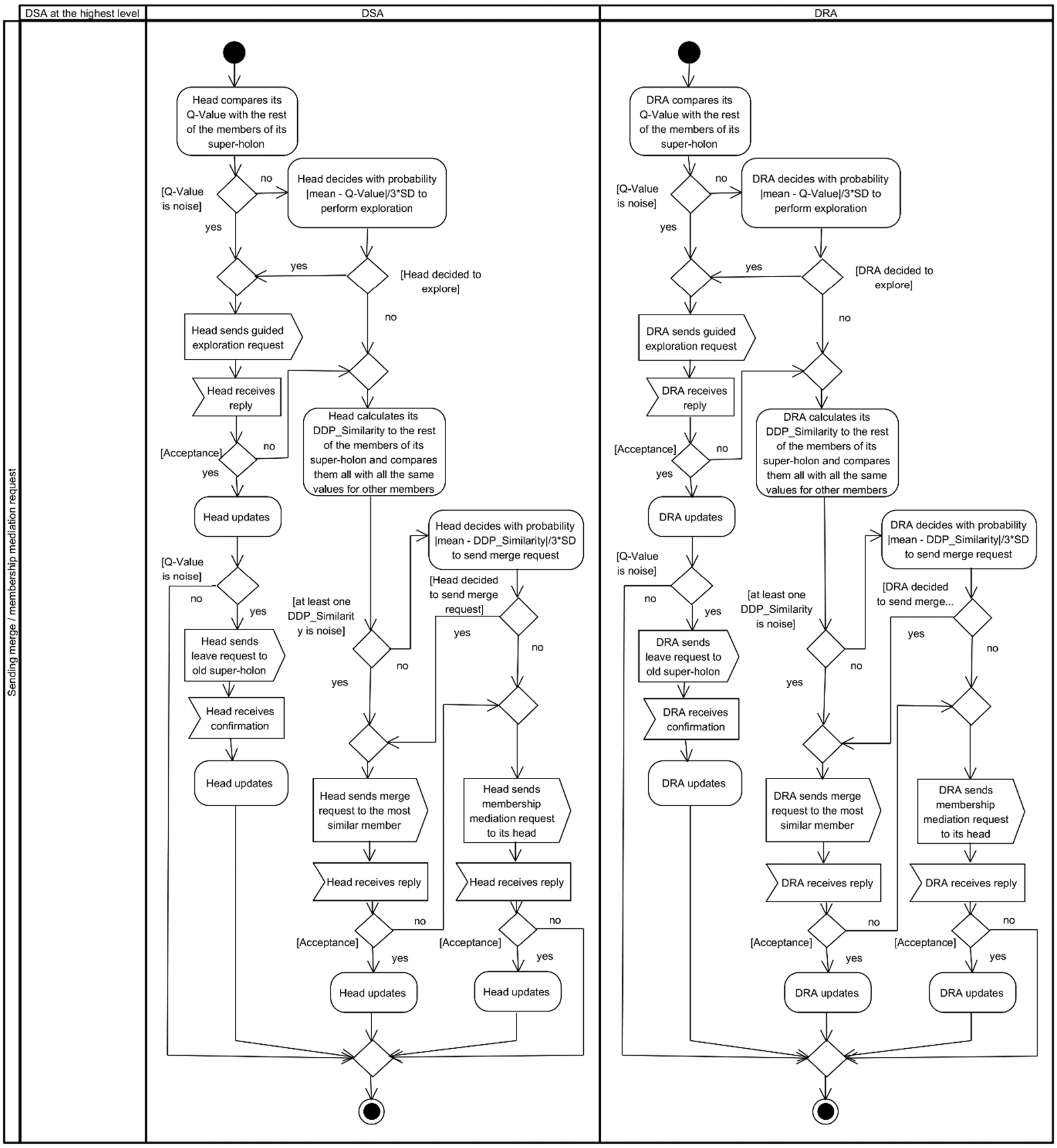
[26] Designed using Visual Paradigm Version 14.0

execution time [12, p. 166]. Two points about this definition bear emphasizing. One is that this process allows agents to change their organization at runtime, and the second one is that this change is conducted solely internally and does not need any control by any external agent. It should be noted that considering this concept it is necessary to distinguish between systems that include no internal and external explicit control from those that include an internal control. For example, in a termite society, the different arches are all located at the same distance from the queen due to a pheromone gradient [12]. The queen broadcasts this information and this action is an internal control. By consequence, the following definitions have been given:

- **Strong self-organizing systems** are those systems where there is no explicit central control either internal or external.
- **Weak self-organizing systems** are those systems where, from an internal point of view, there is re-organization maybe under an internal (central) control or planning. This kind of system can be illustrated by the example of the termites putting the bullet in a circle under the control of the queen. [12, p. 166]

With this regard, one of the characteristics of the self-organization process is the type of the internal control that can be: central, totally decentralized, or hybrid. In case of a holonic multi-agent system, we have a self-organizing behavior with hybrid control (bottom-up / top-down). On the one hand, the process is bottom-up as in general the DRAs decide on their moves and have decentralized control over the process. On the other hand, since for the new memberships the agents need a permission from the head of the new super-holon, the system includes local top-down controls as well. As a result, it can be concluded that the system exhibits a weak self-organizing behavior with hybrid control.

The clustering process in the HMDS has already been described in section 5.1.4 (for more information see section 6.1.1). The RL process will also be introduced in detail in section 6.1.3. This technique introduces a value called the Q-value for each of the connections in the holarchy, which indicates how promising the connection is according to the current information. This section will show how these values are used for the purpose of self-organization. The goal diagrams presented in Figure 26, Figure 27, and Figure 28 describe the self-organization process in the HMDS. The agents decide on their actions, i.e., to leave, stay in or join a super-holon, according to the similarity between their symptoms and some learned values, i.e., Q-values calculated based on Q-Learning (see chapter 6). In general, the decision-making idea is based on some statistical techniques for outlier detection, called the empirical rule [143], i.e., an agent may decide to join a holon if it's not an noise considering its members, and may decide to leave its super-holon in the opposite case. It is to be noted that the whole process is initiated and ended in Figure 26, however, the goal diagrams in Figure 27 and Figure 28 illustrate the activities that will be initiated in response to the requests sent in the first goal diagram. An agent can decide to merge with any member of its super-holon and move downwards in the holarchy, however, to move upwards head's membership mediation is needed.



DSA at the highest level
DSA
DRA
Sending merge / membership mediation request
Head compares its Q-Value with the rest of the members of its super-holon
[Q-Value is noise]
no
yes
Head decides with probability |mean - Q-Value|/3*SD to perform exploration
[Head decided to explore]
Head sends guided exploration request
Head receives reply
[Acceptance]
Head updates
Head calculates its DDP_Similarity to the rest of the members of its super-holon and compares them all with all the same values for other members
Head decides with probability |mean - DDP_Similarity|/3*SD to send merge request
[Head decided to send merge request]
[at least one DDP_Similarity is noise]
Head sends leave request to old super-holon
Head receives confirmation
Head sends merge request to the most similar member
Head sends membership mediation request to its head
Head receives reply
DRA compares its Q-Value with the rest of the members of its super-holon
DRA decides with probability |mean - Q-Value|/3*SD to perform exploration
[DRA decided to explore]
DRA sends guided exploration request
DRA receives reply
DRA updates
DRA calculates its DDP_Similarity to the rest of the members of its super-holon and compares them all with all the same values for other members
DRA decides with probability |mean - DDP_Similarity|/3*SD to send merge request
[DRA decided to send merge...
DRA sends leave request to old super-holon
DRA receives confirmation
DRA sends merge request to the most similar member
DRA sends membership mediation request to its head

Figure 26. The goal diagram for self-organization in the HMDS – part 1

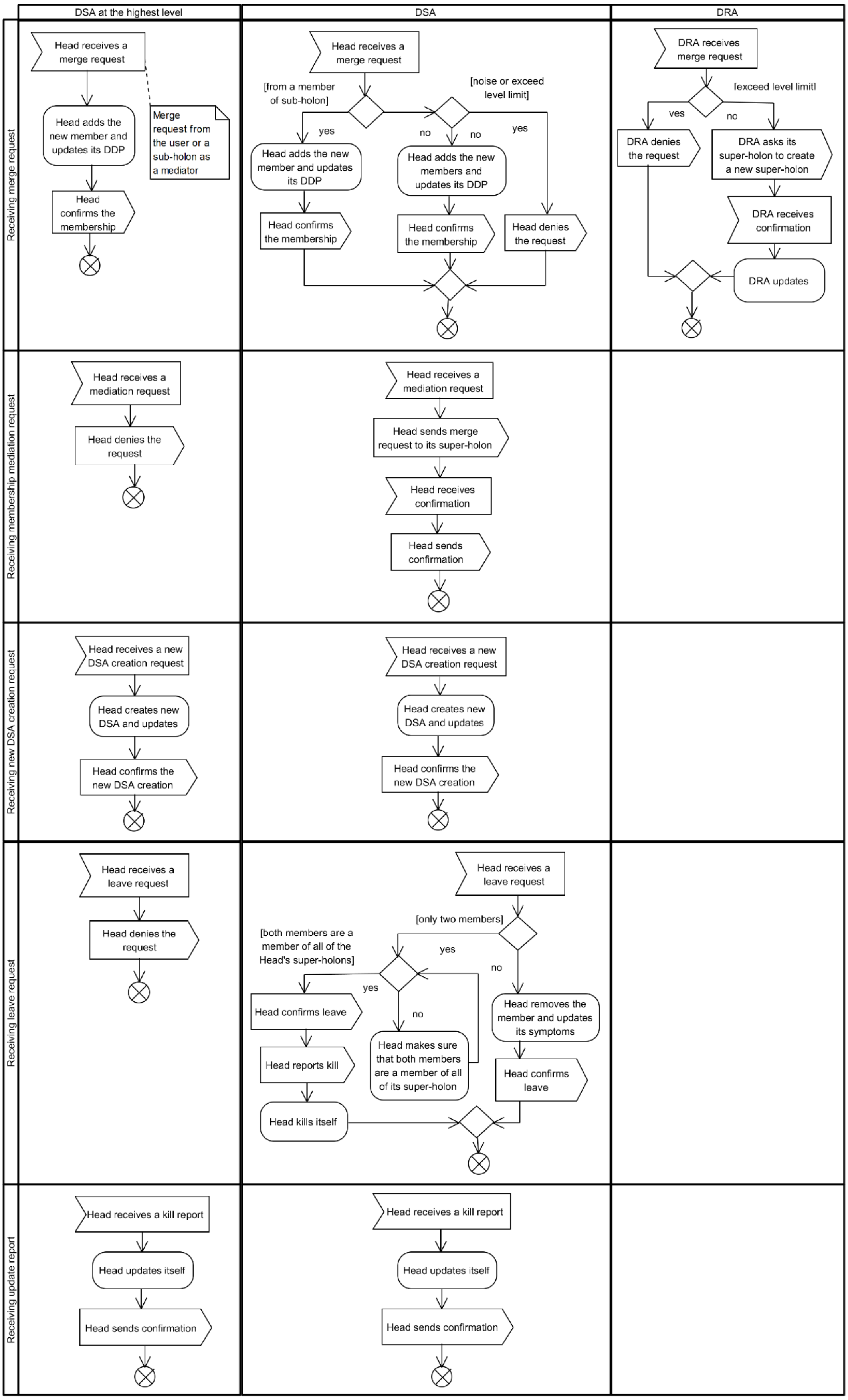

DSA at the highest level
DSA
DRA
Receiving merge request
Head receives a merge request
Head adds the new member and updates its DDP
Merge request from the user or a sub-holon as a mediator
Head confirms the membership
Head receives a merge request
[from a member of sub-holon]
[noise or exceed level limit]
yes
no
no
yes
Head adds the new member and updates its DDP
Head adds the new members and updates its DDP
Head confirms the membership
Head confirms the membership
Head denies the request
DRA receives merge request
[exceed level limit]
ves
no
DRA denies the request
DRA asks its super-holon to create a new super-holon
DRA receives confirmation
DRA updates
Receiving membership mediation request
Head receives a mediation request
Head denies the request
Head receives a mediation request
Head sends merge request to its super-holon
Head receives confirmation
Head sends confirmation
Receiving new DSA creation request
Head receives a new DSA creation request
Head creates new DSA and updates
Head confirms the new DSA creation
Head receives a new DSA creation request
Head creates new DSA and updates
Head confirms the new DSA creation
Receiving leave request
Head receives a leave request
Head denies the request
Head receives a leave request
[only two members]
[both members are a member of all of the Head's super-holons]
yes
no
yes
Head confirms leave
no
Head reports kill
Head makes sure that both members are a member of all of its super-holon
Head removes the member and updates its symptoms
Head confirms leave
Head kills itself
Receiving update report
Head receives a kill report
Head updates itself
Head sends confirmation
Head receives a kill report
Head updates itself
Head sends confirmation

Figure 27. The goal diagram for self-organization in the HMDS – part 2

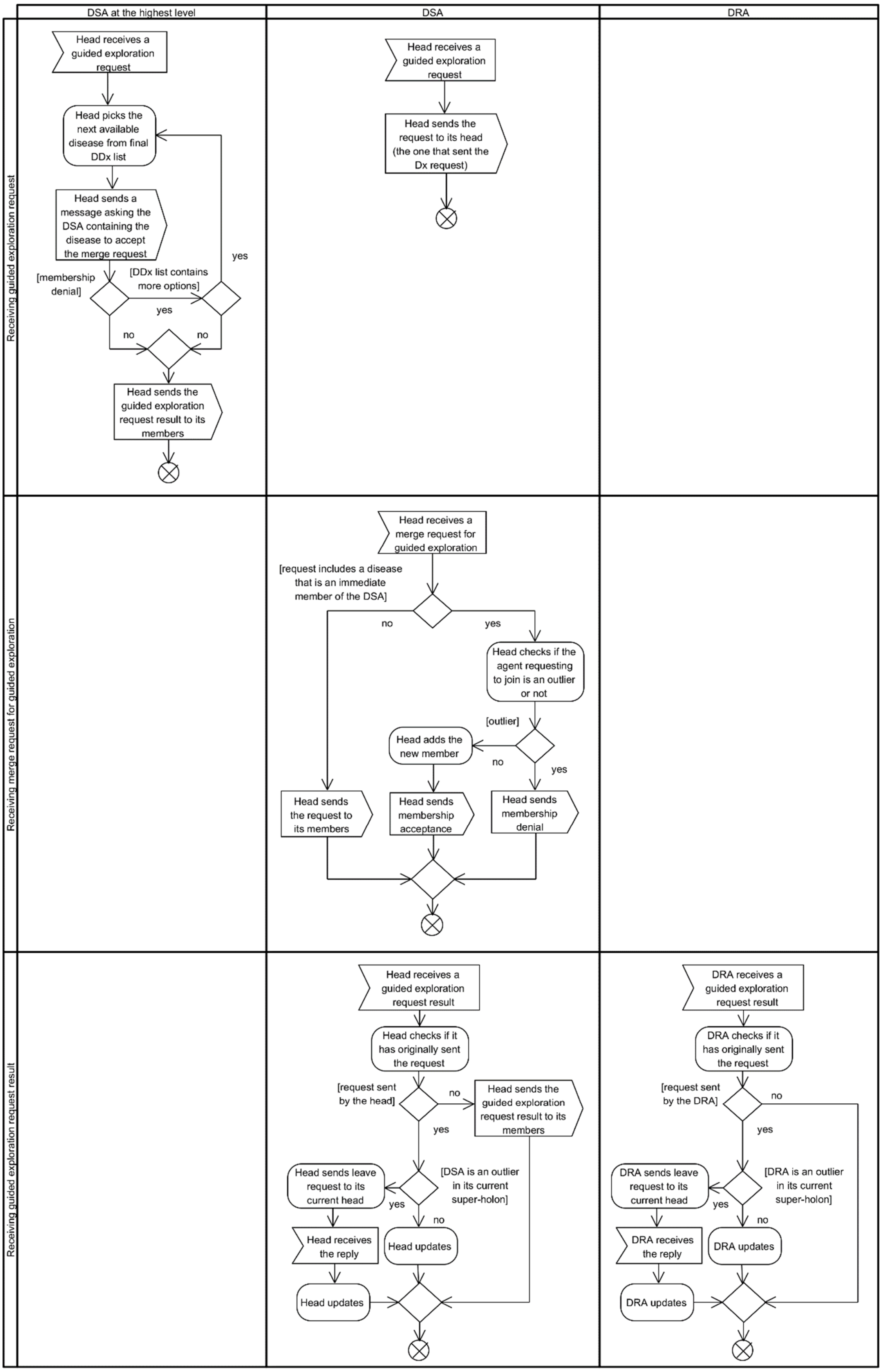
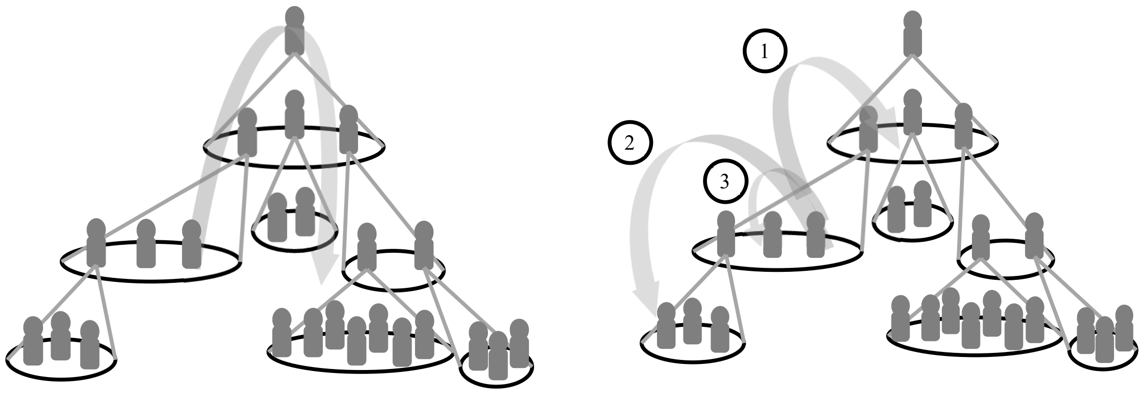

DSA at the highest level
DSA
DRA
Receiving guided exploration request
Receiving merge request for guided exploration
Receiving guided exploration request result
Head receives a guided exploration request
Head picks the next available disease from final DDx list
Head sends a message asking the DSA containing the disease to accept the merge request
[membership denial]
[DDx list contains more options]
yes
yes
no
no
Head sends the guided exploration request result to its members
Head receives a guided exploration request
Head sends the request to its head (the one that sent the Dx request)
Head receives a merge request for guided exploration
[request includes a disease that is an immediate member of the DSA]
no
yes
Head checks if the agent requesting to join is an outlier or not
[outlier]
Head adds the new member
no
yes
Head sends the request to its members
Head sends membership acceptance
Head sends membership denial
Head receives a guided exploration request result
Head checks if it has originally sent the request
[request sent by the head]
no
Head sends the guided exploration request result to its members
yes
Head sends leave request to its current head
[DSA is an outlier in its current super-holon]
yes
no
Head receives the reply
Head updates
Head updates
DRA receives a guided exploration request result
DRA checks if it has originally sent the request
[request sent by the DRA]
no
yes
DRA sends leave request to its current head
[DRA is an outlier in its current super-holon]
yes
no
DRA receives the reply
DRA updates
DRA updates

Figure 28. The goal diagram for self-organization in the HMDS – part 3

As mentioned and illustrated in Figure 26, an agent will decide to leave it's super-holon in case it is a noise considering the rest of the members of its super-holon. To this end, after each diagnosis process agents will update their Q-values (see chapter 6) and having these new values they may decide to explore new opportunities. Two different options may be available here: guided exploration or random exploration (see Figure 29 and compare to Figure 26).

In guided exploration the agent which has reacted to some diagnosis request will ask its heads to mediate a new membership for it on its behalf in order to join any super-holon with similar interests. The heads will then transfer the message to any other active super-holon and in case this agent is not an outlier this super-holon will accept its membership. This method ensures that the agent is joining a group with similar interests. The other option is the random exploration. In this approach the agent will move vertically, either to a higher super-holon of would join any other DSAs or DRAs in its super-holon. In this case the agent will again consider its similarity to the candidates.



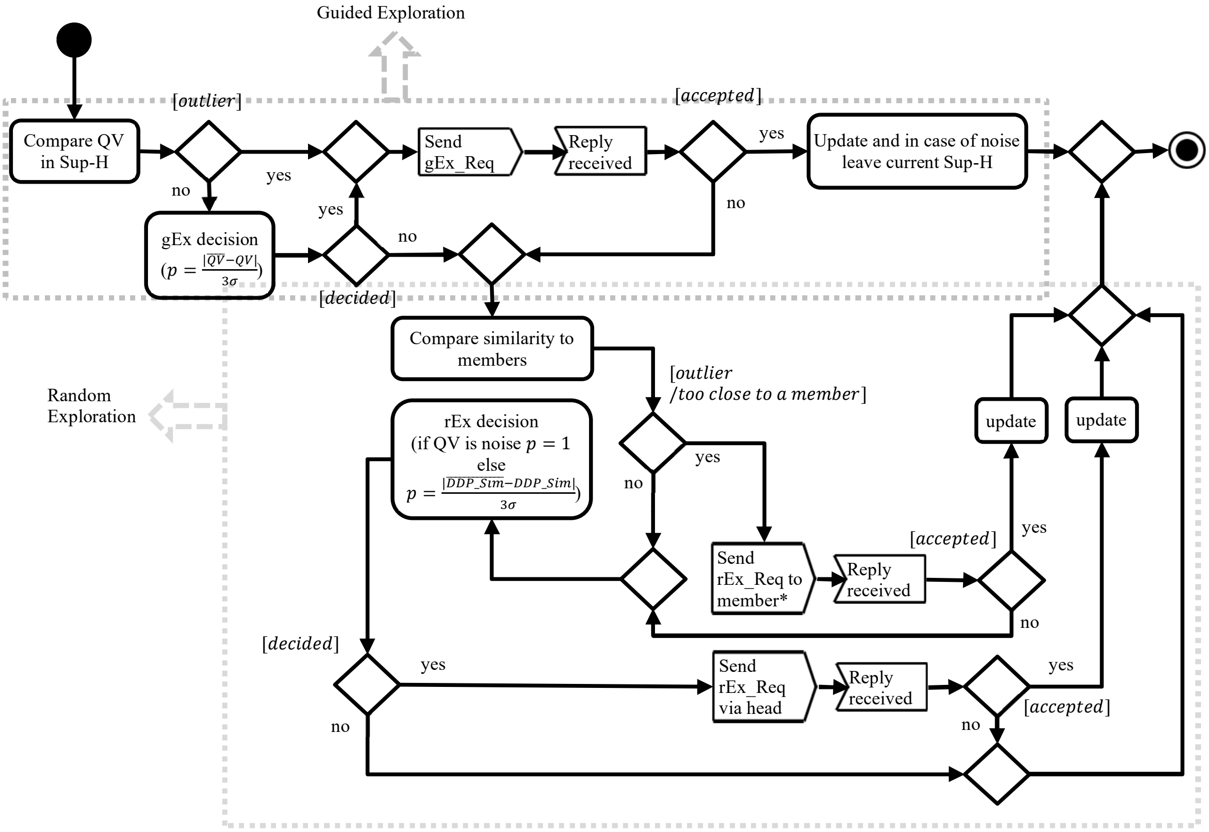
Guided exploration
Joining another DSA with similar interests.

Random exploration
1) Joining head's super-H
2) Joining another DSA in the same super-H
3) Joining another DRA in the same super-H (creating a new DSA)

Figure 29. The Guided and the random exploration in the HMDS

It should be noted that in guided exploration the agent will send a guided exploration request to its head and the head will pass this request to its head until the request reaches the highest DSA. This DSA will then send a message to its members, asking the super-holon containing the final diagnosis or the first disease in the DDx list which is different from the requesting agent to accept the merge request from this agent. The members of the highest DSA that where participating in the diagnosis process will take this message and check if the mentioned disease is their immediate member. If not, they will send the message to their active members, until the message reaches the right super-holon. The highest DSA may try all the diseases in the DDX list one by one in case of a membership denial from the corresponding super-holon, which as mentioned is based on statistical calculations for outlier detection.

The goal diagram in Figure 30 shows how an agent chooses between a guided or random exploration. The agent will compare its Q-value in its super-holon and in case of an outlier will start with the possibility of a guided exploration and if not successful will then try random exploration. It should be noted that in order to implement the exploration/exploitation possibility, even if the Q-value is not an outlier, the agent may still try to explore new opportunities to avoid biased actions. The probability of this action would however be higher if the Q-value is closer to the outlier threshold. As a result, the probability of exploration is $\frac{|\overline{QV}-QV|}{3\sigma}$, where QV indicates the Q-value and $\overline{QV}$ stands for the mean value of the Q-values.


Guided Exploration
[outlier]
[accepted]
Compare QV in Sup-H
yes
no
Send gEx_Req
Reply received
yes
Update and in case of noise leave current Sup-H
no
gEx decision ($p = \frac{|\overline{QV}-QV|}{3\sigma}$)
yes
no
[decided]
Compare similarity to members
[outlier /too close to a member]
Random Exploration
rEx decision (if QV is noise $p = 1$ else $p = \frac{|\overline{DDP\_Sim}-DDP\_Sim|}{3\sigma}$)
yes
no
update
update
Send rEx_Req to member*
Reply received
[accepted]
yes
no
[decided]
yes
Send rEx_Req via head
Reply received
yes
[accepted]
no
no

* DSA checks for outlier and number of levels / DRA solely checks the number of levels

Figure 30. Choosing the guided or the random exploration in the HMDS

Chapter 6
SYSTEM DESIGN: LEARNING IN THE HMDS

“Machine intelligence is the last invention that humanity will ever need to make.”

- Nick Bostrom (born in 1973)

Machine learning is one of the core fields of artificial intelligence and is concerned with the question of how to construct “computer programs that automatically improve with experience” [144, p. Editorial Reviews]. Depending on the nature of the learning data available to the learning system, machine learning methods are typically classified into three main categories [145, 146]: supervised, unsupervised and Reinforcement Learning (RL).

In supervised learning example inputs and their desired outputs are given and the goal is to learn a general rule that maps these inputs to their desired outputs. In unsupervised leaning, on the other hand, no labels are given to the learning algorithm, leaving it on its own to find the hidden structure of the data, e.g. to look for the similarities between the data instances (i.e., clustering [147]), or to discover the dependencies between the variables in large databases (i.e., association rule mining [148]). In reinforcement learning the desired input/output pairs are again not presented, however, the algorithm is able to estimate the optimal actions by interacting with a dynamic environment and based on the outcomes of the more recent actions, while ignoring experiences from the past, that were not reinforced recently.

As mentioned in section 5.1.1, the initial holarchy of the system can be created using clustering in different levels of the holarchy. Clustering is an unsupervised learning technique and therefore doesn't require any learning feedback. After having the initial holarchy, however, it is still essential to support the system in learning and updating based on the new observations. Medical knowledge demonstrates a steady upward growth, and diagnosis is also very much affected by the geographical regions. As a result, in order to adapt and improve its behavior, the system needs to: (1) Update its medical knowledge based on the new instances, (2) Reorganize its holarchy according to the new experiences and the feedback.

In the HMDS, holon identifiers are updated applying the exponential smoothing method, as a supervised learning method, and the self-organization of the holarchy is supported by Q-learning, as a reinforcement learning technique. Since both methods require feedback from the environment, it is clear that the quality of the feedback is of central importance. The rest of this section covers the above mentioned techniques and provides an experimental example, however, the discussion on the learning feedback and the reward engineering can be followed in section 6.2 and 6.3.

6.1 The Machine Learning Methods used in the HMDS

6.1.1 Clustering in the HMDS

Clustering is the most common unsupervised learning method [147, 149], and is the task of grouping a set of objects such that similar objects end up in the same group and dissimilar objects are diverted into different groups. Clustering, in the HMDS, establishes a starting point for super-holons in the holarchy. These super-holons are later updated according to the rest of the self-organization techniques. In order to cluster the diseases, exclusively according to the similarity between their signs and symptoms, the clustering algorithm should not require the specification of the number of the clusters. The Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [13] is one of the best algorithms matching this condition. However, despite its strengths, DBSCAN has a shortcoming in parameter detection, which is done in interaction with user, presenting some graphical representation of the data. For this reason, the presentation of a simple and effective method that could automatically detect the input parameter of DBSCAN was included in the design agenda for the implementation of the HMDS. The result is already published in [7]. The automated detection of the input parameter helps best to deal with complicated data such as diseases.

6.1.1.1 The Density-Based Spatial Clustering of Applications with Noise

Due to their diversity, clustering methods are classified into different categories in the scientific literature [150, 151, 152, 153]. However, despite the slight differences between these classifications, they all mention the DBSCAN algorithm as one of the eminent methods available. DBSCAN owes its popularity to the group of capabilities it offers [13]: (1) it does not require the specification of the number of clusters in the dataset beforehand, (2) it requires little domain knowledge to determine its input parameter, (3) it can find arbitrarily shaped clusters, (4) it has good efficiency on large datasets, (5) it has a notion of noise, and is robust to outliers, (6) it is designed in a way that it can be supported efficiently by spatial access methods such as R*-trees [154], and so on.

DBSCAN algorithm requires two input parameters, namely Eps and $MinPts$, which are considered to be the density parameters of the thinnest cluster acceptable, specifying the lowest density which is not considered to be noise. These parameters are hence respectively the radius and the minimum number of data objects of the least dense cluster possible. The algorithm supports the user in determining the appropriate values for these parameters offering a heuristic method, which imposes the user interaction based on some graphical representation of the data (represented in section 0). However, since DBSCAN is sensitive to its input parameters and the parameters have significant influences on the clustering result, an automated and more precise method for the determination of the input parameters is needed [7].

Some notable algorithms targeting this problem are: (1) GRPDBSCAN, which combines the grid partition technique and DBSCAN algorithm [155], (2) DBSCAN-GM, that combines

Gaussian-Means and DBSCAN algorithms [156], and (3) BDE-DBSCAN, which combines Differential Evolution and DBSCAN algorithms [157].

Opposed to these methods, which all intend to solve the problem using some other techniques, the method presented in [7] remains with the original idea of the DBSCAN algorithm and just tries to omit the user interaction needed, allowing the algorithm to detect the appropriate value itself. This is done using some basic statistical techniques for outlier detection. Two different approaches are mentioned in this paper, which apply the concept of Standard Deviation (SD) to the problem of outlier detection, namely the empirical rule for normal distributions and the Chebyshev's inequality for non-normal distributions [143, 158]. However, the paper mainly focuses on the application of the empirical rule to outlier detection in normal distributed data and addresses the Chebyshev's inequality only as a possible solution for non-normal distributions.

According to [13, p. 227], the key idea of DBSCAN algorithm is that "for each point of the cluster the neighborhood of a given radius has to contain at least a minimum number of points, i.e., the density in the neighborhood has to exceed some threshold". The following definitions from [13, pp. 227-228] support the realization of this idea:

Definition 1: ($Eps - neighborhood$ of a point) The $Eps - neighborhood$ of a point p, denoted by $N_{Eps}(p)$, is defined by $N_{Eps}(p) = \{q \in D \mid dist(p, q) \leq Eps \}$.

Definition 2: (directly density-reachable) A point p is directly density-reachable from a point q, w.r.t. Eps and $MinPts$, if

1. $p \in N_{Eps}(q)$ and
2. $|N_{Eps}(q) \geq MinPts|$

The second condition is called core point condition (There are two kinds of points in a cluster, points inside of the cluster, called core points, and points on the border of the cluster, called border points).

Definition 3: (density-reachable) A point p is density-reachable from a point q, w.r.t. Eps and $MinPts$, if there is a chain of points $p_1, \dots, p_n, p_1 = q, p_n = p$ such that p_{i+1} is directly density-reachable from p_i.

Definition 4: (density-connected) A point p is density-connected to a point q, w.r.t. Eps and $MinPts$, if there is a point o such that both, p and q are density-reachable from o, w.r.t. Eps and $MinPts$.

Definition 5: (cluster) Let D be a database of points. A cluster C, w.r.t. Eps and $MinPts$, is a non-empty subset of D satisfying the following conditions:

1. p, q: $if\ p \in C$ and q is density-reachable from p, w.r.t. Eps and $MinPts$, then $q \in C$. (Maximality)
2. $p, q \in C$: p is density-connected to q, w.r.t. EPS and $MinPts$. (Connectivity)

Definition 6: (noise) Let $C_1, \dots, C_k$ be the clusters of the database D, w.r.t. parameters Eps_i and $MinPts_i$, $i = 1, \dots, k$. Then the noise is defined as the set of points in the database D not belonging to any cluster C_i, i.e., $noise = \{p \in D \mid \forall i: p \notin C_i)$.

As stated in [13, p. 228], the following lemmata are important for validating the correctness of the algorithm. Intuitively, they state that having the parameters Eps and $MinPts$, a cluster can be discovered in a two-step approach. First, choose an arbitrary point from the database satisfying the core point condition as a seed. Second, retrieve all points that are density-reachable from the seed, obtaining the cluster containing the seed.

Lemma 1: Let p be a point in D and $|N_{Eps}(p)| \geq MinPts$. Then the set $O = \{o \mid o \in D\ and\ o\ is\ density - reachable\ from\ p, w.r.t. Eps\ and\ MinPts\}$ is a cluster, w.r.t. Eps and $MinPts$.

Lemma 2: Let C be a cluster, w.r.t. Eps and $MinPts$, and let p be any point in C with $|N_{Eps}(p)| \geq MinPts$. Then C equals to the set $O = \{o \mid o\ is\ density - reachable\ from\ p,$ $w.r.t. Eps\ and\ MinPts\}$.

6.1.1.1.1 The Algorithm

The DBSCAN algorithm [13] can be described as follows:

Table 11. Algorithm 1: Pseudo-code of the DBSCAN

DBSCAN Algorithm (Input: $D, Eps, MinPts$)

```
While ($D$ has an unclassified[27] point)
    Select an arbitrary unclassified point $p$.
    If $p$ does not satisfy the core point condition, mark it as a noise.
    Else retrieve all the density-reachable points from $N_{Eps}(p)$ forming a cluster
    containing $N_{Eps}(p)$ and mark all the member of this cluster as classified.
End While
```

6.1.1.1.2 Determining the Parameters Eps and MinPts

DBSCAN [13] offers a simple but effective heuristic method to determine the parameters Eps and $MinPts$ of the thinnest cluster in the dataset. For a given k function $k - dist$ is defined from the database D to the real numbers, mapping each point to the distance from its $k - th$ nearest neighbor. Based on this function, the algorithm then defines the sorted $k - dist$ graph, which displays the points of the dataset sorted in descending order of their $k - dist$ values. It is clear that the first point in the first valley of the sorted $MinPts - dist$ graph can be the threshold point with the maximal $MinPts - dist$ value in the thinnest cluster. All points with a larger $MinPts - dist$ values are considered to be noise, and all the other points are assigned to some clusters.

DBSCAN [13] states that according to experiments, the $k - dist$ graphs for $k > 4$ do not significantly differ from the $4 - dist$ graph and, furthermore, they need considerably more computation. Therefore, it eliminates the parameter $MinPts$ by setting it to 4 for all datasets (for 2-dimensional data). The parameter determination method also explains, that since in general, it is very difficult to detect the first valley of the sorted $MinPts - dist$ graph automatically, but it is relatively simple for the user to see this valley in a graphical representation, it is suggested to follow an interactive approach for determining the threshold point.

[27] Note that the term unclassified here indicates that it is not determined yet if the point is a noise or not.

6.1.1.2 Automated Determination of the Parameter Eps

This subsection describes the method presented in [7], which as mentioned is introduced to automatically detect the input parameter of DBSCAN. The term noise in DBSCAN is equivalent to an outlier in statistics, which is "an observation that is far removed from the rest of the observations" [159, p. 89]. One of the basic statistical techniques for outlier detection is the empirical rule. The empirical rule is an important rule of thumb, that is used to state the approximate percentage of values that lie within a given number of standard deviations from the $mean$ of a set of data if the data are normally distributed. The empirical rule, also called the 68-95-99.7 rule or the three-sigma rule of thumb states that 68.27%, 95.45%, and 99.73% of the values in a normal distribution lie within one, two, and three standard deviations of the mean [143] (see Figure 31 [160]). One of the practical uses of the empirical rule is as a definition of outliers as the data that fall more than three standard deviations from the norm in normal distributions [161].

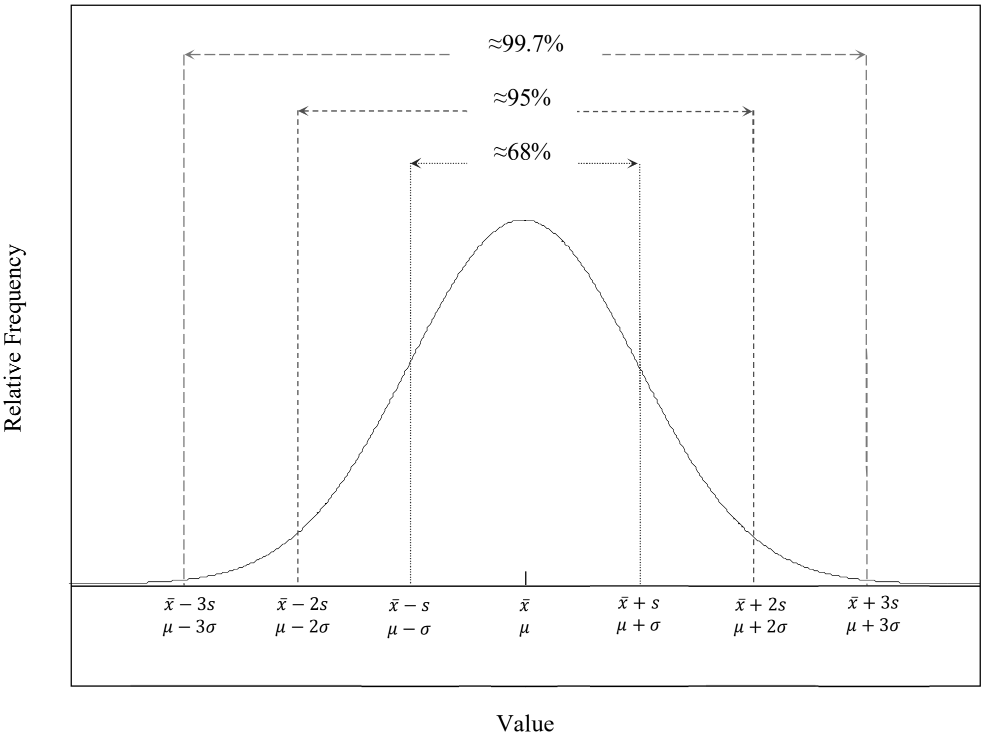


Figure 31. The Empirical Rule

If there are many points that fall more than three standard deviations from the norm, then the distribution is most likely non-normal. In this case, Chebyshev's inequality, which applies to non-normal distributions, is applicable. Chebyshev's inequality states that in any probability distribution, at least $1-\frac{1}{k^2}$ of the values are within k standard deviations of the $mean$ [143] (e.g. in non-normal distributions at least 99% of the values lie within 10 standard deviations of the $mean$). As a result, using the Chebyshev's inequality, the outlier can also be defined as the data that fall outside an appropriate number of standard deviations from the mean [162][28].

Determining the parameter Eps, the algorithm is aiming a radius that covers the majority of the $MinPts-dist$ values and stands well as a threshold for the specification of the noise values. As mentioned above, the term noise in DBSCAN algorithm is equivalent to an outlier in statistics, which is "an observation that is far removed from the rest of the observations" [159, p. 89]. Thus, the idea here is to use statistical rules in order to find the threshold value between the accepted $MinPts-dist$ values and the values considered for the noise points.

As mentioned above, one of the practical uses of the empirical rule is as a definition of outliers as the data that fall more than three standard deviations from the norm in normal distributions [161]. Thus, considering the $MinPts-dist$ values, the value of parameter Eps can be set to their $mean$ plus three standard deviations. This would cover even more than 99.73% of the calculated $MinPts-dist$ values, since the $MinPts-dist$ values smaller than $mean-3\times SD$ are also covered here.

Border points and even in general, points closer to the border of the clusters usually have greater $MinPts-dist$ values, which lead to larger Eps values and thus might cause two close clusters to be detected as one cluster (Since the parameter $MinPts$ is generally set to 4, this problem may be caused mostly by the border points). These relatively greater $MinPts-dist$ values, however, do not have any positive effect on the process of cluster detection, as the $MinPts-dist$ values of the core points are actually the ones forming the right clusters and at the same time covering the border points. Figure 32 shows a case in which the $4-dist$ value of border point p is much larger than the $4-dist$ value of the core point q, which can actually cover p in its $4-dist-neighborhood$.

[28] This work focuses solely on the empirical rule and the normal distributions. However, the possibility of using the Chebyshev's inequality is given here, in order to show that the general idea of using outlier detection techniques for the reason of parameter determination in DBSCAN is not limited to the distribution of the data.

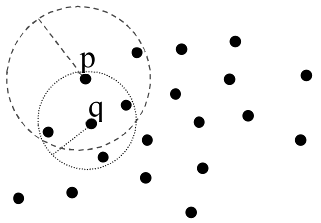


Figure 32. The $4 - dist$ values for example core (q) and border point (p)

In order to eliminate the negative effect of the $MinPts - dist$ values of the border points, the algorithm presented here considers any point with minimum $MinPts - dist$ value which covers the border point in its $MinPts - dist - neighborhood$ and replaces the $MinPts - dist$ value of this border point with the $MinPts - dist$ value of this core point. Thus for a given k, function $k - dist'$ is defined from the database D to the real numbers, mapping each point to the $k - dist$ value of any core point, covering this point in its $k - dist - neighborhood$, with minimum $k - dist$ value.

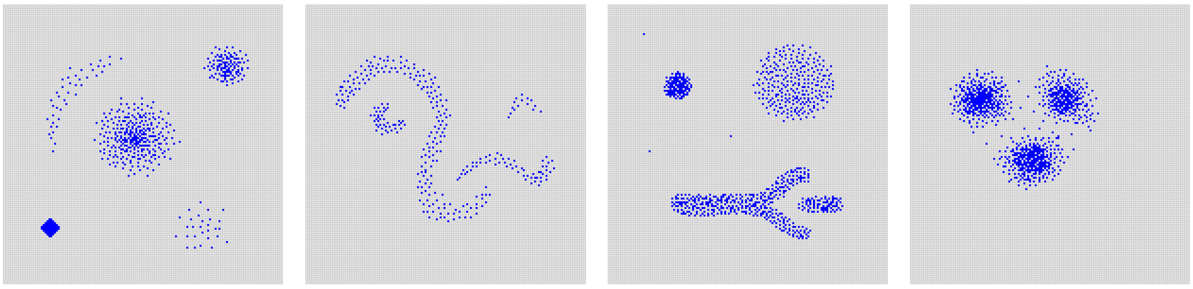
In fact, following this technique, points are considered in ascending order of their $MinPts - dist$ values, then taking each point p, if the $MinPts - dist'$ value for any point in its $MinPts - dist - neighborhood$ is not set so far, this value will be set to the $MinPts - dist$ value of point p. Using this technique for each point, the $MinPts - dist$ value of the smallest cluster, the point can join, would be considered. At the end the $mean$ and the standard deviation of these $MinPts - dist'$ values which are saved for all points are calculated and the Eps' value is set to $mean + 3 \times SD$. The following pseudo-code indicates this method (Table 12). For more information on the performance and the time complexity of the algorithm please refer to [7].

Table 12. Algorithm 2: Pseudo-code of the $EpsFinder$

```
**$EpsFinder$ (Input: $D$)**

1. For each point $p$ calculate the $MinPts - dist$ value.
2. Sort the points in ascending order of theirs $MinPts - dist$ values.
3. Following the ascending order, take each point $p$ and if the $MinPts - dist'$
   value for the point itself or any point in its $MinPts - dist - neighborhood$
   is not set so far, set this value to the $MinPts - dist$ value of the point $p$.
4. Calculate the $mean$ of the $MinPts - dist'$ values: $mean$
5. Calculate the standard deviation of the $MinPts - dist'$ values: $SD$
6. Set the $Eps'$ value to $mean + 3 \times SD$.
```

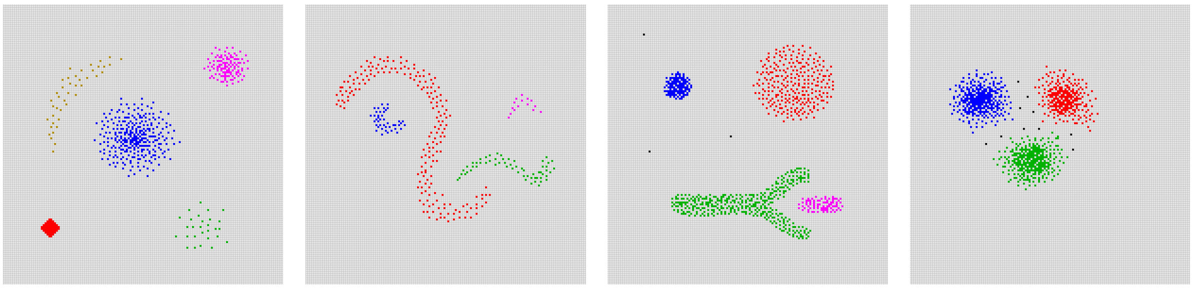
In order to demonstrate the improvement of the algorithm both methods were applied to some datasets in [7]. All the experiments were performed on Intel(R) Celeron(R) CPU 1.90GHz with 2 GB RAM on the Microsoft Windows 8 platform. The algorithm and the datasets were implemented in Java on Eclipse IDE, MARS.1.

Sample datasets are depicted in Figure 33. The noise percentage for datasets 1 and 2 is 0%, however, datasets 3 and 4 do have noise values. Figure 34 shows the results of the clustering. For each dataset clusters are marked with different colors and noise points are shown with black color.



Dataset 1 Dataset 2 Dataset 3 Dataset 4

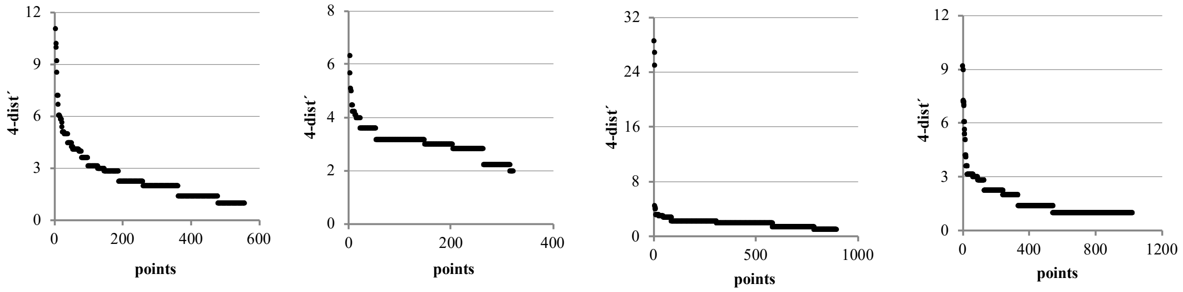
Figure 33. Sample datasets



Dataset 1 Dataset 2 Dataset 3 Dataset 4

Figure 34. Detected clusters

The sorted $4 - dist'$ graphs of the sample datasets are displayed in Figure 35. Here, Eps indicates the value determined by the user, according to the visual representation of the data, and Eps' represents the value calculated automatically by the algorithm presented in Section 4 $(EpsFinder)$.



Dataset 1 $Eps = 6.08$ $Eps' = 6.61$

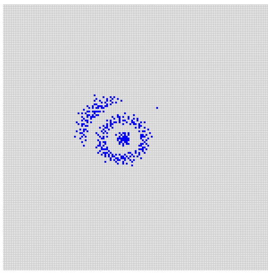
Dataset 2 $Eps = 4.47$ $Eps' = 4.71$

Dataset 3 $Eps = 4.47$ $Eps' = 6.63$

Dataset 4 $Eps = 4.24$ $Eps' = 4.45$

Figure 35. Sorted $4 - dist'$ graphs for sample datasets[29]

In order to illustrate the problem that may occur with the $k - dist$ value of the border points dataset 5 is presented in Figure 36. This dataset is defined in a way that nested and very close clusters are available in it.



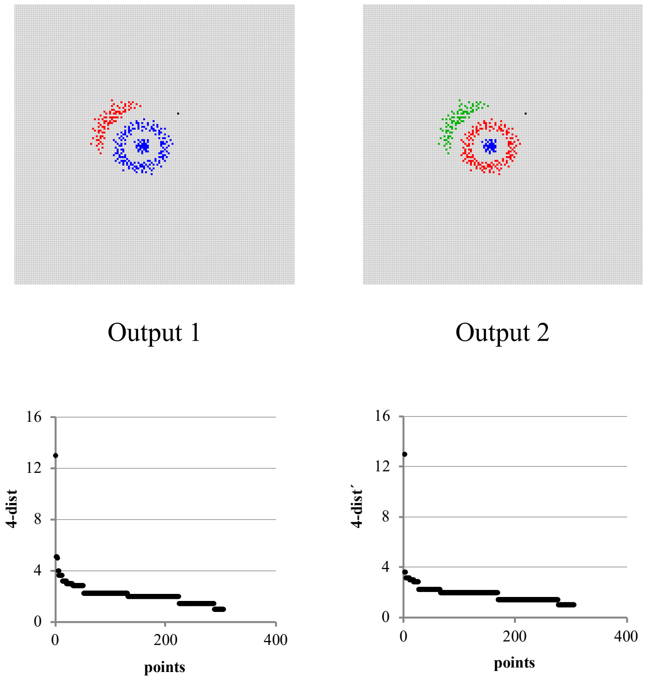
Dataset 5

Figure 36. Dataset 5

Figure 37 illustrates the output results applying the two different methods to Dataset 5. Output 1 shows the clustering result according to the $4 - dist$ values, which were considered by the old method. In this experiment the algorithm has failed to distinguish the nested clusters. Output 2, on the other hand, shows the clustering result according to the $4 - dist'$ values. Here, the threshold values calculated are smaller and hence the algorithm has detected the nested clusters

[29] Note that the larger difference between Eps and Eps' for Dataset 3 is caused by the larger difference between the $4 - dist'$ values of those data instances considered as noise and the rest of the data instances. This difference has no effect on the clustering result, since Eps and Eps' are actually threshold values and since there are no data instances with $4 - dist'$ values between Eps and Eps', the clustering result would remain the same.

easily. Graph 1 and Graph 2 in Figure 7 show here the $4-dist$ and $4-dist'$ values calculated using each of the techniques, together with the corresponding Eps and Eps' values.



Output 1

Output 2

Graph 1

$Eps = 5.10$

$Eps' = 4.82$

Graph 2

$Eps = 3.61$

$Eps' = 4.34$

Figure 37. Different clustering results for dataset 5

It should be pointed out that even though the experiments presented here were all for 2-dimensional datasets, the idea can be applied to high-dimensional datasets as well. This is clearly possible, since the calculation of the distance between the points and the application of standard deviation remains the same for high-dimensional datasets. The only point that must be considered is that, the DBSCAN has suggested 4 as the $MinPts$ value just for 2-dimensional datasets. However, as mentioned before, Eps and $MinPts$ are the density parameters of the thinnest cluster; therefore it is always possible to determine the Eps by keeping the $MinPts$ parameter small enough. The diversity of the density may always be described with different radii containing a predefined number of points ($MinPts$).

6.1.2 Exponential Smoothing in the HMDS

Exponential smoothing is a very popular scheme for producing smoothed time series [163]. Using this technique, the past observations are assigned exponentially decreasing weights and recent ones are given relatively higher weights:

$$s_t = \alpha . x_t + (1 - \alpha) . s_{t-1} \tag{3}$$

where α is the smoothing factor, and $0 < \alpha < 1$. As it is clear, the smoothed statistic s_t is a weighted average of the current observation x_t and the previous smoothed statistic s_{t-1}.

In the HMDS, this method is used in order to update the holon identifiers. Every time a new case of a disease is diagnosed and then confirmed by the physician using the feedback loop, the system considers the signs and symptoms of the patient to update its DDP. This method is clearly a supervised learning technique. The smoothing factor indicates the learning rate and shows how the algorithm considers the previous observations in calculating the current value. Larger values of α actually reduce the level of smoothing, and in the limiting case with $\alpha = 1$ the output series is just the current observation. Using this learning technique, the system will be capable of diagnosing the disease in patients that do not exhibit all the signs and symptoms that are known for a disease. In the HMDS the smoothing factor can be equal to the learning rate in the RL method (see section 6.1.3).

6.1.3 Reinforcement Learning (RL) in the HMDS

In the HMDS, the holarchy is in fact keeping track of the best decisions made by the system performing the self-organization. Due to the holonic memberships this process can be regarded as a sequential decision-making problem. This process should be supported by an appropriate machine learning technique. Considering the nature of the problem, the absence of desired input/output pairs, and the accessibility of a dynamic environment, reinforcement learning is the best match for our problem. There are many different reinforcement learning techniques in the literature and in the next stage we need to choose the best fitting one for our system. However, in order to apply reinforcement learning to our problem, we first need to model it in a way that the algorithms can be applied. For this reason, the framework of Markov Decision Processes (MDPs) [164], [165] is used [166].

Decision-making tasks that involve delayed consequences can be formulated as sequential decision-making problems, for which decision-making strategies must be found that take into account the expectations of both short-term and long-term consequences of the decisions. Dealing with such problems, at each time step, the Decision Maker (DM), observes the system's current state and selects an action, and then makes a transition to a successor state, which is determined by current state, the chosen action and a random disturbance that aims the exploration-exploitation trade-off. For each action, the DM will then receive a certain amount of payoff that depends on the action and the current state. The goal is to find a rule, i.e., an action selection policy, for the DM that maximizes the total amount of the accumulated payoff.

Markov Decision Processes (MDPs) [165] provide a framework for modeling such sequential decision-making problems. The optimization problem is then to be solved applying the Dynamic Programming (DP) approach and Reinforcement Learning (RL). Some algorithms for sequential decision-making have been studied in [167], which aim to produce policies that maximize a measure of the long-term reward to an agent following it in a specific environment.

According to [167], these policies can be produced under two different problem scenarios that differ in the information available for constructing the policy: Planning and Reinforcement Learning. In planning, a complete model of the environment is known in advance and the produced policy is typically stationary. Reinforcement learning, however, can be used when a model of the environment is unknown or difficult to work with directly. Considering the self-organization problem in the HMDS, this case falls into the latter problem scenario category.

6.1.3.1 Markov Decision Processes (MDPs)

"Markov Decision Processes (MDPs), also referred to as stochastic dynamic programs or stochastic control problems, are models for sequential decision-making when outcomes are uncertain" [165, p. XV]. They are defined as controlled stochastic processes satisfying the Markov property and assigning reward value to state transitions [165]. As stated in [166, p. 19], "A stochastic process has the Markov property if the conditional distribution of the next state of the process depends only on the current state of the process".

"The Markov decision process model consists of decision epochs, states, actions, rewards, and transition probabilities. Choosing an action in a state generates a reward and determines the state at the next decision epoch through a transition probability function" [165, p. XV]. According to [168], a Markov decision process is a model of an agent operating in an uncertain but observable world. The definition of Markov decision process presented here, and the notations are according to the ones used in [168] and [169].

Formally, an MDP is a 5-tuple (S, A, T, R, γ):

- S is a set of agent-environment states.
- A is a set of actions the agent can take.
- $T(s_i, a, s_j): S \times A \times S \rightarrow [0,1]$ is a state transition function which gives the probability that taking action a in state s_i results in state s_j.
- $R(s_i, a, s_j): S \times A \times S \rightarrow \mathbb{R}$ is a reward function that quantifies the reward the agent gets for taking action a in state s_i resulting in state s_j.
- $\gamma \in [0,1)$ is a discount factor that trades off between immediate reward and potential future reward.

In this representation of the problem, for each agent, the world is its current state at any given time. At each time step, the agent observes its world and itself, and based on the results chooses and executes an action from its set of actions. The agent then receives a reward for its action choice in that state, which is calculated using a reward function. Another function, called the state transition function, then probabilistically transitions the agent's world to a new state. The objective of an MDP agent is to take the actions that maximize its future expected reward [168].

The solution to an MDP, also called a policy, is a function $\pi(s_i): S \rightarrow A$, that maps states to actions. According to [168], the optimal policy over an infinite horizon is one inducing the value function:

$$V^*(s_i) = \max_{a \in A} \left[R(s_i, a, s_j) + \gamma \sum_{s_j \in S} T(s_i, a, s_j) V^*(s_j) \right] \tag{4}$$

As explained in [168], this famous equation, known as the Bellmann Equation, quantifies the value of being in each state based on immediate reward and discounted expected future reward. The future reward is calculated based on the action the agent can take from its next state and the reward it receives in future states it may move to, discounted more as time passes. This recursion implements a dynamic programming method, which breaks the optimization problem into subproblems that are solved sequentially in a recurrent fashion. In other words, according to [170], the core problem of MDPs is to find a policy for the decision maker that can be best described as an action selection rule to be followed by the agent, given the state it is in: a function π that specifies the action $\pi(s)$, which the decision maker will choose when in state s. As a result, a policy defines the agent's action selection with respect to changes in the environment and the goal is to choose a policy π that will maximize expected discounted cumulative function of the random rewards over a potentially infinite horizon.

According to [171], reinforcement learning can be formalized in terms of Markov decision processes, in which the agent is initially only aware of the set of possible states and the set of possible actions. Thus, the state transition function, $T(s_i, a, s_j)$, and the reward function, $R(s_i, a, s_j)$, are initially unknown. An agent can act in a world and after each step observe the state of the world it has entered and the reward it has obtained. In this formalization, the agent acts to achieve the optimal discounted reward with a discount factor γ. According to [172], the cumulative value $V^\pi(s_t)$, achieved by following an arbitrary policy π from an arbitrary initial state s_t, can be defined as follows:

$$V^\pi(s_t) \equiv r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \cdots \equiv \textstyle\sum_{i=0}^{\infty} \gamma^i \, r_{t+i} \tag{5}$$

where the sequence of rewards r_{t+i} is generated by beginning at state s_t and by repeatedly using the policy π to select actions. The rewards are discounted exponentially by a factor of γ^i. Note if $\gamma = 0$, only the immediate reward is considered, however, if γ is set closer to 1, future rewards are given greater emphasis relative to the immediate reward. The quantity $V^\pi(s)$ defined in equation 5 is often called the discounted cumulative reward achieved by policy π from initial state s. The agent's task is to learn a policy π that maximizes $V^\pi(s)$ for all states s. Such a policy is called an optimal policy and is denoted by π^*.

$$\pi^* \equiv \underset{\pi}{\operatorname{argmax}} V^\pi(s), \qquad \forall s \tag{6}$$

To simplify the notation, the value function $V^{\pi^*}(s)$ of such an optimal policy is referred to as $V^*(s)$ (cf. equation 4).

Reinforcement learning algorithms aim to learn an optimal policy for an MDP, using solely the reward signals received through the iterating interactions with the environment. These algorithms can be divided into two classes: Value iteration and policy iteration algorithms [173].

Value iteration algorithms learn the optimal value function and attempt to derive an optimal policy from this learnt value function. Alternatively, policy iteration algorithms directly build an optimal policy by interleaving two phases: policy evaluation, during which the value of the current policy is estimated, and policy improvement, during which based on the results of the policy evaluation phase the policy is then locally improved. This process continues until no further improvement is possible and an optimal policy is reached [173].

6.1.3.2 The Exploitation-Exploration Trade-Off

Since RL algorithms do not assume a given model for the MDP, one of the key issues in applying them is that agents need to explore the environment in order to observe the effects of their actions. This implies that agents cannot simply take the actions that are associated with the current highest estimated rewards, but also need to try new actions in an attempt to discover better strategies. This problem is known as exploitation-exploration trade-off in RL [173].

Two basic approaches exist to address this problem. On-policy methods attempt to evaluate or improve the policy that is used to make decisions. The distinguishing feature of on-policy methods is that they estimate the value of a policy while using it for control. “In off-policy method these two functions are separated. The policy used to generate behavior, called the behavior policy, may in fact be unrelated to the policy that is evaluated and improved, called the estimation policy. An advantage of this separation is that the estimation policy may be deterministic (e.g. greedy), while the behavior policy can continue to sample all possible actions” [169]. In other words, the algorithm learns the values associated with taking the exploitation policy while following an exploration/exploitation policy.

6.1.3.3 Action-Selection Strategies

As mentioned above, a common challenge in reinforcement learning is to find a trade-off between exploration (discovering new features about the world by selecting sub-optimal actions) and exploitation (using already known actions that lead to best known results) [174].

The agents may follow different action-selection strategies in order to choose their actions, each of them dealing differently with this trade-off:

(1) **Greedy action-selection:** The simplest action-selection strategy is greedy selection. In this strategy the agent always selects the action with the highest state-action value. This method is pure exploitation. The rest of the methods, in fact aim to achieve a balance between exploration and exploitation [174].

(2) ***ε*-Greedy action-selection:** ε-Greedy is a variation on normal greedy selection. In both cases, the agent identifies the best move according to the state-action values. However, in ε-greedy strategy with a small probability of ε, rather than taking the best action, the agent will uniformly select an action from the remaining actions [174].

(3) **Softmax (Boltzmann) action-selection:** Although ε-greedy action selection is an effective and popular means of balancing exploration and exploitation in reinforcement learning, one drawback is that when it explores it chooses equally among all actions. This means that it is as likely to choose the worst-appearing action as it is to choose the next-to-best action. The obvious solution is to vary the action probabilities as a graded function of estimated value. In Softmax approach the greedy action is still given the highest selection probability, but all the others are ranked and weighted according to their estimated values. The most common softmax method uses a Boltzmann (Gibbs) distribution and chooses action a on the t-th time step with probability:

$$p = \frac{e^{Q_t(a)/_\tau}}{\sum_{i=1}^{n} e^{Q_t(i)/_\tau}} \tag{7}$$

where τ is a positive parameter called the temperature. High temperatures cause the actions to be all (nearly) equiprobable. Low temperatures cause a greater difference in selection probability for actions that differ in their value estimates. In the limit as τ → 0, softmax action selection becomes the same as greedy action selection [169].

6.1.3.4 RL Methods for MDPs (single-agent, MARL, swarm RL)

Table 13 lists the notable RL techniques and demonstrates, whether they are an on-policy or an off-policy approach.

Table 13. The notable RL techniques

RL Technique	Policy	Year	Reference
Actor-Critic (AC)	on-policy	1983	[175]
Temporal Difference (TD)	on-policy	1988	[176]
Q-Learning	off-policy	1989	[177]
R-Learning	off-policy	1993	[178]
State-Action-Reward-State-Action (SARSA)	on-policy	1994	[179]
Actor Critic Learning Automaton (ACLA)	on-policy	2007	[180]
QV(λ)-Learning	on-policy	2007	[180]
Monte Carlo (MC) for RL	on-policy	2008	[181]

As most researches in the fields of MARL (Multi-Agent RL) [182] and swarm RL have focused on Q-learning, this study concentrates on the Q-learning method and demonstrates the ideas using this technique. It should, however, be noted that the general idea can be applied to any RL technique based on the problem at hand. The key reason behind the wide use of Q-learning may be due to its off-policy approach (R-learning is a variant of Q-learning method for non-discounted, non-episodic problems). The MASs consist of several autonomous agents that may have different policies and may even change their policies constantly. In off-policy methods even if an agent changes its policy the system is still able to use what it has learnt so far. Q-learning tends to converge a bit slower; however, it has the capability to continue learning while changing policies and is more flexible if alternative routes appear. Q-learning learns the Q-values associated with taking the exploitation policy while following an exploration/exploitation policy.

6.1.3.4.1 Q-Learning

Q-learning [177] is a model-free reinforcement learning technique that employs off-policy learning method and can be used to find an optimal action-selection policy for any MDP. The algorithm proposes the following function in order to calculate the Quantity of a state-action combination:

$$Q: S \times A \to \mathbb{R}$$
$$Q(s_t, a_t) \leftarrow (1 - \alpha_t) Q(s_t, a_t) + \alpha_t (r_{t+1} + \gamma \max_a Q(s_{t+1}, a)) \tag{8}$$

where r_{t+1} is the reward received after performing a_t in s_t, and α_t is the learning rate ($0 < \alpha_t \leq 1$), which determines to what extent the newly acquired information will override the old information. Before learning has started, an (arbitrary) fixed value, chosen by the designer is assigned to every $Q(s_t, a_t)$. Then, each time the agent selects an action, and observes a reward and a new state that may depend on both the previous state and the selected action, $Q(s_t, a_t)$ is updated.

6.1.3.4.2 Q-Learning for Agent-Based Systems

A significant part of the research on learning in agent-based systems concerns reinforcement learning. The algorithms can be divided into three different classes: single-agent RL, multi-agent RL (a combination of Game Theory[30] and RL), and swarm RL (a combination of Swarm Intelligence (SI)[31] and RL). A comprehensive overview of single and multi-agent RL is presented in [182].

The formal model of single-agent RL is the MDP, i.e., a 5-tuple (S, A, T, R, γ), and as mentioned the agent aims to find an optimal policy, which specifies how the agent chooses its actions given any state. In single-agent RL a single agent applies the RL algorithm, e.g. Q-learning, and updates the values following its action selection strategy.

The application of RL to an MDP allows a single agent to learn a policy that maximizes a cumulative reward that is received from the environment. However, when multiple agents apply RL in a shared environment, the optimal policy of an agent depends not only on the environment, but on the actions of the other agents as well. In this case, there are two main groups to be considered: Multi-agent systems, consisting of agents that have individual goals and decision-making capabilities, which are influenced by other agent's decisions, and on the other hand multi-agent systems behaving as a swarm, consisting of agents that collaboratively learn a single objective. The Multi-Agent RL (MARL) is based on Game Theory concepts, which induces

[30] Game theory is the study of mathematical models of strategic interaction between rational decision-makers [241].

[31] The term Swarm Intelligence (SI) refers to self-organized collective behavior of natural and artificial systems composed of many individuals following simple rules that exploit only local information [187].

that agents are capable of discovering good solutions to the problem at hand either by coordinating with other learners or by competing with them. However, swarm RL considers Swarm Intelligence (SI) concepts, allowing the agent to share their knowledge by updating the policy the swarm is following.

In case of a multi-agent system a generalization of the MDP, i.e., stochastic game is used to describe the problem. As described in [182], a stochastic game is a tuple $(S, A_1, \ldots, A_n, T, R_1, \ldots, R_n, \gamma)$, where:

- n is the number of agents.
- S is the set of the environment states.
- $A_i, i = 1, \ldots, n$ are the set of actions that available to the agents, yielding the joint action set $\boldsymbol{A} = A_1 \times \ldots \times A_n$.
- $T: S \times \boldsymbol{A} \times S \rightarrow [0,1]$ is the state transition probability function.
- $R_i: S \times \boldsymbol{A} \times S \rightarrow \mathbb{R}, i = 1, \ldots, n$ are reward functions of agents.
- $\gamma \in [0,1)$ is a discount factor that trades off between immediate reward and potential future reward.

In the multi-agent case, the state transitions are the result of the joint actions and since the rewards of the agents depend on the joint action, the cumulative reward depend on the joint policy. As the concentration of this research is on swarm RL techniques, for further information on MARL please refer to [182].

As Tuyls and Weiss note:

> Rather than developing complex behaviors for single individuals, in swarm RL swarm intelligence investigates the emerging (intelligent) behavior of a group of simple individuals that achieve complex behavior through their interactions with one another. Consequently, swarm intelligence can be considered as a cooperative multiagent learning approach in that the behavior of the full set of agents is determined by the actions of and interactions among the individuals. SI and RL are closely related, as both techniques use iterative learning algorithms based on trial and error and a reinforcement signal to find optimal solutions. The key difference though is how the reinforcement signal is used to modify an individual's behavior. [183, p. 49]

As a result, similar to single-agent RL, in swarm RL state transition function is described with a single agent's action. On the other hand, similar to multi-agent RL, swarm RL recognizes other agents' actions, however, solely implicitly, by considering their effects.

The most well-known swarm reinforcement learning algorithms are based on Ant Colony Optimization (ACO)[32]:

(1) **Ant-Q** [184]: This algorithm applies the idea of swarm intelligence by calculating the reward considering actions taken by other agents.

(2) **Pheromone-Q-Learning (Phe-Q)** [185]: In this algorithm, the action-selection does not just look for the highest state-action value but it also recognizes the other agents considering a belief factor (B) that shows the extent to which an agent believes in the pheromone that it detects. Belief factor in fact indicates the ratio of pheromone concentration in one state to the neighboring states.

(3) **Swarm RL based on ACO** [186]: In this algorithm each agent performs its own Q-Learning, however, it corrects its values according to the findings of the other agents.

So far swarm RL has focused solely on homogeneous swarms and systems introduced as Heterogeneous Swarm (HetSs) merely include very few, i.e., two or three sub-swarms of homogeneous agents with crisp borderlines for their swarm behaviors. As the HMDS can be regarded as a swarm of heterogeneous agents, this study has also concentrated on the introduction of a novel approach that allows individuals with higher degrees of heterogeneity, to behave as a swarm in case they have identical sub-problems to solve. Some results of these attempts have been already published in [8, 11].

In fact, if there is an overlap between the problems two different agents are solving, or between the data they are aiming for, they may have valuable knowledge to share and therefore a reason to behave as a swarm. Following subsection takes a closer look at the heterogeneous swarms and section 6.1.3.6 describes the Heterogeneous Swarm RL (HetSRL) method [11] that can be used by heterogeneous group of agents with diverse but overlapping goals in order to exhibit

[32] ACO algorithms are probabilistic techniques that can be used for problems which can be reduced to finding good paths in graphs. The main idea of these algorithms, first introduced in [240], is based on the behavior of ants seeking a path between their nest and a food source.

swarm behavior. A carefully adapted version of this method can also be used by the HMDS for the purpose of self-organization (see section 6.1.3.8), which has already been introduced in [8].

6.1.3.5 Swarm Intelligence in Homogeneous and Heterogeneous Swarms

As mentioned, the term Swarm Intelligence (SI), first introduced in [187], refers to self-organized collective behavior of natural and artificial systems composed of many individuals following simple rules that exploit only local information. Examples of natural systems exhibiting swarm intelligence are colonies of ants and termites, schools of fish, flocks of birds, and herds of land animals. Inspired by the natural systems, some artificial systems are also benefiting from the power of swarm intelligence, notably some multi-robot systems (e.g., [188]), and also certain computer programs that are developed to solve optimization and data analysis problems (e.g., [9]). The artificial SI systems consist typically of population of simple robots or software agents, however, recently, some SI systems have been developed that allow human swarming (e.g., [189]). As stated by Dorigo and Birattari, a typical swarm intelligence system has the following properties:

- it is composed of many individuals;
- the individuals are relatively homogeneous (i.e., they are either all identical or they belong to a few typologies);
- the interactions among the individuals are based on simple behavioral rules that exploit only local information that the individuals exchange directly or via the environment (stigmergy);
- the overall behavior of the system results from the interactions of individuals with each other and with their environment, that is, the group behavior self-organizes. [190]

Homogeneous swarms are attractive due to their conceptual simplicity [191], thus most existing studies consider homogeneous swarms, however, “real-world swarms often include agents with varying dynamical properties, which leads to new collective behaviors” [192, p. 810]. This heterogeneity ranges from intra-species behavioral variations caused by morphologic or age differences, to inter-species cooperation, such as symbiosis [191]. In [192] interesting examples of heterogeneous natural systems are given:

(1) Age-structured swarms, in which heterogeneity arises when motion or sensing capabilities vary significantly with age.

(2) Predator-prey swarming, where there are distinct time-scale differences in the motion of predator and prey animals.

(3) Segregation of intermingled cell types during growth and development of an organism by introducing heterogeneity in inter cell adhesion properties.

Inspired by heterogeneous natural swarms, heterogeneous artificial swarms have also been designed and implemented. Some notable studies in this field have concentrated on robotic systems, where individual robots with varying capabilities are segregated in two to three populations and then are used together in order to achieve a common goal [188, 192, 193, 194, 195]. The varying capabilities that causes the heterogeneity in such systems may be due to the lack of capabilities that are costly to be implemented in all agents or may arise over time as some agents in the swarm malfunction [192]. Heterogeneity has also been used in approaches based on Particle Swarm Optimization (PSO) technique [191, 196, 197]. According to [191], in this technique differences along any of the aspects of the configuration of a particle give rise to a taxonomy based on four types of heterogeneity: neighborhood heterogeneity, model-of-influence heterogeneity, update-rule heterogeneity, and parameter heterogeneity.

Heterogeneous systems have also attracted many researchers differently as “designing task-specific agents is often easier than designing versatile, multipotent ones” [191, p. 699]. In order to keep a swarm homogeneous all the agents should be equipped with all the capabilities needed, which would in many cases eventually lead to complicated agents. As a result, heterogeneity clearly supports the very important property of the SI systems and that is the fact that in such systems the agents are to be kept as simple as possible.

As mentioned, so far, systems introduced as Heterogeneous Swarms (HetSs) merely include very few, i.e., two or three sub-swarms of homogeneous agents, which either according to their capabilities deal with specific sub-problem of the general problem (e.g., [188]), or exhibit different behaviors in order to try a variety of different approaches to the same problem, and therefore “reduces the risk of using a homogeneous swarm of the wrong type for the problem at hand” [191]. Following subsection describes an approach that allows agents, which are originally designed to solve different problems and as a result have higher degrees of heterogeneity, to behave as a swarm when addressing identical sub-problems.

6.1.3.6 A Novel Heterogeneous Swarm RL (HetSRL) Method[33]

In homogeneous swarms, agents use the collective intelligence of their group in order to make decisions. Indeed, the homogeneity in capabilities and goals between the agents make this collaboration argumentative. Accordingly, in HetSs the agents should be able to measure the similarity in their capabilities and goals, i.e., affinity, to check the possibility of exhibiting swarm behavior. One of the famous examples of such approach can be seen in human decision-making and the extent to which a person follows suggestions from different groups of people based on their affinity. Accordingly, some recommender systems are considering the social affinity between the users to predict user preferences [198, 199].

According to the American Psychological Association (APA)[34] ingroup bias or ingroup favoritism is “the tendency to favor one’s own group, its members, its characteristics, and its products, particularly in reference to other groups. The favoring of the ingroup tends to be more pronounced than the rejection of the outgroup, but both tendencies become more pronounced during periods of intergroup contact” [200]. As stated in [201, p. 1739], this tendency “often results from a greater propensity to trust those who are similar to oneself in background or values”. In this case, commonality builds affinity and trust.

In the context of multi-agent systems affinity can be defined as a value that indicates, according to the application’s objectives, the compatibility of two agents to work together (see section 4.2.2.2.2). In order to be able to measure the affinity between two agents, the determining factors should be identified and eventually measured for each of the agents. In this context, the term agent profile can be defined as a list of values representing the extent to which an agent exhibits interest in the characteristics under consideration. As a result, in order to measure the affinity between two agents, the similarity between their profiles can be measured (e.g. calculating the distance between the profiles). In HetSs, an agent may check for similarities between its own profile and the profile of some relevant agents, in order to make its own decision. Therefore, the environment should be able to provide the agents with the relevant information.

[33] The contents of this sub-section present the result of a study that has been conducted as a part of this doctoral research and have been already published in [11].

[34] APA is the leading scientific and professional organization representing psychology in the United States, with more than 118,000 researchers, educators, clinicians, consultants and students as its members [242].

In order to model such sequential decision-making problems an extension of the MDP can be introduced, which is augmented by assuming a new element that allows the measurement of the affinity between the agent profiles. This model is described as a 6-tuple (S,P,A,T,R, γ), where:

- S is a set of agent-environment states.
- P is a set of profiles that for each state indicates its visitors' collective profile.
- A is a set of actions the agent can take.
- $T(s_i, a, s_j): S \times A \times S \rightarrow [0,1]$ is a state transition function which gives the probability that taking action a in state s_i results in state s_j.
- $R(s_i, a, s_j): S \times A \times S \rightarrow \mathbb{R}$ is a reward function that quantifies the reward the agent gets for taking action a in state s_i resulting in state s_j.
- $\gamma \in [0,1)$ is a discount factor that trades off between immediate reward and potential future reward.

In this representation of the problem, the optimal solution should still solely maximize the reward; however, at each time step the best action will be the one that maximizes the reward and affinity simultaneously. As a result, it should be noted that this problem cannot be regarded as a multi-objective optimization problem that has two objectives.

Several researches have concentrated on designing MDPs for optimization problems with multiple objectives (e.g. [202, 203, 204]). According to [203, p. 70], "a Multi-Objective MDP (MOMDP) is an MDP in which the reward function $\boldsymbol{R}: S \times A \times S \rightarrow \mathfrak{R}^n$ describes a vector of n rewards, one for each objective, instead of a scalar. Similarly, a value function V^π in an MOMDP specifies the expected cumulative discounted reward vector:

$$\boldsymbol{V}^\pi = E[\sum_{k=0}^{\infty} \gamma^k \boldsymbol{r}_{k+1} \,|\pi] \tag{9}$$

$$\boldsymbol{V}^\pi(s) = E[\sum_{k=0}^{\infty} \gamma^k \boldsymbol{r}_{t+k+1} \,|\pi, s_t = s] \tag{10}$$

where $\boldsymbol{r}_t$ is the vector of rewards received at time t."

As mentioned, the profile-based or affinity-based extension of MDP is not a MOMDP, however, at each time step, action selection is an optimization problem with two objectives: maximizing the reward and the affinity. In this case, the MDP remains single-objective; however, the action selection will be multi-objective.

As stated in [205, p. 77], a Multi-Objective Optimization Problem (MOOP) is a multiple criteria decision-making problem, "which is concerned with mathematical optimization problems involving more than one objective function to be optimized simultaneously". For a nontrivial multi-objective optimization problem, there does not exist a single solution, but a set of non-dominated or Pareto optimal solutions that without additional preferences, i.e. degrading some of the other objective values, all can be considered equally good as solution vectors cannot be ordered [205]. In our case, any of the actions leading to a Pareto optimal solution to maximizing the reward and affinity can hence be chosen as a Pareto optimal action at each state, however, only the resulting reward is to be considered for the overall optimization problem.

Scalarization is one the most commonly useful methods for finding the Pareto optimal solutions for multi-objective optimization problems. "Scalarizing a multi-objective optimization problem means formulating a single-objective optimization problem such that optimal solutions to the single-objective optimization problem are Pareto optimal solutions to the multi-objective optimization problems" [206, p. 7484]. As suggested in [204], assuming that X is the set of all feasible inputs to a MOOP with objective $\boldsymbol{f}(x)$, one approach to finding the Pareto optimal solutions is to solve a set of optimization problems that are scalarized versions of the MOOP at hand. "A scalarization function ρ can hence be chosen which maps vector-valued outcomes scalars and then one solves:

$$\max_{x \in X} \rho\big(\boldsymbol{f}(x)\big), \tag{11}$$

the scalar optimization defined by composing the vector-valued outcome function with the scalarization function" [204, p. 7382]. Generally, having no preferences in choosing the final Pareto optimal solution would help to implement randomness, which helps to avoid bias. As a result, the scalar optimization function will be an objective function, i.e., a loss function to be minimized or its negative to be maximized.

In the affinity-based HetSRL, the objective is to maximize both Q-value and affinity for each action simultaneously, however, since the scales are not same, and the maximum value of both values are not clear, the multiplication of the two values is considered for the scalar optimization function:

$$\max_{a \in A}(Q(s_t, a). Aff\ (agtP, P(s_{t+1}))) \tag{12}$$

where $Aff(agtP, P(s_{t+1}))$ indicates the affinity between the agent performing action a and the ones who have already done this action and contributed to the calculation of the cumulative value given by $Q(s_t, a)$. One method for calculating this value is to calculate the Euclidean distance between the agent's profile, i.e., $agtP$ and the collective profile of the state's visitors $P(s_{t+1})$.

As mentioned above, even though at each time step the best action will be the one that maximizes the reward and the affinity simultaneously, the overall optimal solution should still solely maximize the cumulative reward value. As a result, in the affinity-based HetSRL in order to update the Q-value for the estimation of optimal future values instead of max function $argmax$ function is used:

$$\begin{aligned} Q_{new}(s_t, a_t) \leftarrow (1 - \alpha_t) Q_{old}(s_t, a_t) + \alpha_t (R_{new}(s_t, a_t) \\ + \gamma \operatorname*{argmax}_{Q_{old}(s_{t+1}, a)} (Q_{old}(s_{t+1}, a). Aff(agtP, P(s_{t+2})))) \end{aligned} \tag{13}$$

In this extension of Q-learning α can be defined as a coefficient of affinity in order to show the relevance of the new information.

To prove the convergence of this extension of Q-learning, the approach presented in [207] can be adapted to this method. In section 6.1.3.9 an example of such an approach is given that proves the convergence of a practical application of the affinity-based HetSRL method, i.e., the Holonic-Q-Learning (see section 6.1.3.8). Before introducing the Holonic-Q-Learning and proving the convergence of this method, however, the effectiveness of the affinity-based HetSRL method is shown by applying this method to the Shortest Path Problem (SPP) (section 6.1.3.7).

6.1.3.7 The Application of Affinity-Based HetSRL Method to the SPP

In order to demonstrate the performance of the affinity-based HetSRL method introduced in the previous section in a straightforward example, the shortest path problem from the start cell to targets in an n by n grid environment is considered here. Targets may be placed in arbitrary locations on the grid. Both targets and the agents are represented by a profile, i.e. an array of values, and the agents' goal is to find the shortest path to the most relevant target. This problem resembles a recommendation system in which agents represent user preferences and try to predict the rating or preference a user would give to an item. This simulation considers a 20 by 20 environment, 10 agents with different profiles, and two different targets. Figure 38. illustrates the problem environment. Targets are located at (10,20) and (20,10), and the starting point is at (1,1), letting the coordinates at the top left be (1,1). Reaching the targets, an agent will receive a reward that is directly proportional to its affinity to the target, i.e. $r = Aff\big(agtP, P(target)\big) \times 100$.

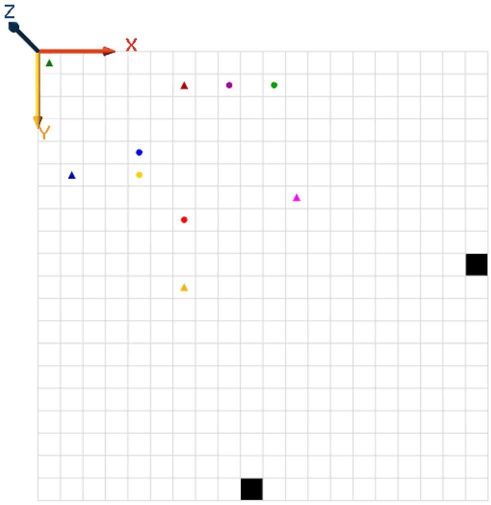
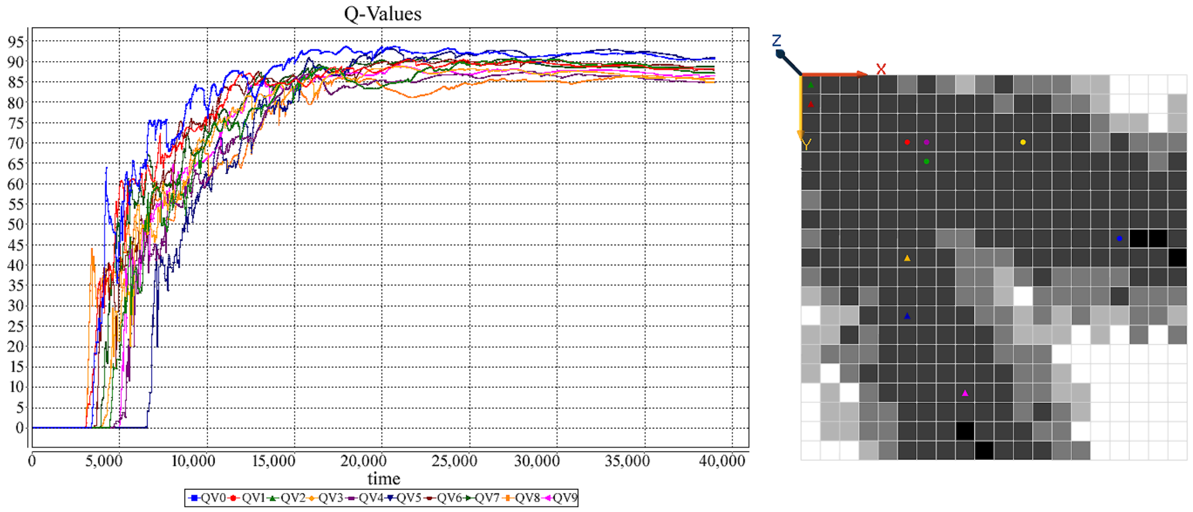


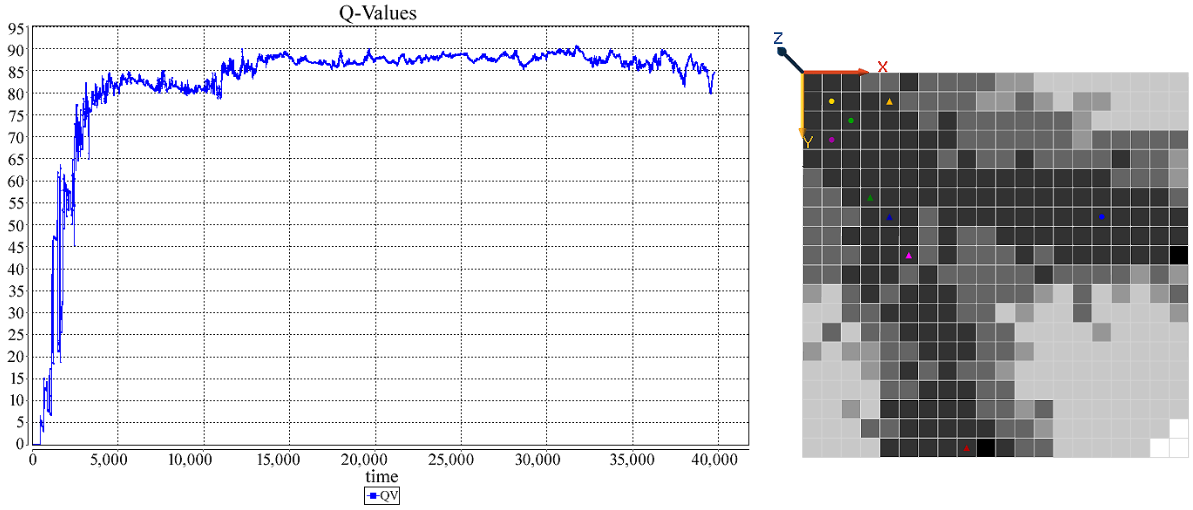
Figure 38. The problem environment.

All the experiments were performed on Intel(R) Celeron(R) CPU 1.90GHz with 10 GB RAM on the Microsoft Windows 10 platform and the simulations have been conducted using the GAMA platform, which is a modeling and simulation development environment for building spatially explicit agent-based simulations [21] (For further information on GAMA platform please refer to chapter 7). Figure 39–Figure 42 demonstrate the changes of the Q-value at (1,1). Basically, the Q-Value remains zero until the first path from the starting point to a target is found. This value then remains almost stable when the agent follows almost the same path (i.e. the best path found within a constrained time frame) and has less intention to explore other possibilities.



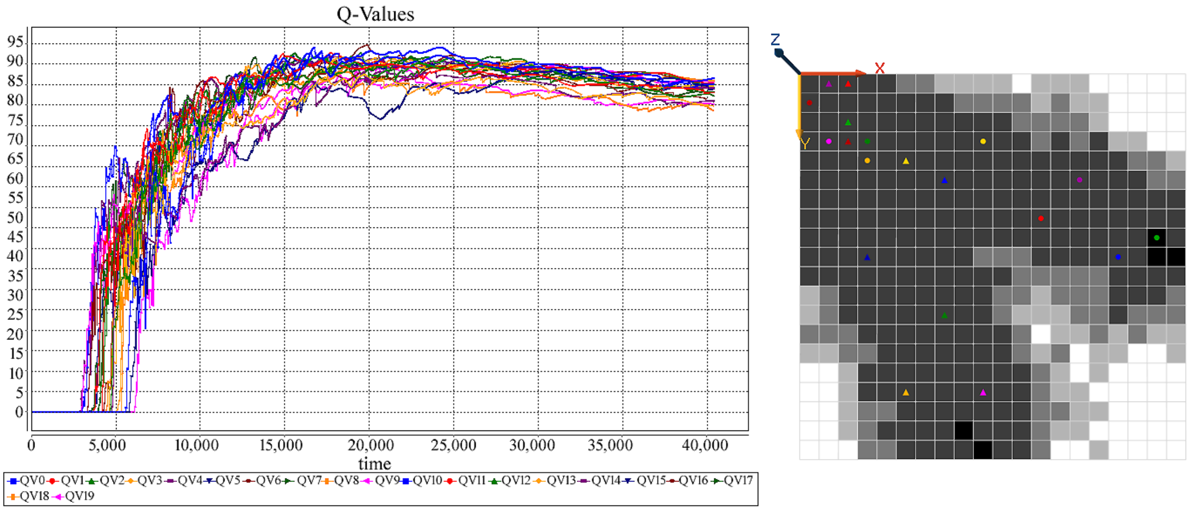
(a) (b)

Figure 39. (a) QV chart for 10 single agents. (b) The QVs for different cells.



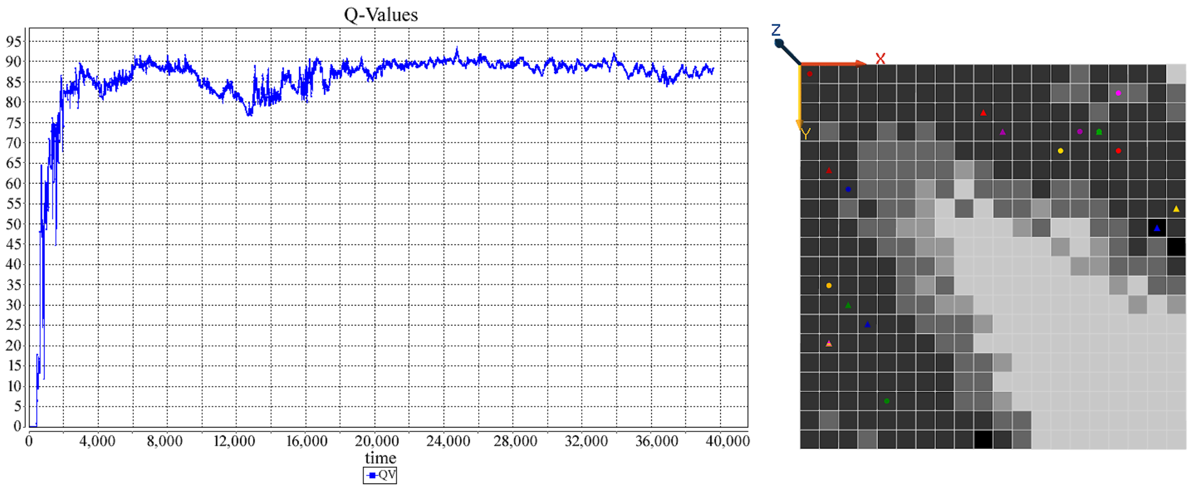
(a) (b)

Figure 40. (a) QV chart for a HetS of 10 agents. (b) The QVs for different cells.



(a) (b)

Figure 41. (a) QV chart for 20 single agents. (b) The QVs for different cells.



(a) (b)

Figure 42. (a) QV chart for a HetS of 20 agents. (b) The QVs for different cells.

Table 14 compares the approximate time of the first path detection and the approximate time needed for the optimal path detection for all of the four experiments. The results indicate that HetSRL has significantly reduced the search time, and the impact is more significant as the number of agents increases.

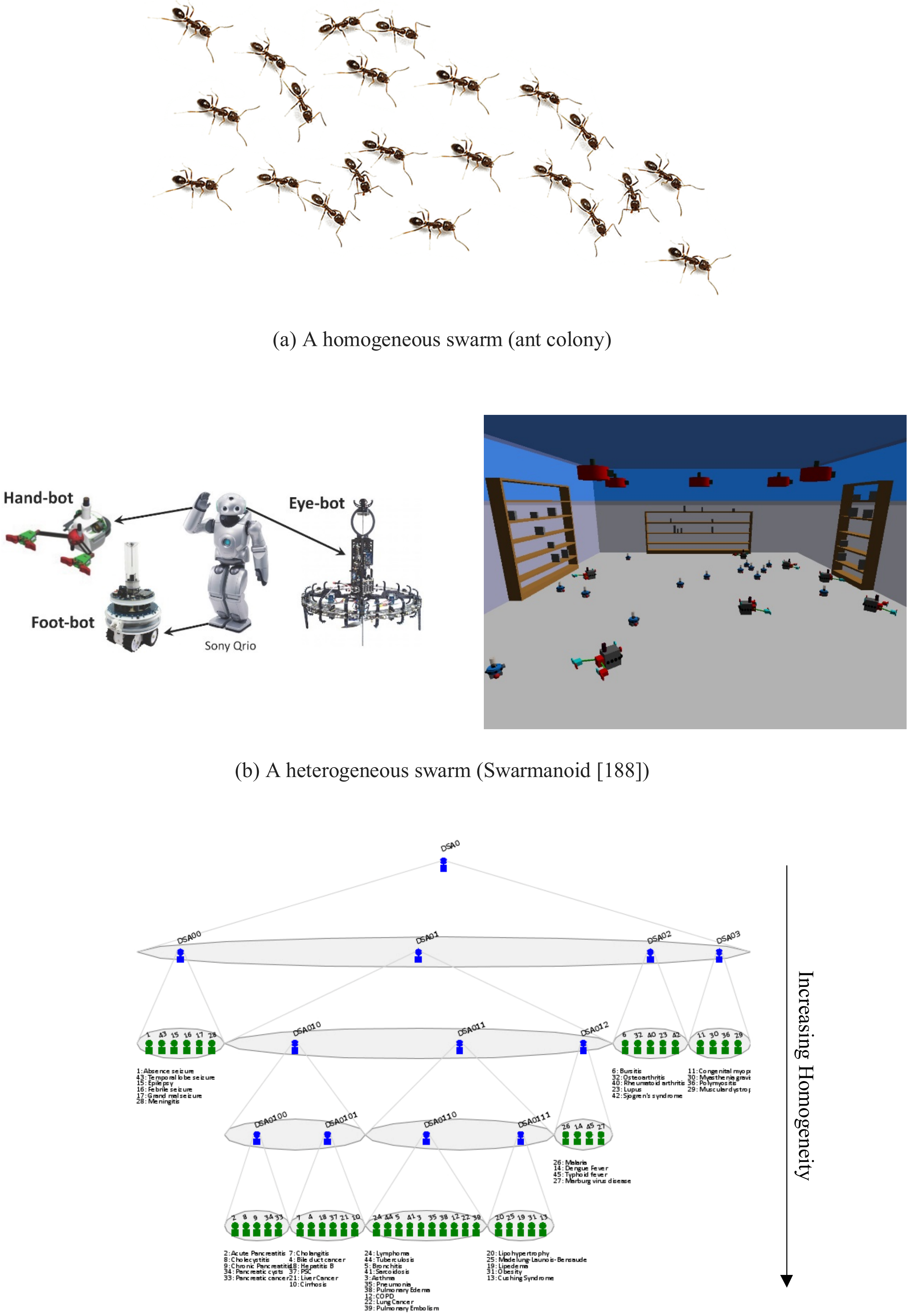
Table 14. Path detection time.

	Approximate Time Needed for the First Path Detection	**Approximate Time Needed for the Optimal Path Detection**
Figure 39: Single-agent (10 agents)	35 seconds	150 seconds
Figure 40: HetS (10 agents)	5 seconds	35 seconds
Figure 41: Single-agent (20 agents)	35 seconds	150 seconds
Figure 42: HetS (20 agents)	5 seconds	20 seconds

6.1.3.8 The Holonic-Q-Learning (HQL)

This subsection aims to show how a carefully adapted affinity-based HetSRL method can support the HMDS in performing the self-organization process. The HMDS is a HetS, consisting of agents with different information and goals, which consequently leads to different behaviors. In this system, super-holons are mapping the sub-swarms. In contrast to the HetSs under consideration so far (see section 6.1.3.5) the sub-swarms here are not homogeneous but again heterogeneous and may again include heterogeneous sub-swarms. Even though these sub-swarms are not homogeneous, the information and goals (behaviors) of their members are relatively more similar in the lower levels, such that they can exhibit swarm behavior. To this end, the degree of homogeneity, in our terminology the affinity between the agents, should be measured and considered.

Figure 43 shows the HMDS in comparison with the well-known homogeneous and heterogeneous swarms. As mentioned, the ant colony is the famous example of homogeneous systems that exhibit swarm behavior. In such systems all the agents are identical. So far, systems introduced as heterogeneous swarms merely include very few, i.e., two or three sub-swarms of homogeneous agents. The famous example of this category is the Swarmnoid [188], which is a heterogeneous swarm of robots. As mentioned above, in the HMDS, and in general HMASs, sub-swarms are again heterogeneous, however, with higher degree of homogeneity, such that they can measure the appropriateness of exhibiting swarm behavior and based on the results decide to collaborate. When two agents have high affinity, they will have more relevant information to share and therefore they have reasons to collaborate and exhibit swarm behavior.



(a) A homogeneous swarm (ant colony)

(b) A heterogeneous swarm (Swarmanoid [188])

(c) Heterogeneity in different levels of the HMDS

Figure 43. The HMDS in comparison with well-known swarms

The Holonic-QL is a Q-learning technique introduced for self-organization in the HMDS [8]. In HQL, the Q-value is in fact measuring how favorable it is for a holon to be a member (i.e., sub-holon), of another holon. In this case, the states are the existing holons $\{h_i\}$ and $\mathfrak{h}_i$ denotes the associated actions of trying the membership in holon i[35]:

$$\begin{aligned} Q_t(sub(h),\mathfrak{h}) &\leftarrow (1-\alpha_t)Q_{t-1}(sub(h),\mathfrak{h}) + \alpha_t(R_t(sub(h),\mathfrak{h}) \\ &+ \gamma \underset{Q_{t-1}(h,\mathfrak{sup}(\mathfrak{h}))}{\operatorname{argmax}} (Q_{t-1}(h,\mathfrak{sup}(\mathfrak{h}))).Aff(agtP,P(sup(h))))) \end{aligned} \tag{14}$$

where,

$$\alpha_t = \frac{1}{1+visits_t(sub(h),\mathfrak{h})} \tag{15}$$

$$Aff(agtP,P(h)) = 1 - \frac{d(agtP,P(h))}{\max d(agtP,P(h))} \tag{16}$$

and the reward is calculated by the head of a super-holon (see [9] or section 6.3).

6.1.3.9 The Proof of Convergence of the Holonic-Q-Learning

In case of a deterministic reward function, the convergence of the Q-learning method can be proven following the same simple approach as the one taken in [208], which uses a theorem on random iterative processes convergence introduced in [209, 210]. For Q-learning methods with nondeterministic reward functions, however, the general convergence theorem for Q-learning presented in [207] can be adapted to the Q-learning method in question. As stated in [207], the most important condition in this theorem is that the sequence of episodes (not necessarily continuous) that form the basis of learning must include an infinite number of episodes for each starting state and action. The theorem given bellow is defined and eventually proved following the same approach as in [207].

[35] In comparison to general affinity-based HetSRL model that is described as a 6-tuple (S,P,A,T,R, γ), HQL can be defined as a finite $MDP(H,A,T,R,\gamma)$, where H is the set of holons. The reason behind is that each holon is represented by a holon-identifier that carries the profile information.

Theorem: Given bounded rewards[36] $|r_n| \leq \mathcal{R}$, learning rates $0 \leq \alpha_n \leq 1$, and

$$\sum_{i=1}^{\infty} \alpha_{n^{i}(sub(h),\mathfrak{h})} = \infty, \sum_{i=1}^{\infty} \left[\alpha_{n^{i}(sub(h),\mathfrak{h})}\right]^2 < \infty, \forall sub(h), \mathfrak{h}, \tag{17}$$

where $n^i(sub(h), \mathfrak{h})$ is the index of the i^th^ time $sub(h)$ has tried the membership in holon h, then $Q_n(sub(h), \mathfrak{h}) \rightarrow Q^*(sub(h), \mathfrak{h})$ as $n \rightarrow \infty, \forall sub(h), \mathfrak{h}$, with probability 1.

The convergence proof: As stated in [207, p. 282], "the key to the approach is an artificial controlled Markov process called the Action-Replay Process (ARP), which is constructed from the episode sequence and the learning rate sequence α_n." To prove the convergence of Q-learning the ARP is constructed in a way that $Q_n(x, a)$ are the optimal action values for ARP states $\langle x, n\rangle$ and ARP actions a, and that the ARP convergences to the real process. Two lemmas address these conditions respectively and form the heart of the proof. These lemmas are then used to prove that $Q_n(x, a)$ tend to $Q^*(x, a)$. In order to prove the convergence of HQL the ARP definition, and the lemmas are adapted to this method.

The Action-Replay Process (ARP): "The definition of the ARP is contingent on a sequence of episodes observed in the real process" [207, p. 287]. The state space of the ARP is $\{\langle sub(h), n\rangle\}$, for state $sub(h)$ of the real process and $n \geq 1$, together with an absorbing state, and the action space is $\{h\}$ for action h from the real process.

In order to establish the stochastic reward and state transition consequent on performing action h at state $\langle sub(h), n\rangle$ the following definitions are given. For convenience, define $n^i \equiv n^i(sub(h), \mathfrak{h})$, as the index of the i^th^ time $sub(h)$ has tried the membership in holon h. Define

$$i_* = \begin{cases} argmax_i\{n^i < n\} & if\ sub(h), \mathfrak{h}\ has\ been\ executed\ before\ episode\ n \\ 0 & otherwise \end{cases} \tag{18}$$

such that n^{i_*} is the last time before episode n that $sub(h), h$ was executed in the real process. If $i_* = 0$, the reward is set to $Q_0(sub(h), \mathfrak{h})$, and the ARP absorbs. Otherwise, let

[36] The reward function of HQL is engineered so that all rewards are bounded by 1. However, to prove the convergence for a more general case, value $\mathcal{R}$ is considered here as the upper bound of the reward values.

$$i_e = \begin{cases} i_* & with\, probability\ \alpha_{n^{i_*}} \\ i_* - 1 & with\, probability\ \left(1 - \alpha_{n^{i_*}}\right)\alpha_{n^{i_*-1}} \\ i_* - 2 & with\, probability\ \left(1 - \alpha_{n^{i_*}}\right)\left(1 - \alpha_{n^{i_*-1}}\right)\alpha_{n^{i_*-2}} \\ \vdots & \vdots \\ 0 & with\, probability\ \prod_{i=1}^{i_*}\left(1 - \alpha_{n^i}\right) \end{cases} \tag{19}$$

be the index of the episode that is replayed or taken, chosen probabilistically from the sample episodes of the real process. As stated above, if $i_e = 0$, the reward is set to $Q_0(sub(h), \mathfrak{h})$, and the ARP absorbs. Otherwise, taking i_e provides reward $r_{n^{i_e}}$, and causes a state transition to $\langle h_{n^{i_e}}, n^{i_e} - 1\rangle$, which is at level $n^{i_e} - 1$. As a result, taking an action in the ARP always causes a state transition to a lower level and eventually a termination. It should be noted that the same as in the real process, the discount factor in the ARP is set to γ.

Lemma A: $Q_n(sub(h), \mathfrak{h})$ are the optimal action values for ARP states $\langle sub(h), n\rangle$ and ARP actions $\mathfrak{h}$. That is

$$Q_n(sub(h), \mathfrak{h}) = Q_{ARP}^*(\langle sub(h), n\rangle, \mathfrak{h}), \forall \mathfrak{h}, sub(h), and\ n \geq 0. \tag{20}$$

Proof: By induction. From the construction of ARP, $Q_0(sub(h), \mathfrak{h})$ is the only possible and hence the optimal action value of $\langle sub(h), 0\rangle, \mathfrak{h}$. Therefore,

$$Q_0(sub(h), \mathfrak{h}) = Q_{ARP}^*(\langle sub(h), 0\rangle, \mathfrak{h}).$$

Thus, the theorem holds for $n = 0$.

Suppose that the values of Q_{n-1}, as produced by the HQL rule, are the optimal action value for the ARP at level $n - 1$, that is

$$Q_{n-1}(sub(h), \mathfrak{h}) = Q_{ARP}^*(\langle sub(h), n - 1\rangle, \mathfrak{h}), \forall \mathfrak{h}, sub(h).$$

This implies that the value of state $sub(h)$ after $n - 1$ episodes are the optimal values for the ARP at the $n - 1$th level, that is

$$V_{n-1}\left(sub(h)\right) = V^*(\langle sub(h), n - 1\rangle).$$

Now consider the cases in trying to perform action $\mathfrak{h}$ in $\langle sub(h), n\rangle$. If $sub(h), \mathfrak{h} \neq sub(h)_n, \mathfrak{h}_n$, then this is the same as performing $\mathfrak{h}$ in $\langle sub(h), n-1\rangle$, and $Q_n(sub(h), \mathfrak{h}) = Q_{n-1}(sub(h), \mathfrak{h})$. Therefore,

$$Q_n(sub(h), \mathfrak{h}) = Q_{n-1}(sub(h), \mathfrak{h}) = Q^*_{ARP}(\langle sub(h), n-1\rangle, \mathfrak{h}) = Q^*_{ARP}(\langle sub(h), n\rangle, \mathfrak{h}).$$

Otherwise, performing h_n in $\langle sub(h)_n, n\rangle$

- with probability $1-\alpha_n$ is exactly the same as performing $\mathfrak{h}_n$ in $\langle sub(h)_n, n-1\rangle$, or
- with probability α_n yields immediate reward r_n and new state $\langle h_n, n-1\rangle$.

Therefore, the optimal action value in the ARP of $\langle sub(h)_n, n\rangle$, $\mathfrak{h}_n$ is

$$\begin{aligned}
&Q^*_{ARP}(\langle sub(h)_n, n\rangle, \mathfrak{h}_n)\\
&\qquad = (1-\alpha_n)Q^*_{ARP}(\langle sub(h)_n, n-1\rangle, \mathfrak{h}_n) + \alpha_n\big(r_n + \gamma V^*(\langle \mathrm{h}_n, n-1\rangle)\big)\\
&\qquad = (1-\alpha_n)Q_{n-1}(sub(h)_n, \mathfrak{h}_n) + \alpha_n\big(r_n + \gamma V_{n-1}(h_n)\big)\\
&\qquad = (1-\alpha_n)Q_{n-1}(sub(h)_n, \mathfrak{h}_n)\\
&\qquad + \alpha_n\left(r_n + \gamma \underset{Q_{n-1}(h_n, \mathrm{sup}(\mathfrak{h})_n)}{\operatorname{argmax}} Q_{n-1}(h_n, \mathrm{sup}(\mathfrak{h})_n).Aff(agtP, P(sup(h)_n))\right)\\
&\qquad = Q_n(sub(h)_n, \mathfrak{h}_n)
\end{aligned}$$

from the induction hypothesis and the HQL equation. Hence $Q_n(sub(h), \mathfrak{h}) = Q^*_{ARP}(\langle sub(h), n\rangle, \mathfrak{h}), \forall \mathfrak{h}, sub(h)$, as required.

Lemma B [207, p. 283]**:** This lemma concerns the convergence of the ARP to the real process. The first two lemmas are preparatory, however, the next two specify the form of the convergence and provide foundations for proving that it occurs.

B.1 (Discounting infinite sequences): Consider a discounted, bounded-reward, finite Markov process with transition matrix $T(sub(h), \mathfrak{h}, h)$. From any starting $sub(h)$, the difference between the value of that state under any set of s actions and under those same s actions followed by any arbitrary policy tends to 0 as $s \to \infty$.

Proof: Ignoring the value of the $(s+1)^{\text{th}}$ state incurs a penalty of

$$\delta \equiv \gamma^s \sum_{h_{s+1}} T(sub(h)_s, \mathfrak{h}_s, h_{s+1}) V^\pi(h_{s+1}).$$

But if all rewards are bounded by $\mathcal{R}$, $V^\pi(h) < \mathcal{R} \sum_{i=0}^{\infty} \gamma^i = \mathcal{R}/1 - \gamma$, and so

$$|\delta| < \gamma^s \frac{\mathcal{R}}{1-\gamma} \to 0 \text{ as } s \to \infty.$$

B.2 (The probability of straying below level low after executing s actions can be made arbitrarily small): Given any level low, there exist another yet higher level, $high$, such that the probability of straying below low after taking s actions in ARP starting from above $high$ can be made arbitrarily small.

Proof: Define i_{high} as the largest i such that $n^i(sub(h), \mathfrak{h}) \le n$, and i_{low} as the smallest such that $n^i(sub(h), \mathfrak{h}) \ge low$. Then, defining $\alpha_{n^0} = 1$, the probability of straying below low starting from $\langle sub(h), n \rangle, n > low$ executing action $\mathfrak{h}$ is:

$$\left[\prod_{i=i_{low}}^{i_{high}} (1 - \alpha_{n^i})\right] \sum_{j=0}^{i_{low}-1} \left\{\alpha_{n^j} \prod_{k=j+1}^{i_{low}-1} (1 - \alpha_{n^k})\right\} \le \prod_{i=i_{low}}^{i_{high}} (1 - \alpha_{n^i})$$

The exponential inequality $1 + x \le e^x$ implies that $\prod_{i=i_{low}}^{i_{high}} (1 - \alpha_{n^i}) < e^{-\sum_{i=i_{low}}^{i_{high}} \alpha_{n^i}} \to 0$ as n and hence $i_{high} \to \infty$. Furthermore, since the state and action spaces are finite, given η, there exist some level n_1 such that starting above there from any $(sub(h), \mathfrak{h})$ leads to a level above low with probability at least $1 - \eta$. This argument iterates for the second action with n_1 as the new lower limit. In fact, any increase in $|high - low|$ causes the probability of straying below low to become subject to exponential decay. η can be chosen appropriately to set the overall probability of straying below low less than any arbitrary $\epsilon > 0$.

B.3 (Rewards and transition probabilities converge with probability 1): Define $T^{ARP}(\langle sub(h), n \rangle, \mathfrak{h}, \langle h, m \rangle)$ and $R^{(n)}(sub(h), \mathfrak{h})$ as the transition-probability matrices and expected rewards of ARP. Also define

$$T^{(n)}(sub(h), \mathfrak{h}, h) = \sum_{m=1}^{n-1} T^{ARP}(\langle sub(h), n \rangle, \mathfrak{h}, \langle h, m \rangle) \tag{21}$$

as the probability that for each $sub(h)$, n and $\mathfrak{h}$, executing action $\mathfrak{h}$ at state $\langle sub(h), n\rangle$ in ARP leads to state h of the real process at some lower level in th deck. With probability 1, as the level n increases to infinity, the probabilities $T^{(n)}(sub(h), \mathfrak{h}, h)$ and expected rewards $R^{(n)}(sub(h), \mathfrak{h})$ in the ARP converge and tend to the transition matrices and the expected rewards in the real process.

Proof: Theorem 2.3.1 of [211] states that if X_n are updated according to

$$X_{n+1} = X_n + \beta_n(\xi_n - X_n)$$

Where $0 \leq \beta_n < 1, \sum_{i=1}^{\infty}\beta_n = \infty, \sum_{i=1}^{\infty}{\beta_n}^2 < \infty$, and ξ_n are bounded random variables with mean Ξ, then $X_n \rightarrow \Xi$, as $n \rightarrow \infty$, with probability 1.

If $R(\langle sub(h), n\rangle, \mathfrak{h})$ is the expected immediate reward for performing action $\mathfrak{h}$ from state $sub(h)$ at level n in the ARP, then

$$\begin{aligned} R\big(\langle sub(h), n^{i+1}\rangle, \mathfrak{h}\big) &= \alpha_{n^{i+1}}.r_{n^{i+1}} + \big(1 - \alpha_{n^{i+1}}\big)R\big(\langle sub(h), n^{i}\rangle, \mathfrak{h}\big) \\ &= R\big(\langle sub(h), n^{i}\rangle, \mathfrak{h}\big) + \alpha_{n^{i+1}}(r_{n^{i+1}} - R\big(\langle sub(h), n^{i}\rangle, \mathfrak{h}\big)) \end{aligned}$$

Where R and α satisfy the conditions of the theorem with $\Xi = R(sub(h), \mathfrak{h})$. Therefore $R(\langle sub(h), n\rangle, \mathfrak{h}) \rightarrow R(sub(h), \mathfrak{h})$ as $n \rightarrow \infty$, with probability 1. Also, since there are only a finite number of states and actions, the convergence is uniform.

As α_{n^i} itself indicates the probability that action $\mathfrak{h}$ is executed from $sub(h)$ at level i, define

$$t_n = \begin{cases} 1 & if\ h_n = h \\ 0 & otherwise \end{cases}$$

as a random variable indicator function of the n^th^ transition, with $\Xi = T(sub(h), \mathfrak{h}, h)$. Then, with $T^{(n)}(sub(h), \mathfrak{h}, h)$ as the probability of ending up at state h based on transition from state $sub(h)$ using action $\mathfrak{h}$ at level n in the ARP,

$$\begin{aligned} T^{(n^{i+1})}(sub(h), \mathfrak{h}, h) &= \alpha_{n^{i+1}}.t_{n^{i+1}} + \big(1 - \alpha_{n^{i+1}}\big)T^{(n^{i})}(sub(h), \mathfrak{h}, h) \\ &= T^{(n^{i})}(sub(h), \mathfrak{h}, h) + \alpha_{n^{i+1}}(t_{n^{i+1}} - T^{(n^{i})}(sub(h), \mathfrak{h}, h)) \end{aligned}$$

and so, $T^{(n)}(sub(h), \mathfrak{h}, h) \rightarrow\ T(sub(h), \mathfrak{h}, h)$ as $n \rightarrow \infty$, with probability 1. Since in addition, all observation from the real process are independent, and by B.2, the probability of straying below a fixed level k can be made arbitrarily small, the transition probability and expected rewards for a single step conditional on ending up at a level greater than k also converges to $T(sub(h), \mathfrak{h}, h)$ and $R(sub(h), \mathfrak{h})$ as $n \rightarrow \infty$.

B.4 (Close rewards and transitions imply close values): Let $T^i(sub(h), \mathfrak{h}, h)$ for $i = 1 \dots s$ be the transition matrices of s Markov chains, and $R^i(sub(h), \mathfrak{h})$ be the reward functions. Consider the s-step chain formed from the concatenation of these, i.e., starting from state $sub(h)$, move to state h according to $T^1(sub(h), \mathfrak{h}, h)$, then state $sup(h)$ according to $T^2(h, \mathfrak{sup}(\mathfrak{h}), \mathrm{sup}\,(h))$, and so on, with corresponding rewards. Given $\eta > 0$, if $T^i(sub(h), \mathfrak{h}, h)$ are within $\eta/\mathcal{R}$ of $T(sub(h), \mathfrak{h}, h)$, $\forall \mathfrak{h}, sub(h), h$, and $R^i(sub(h), \mathfrak{h})$ are within η of $R(sub(h), \mathfrak{h})$, $\forall \mathfrak{h}, sub(h)$, then the value of the s actions in the concatenated chain is within $\eta s(s+1)/2$ of their value in the real process.

Proof: Define:

$$\bar{Q}\big(sub(h), \mathfrak{h}, \mathfrak{sup}(\mathfrak{h})\big) = R(sub(h), \mathfrak{h}) + \gamma \sum_h T(sub(h), \mathfrak{h}, h)\, R(h, \mathfrak{sup}(\mathfrak{h}))$$

as the expected reward in the real process for executing two actions, $\mathfrak{h}$ and $\mathfrak{sup}(\mathfrak{h})$ at state $sub(h)$, and

$$\bar{Q}'\big(sub(h), \mathfrak{h}, \mathfrak{sup}(\mathfrak{h})\big) = R^1(sub(h), \mathfrak{h}) + \gamma \sum_h T^1(sub(h), \mathfrak{h}, h)\, R^2(h, \mathfrak{sup}(\mathfrak{h}))$$

as the equivalent in the concatenated chain for exactly the same action.

Then, since in any given s-step chain between 0 to s different rewards might be received, $\left|R^i(sub(h), \mathfrak{h}) - R(sub(h), \mathfrak{h})\right| < \eta$, and $\left|T^i(sub(h), \mathfrak{h}, h) - T(sub(h), \mathfrak{h}, h)\right| < \eta/\mathcal{R}$, $\forall \mathfrak{h}, i, sub(h), h$,

$$\left|\bar{Q}'(sub(h), \mathfrak{h}_1, \dots, \mathfrak{h}_s) - \bar{Q}(sub(h), \mathfrak{h}_1, \dots, \mathfrak{h}_s)\right| < \eta \sum_{i=1}^{s} \frac{\binom{s}{i}}{2^s} i < \eta \sum_{i=1}^{s} i = \eta \frac{s(s+1)}{2}$$

This applies to the ARP if the rewards and transition matrices at the successively lower levels are sufficiently close to those in the real process (the main body of the theorem quantifies the cost of this condition failing).

The theorem: Putting these together, the ARP tends towards the real process, and so its optimal Q-values do too. $Q_n(sub(h), \mathfrak{h})$ are the optimal values for the nth level of ARP (by Lemma A), and so tend to $Q^*(sub(h), \mathfrak{h})$.

Proof: Assume, without loss of generality, that $Q_0(sub(h), h) < \mathcal{R}/(1-\gamma)$ and that $\mathcal{R} \geq 1$.

Given $\varepsilon > 0$, choose s such that

$$\gamma^s \frac{\mathcal{R}}{1-\gamma} < \frac{\varepsilon}{6}.$$

By B.3 with probability 1, it is possible to choose low sufficiently large such that for $n > low$, and $\forall \mathfrak{h}, sub(h), h$,

$$\left|T^{(n)}(sub(h), \mathfrak{h}, h) - T(sub(h), \mathfrak{h}, h)\right| < \frac{\epsilon}{3s(s+1)\mathcal{R}},$$

$$\text{and } \left|R^{(n)}(sub(h), \mathfrak{h}) - R(sub(h), \mathfrak{h})\right| < \frac{\epsilon}{3s(s+1)}.$$

By B.2, choose $high$ sufficiently large such that for $n > high$, the probability, after taking s actions, of ending up at a level lower than low is less than $min\{(\varepsilon(1-\gamma)/6s\mathcal{R}), (\varepsilon/3s(s+1)\mathcal{R})\}$. This means that

$$\left|T'^{(n)}(sub(h), \mathfrak{h}, h) - T(sub(h), \mathfrak{h}, h)\right| < \frac{2\epsilon}{3s(s+1)\mathcal{R}},$$

$$\text{and } \left|R'^{(n)}(sub(h), \mathfrak{h}) - R(sub(h), \mathfrak{h})\right| < \frac{2\epsilon}{3s(s+1)},$$

where the primes on $T'^{(n)}$ and $R'^{(n)}$ indicate that these are conditional on the level in the ARP after the sth step being greater than low.

Then for n>high, by B.4, compare the value $\bar{Q}_{ARP}(\langle sub(h), n\rangle, \mathfrak{h}_1, \dots, \mathfrak{h}_s)$ of taking actions $\mathfrak{h}_1, \dots, \mathfrak{h}_s$ at state $sub(h)$ in the ARP, with $\bar{Q}(sub(h), \mathfrak{h}_1, \dots, \mathfrak{h}_s)$ of taking them in the real process:

$$\begin{aligned} &|\bar{Q}_{ARP}(\langle sub(h), n\rangle, \mathfrak{h}_1, \dots, \mathfrak{h}_s) - \bar{Q}(sub(h), \mathfrak{h}_1, \dots, \mathfrak{h}_s)| \\ &\qquad < \frac{\epsilon(1-\gamma)}{6s\mathcal{R}} \frac{2s\mathcal{R}}{1-\gamma} + \frac{2\epsilon}{3s(s+1)} \frac{s(s+1)}{2} = \frac{2\epsilon}{3} \end{aligned}$$

where, the first term is the cost of conditions for B.2 not holding, as the cost of straying below low is bounded by $2s\mathcal{R}/(1-\gamma)$. The second term is the cost, from B.4, of the incorrect rewards and transition probabilities.

Since this inequation applies to any set of actions, it applies perforce to a set of actions optimal for either the ARP or the real process. Therefore,

$$|Q^*_{ARP}(\langle sub(h), n\rangle, \mathfrak{h}) - Q^*(sub(h), \mathfrak{h})| < \epsilon.$$

So, as required, with probability 1, $Q_n(sub(h), \mathfrak{h}) \to Q^*(sub(h), \mathfrak{h})$ as $n \to \infty$.

As stated in [207], it should be noted that "changing more than one Q-value on each iteration requires just a minor modification to the ARP such that an action can be taken at any level at which it was executed in the real process, i.e., more than one action can be taken at each level." As long as the stochastic convergence condition in theorem are still satisfied, the proof requires no non-trivial modification. The $Q_n(sub(h), \mathfrak{h})$ values are still optimal for the modified ARP, and this still tends to the real process. In fact, as the proof relies on the ARP estimating rewards and transition functions based on many episodes, this process will just be accelerated by changing more than one Q-value per iteration.

6.2 Learning Feedback in the HMDS

In the HMDS, the system will announce the result in form of a list of top 10 possible diseases, and will then receive a feedback, i.e., the final diagnosis, from the physician. However, it should be noted that the system cannot simply take this diagnosis to calculate the rewards but should have a strong policy to reduce the possibility of errors and biased diagnoses. The HMDS is meant to be used in hospitals, and therefore, is multi-user and receives the feedback from different physicians. Moreover, the system will consider counterfactual learning, which refers to the ability to learn from forgone outcomes, i.e., the outcome of the option(s) that were not chosen. In order to design a mechanism to consider the other options next to the physician's final diagnosis, the probability of selecting the options is estimated with a combination of ε-greedy and softmax rules (see Figure 44).

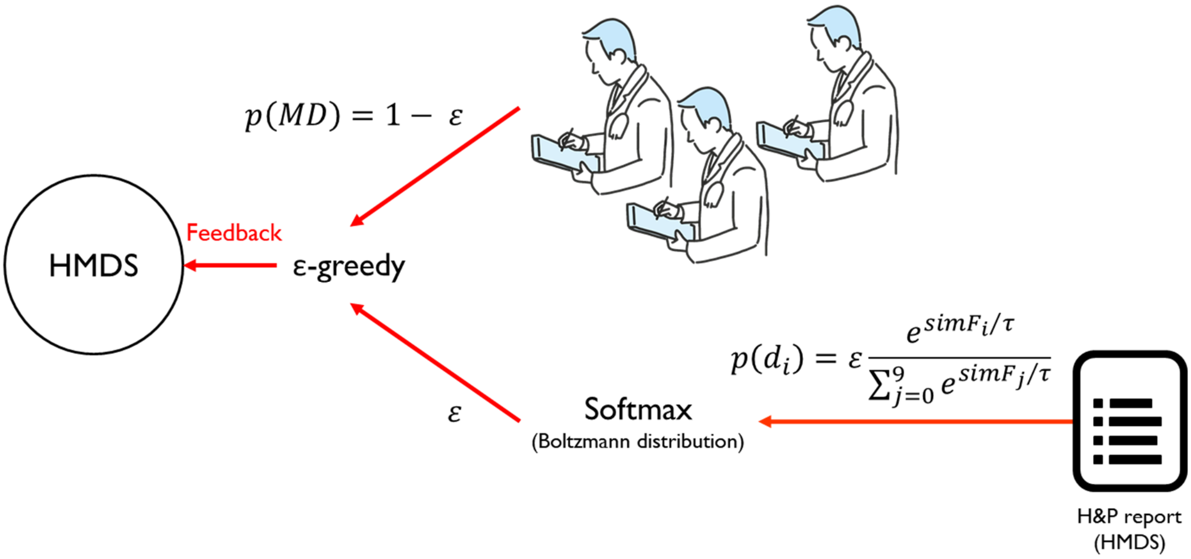

$p(MD) = 1 - \varepsilon$
HMDS
Feedback
ε-greedy
ε
Softmax
(Boltzmann distribution)
$p(d_i) = \varepsilon \frac{e^{simF_i/\tau}}{\sum_{j=0}^{9} e^{simF_j/\tau}}$
H&P report
(HMDS)

Figure 44. Counterfactual learning in the HMDS

Using this method, the physician's diagnosis is given the highest selection probability, i.e., $1 - \varepsilon$, and all the others are ranked and weighted using the softmax rule. For this reason, the most common softmax method, which uses the Gibbs or Boltzmann distribution, is used and have been adapted to the problem as follows:

$$p(d_i) = \varepsilon \frac{e^{simF_i/\tau}}{\sum_{j=0}^{9} e^{simF_j/\tau}} \tag{22}$$

where d_i for $0 \leq i \leq 9$ is a disease in the system's output list, $simF_i$ is the product of the similarity of d_i to the diagnosis request and the frequency of it, and τ is a positive parameter called temperature. Low temperatures cause a greater difference in selection probability of the diseases. ε can be defined in a way that it increases as the number of diagnoses grows. And if τ is defined in a way that it decreases as the time passes, the system will at the beginning mainly concentrate on the physicians' input, however, it tends to trust its own knowledge more as the time passes.

$$\varepsilon = 1 - \frac{1}{number\ of\ diagnoses/c + 1} \tag{23}$$

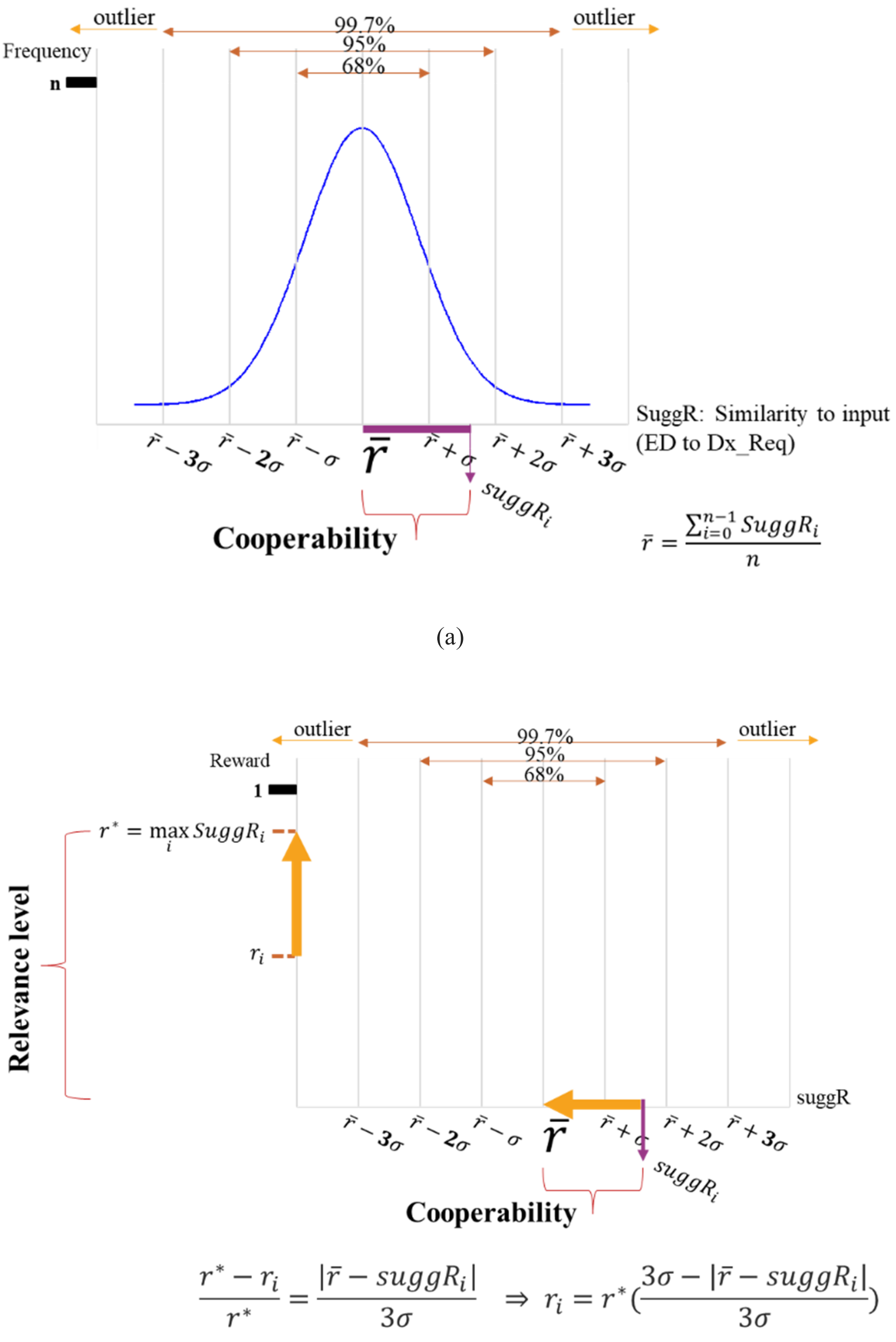
$$\tau = 1 - \varepsilon \tag{24}$$

The system can then use this feedback to update the holon identifier of the DRA acting for the final diagnosis, and to calculate the rewards for the reinforcement learning technique.

6.3 Reward Engineering in the HMDS

The term reward engineering, first coined by Daniel Dewey in [212], refers to the engineering of the agent's environment in order to make the reward assignment more reliable. According to [212, p. 14], in reward engineering "the reinforcement learning agent's goal is not being changed, but the environment is being partially designed so that reward maximization leads to desirable behaviour". For this reason, the human factor may even be removed from the loop and the reward may be assigned via an automatic mechanism [212].

In the HMDS, the holon on the top level will receive the feedback from the environment and then will pass it to its members and this action is repeated until the feedback is announced to all the DSAs in the system. These DSAs then act as the environment of their members (see section 5.1.3) and calculate their rewards. There are different factors to be considered by the heads to calculate the rewards. At the end of each diagnosis process, the heads will receive the identifier of the final diagnosis. Calculating the rewards, the similarity between the identifier of the final diagnosis and the identifier of the participating agents is clearly a decisive factor. This means that the head can check the similarity between the member and the final diagnosis to assign the appropriate reward to it. This is the case because members are involved in the decision made by the head to join the diagnosis process at the very first step. However, there are some other factors to be considered here too. A member of a super-holon is supposed to have good collaboration with the rest of the members. And this should be the case if the problem is relevant. Therefore, calculating the reward, the head should also consider the cooperability and the relevance level both (see Figure 45).



(a)

(b)

Figure 45. Cooperability and relevance level factors in reward engineering

As a result, having the final diagnosis the head will announce this on its blackboard and the members then send back their similarity. This value is called the suggested reward ($suggR_i$). Considering the distribution of these values, statistically those values that are more than three standard deviation far from the mean value are considered as outliers. And of course, the closer they are to the mean the cooperability has been higher.

As mentioned, the second factor to be considered is the relevance level. The value r^* is defined here as the maximum suggested reward. The higher this value is, the more relevant the problem is and therefore incompatibilities should be penalized more. As a result, the difference between the reward and the r^* is directly proportional to the difference between the suggested reward and the average reward and the proportionality constant is r^* over 3σ:

$$\frac{r^* - r_i}{r^*} = \frac{|\bar{r} - suggR_i|}{3\sigma} \tag{25}$$

Therefore,

$$r_i = \begin{cases} 0 & suggR_i \; is \; an \; outlier \\ r^* \left(3\sigma - |\bar{r} - suggR_i|\right)/3\sigma & else \end{cases} \tag{26}$$

As a result, if the suggested reward is very close to the average reward the final reward is also very close to the maximum reward. On the other hand, if the suggested reward is far from the average the final reward would also be relatively low. Moreover, when the problem is relevant to the super-holon, the r^* will be greater and therefore incompatibilities could be penalized more (see Figure 46). To have a uniform reward metrics, however, the reward value can be adjusted to

$$r_i = \begin{cases} 0 & suggR_i \; is \; an \; outlier \\ 1 - r^* \left(1 + \dfrac{3\sigma - |\bar{r} - suggR_i|}{3\sigma}\right) & else \end{cases} \tag{27}$$

so that Q-values are updated comparably.

It should be noted that, a super-holon that is participating in an ongoing diagnosis process will use the reward function to calculate the rewards. However, if a super-holon, which had originally not participated in the diagnosis process, realizes that the final diagnosis matches one of its members, it will assign the reward -1 (penalty) to the DRA representing the final diagnosis. This will cause the Q-value to deviate from the mean value, which will force the member to start exploring. This exploration is here essential, since it is clear that the DRA is in a wrong place and this is the reason for not receiving the diagnosis request in order to react. To this end, DSAs that have not participated in the ongoing diagnosis process and do not include the final diagnosis as one of their members will, too, pass the final diagnosis, however, only to their DSA members.

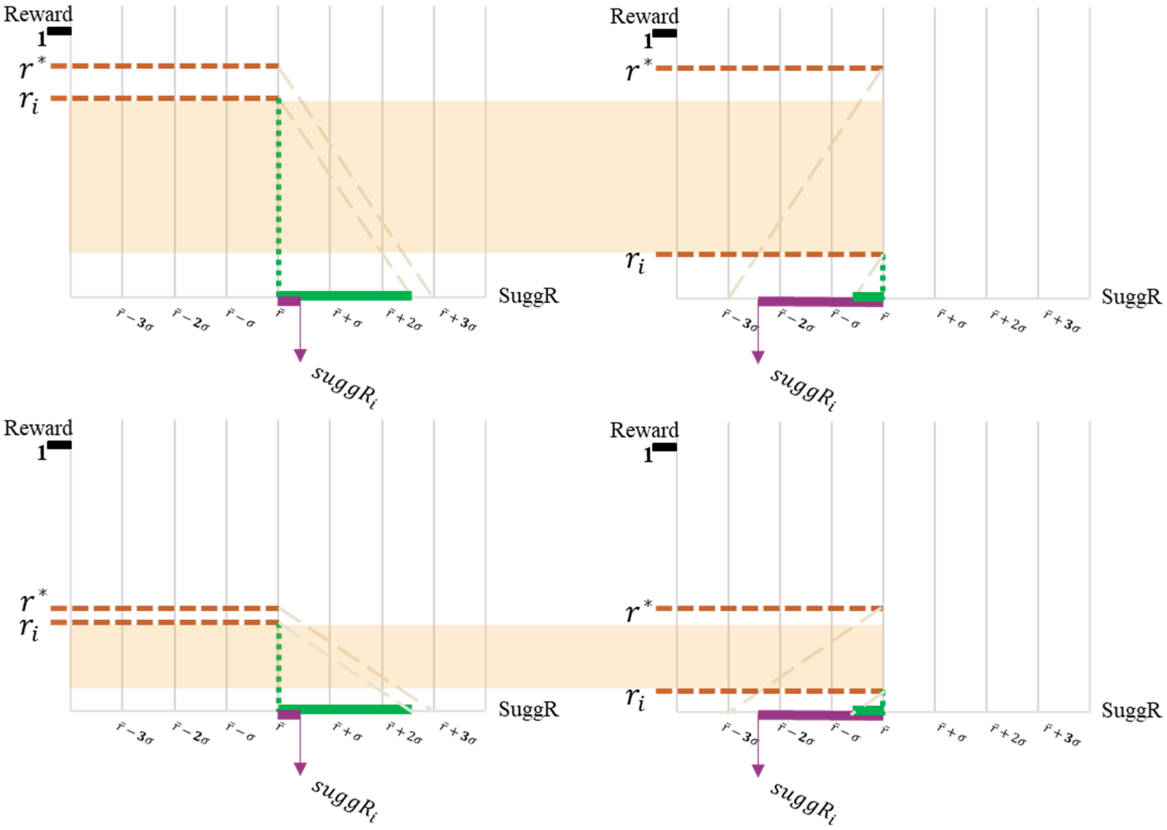


Figure 46. The calculation of reward in the HMDS

Chapter 7
SYSTEM SIMULATION

In order to simulate the HMDS the first step is to choose the right agent-oriented programming language and the suitable platform. Based on the characteristics of the desired software system, GAMA platform[37] [21] has been used in this project to build the simulations[38]. GAMA is a modeling and simulation development environment for building spatially explicit agent-based simulations using GAML (GAma Modeling Language), a high-level and intuitive agent-based language. Simplicity and support for holonic concept were the main reasons to choose this simulation platform. In addition, a comprehensive comparison between the agent-oriented programming languages and platforms has been conducted in [213], according to which GAMA is associated with highest rates of responses to the evaluation criteria generally linked the agent-oriented platforms (see section 7.1). This chapter provides a brief introduction to GAML (see section 7.2) and shows how the HMDS has been simulated using this platform (see section 7.3 and 7.3). Chapter 8 will then demonstrate how the simulated system behaves and uses a number of real medical cases as benchmarks to assess the functionality of the system.

[37] GAMA is developed by several teams under the umbrella of the IRD/UPMC international research unit UMMISCO [243].

[38] Even though for simulation development GAMA platform has been suggested and used in this study, for application development Janus platform [217], which provides a comprehensive set of features to develop, run, display and monitor multiagent-based and holonic applications, is highly recommended.

7.1 The Agent-Oriented Programming Languages and Platforms

As mentioned already, a comprehensive comparison between the agent-oriented programming languages and platforms has been conducted in [213], according to which GAMA is associated with highest rates of responses to the evaluation criteria generally linked the agent-oriented platforms (see Table 15 extracted from [213, p. 1083]). Support for inter-agent communication, support for holonic MASs, support for organizational modeling approaches, support for agent environment, graphic support for development and implementation, and support for the management of the MAS were the most important criteria in choosing the right platform for the simulation of the HMDS and GAMA has received high scores for all these criteria.

Table 15. Scores to each of the studied agent programming languages or frameworks

Language Criteria	AgentSpeak	GAML	Jade	NetLogo	SARL	Max grade
Run-time Platform	JASON [214]	GAMA [21]	Jade [215]	NetLogo [216]	Janus [217]	
Fields of application	3	3	3	3	3	/4
Inter-agent communication	1	2	1	1	3	/3
Code extensibility	4	3	4	2	4	/4
Support hierarchical or holonic multiagent systems	1	1	2	1	2	/2
Support for organizational modeling approaches	2	2	1	1	2	/2
Support for agent environment	1	3	1	2	1	/3
Facilitating the transition between design and implementation	2	2	2	2	0	/3
Graphic support for development and implementation	3	3	2	0	2	/3
Documentation	3	3	2	3	3	/4
Facilitating the learning of the tool	3	2	2	3	3	/4
Deployment	2	2	3	3	2	/4
Debugging tools	0	0	1	0	1	/1
Support for the management of the MAS	2	3	1	0	2	/3
Total (Σ)	27	29	25	21	28	/40

7.2 A Brief Introduction to GAML

According to online documentation of GAMA Platform [218], GAML is an agent-oriented language dedicated to the definition of agent-based simulations, which takes its roots in object-oriented languages such as Java and Smalltalk, but extends the object-oriented programming approach with powerful concepts such as skills, declarative definitions, or agent migration to support a better expressivity in models.

As stated in [218], despite some obvious similarities with available agent based modeling languages, e.g., NetLogo, GAML exhibits a number of distinguishable characteristics: (1) it enriches the traditional representation of agents with modern computing notions like inheritance, type safety, or multi-level agency, (2) it provides the possibility to use different behavioral architectures for programming agents, and (3) it extends the agent-based paradigm to eliminate the boundaries between the domain of a model, which in Agent-Based Modeling (ABM) is represented with agents, and the experimental processes surrounding its simulations, which are usually not represented with agents, including, for example, visualization processes.

The definition of the concepts on which GAMA (and GAML) are based today have been presented in [219]. The two most interesting consequences of this orientation are:

- Since simulations, or experiments, are represented by agents, GAMA is bound to support high-level model compositionality, i.e., the definition of models that can use other models as inner agents, leveraging multi-modeling or multi-paradigm modeling as particular cases of composition.
- The visualization of models can be expressed by models of visualization, composed of agents entirely dedicated to visually represent other agents, allowing for a clear separation of concerns between a simulation and its representation and, hence the possibility to play with multiple representations of the same model at once. [218]

7.2.1 Lexical semantics of GAML

The vocabulary of GAML is described as follows:

1. The role of GAML is to support modelers in writing models, which are specifications of simulations that can be executed and controlled during experiments, themselves specified by experiment plans.

2. The agent-oriented modeling paradigm means that everything "active" (entities of a model, systems, processes, activities, like simulations and experiments) can be represented in GAML as an agent (which can be thought of as a computational component owning its own data and executing its own behavior, alone or in interaction with other agents).

3. Like in the object-oriented paradigm, where the notion of class is used to supply a specification for objects, agents in GAML are specified by their species, which provide them with a set of attributes (what they know), actions (what they can do), behaviors (what they actually do) and also specifies properties of their population, for instance its topology (how they are connected) or schedule (in which order and when they should execute).

4. Any species can be nested in another species (called its macro-species), in which case the populations of its instances will imperatively be hosted by an instance of this macro-species. A species can also inherit its properties from another species (called its parent species), creating a relationship similar to specialization in object-oriented design. In addition to this, species can be constructed in a compositional way with the notion of skills, bundles of attributes and actions that can be shared between different species and inherited by their children.

5. Given that all agents are specified by a species, simulations and experiments are then instances of two species which are, respectively, called model and experiment plan. Think of them as "specialized" categories of species.

6. The relationships between species, models and experiment plans are codified in the meta-model of GAML in the form of a framework composed of three abstract species respectively called agent (direct or indirect parent of all species), model (parent of all species that define a model) and experiment (parent of all species that define an experiment plan). In this meta-model, instances of the children of agent know the instance of the child of model in which they are hosted as their world, while the instance of experiment plan identifies the same agent as one of the simulations it is in charge of. The following diagram summarizes this framework [218]:

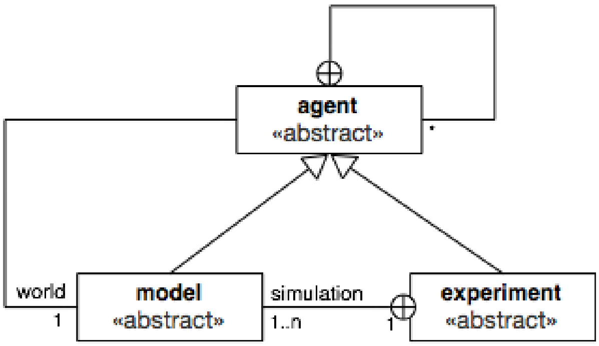


Figure 47. The relationships between the abstract species in the meta-model of GAML

As a result, writing a model in GAML involves defining a species which inherits from model, in which other species, inheriting (directly or indirectly) from agent and representing the entities that populate this model, will be nested. The model itself is nested in one or several experiment plans among which a user will be able to choose which experiment he/she wants to execute [218] (Figure 48 [218]).

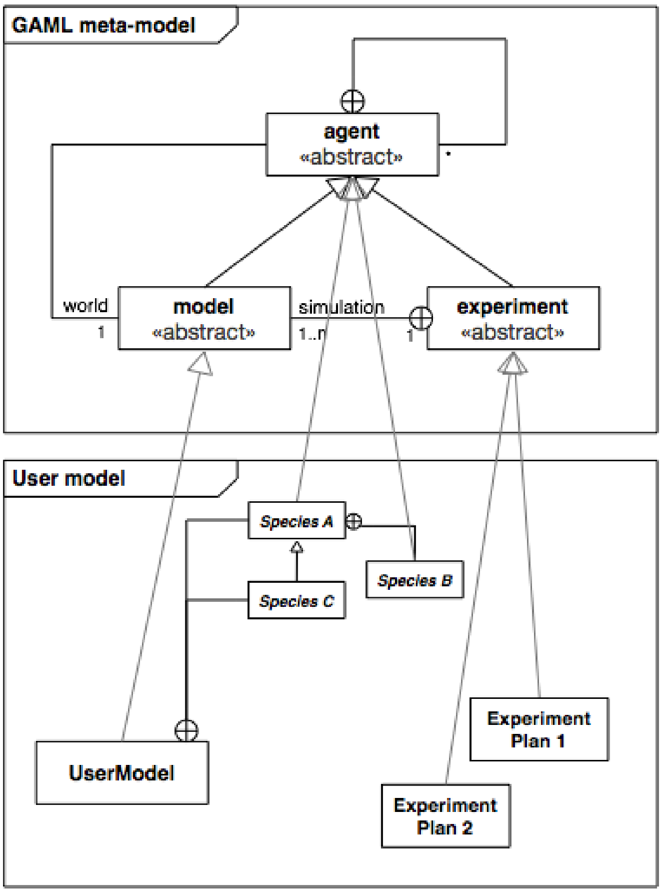


Figure 48. The mapping between the GAML meta-model and user model

Figure 49 shows the abstract species of GAML in the HMDS model and Figure 50 illustrates the mapping between the GAML meta-model and the HMDS model.

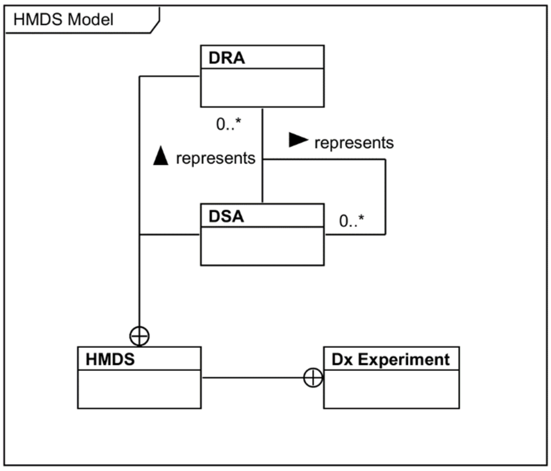


Figure 49. The abstract species of GAML in the HMDS model

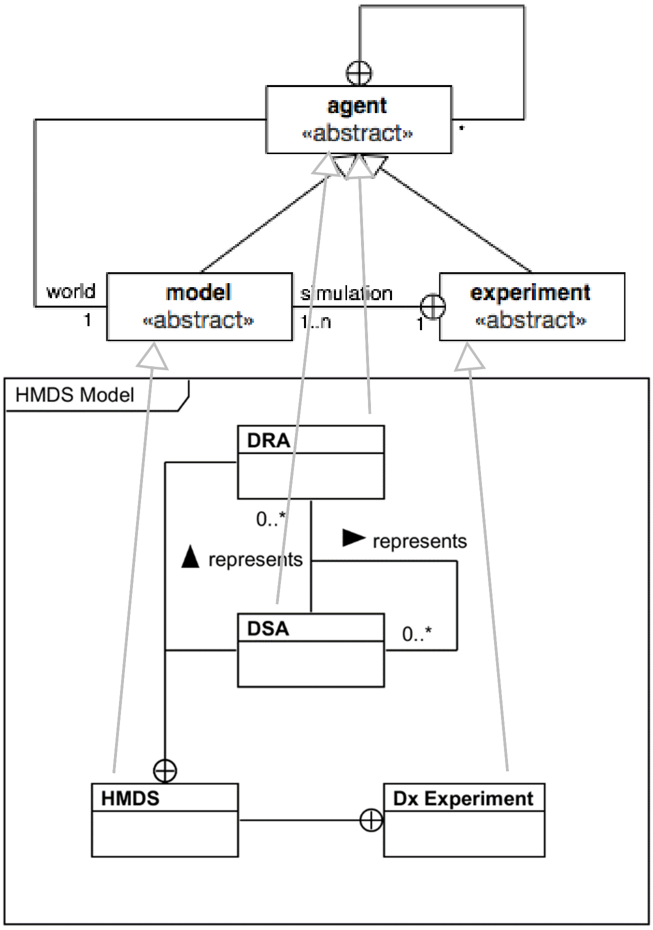


Figure 50. The mapping between the GAML meta-model and the HMDS model

7.2.2 Organization of a model

According to [218], “defining a model in GAML amounts to defining a model species, which later allows to instantiate a model agent (aka a simulation), which may or may not contain micro-species, and which can be flanked by experiment plans in order to be simulated”. This conceptual structure is respected in the definition of model files that follows a similar pattern:

1. Definition of the global species, preceded by a header, in order to represent the model species
2. Definition of the different micro-species (either nested inside the global species or at the same level)
3. Definition of the different experiment plans that target this model [218]

7.2.2.1 Model Header (model species)

The header of a model file begins with the declaration of the name of the model, which contrarily to other statements, does not end with a semi-colon.

```
model HMDS
```

7.2.2.2 Species declarations

The model header is followed by the declaration of the different species of agents that populate the model. It should be noted that the special species global is the world species in which all the global attributes, actions, and behaviors will be declared. This species does not have a name as it is unique in its model.

```
model HMDS

global {
       // The definition of global attributes, actions, and behaviors
}
// Agents

// DSA
species DSA {…}

// DRA
species DRA {…}
```

7.2.2.3 Experiment declarations

Experiments are usually declared at the end of the model file, starting with the keyword experiment. They contain the simulation parameters, and the definition of the output. A model may contain one to multiple experiments as needed.

```
model HMDS

global {
	// definition of global attributes, actions, behaviors
}

// Agents
//DSA
species DSA {…}
//DRA
species DRA {…}
//input and output
experiment Diagnosis type: gui {
	// definition of parameters (inputs)
	// definition of output
	output {...}
}
```

7.3 A GAML model of the HMDS

Following GAML model shows a simple representation of the HMDS. It should be noted that a behavior, or “reflex”, is an action which is called automatically at each time step by an agent, and “init” is a special reflex, that occurs only when the agent is created.

```
/*
* Name: Holonic Medical Diagnosis System (HMDS)
* Author: Zohreh Akbari
*/
model HMDS

global {
	// Definition of global attributes
	[…]
	// Diagnosis request signs and symptoms: float variables
	[…]
	// Input list of integers that indicate any changes in the input parameters
```

```
        […]

        // The list of diseases: string variables
        […]

        // The list of symptoms: string variables
        […]

        // The list of medical tests: string variables
        […]

        // The medical tests for each of the diseases: a list of lists of integers
        […]

        // *----------*----------* Diseases / DRAs *----------*----------*

        // DRA IDs: list of float variables (DDP)
        […]

        // *----------*----------* Initialization *----------*----------*

        init {
                // The initialization and creation of DRAs
                […]

                // The initialization and creation of the highest DSA
                // At the beginning all the DRAs will become a member of the highest DSA
                […]

        }
}

// *----------*----------* Agents *----------*----------*

// DSA

species DSA {

        // *----------*----------* Blackboard *----------*----------*

        // All the attributes of a DRA and the variables needed to conduct its behavior.
        […]

        // *----------*----------* Clustering *----------*----------*

        init clustering {…}

        // *----------*----------* Reflexes *----------*----------*

        reflex buttonTrue {…}

        reflex inputchanged {…}

        reflex joinDiagnosis when: run {…}

        reflex allMem_haveResponded when: (!all_Members_haveResponded) {…}

        reflex DDxList when: (all_Members_haveResponded and joinDx) {…}

        reflex allMem_SuggestedReward when: (finalDxAnnounced_self and
        !all_Members_haveSent_SuggestedReward and !membersRewarded) {…}

        // Calculate the suggested reward
        reflex CalSuggestedReward when: (all_Members_haveSent_SuggestedReward and
        !membersRewarded) {…}

        // Calculate the initial Q-Value
        reflex CalInitQValue when: newHead {…}

        // Calculate the Q-Value
        reflex CalQValue when: newReward {…}

        // Check the necessity of exploration
        reflex checkExpNecessity when: QV_updated {…}

        // Guided exploration
        reflex gExp when:(gExpMergeReq or gExpReqResult) {…}

        // Random exploration
        reflex rExp when:(rExpMergeReq or memMedReq) {…}

        // Create new DSA
        reflex createNewDSA when:(newDSA_Req) {…}

        // Receive update report
        reflex recUpdateRep when:(updateRep) {…}

        // Accept leave request from a member
        reflex accLeave when: leaveReq {…}

        // *----------*----------* Aspect *----------*----------*

        // Express how DRAs will be drawn
        aspect base {…}

        // Express how output will be displayed
        aspect output {…}
}

//DRA

species DRA {

        // *----------*----------* Blackboard *----------*----------*

        // All the attributes of a DRA and the variables needed to conduct its behavior
        […]

        // *----------*----------* Reflexes *----------*----------*

        // Calculate the similarity to the input and send it to the head
        reflex CalSimilarityToInput when: joinDx_Head and !finalDxAnnounced_head {…}

        // Calculate the similarity to the final diagnosis and send it to the head
        reflex CalSimilarityToFinalDx when: finalDxAnnounced_head and joinDx_Head {…}

        // Calculate the initial Q-Value
        reflex CalInitQValue when: newHead {…}

        // Calculate the Q-Value
        reflex CalQValue when: newReward {…}

        // Check the necessity of Exploration
        reflex checkExpNecessity when: QV_updated {…}

        // Guided exploration
        reflex gExp when:(gExpReqResult) {…}

        // Random exploration
        reflex rExp when:(rExpMergeReq) {…}


        // Leave current super-holon through coordination with the head
        reflex leave when: leave {…}

        // *----------*----------* Aspect *----------*----------*

        // Express how DRAs will be drawn
        aspect base {…}
}

//input and output
experiment Diagnosis type: gui {

        // Input parameters
        […]

        // Outputs
        output {

                // The holarchy
                display HMDS {
                        species DSA aspect: base;
                        species DRA aspect: base;
                }

                // The ordered DDx list, the suggested signs and symptoms to be checked,
                // and the suggested medical tests
                display output {
                        species DSA aspect: output;
                }

                // The Q-Value chart
                display DRAs_QValues {…}
        }
}
```

7.4 An example of the System Simulation

This section covers a simple simulation of the HMDS with a limited number of diseases (twenty diseases as listed in Table 16). The information regarding the diseases has been gathered from Mayo Clinic [220] website, which provides detailed information about the diseases and their related information.

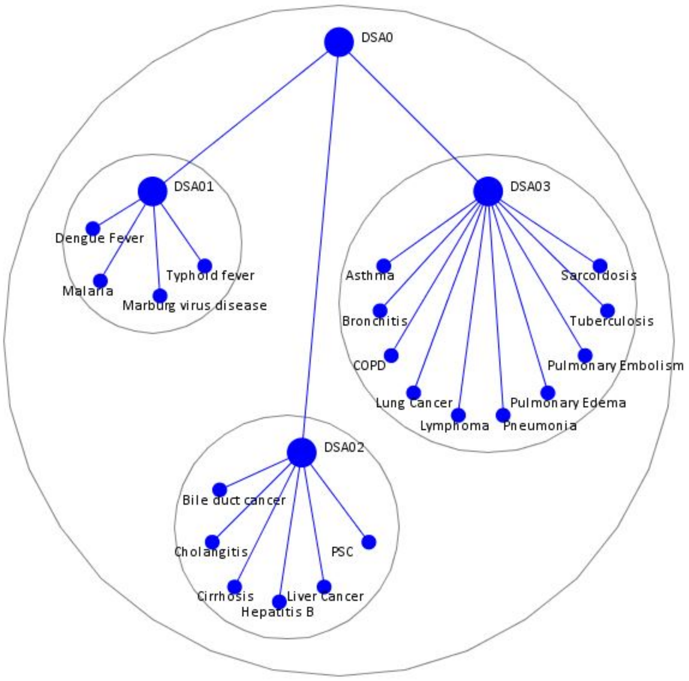


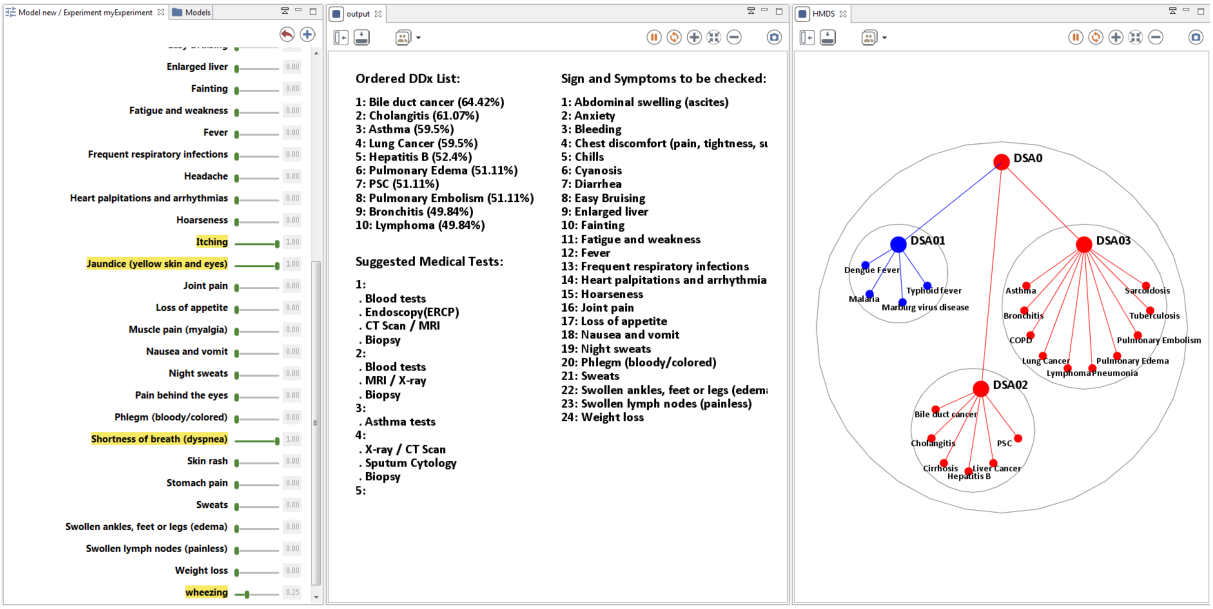
Figure 51. The holarchy of the simulated HMDS with twenty diseases

Figure 51 illustrates the initial holarchy of the simulated system, i.e., the result of the clustering that has been done based on the signs and the symptoms of the given diseases, i.e., the distances between the DDP or the holon identifier of each of the DRAs. Each of the lowest DSAs has in fact grouped the diseases that should be considered in the differential diagnosis of each of its members. For complete list of diseases that should be covered in the DDx of each of the diseases please refer to [17].

Table 16. The diseases and the signs and symptoms covered by the simulated system

	Asthma	Bile duct cancer	Bronchitis	Cholangitis	Cirrhosis	COPD	Dengue Fever	Hepatitis B	Liver Cancer	Lung Cancer	Lymphoma	Malaria	Marburg virus disease	Pneumonia	Primary Sclerosing Cholangitis	Pulmonary Edema	Pulmonary Embolism	Pulmonary Tuberculosis	Sarcoidosis	Typhoid fever
Abdominal pain	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
Abdominal pain (upper part)	0.0	1.0	0.0	1.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0
Abdominal swelling (ascites)	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
Anxiety	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0	0.0
Bleeding	0.0	0.0	0.0	0.0	1.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Chest discomfort (pain, tightness, suffocating)	1.0	0.0	1.0	0.0	0.0	1.0	0.0	0.0	0.0	1.0	0.0	0.0	1.0	1.0	0.0	1.0	1.0	1.0	1.0	0.0
Chills	0.0	0.0	1.0	1.0	0.0	0.0	1.0	0.0	0.0	0.0	1.0	1.0	1.0	1.0	1.0	0.0	0.0	1.0	0.0	1.0
Clay-colored stools	0.0	1.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0
Conjunctivitis (red eyes)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Constipation	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
Cough	1.0	0.0	1.0	0.0	0.0	1.0	1.0	0.0	0.0	1.0	0.0	0.0	1.0	1.0	0.0	1.0	1.0	1.0	1.0	1.0
Cyanosis	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
Dark urine	0.0	1.0	0.0	1.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0
Diarrhea	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	1.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0	1.0
Easy Bruising	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Enlarged liver	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0
Fainting	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
Fatigue and weakness	0.0	1.0	1.0	0.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.0	0.0	1.0	1.0	1.0
Fever	0.0	1.0	1.0	1.0	0.0	0.0	1.0	1.0	0.0	0.0	1.0	1.0	1.0	1.0	1.0	0.0	0.0	1.0	1.0	1.0
Frequent respiratory infections	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Headache	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
Heart palpitations and arrhythmias	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	1.0	1.0	0.0	0.0	0.0
Hoarseness	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Itching	0.0	1.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0
Jaundice (yellow skin and eyes)	0.0	1.0	0.0	1.0	1.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0
Joint pain	0.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Loss of appetite	0.0	0.0	0.0	0.0	1.0	0.0	0.0	1.0	1.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	1.0
Muscle pain (myalgia)	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
Nausea and vomit	0.0	1.0	0.0	1.0	1.0	0.0	1.0	1.0	1.0	0.0	0.0	1.0	1.0	1.0	1.0	0.0	0.0	0.0	0.0	1.0
Night sweats	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	1.0	0.0	0.0	1.0	0.0	0.0
Pain behind the eyes	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Phlegm (bloody/colored)	0.0	0.0	1.0	0.0	0.0	1.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	1.0	0.0	1.0	1.0	1.0	0.0	0.0
Shortness of breath (dyspnea)	1.0	0.0	1.0	0.0	0.0	1.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	1.0	0.0	1.0	1.0	0.0	1.0	0.0
Skin rash	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
Stomach pain	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Sweats	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	1.0	0.0	0.0	1.0	0.0	0.0	1.0
Swollen ankles, feet or legs (edema)	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
Swollen lymph nodes (painless)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0
Weight loss	0.0	1.0	0.0	0.0	1.0	1.0	0.0	0.0	1.0	1.0	1.0	0.0	1.0	0.0	1.0	0.0	0.0	1.0	1.0	1.0
Wheezing	1.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0	1.0	0.0

The user interface of a simulated system that covers the given diseases is shown in Figure 52. The signs and symptoms, i.e., the diagnosis request, can be given using the range inputs on the left section. The right section shows the actual holarchy. For each diagnosis request those holons that have been activated to participate in the diagnosis process will be highlighted. The middle section includes the ordered DDx list, suggested medical tests, and further signs and symptoms to be checked. This small system has been used in chapter 8 in section 8.1 to demonstrate the behavior of the system and to assess its diagnosis abilities.



Enlarged liver	0.00
Fainting	0.00
Fatigue and weakness	0.00
Fever	0.00
Frequent respiratory infections	0.00
Headache	0.00
Heart palpitations and arrhythmias	0.00
Hoarseness	0.00
Itching	1.00
Jaundice (yellow skin and eyes)	1.00
Joint pain	0.00
Loss of appetite	0.00
Muscle pain (myalgia)	0.00
Nausea and vomit	0.00
Night sweats	0.00
Pain behind the eyes	0.00
Phlegm (bloody/colored)	0.00
Shortness of breath (dyspnea)	1.00
Skin rash	0.00
Stomach pain	0.00
Sweats	0.00
Swollen ankles, feet or legs (edema)	0.00
Swollen lymph nodes (painless)	0.00
Weight loss	0.00
wheezing	0.25

```
Ordered DDx List:

1: Bile duct cancer (64.42%)
2: Cholangitis (61.07%)
3: Asthma (59.5%)
4: Lung Cancer (59.5%)
5: Hepatitis B (52.4%)
6: Pulmonary Edema (51.11%)
7: PSC (51.11%)
8: Pulmonary Embolism (51.11%)
9: Bronchitis (49.84%)
10: Lymphoma (49.84%)


Suggested Medical Tests:

1:
 . Blood tests
 . Endoscopy(ERCP)
 . CT Scan / MRI
 . Biopsy
2:
 . Blood tests
 . MRI / X-ray
 . Biopsy
3:
 . Asthma tests
4:
 . X-ray / CT Scan
 . Sputum Cytology
 . Biopsy
5:
```

```
Sign and Symptoms to be checked:

1: Abdominal swelling (ascites)
2: Anxiety
3: Bleeding
4: Chest discomfort (pain, tightness, su
5: Chills
6: Cyanosis
7: Diarrhea
8: Easy Bruising
9: Enlarged liver
10: Fainting
11: Fatigue and weakness
12: Fever
13: Frequent respiratory infections
14: Heart palpitations and arrhythmia
15: Hoarseness
16: Joint pain
17: Loss of appetite
18: Nausea and vomit
19: Night sweats
20: Phlegm (bloody/colored)
21: Sweats
22: Swollen ankles, feet or legs (edema
23: Swollen lymph nodes (painless)
24: Weight loss
```

Figure 52. The user interface of the simulated HMDS with 20 DRAs

It should be noted that the system builds its holarchy based on the given diseases at the initialization stage. To this end, each DSA will perform clustering at its initialization stage (see the GAML model provided in section 7.3). Thus, it is always possible to provide the system with more diseases and build a larger system for simulation (see section 8.2.1).

Chapter 8
SYSTEM ASSESSMENT

This chapter includes the assessment simulations of the HMDS. The tests monitor the general behavior of the system in performing the H&P (section 8.1). Additionally, learning abilities of the system is examined by providing the system with appropriate inputs and evaluating the corresponding outputs (section 8.2).

8.1 The Assessment of the Diagnosis Abilities

Case 1: Lung Cancer

The first example is based on an actual H&P report, which is provided in [221] for study purposes. This report is the final H&P report, so having the CC, the relevant signs and symptoms have been checked and the necessary medical test and the final DDx list was given. In this simulation, the CC is given as the diagnosis request and the system reaction is monitored using the actual report as a benchmark. In order to review the original H&P report please refer to appendix B.

Briefly, the H&P report includes:

Chief Complaint (CC): Shortness of Breath (SOB)

History of Present Illness (HPI): chest pain, chills, cough, fever, history of breathing troubles / asthma / pneumonia / TB exposure, history of cancer in family, history with tobacco, night sweats, productive cough, vomit, weight loss, wheezing

Review of Systems (ROS): routine with focus on anxiety, fainting, fatigue and weakness, heart palpitations and arrhythmias, change in skin color, changes in appetite

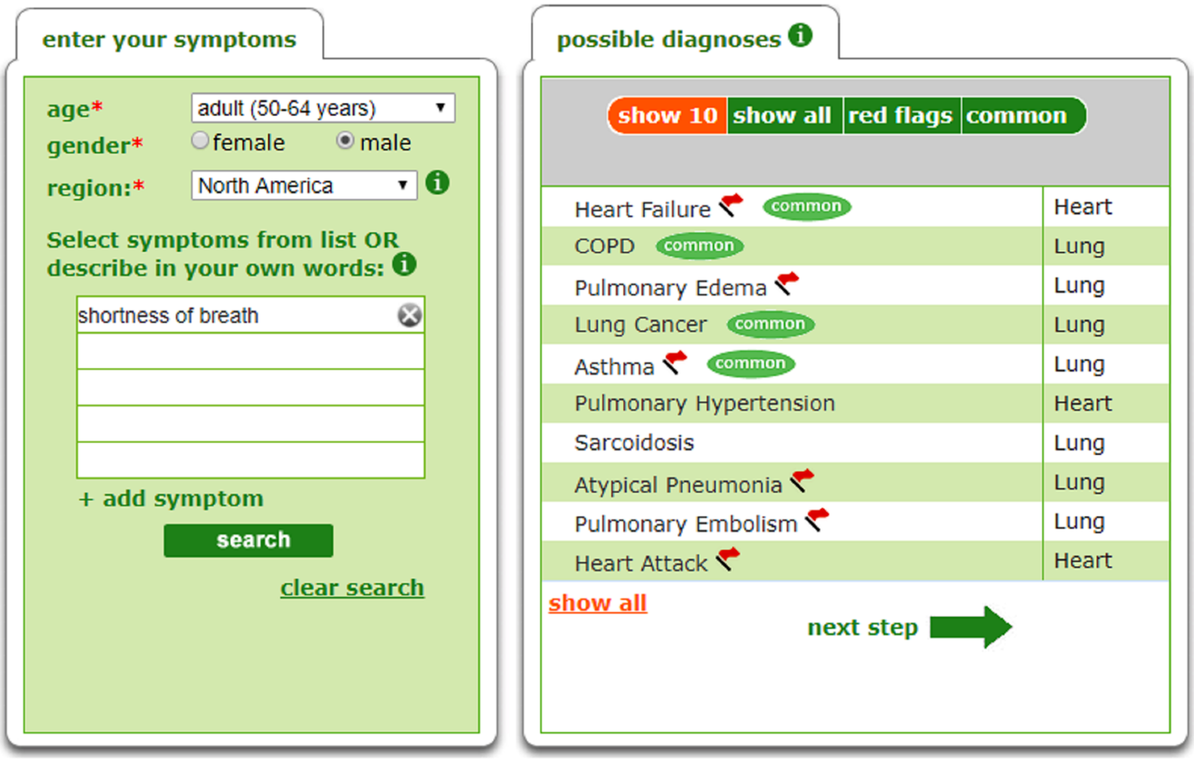
Physical Examination (PE): normal with focus on cyanosis, edema, swollen lymph nodes

Diagnostic Tests: blood test, CT scan

Assessment and Plan: (1) asthma (asthma tests), (2) lung cancer (X-ray / CT scan, biopsy), (3) pneumonia (blood test), (4) sarcoidosis (blood test), (5) tuberculosis (PPD: Purified Protein Derivative skin test for tuberculosis)

Having this information, it is possible to provide the simulated version of the HMDS with the CC and monitor its output. Most specifically, in this simulation the ability of the system in guiding the H&P is to be checked. For this reason, it is suggested to control how the signs and symptoms that are announced by the system to be checked match the ones being mentioned in the original H&P report. Same input is also given to the Isabel in order to demonstrate the actual aim of the HMDS and to once again emphasize its difference with the state of the art of the MDSs. However, it should be noted that this comparison does not intend to show which system is better in diagnosis, but to show that they are designed for different reasons.

Figure 53 shows how Isabel reacts if it is solely provided with the CC, i.e., the SOB. As demonstrated, the output includes lung cancer as the fourth diagnosis, and the system does not suggest any additional symptoms to be questioned or signs to be checked. Supposing that a doctor has performed the H&P step and giving all the signs and symptoms found in this stage to the system, the first suggestion of the system will be lung cancer (Figure 54). It should be noted that as mentioned before, in this scenario the Isabel is actually used to obtain a second opinion, as the H&P is already done by the physician.



age*: adult (50-64 years)

gender*: ○ female ● male

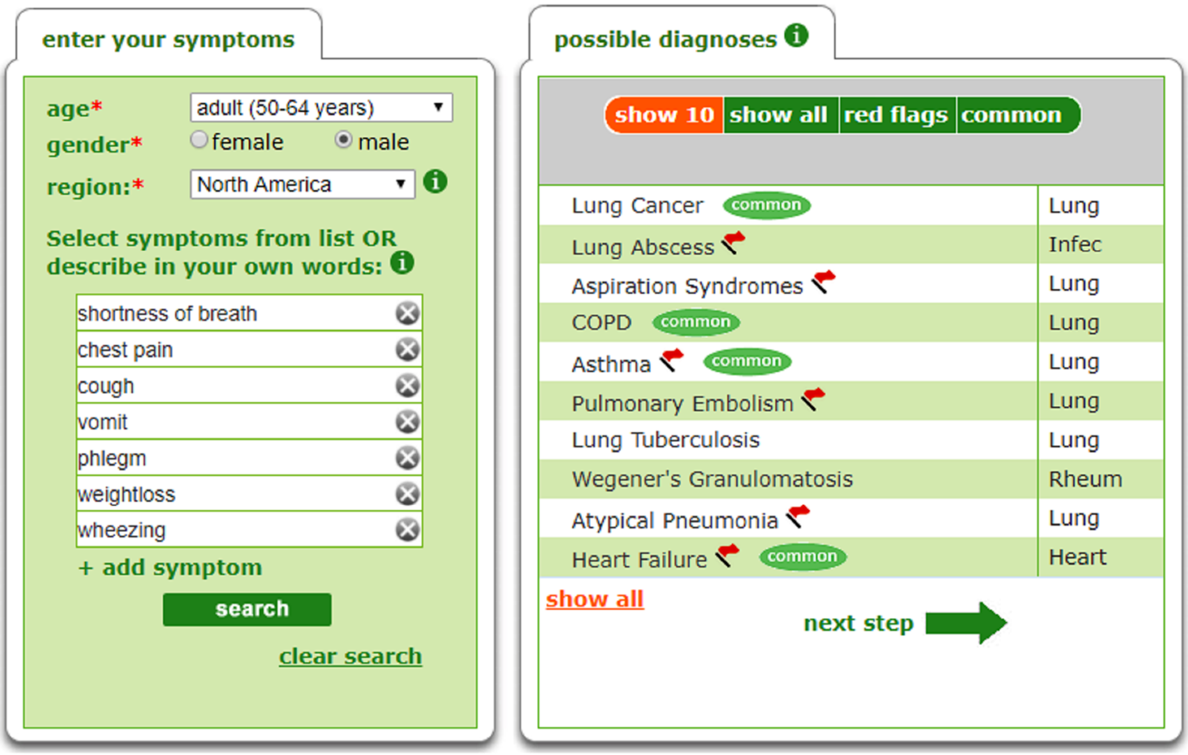
region:*: North America

Select symptoms from list OR describe in your own words:

shortness of breath

Heart Failure 🚩 common	Heart
COPD common	Lung
Pulmonary Edema 🚩	Lung
Lung Cancer common	Lung
Asthma 🚩 common	Lung
Pulmonary Hypertension	Heart
Sarcoidosis	Lung
Atypical Pneumonia 🚩	Lung
Pulmonary Embolism 🚩	Lung
Heart Attack 🚩	Heart

Figure 53. Isabel output for shortness of breath



age*: adult (50-64 years)

gender*: ○ female ● male

region:*: North America

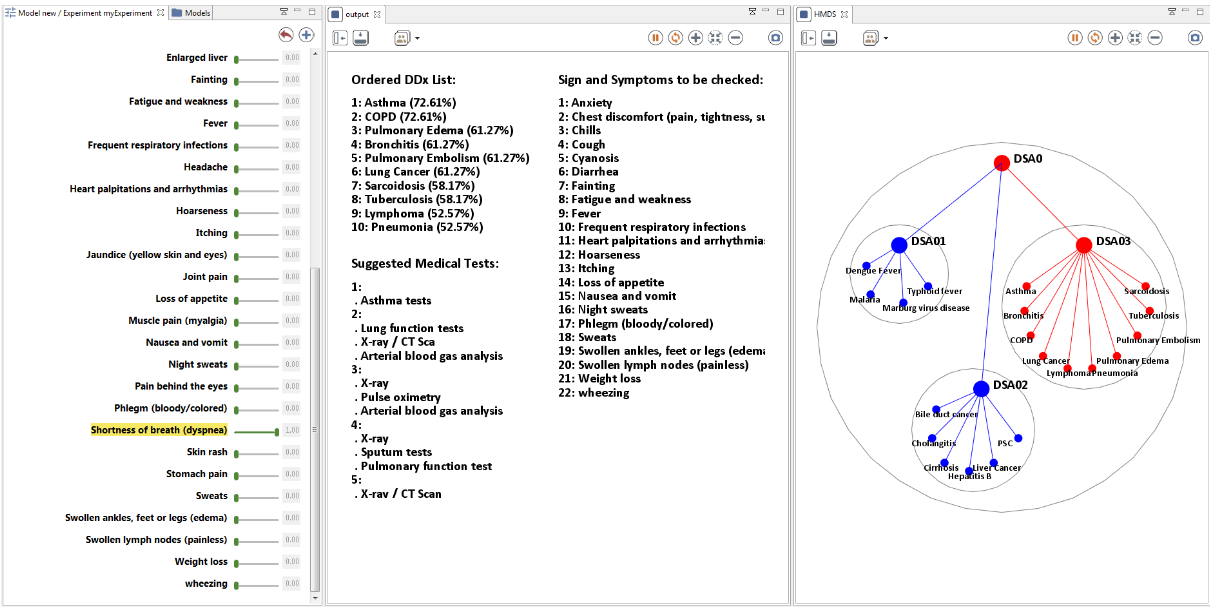
Select symptoms from list OR describe in your own words:

shortness of breath
chest pain
cough
vomit
phlegm
weightloss
wheezing

Lung Cancer common	Lung
Lung Abscess 🚩	Infec
Aspiration Syndromes 🚩	Lung
COPD common	Lung
Asthma 🚩 common	Lung
Pulmonary Embolism 🚩	Lung
Lung Tuberculosis	Lung
Wegener's Granulomatosis	Rheum
Atypical Pneumonia 🚩	Lung
Heart Failure 🚩 common	Heart

Figure 54. Isabel output based on already performed H&P

Providing the HMDS with the shortness of breath as the CC, the DSA of the pulmonary diseases will be activated and the initial DDx given by the system is: (1) asthma, (2) COPD, (3) pulmonary edema, (4) bronchitis, (5) pulmonary embolism, (6) lung cancer, (7) sarcoidosis, (8) tuberculosis, (9) lymphoma, and (10) pneumonia. It should be noted that in reality the initial DDx list cannot be found on an H&P report, since this is actually something the physician would have in mind, according to which s(he) will start checking the signs and symptoms to improve the list. The suggested signs and symptoms to be checked include: anxiety, chest discomfort, chills, cough, cyanosis, diarrhea, fainting, fatigue and weakness, fever, presence of frequent respiratory infections, heart palpitations and arrhythmias, hoarseness, itching, loss of appetite, nausea and vomit, night sweats, phlegm (bloody/colored), sweats, edema, swollen lymph nodes, weight loss, wheezing. These signs and symptoms very much match the ones mentioned in HPI, ROS and PE sections of the original H&P report (Figure 55).



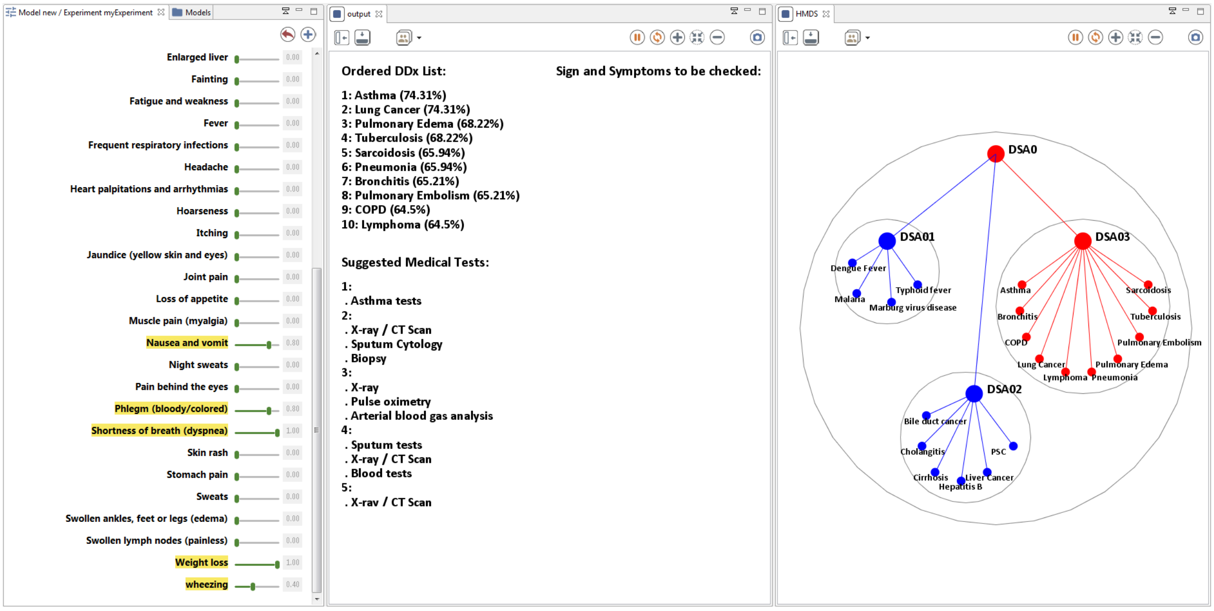
Enlarged liver	0.00
Fainting	0.00
Fatigue and weakness	0.00
Fever	0.00
Frequent respiratory infections	0.00
Headache	0.00
Heart palpitations and arrhythmias	0.00
Hoarseness	0.00
Itching	0.00
Jaundice (yellow skin and eyes)	0.00
Joint pain	0.00
Loss of appetite	0.00
Muscle pain (myalgia)	0.00
Nausea and vomit	0.00
Night sweats	0.00
Pain behind the eyes	0.00
Phlegm (bloody/colored)	0.00
Shortness of breath (dyspnea)	1.00
Skin rash	0.00
Stomach pain	0.00
Sweats	0.00
Swollen ankles, feet or legs (edema)	0.00
Swollen lymph nodes (painless)	0.00
Weight loss	0.00
wheezing	0.00

```
Ordered DDx List:

1: Asthma (72.61%)
2: COPD (72.61%)
3: Pulmonary Edema (61.27%)
4: Bronchitis (61.27%)
5: Pulmonary Embolism (61.27%)
6: Lung Cancer (61.27%)
7: Sarcoidosis (58.17%)
8: Tuberculosis (58.17%)
9: Lymphoma (52.57%)
10: Pneumonia (52.57%)

Suggested Medical Tests:

1:
 . Asthma tests
2:
 . Lung function tests
 . X-ray / CT Sca
 . Arterial blood gas analysis
3:
 . X-ray
 . Pulse oximetry
 . Arterial blood gas analysis
4:
 . X-ray
 . Sputum tests
 . Pulmonary function test
5:
 . X-rav / CT Scan
```

```
Sign and Symptoms to be checked:

1: Anxiety
2: Chest discomfort (pain, tightness, su
3: Chills
4: Cough
5: Cyanosis
6: Diarrhea
7: Fainting
8: Fatigue and weakness
9: Fever
10: Frequent respiratory infections
11: Heart palpitations and arrhythmia:
12: Hoarseness
13: Itching
14: Loss of appetite
15: Nausea and vomit
16: Night sweats
17: Phlegm (bloody/colored)
18: Sweats
19: Swollen ankles, feet or legs (edema
20: Swollen lymph nodes (painless)
21: Weight loss
22: wheezing
```

Figure 55. The output of the simulated HMDS for shortness of breath

After entering the value of these signs and symptoms according to their presence or absence, the final DDx list would be: (1) asthma, (2) lung cancer, (3) pulmonary edema, (4) tuberculosis, (5) sarcoidosis, (6) pneumonia, (7) bronchitis, (8) pulmonary embolism, (9) COPD, and (10) lymphoma. The suggested medical tests would be asthma tests, X-ray/CT scan, sputum cytology, biopsy, pulse oximetry, arterial blood gas analysis, and sputum test for tuberculosis. This result matches the actual H&P to a considerable degree and may be improved through learning (Figure 56).



Enlarged liver	0.00
Fainting	0.00
Fatigue and weakness	0.00
Fever	0.00
Frequent respiratory infections	0.00
Headache	0.00
Heart palpitations and arrhythmias	0.00
Hoarseness	0.00
Itching	0.00
Jaundice (yellow skin and eyes)	0.00
Joint pain	0.00
Loss of appetite	0.00
Muscle pain (myalgia)	0.00
Nausea and vomit	0.80
Night sweats	0.00
Pain behind the eyes	0.00
Phlegm (bloody/colored)	0.80
Shortness of breath (dyspnea)	1.00
Skin rash	0.00
Stomach pain	0.00
Sweats	0.00
Swollen ankles, feet or legs (edema)	0.00
Swollen lymph nodes (painless)	0.00
Weight loss	1.00
wheezing	0.40

Ordered DDx List:

1: Asthma (74.31%)
2: Lung Cancer (74.31%)
3: Pulmonary Edema (68.22%)
4: Tuberculosis (68.22%)
5: Sarcoidosis (65.94%)
6: Pneumonia (65.94%)
7: Bronchitis (65.21%)
8: Pulmonary Embolism (65.21%)
9: COPD (64.5%)
10: Lymphoma (64.5%)

Suggested Medical Tests:

1:
. Asthma tests
2:
. X-ray / CT Scan
. Sputum Cytology
. Biopsy
3:
. X-ray
. Pulse oximetry
. Arterial blood gas analysis
4:
. Sputum tests
. X-ray / CT Scan
. Blood tests
5:
. X-rav / CT Scan

Sign and Symptoms to be checked:

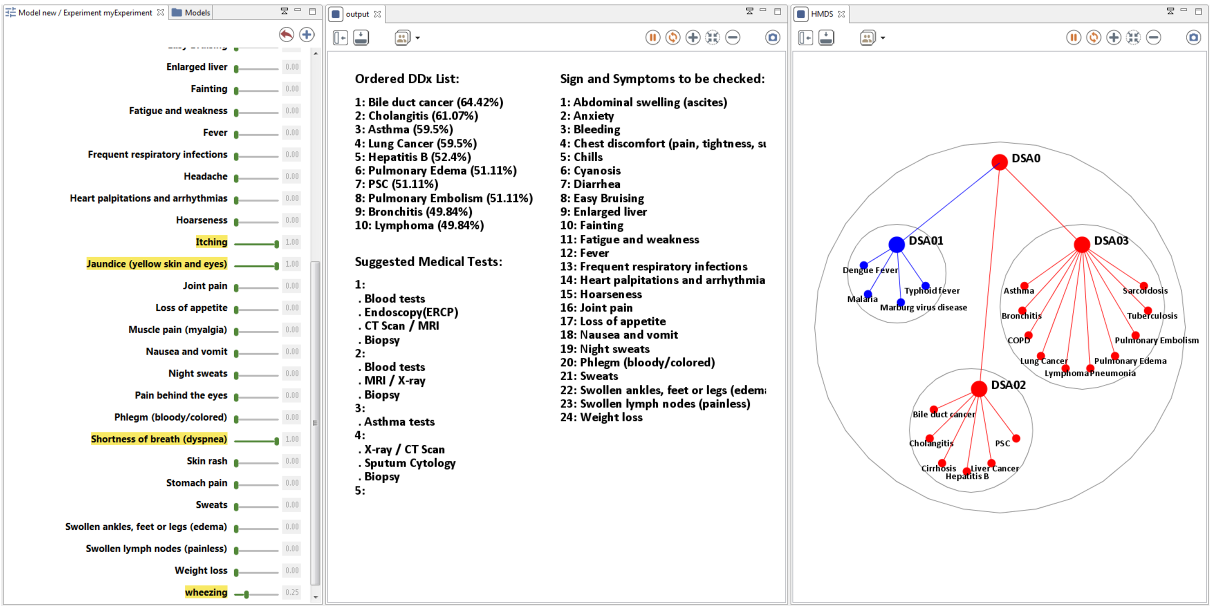
Figure 56. The final output of the simulated HMDS for case 1

Case 2: Metastatic Lung Cancer to Bile Duct Cancer

The HMDS also acts well in the presence of multiple diseases at the same time, like metastasis cases. This example is extracted from a medical paper on cancer metastasis [222]. The signs and symptoms in this case included abdominal pain, coarse breath sounds, dry cough, jaundice, and shortness of breath; and the final diagnosis was metastatic lung cancer to common bile duct cancer. The suggested medical tests were blood test, CT scan, ERCP (Endoscopic Retrograde Cholangiopancreatography), and biopsy.

Giving these symptoms to the HMDS as the diagnosis request two different DSAs will be activated: the DSA of pulmonary diseases and the DSA of hepatology and gastrointestinal disorders. Their super-holon will then put the output of both members in order. The DDx list will include: (1) bile duct cancer, (2) cholangitis, (3) asthma, (4) lung cancer, (5) hepatitis B, (6) pulmonary edema, (7) PSC, (8) pulmonary embolism, (9) bronchitis, (10) lymphoma[39]. As a result, the DDx list includes the bile duct cancer as the first and the lung cancer as the fourth possible diagnosis, and therefore the possibility of metastasis can be clearly mentioned to the physician (see Figure 57).

[39] It should be noted that the order of the final list is based on the similarity between the input and the DDP saved for each disease in the system.



Model new / Experiment myExperiment | Models

Enlarged liver	0.00
Fainting	0.00
Fatigue and weakness	0.00
Fever	0.00
Frequent respiratory infections	0.00
Headache	0.00
Heart palpitations and arrhythmias	0.00
Hoarseness	0.00
Itching	1.00
Jaundice (yellow skin and eyes)	1.00
Joint pain	0.00
Loss of appetite	0.00
Muscle pain (myalgia)	0.00
Nausea and vomit	0.00
Night sweats	0.00
Pain behind the eyes	0.00
Phlegm (bloody/colored)	0.00
Shortness of breath (dyspnea)	1.00
Skin rash	0.00
Stomach pain	0.00
Sweats	0.00
Swollen ankles, feet or legs (edema)	0.00
Swollen lymph nodes (painless)	0.00
Weight loss	0.00
wheezing	0.25

output

Ordered DDx List:

1: Bile duct cancer (64.42%)
2: Cholangitis (61.07%)
3: Asthma (59.5%)
4: Lung Cancer (59.5%)
5: Hepatitis B (52.4%)
6: Pulmonary Edema (51.11%)
7: PSC (51.11%)
8: Pulmonary Embolism (51.11%)
9: Bronchitis (49.84%)
10: Lymphoma (49.84%)

Suggested Medical Tests:

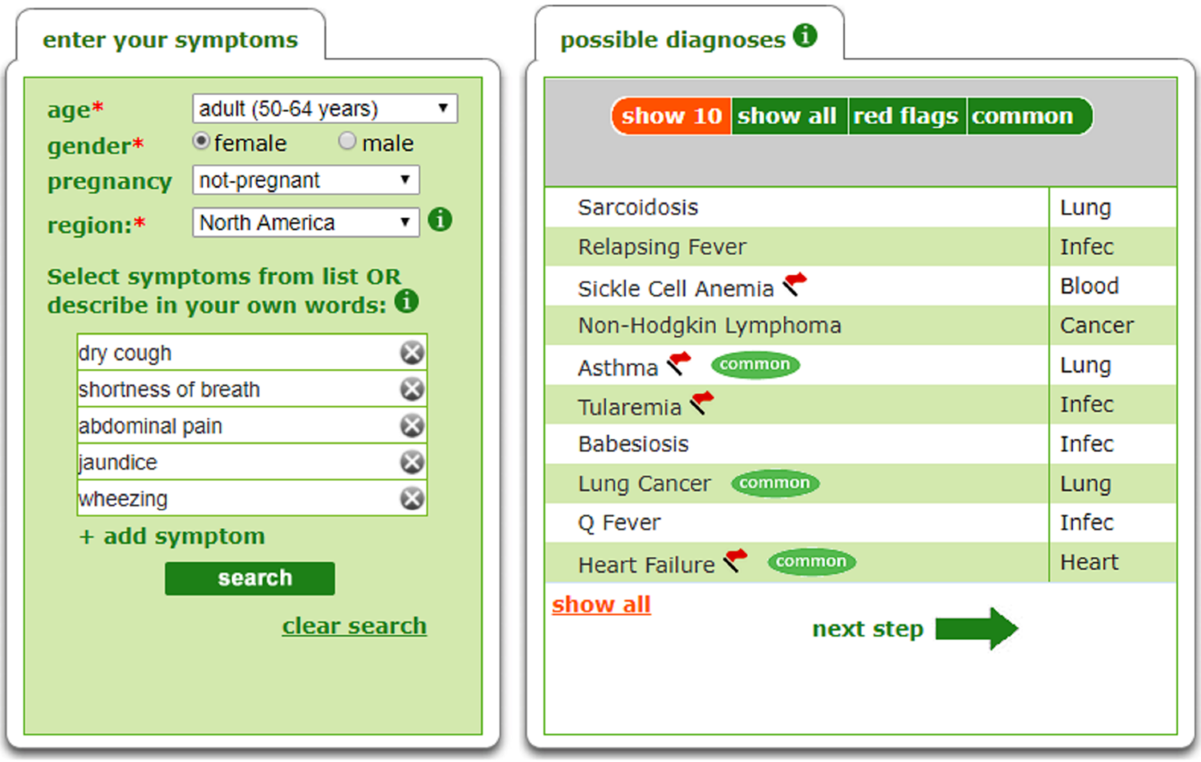
1:
. Blood tests
. Endoscopy(ERCP)
. CT Scan / MRI
. Biopsy
2:
. Blood tests
. MRI / X-ray
. Biopsy
3:
. Asthma tests
4:
. X-ray / CT Scan
. Sputum Cytology
. Biopsy
5:

Sign and Symptoms to be checked:

1: Abdominal swelling (ascites)
2: Anxiety
3: Bleeding
4: Chest discomfort (pain, tightness, su
5: Chills
6: Cyanosis
7: Diarrhea
8: Easy Bruising
9: Enlarged liver
10: Fainting
11: Fatigue and weakness
12: Fever
13: Frequent respiratory infections
14: Heart palpitations and arrhythmia
15: Hoarseness
16: Joint pain
17: Loss of appetite
18: Nausea and vomit
19: Night sweats
20: Phlegm (bloody/colored)
21: Sweats
22: Swollen ankles, feet or legs (edema
23: Swollen lymph nodes (painless)
24: Weight loss

Figure 57. The final output of the simulated HMDS for case 2

Giving these symptoms to Isabel, the system can only diagnose lung cancer and it announces this as suggestion number eighth (see Figure 58).



enter your symptoms

age* adult (50-64 years)
gender* ● female ○ male
pregnancy not-pregnant
region:* North America

Select symptoms from list OR describe in your own words:

dry cough
shortness of breath
abdominal pain
jaundice
wheezing

possible diagnoses

Sarcoidosis	Lung
Relapsing Fever	Infec
Sickle Cell Anemia	Blood
Non-Hodgkin Lymphoma	Cancer
Asthma common	Lung
Tularemia	Infec
Babesiosis	Infec
Lung Cancer common	Lung
Q Fever	Infec
Heart Failure common	Heart

Figure 58. Isabel output for case 2

8.2 The Assessment of the Self-Organization Abilities

8.2.1 The Assessment of Clustering in the HMDS

As mentioned in section 5.1.1, the initial holarchy of the HMDS can be created using clustering in different levels of the holarchy. For this reason, the most common and inclusive DSA of the system accepts the initial description of the diseases in form of DRAs, as its members, clusters them, and defines for each of the clusters (i.e., super-holons) a head. Each new DSA will then perform clustering on its members and defines new DSAs as its own members. This action is repeated recursively until no further clustering is necessary. As mentioned, this step can be performed once as the system is being defined and accelerate the self-organization. Later on, the system can still reorganize its architecture using reinforcement learning (see section 6.1.3). In this section a simulation with 45 diseases and 135 signs and symptoms is presented. Table 17 lists the diseases recognized by the simulated system and Table 18 covers the signs and symptoms. Table 19, Table 20, and Table 21 show the signs and symptoms of each of the diseases.

Table 17. The diseases covered by the simulated system

#	Disease	#	Disease
1	Absence seizure	24	Lymphoma
2	Acute pancreatitis	25	Madelung-Launois-Bensaude
3	Asthma	26	Malaria
4	Bile duct cancer	27	Marburg virus disease
5	Bronchitis	28	Meningitis
6	Bursitis	29	Muscular dystrophy
7	Cholangitis	30	Myasthenia gravis
8	Cholecystitis	31	Obesity
9	Chronic pancreatitis	32	Osteoarthritis
10	Cirrhosis	33	Pancreatic cancer
11	Congenital myopathies	34	Pancreatic cysts
12	COPD	35	Pneumonia
13	Cushing syndrome	36	Polymyositis
14	Dengue fever	37	PSC
15	Epilepsy	38	Pulmonary edema
16	Febrile seizure	39	Pulmonary embolism
17	Grand mal seizure	40	Rheumatoid arthritis
18	Hepatitis B	41	Sarcoidosis
19	Lipedema	42	Sjogren’s syndrome
20	Lipohypertrophy	43	Temporal lobe seizure
21	Liver cancer	44	Tuberculosis
22	Lung cancer	45	Typhoid fever
23	Lupus		

Table 18. The signs and symptoms covered by the simulated system

#	Sign or Symptom	#	Sign or Symptom
1	Abdominal pain	69	Joint warmth
2	Abdominal pain (upper part)	70	Lack of muscle tone
3	Abdominal pain that feels worse after eating	71	Large calf muscles
4	Abdominal pain that radiates to back	72	Learning disabilities
5	Abdominal swelling (ascites)	73	Lip smacking
6	Abdominal tenderness	74	Loss of appetite
7	Acne	75	Loss of bowel and bladder control
8	Altered speaking	76	Loss of consciousness or awareness
9	Anxiety	77	Loss of joint flexibility
10	Belching	78	Malaise
11	Black stools	79	Memory loss
12	Bleeding	80	Muscle cramps or contractions
13	Bloating	81	Muscle pain (myalgia)
14	Blood clots	82	Muscle pain and stiffness
15	Bone spurs	83	Nausea and vomit
16	Breathing problems	84	neck muscle weakness
17	Burning sensation in throat	85	New-onset diabetes
18	Butterfly-shaped rash on face	86	Night sweats
19	Chest discomfort (pain, tightness, suffocating)	87	No thirst
20	Chills	88	Oily, smelly stools (steatorrhea)
21	Clay-colored stools	89	Pain behind the eyes
22	column-like legs	90	Pain in upper abdomen (indigestion)
23	Confusion	91	Pain radiating to right shoulder or back
24	Conjunctivitis (red eyes)	92	Phlegm (bloody/colored)
25	Constipation	93	physical deformity
26	Convulsions (rhythmic contractions)	94	Pink or purple stretch marks (striae)
27	Cough	95	Problems chewing
28	Cyanosis	96	Progressive muscle weakness
29	Dark urine	97	Psychic symptoms
30	Delayed motor skills	98	Raised areas under skin (painful, red, warm)
31	Depression	99	Rapid pulse
32	Diarrhea	100	Raynaud's phenomenon
33	Difficulty getting up from a lying or sitting position	101	Regurgitation (Acid reflux)
34	Difficulty swallowing (dysphagia)	102	Repeated swallowing or chewing
35	Double vision (diplopia	103	Runny or stuffy nose
36	Drooping of one or both eyelid (ptosis)	104	Scream
37	Dry eyes	105	Seizures
38	Dry mouth	106	Sensation of a lump in the throat
39	Dry skin	107	Sensitivity to light
40	Easy Bruising	108	shake or jerk arms and legs
41	Enlarged liver	109	Shortness of breath (dyspnea)
42	Extreme sleepiness	110	Skin lesions (Photosensitivity)
43	Eyelid flutters	111	Skin rash
44	Facial muscle weakness	112	Slow healing of cuts, insect bites and infections)
45	Fainting	113	Sneezing
46	Fatigue and weakness	114	Sore throat
47	Fatty tissue (lower body)	115	Sour taste in mouth
48	Fatty tissue (proximal upper body)	116	Staring
49	Feeling a mass in upper abdomen	117	Stiff neck
50	Feeling especially full after meals	118	Stomach pain
51	Fever	119	Sudden contract of muscles causing the person to fall
52	Food impaction	120	Sudden stop in motion without falling
53	Frequent falls	121	Sweats
54	Frequent respiratory infections	122	Swollen ankles, feet or legs (edema)
55	Grating sensation in joints	123	Swollen lymph nodes (painless)
56	Headache	124	Swollen salivary glands
57	Heart pain (angina)	125	Trouble walking, running and jumping
58	Heart palpitations and arrhythmias	126	Unawareness of having had a seizure
59	Heart rhythm disturbances	127	Unhealthy body fat distribution
60	Heartburn	128	Unusual finger movements
61	High BMI	129	Unusual sensation (aura)
62	Hoarseness	130	Vaginal dryness
63	Itching	131	Vomiting blood
64	Jaundice (yellow skin and eyes)	132	Waddling gait
65	Joint pain	133	Walking on the toes
66	Joint stiffness	134	Weight loss
67	Joint swelling	135	Wheezing
68	Joint tenderness		

Table 19. The signs and symptoms of diseases 1-15 (row: S/Sx, column: disease)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0
2	0.0	1.0	0.0	1.0	0.0	0.0	1.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	1.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0
6	0.0	1.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0
9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
11	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	1.0	0.0
13	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
14	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
15	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
16	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0
17	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
18	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
19	0.0	0.0	1.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
20	0.0	0.0	0.0	0.0	1.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0
21	0.0	0.0	0.0	1.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
22	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
23	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
24	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
25	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
26	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
27	0.0	0.0	1.0	0.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	1.0	0.0
28	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
29	0.0	0.0	0.0	1.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
30	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0
31	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0
32	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0
33	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0
34	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0
35	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
36	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0
37	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
38	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
39	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
40	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	1.0	0.0	0.0
41	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
42	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
43	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
44	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0
45	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
46	0.0	0.0	0.0	1.0	1.0	1.0	0.0	0.0	1.0	1.0	0.0	1.0	1.0	1.0	0.0
47	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
48	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0
49	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
50	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
51	0.0	1.0	0.0	1.0	1.0	1.0	1.0	1.0	1.0	0.0	0.0	0.0	0.0	1.0	0.0
52	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
53	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0
54	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
55	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
56	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0
57	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
58	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
59	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
60	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
61	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
62	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
63	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0
64	0.0	1.0	0.0	1.0	0.0	0.0	1.0	0.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0
65	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0
66	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
67	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
68	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
69	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
70	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0
71	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
72	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0
73	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
74	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0
75	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
76	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
77	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
78	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
79	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
80	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0
81	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0
82	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
83	0.0	1.0	0.0	1.0	0.0	0.0	1.0	1.0	1.0	1.0	0.0	0.0	0.0	1.0	0.0
84	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0
85	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
86	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
87	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
88	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
89	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0
90	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
91	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
92	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
93	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0
94	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0
95	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0
96	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0
97	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
98	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
99	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
100	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
101	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
102	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
103	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
104	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
105	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
106	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
107	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
108	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
109	0.0	0.0	1.0	0.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
110	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
111	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0
112	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0
113	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
114	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
115	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
116	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
117	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
118	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
119	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
120	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
121	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
122	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
123	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
124	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
125	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0
126	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
127	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0
128	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
129	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
130	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
131	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
132	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0
133	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
134	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	1.0	1.0	0.0	1.0	0.0	0.0	0.0
135	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0

Table 20. The signs and symptoms of diseases 16-30 (row: S/Sx, column: disease)

	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0	0.0
2	0.0	0.0	1.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
11	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
13	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
14	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
15	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
16	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0
17	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
18	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
19	0.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
20	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	1.0	1.0	0.0	0.0	0.0
21	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
22	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
23	1.0	1.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0
24	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
25	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
26	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
27	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
28	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
29	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
30	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
31	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
32	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0	0.0
33	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0
34	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0
35	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
36	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
37	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
38	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
39	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
40	0.0	0.0	0.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
41	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
42	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0
43	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
44	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
45	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
46	0.0	1.0	1.0	1.0	0.0	1.0	0.0	1.0	1.0	1.0	1.0	1.0	0.0	0.0	0.0
47	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
48	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0
49	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
50	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
51	1.0	1.0	1.0	0.0	0.0	0.0	0.0	1.0	1.0	0.0	1.0	1.0	1.0	0.0	0.0
52	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
53	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0
54	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
55	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
56	1.0	1.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	1.0	1.0	1.0	0.0	0.0
57	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
58	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
59	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0
60	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
61	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
62	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
63	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
64	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
65	0.0	0.0	1.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
66	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
67	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
68	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
69	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
70	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0
71	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0
72	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0
73	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
74	0.0	0.0	1.0	0.0	0.0	1.0	0.0	0.0	1.0	0.0	0.0	0.0	1.0	0.0	0.0
75	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
76	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0
77	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
78	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
79	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
80	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
81	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0	0.0
82	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0
83	1.0	0.0	1.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	1.0	1.0	1.0	0.0	0.0
84	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
85	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
86	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
87	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0
88	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
89	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
90	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
91	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
92	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
93	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0
94	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
95	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
96	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0
97	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
98	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
99	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
100	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
101	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
102	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
103	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
104	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
105	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
106	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
107	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0
108	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
109	0.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
110	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
111	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0
112	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
113	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
114	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
115	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
116	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
117	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0
118	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
119	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
120	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
121	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0
122	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
123	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
124	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
125	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0
126	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
127	0.0	0.0	0.0	1.0	1.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0
128	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
129	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
130	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
131	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
132	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0
133	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0
134	0.0	0.0	0.0	0.0	0.0	1.0	1.0	1.0	1.0	0.0	0.0	1.0	0.0	0.0	0.0
135	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Table 21. The signs and symptoms of diseases 31-45 (row: S/Sx, column: disease)

	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
2	0.0	0.0	1.0	1.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
11	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
13	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
14	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
15	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
16	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
17	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
18	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
19	0.0	1.0	0.0	0.0	1.0	0.0	0.0	1.0	1.0	1.0	1.0	1.0	0.0	1.0	0.0
20	0.0	0.0	0.0	0.0	1.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0
21	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
22	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
23	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	1.0	1.0	0.0	0.0
24	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
25	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
26	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
27	0.0	1.0	0.0	0.0	1.0	0.0	0.0	1.0	1.0	1.0	1.0	1.0	0.0	1.0	1.0
28	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
29	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
30	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
31	1.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
32	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
33	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
34	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
35	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
36	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
37	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	1.0	0.0	0.0	0.0
38	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
39	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
40	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
41	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
42	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0
43	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
44	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
45	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
46	0.0	1.0	1.0	0.0	1.0	0.0	1.0	0.0	0.0	1.0	1.0	1.0	0.0	1.0	1.0
47	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
48	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
49	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
50	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
51	0.0	1.0	0.0	1.0	1.0	0.0	1.0	0.0	0.0	1.0	1.0	0.0	0.0	1.0	1.0
52	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
53	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
54	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
55	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
56	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
57	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
58	0.0	0.0	0.0	0.0	1.0	0.0	0.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
59	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
60	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
61	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
62	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
63	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
64	0.0	0.0	1.0	1.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
65	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	1.0	0.0	0.0	0.0
66	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	1.0	0.0	0.0	0.0
67	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	1.0	0.0	0.0	0.0
68	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	1.0	0.0	0.0	0.0
69	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	1.0	0.0	0.0	0.0
70	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
71	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
72	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
73	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0
74	0.0	0.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0
75	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
76	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0
77	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0
78	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
79	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	1.0	0.0	0.0	0.0
80	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
81	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
82	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
83	0.0	0.0	1.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
84	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
85	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
86	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0
87	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
88	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
89	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
90	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
91	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
92	0.0	0.0	0.0	0.0	1.0	0.0	0.0	1.0	1.0	0.0	0.0	0.0	0.0	1.0	0.0
93	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0
94	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
95	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
96	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
97	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
98	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
99	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
100	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
101	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
102	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0
103	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
104	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
105	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0
106	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
107	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
108	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
109	0.0	1.0	0.0	0.0	1.0	0.0	1.0	1.0	1.0	1.0	1.0	1.0	0.0	0.0	0.0
110	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0
111	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	1.0	0.0	0.0	1.0
112	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
113	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
114	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
115	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
116	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0
117	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
118	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
119	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
120	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
121	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	1.0
122	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
123	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0
124	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	1.0	0.0	0.0	0.0
125	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
126	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0
127	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
128	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0
129	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0
130	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
131	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
132	0.0	0.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
133	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
134	0.0	0.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0	1.0	0.0	1.0	1.0
135	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0

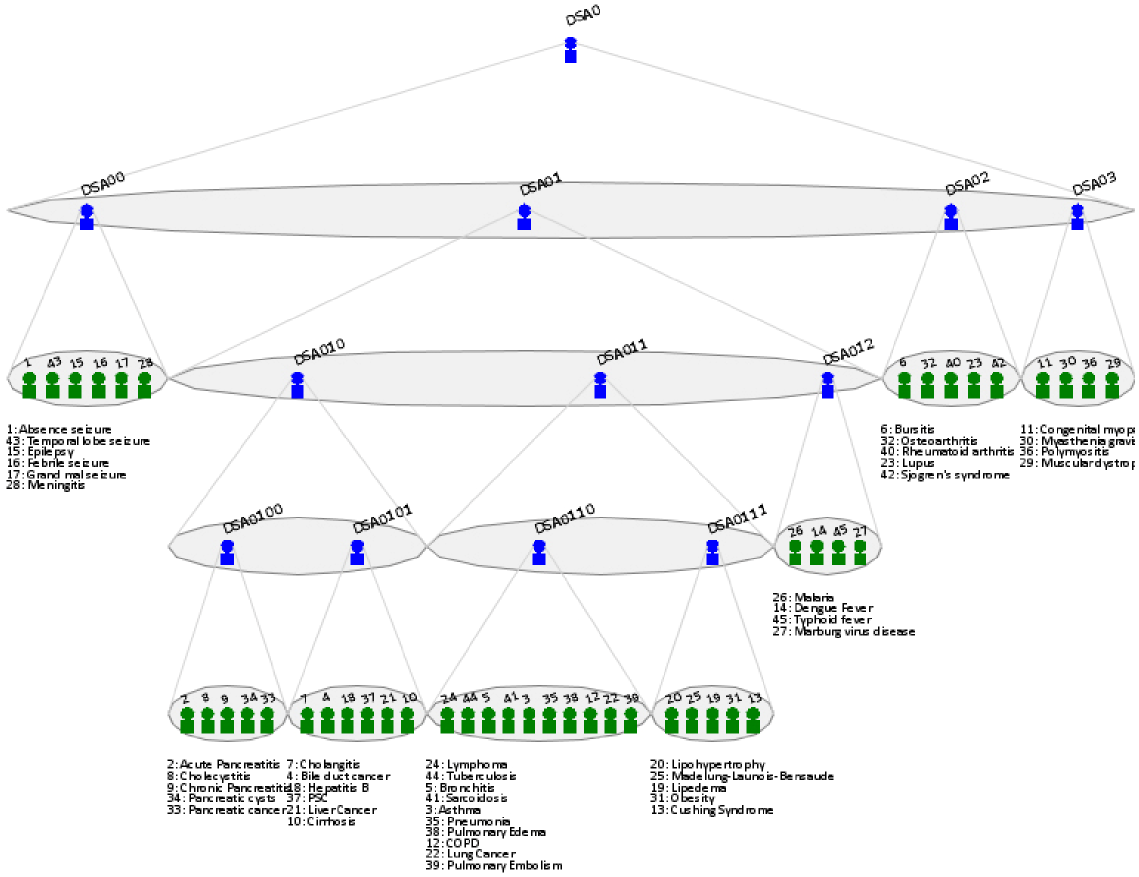


Figure 59. The result of clustering in the HMDS

Figure 59 illustrates the result of clustering in the HMDS. The clustering is done based on the distances between the DDP, i.e., the holon identifier of each of the DRAs. Each DSA that solely includes DRAs has in fact grouped the diseases that should be considered in the differential diagnosis of each of its members. For complete list of diseases that should be covered in the DDx of each of the diseases please refer to [17]. It should be noted that this grouping is done in the range of the diseases that were covered in the simulation. Moreover, the groups formed on the higher levels will help to indicate the right signs and symptoms to be checked in cases where the diagnosis request is not precise enough to reach DSAs that contain DRAs.

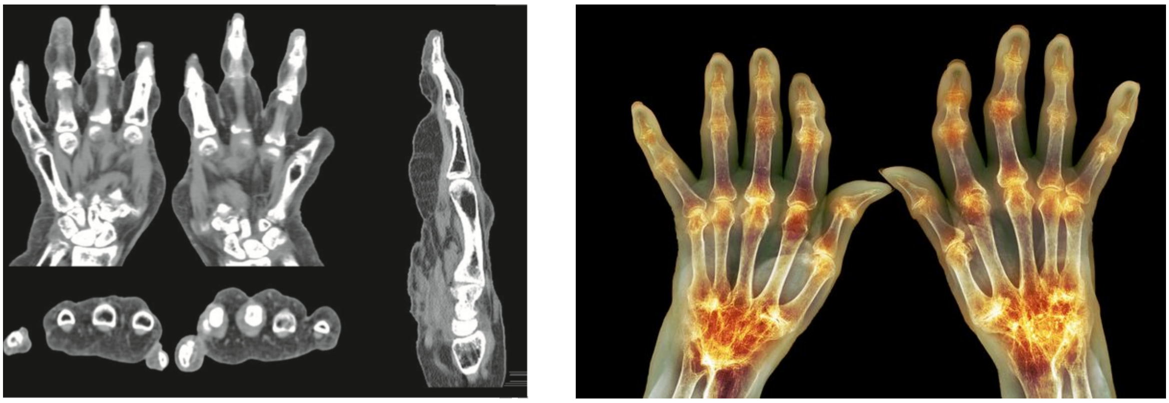
8.2.2 The Assessment of Reinforcement Learning in the HMDS

Case 3: A New Differential Diagnosis for Arthritis

In this simulation, the system covers 45 diseases in a holarchy with four levels. Here, again, a real case is used, in which the Madelung-Launois-Bensaude disease (MLB) is suggested as a new differential diagnosis for arthritis [223]. MLB is a disease that causes the concentration of fatty tissue in proximal upper body. In 2008, for the first time, some instances of this disease have been observed with distal fatty tissue that were misdiagnosed first as arthritis, which normally includes joint pain and joint swelling. Table 22 lists the common signs and symptoms of MLB together with the signs and symptoms observed in the new instances of the disease.

Table 22. The common and new signs and symptoms of MLB

MLB common signs and symptoms	New signs and symptoms
1. Unhealthy body fat distribution 2. Fatty tissue (proximal upper body) 3. Fatigue 4. Physical deformity	1. Joint pain (hands and knees) 2. Joint swelling



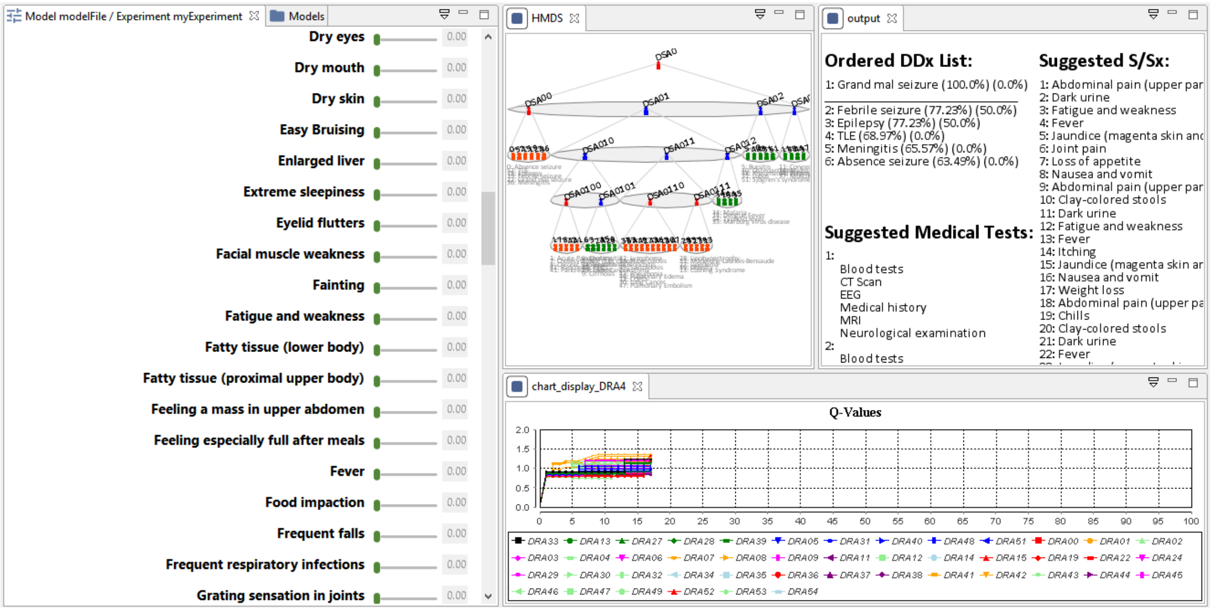
(a) CT scan of MLB

(b) X-ray of rheumatoid arthritis

Figure 60. The similarity between the signs of MLB and arthritis

In this experiment, the new observations will be given to a version of the HMDS, which so far has not considered the MLB disease with arthritis for the reason of DDx. The system should then be able to come to the same conclusion as in [223] and add the DRA acting for the MLB disease to the super-holon containing the DRA acting for arthritis.

In order to demonstrate the system's reactions to this new finding the user interface of the system displays the corresponding Q-values on a specially dedicated diagram (see Figure 61). Essentially, if an agent is not involved in a diagnosis process it will not receive any reward and as a result, its Q-value will remain the same during that round. As the agent participates in a diagnosis process, it will be rewarded and consequently its Q-value will be updated. In case the Q-value of any of the members of the super-holon is getting close to be a noise (close to lower three-sigma limit), the agent will start exploring new opportunities to join new super-holons. One promising approach for this agent is to try to become a member of those super-holons that were activated at the same time with its current super-holon, i.e. guided exploration (see section 5.3). This will guarantee that the agent would have some common interests with the members of its new super-holon(s).



Dry eyes	0.00
Dry mouth	0.00
Dry skin	0.00
Easy Bruising	0.00
Enlarged liver	0.00
Extreme sleepiness	0.00
Eyelid flutters	0.00
Facial muscle weakness	0.00
Fainting	0.00
Fatigue and weakness	0.00
Fatty tissue (lower body)	0.00
Fatty tissue (proximal upper body)	0.00
Feeling a mass in upper abdomen	0.00
Feeling especially full after meals	0.00
Fever	0.00
Food impaction	0.00
Frequent falls	0.00
Frequent respiratory infections	0.00
Grating sensation in joints	0.00

Ordered DDx List:

1: Grand mal seizure (100.0%) (0.0%)

2: Febrile seizure (77.23%) (50.0%)
3: Epilepsy (77.23%) (50.0%)
4: TLE (68.97%) (0.0%)
5: Meningitis (65.57%) (0.0%)
6: Absence seizure (63.49%) (0.0%)

Suggested Medical Tests:

1:
Blood tests
CT Scan
EEG
Medical history
MRI
Neurological examination
2:
Blood tests

Suggested S/Sx:

1: Abdominal pain (upper par
2: Dark urine
3: Fatigue and weakness
4: Fever
5: Jaundice (magenta skin anc
6: Joint pain
7: Loss of appetite
8: Nausea and vomit
9: Abdominal pain (upper par
10: Clay-colored stools
11: Dark urine
12: Fatigue and weakness
13: Fever
14: Itching
15: Jaundice (magenta skin ar
16: Nausea and vomit
17: Weight loss
18: Abdominal pain (upper pa
19: Chills
20: Clay-colored stools
21: Dark urine
22: Fever

Figure 61. The Q-value diagram on the user interface

In this simulation before entering the signs and symptoms of distal MLB as the diagnosis request the system is first provided with some random inputs for diagnosis. As a result, it has become easier to monitor the changes of the Q-values of the DRA reacting for MLB on the Q-value diagram. Figure 62 shows the changes in the Q-values of the different DRAs in case 3. Entering the distal MLB instances into the system, as the DRA acting for the MLB disease now represents a disease with signs and symptoms that are not common in its super-holon, its Q-value will get closer to the lower outlier threshold. Working on these inputs, the super-holon of the arthritis disease has also been activated and as the DRA acting for the MLB disease will eventually start looking for a chance to join some new super-holons, it will try to become a member of this super-holon. Considering the holon identifier of its members, this super-holon will then check whether the DRA acting for the MLB disease would be an outlier and since the case is negative, it will accept this new member. At this stage, the Q-value being displayed for the DRA acting for the MLB will be the maximum value between the Q-values to its super-holons, which in this case is the Q-value to its new super-holon. As it can be followed on the diagram, since this value is now not close to the outlier threshold of at least one of its super-holons, the DRA acting for MLB will stop exploring at this point.

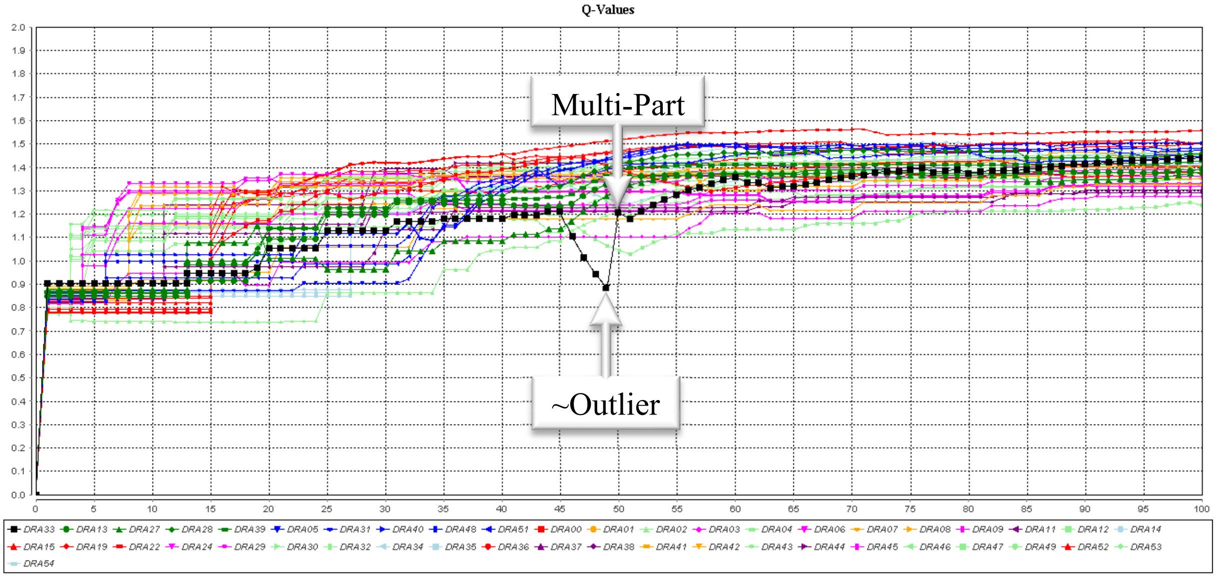
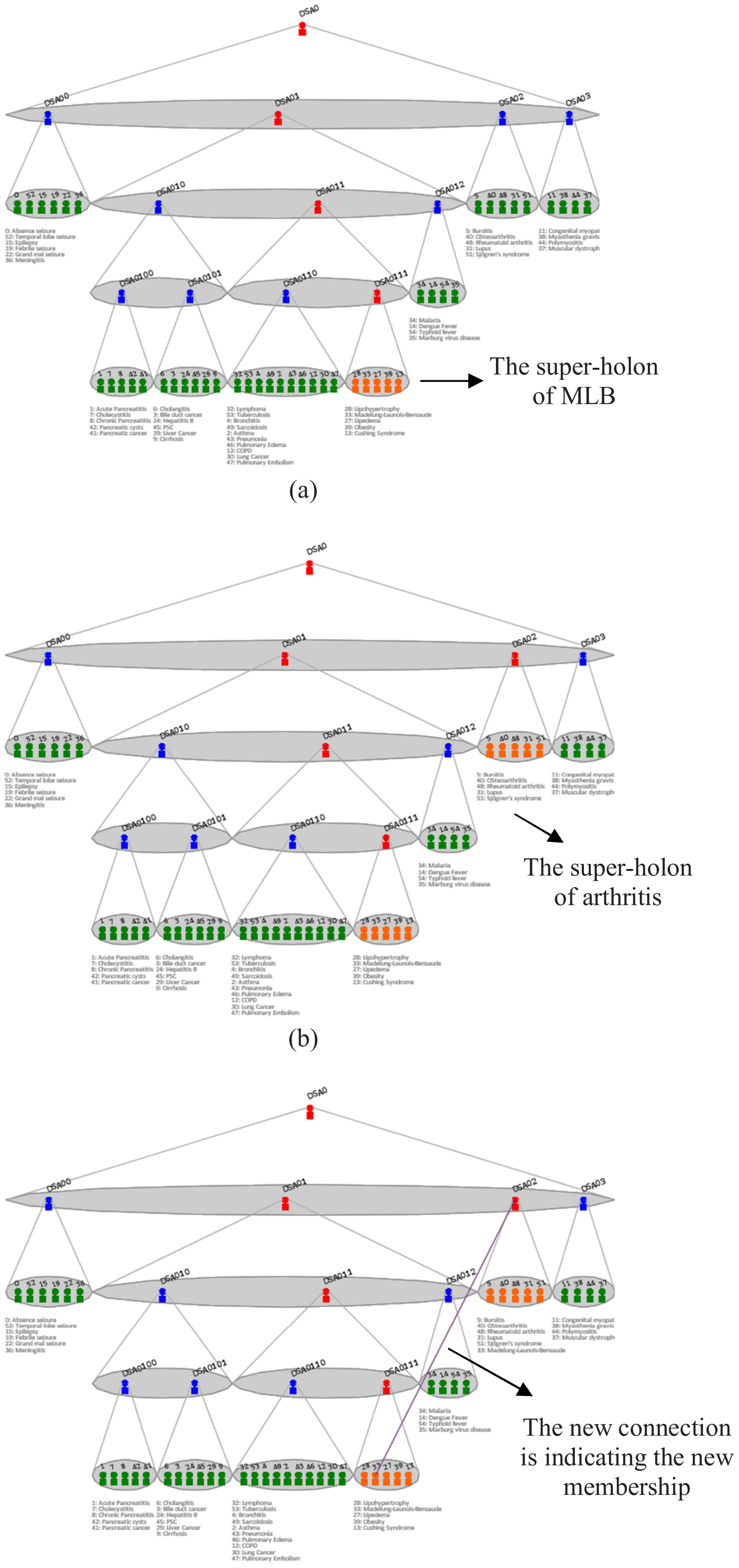


Figure 62. Changes in the Q-values of the different DRAs in case 3

Figure 63 illustrates the self-organization in the HMDS caused by the new instances of the MLB disease. Figure 63(a) shows the active super-holon in the HMDS while receiving the common instances of MLB disease[40]. Figure 63(b) shows that the super-holons of MLB and arthritis are both activated if the system receives any instances of the distal MLB[41]. It should be noted that this means that even the untrained system could have been able to suggest both diseases in its final DDx list and as a result help the physician to consider the possibility of MLB. As mentioned above, after receiving a few instances of the distal MLB the Q-value of the DRA acting for MLB will get closer to the lower outlier threshold and as a result the DRA will start looking for a chance to join some new super-holons with similar signs and symptoms. As the super-holon of the DRA acting for arthritis is active at this stage the DRA acting for MLB will try to become a member of this super-holon (Figure 63 (c)). This simulation has shown an example of the guided exploration in which an agent would try to join the super-holons that are reacting to a diagnosis request it has responded to. For more information on the guided exploration please refer to section 5.3.

[40] The super-holon of the DRA acting for MLB includes the DRAs acting for the following diseases: Lipohypertrophy, MLB, Lipedema, Obesity, and Cushing syndrome.

[41] The super-holon of the DRA acting for arthritis includes the DRAs acting for the following diseases: Bursitis, Osteoarthritis, Rheumatoid arthritis, Lupus, and Sjogren's syndrome.



The super-holon of MLB

(a)

The super-holon of arthritis

(b)

The new connection is indicating the new membership

(c)

Figure 63. Self-organization in the HMDS caused by new instances of MLB disease

8.2.3 The Assessment of System's Behavior in Integrating a New DRA

Medical knowledge demonstrates a steady upward growth and many new diseases are still being discovered. In recent years, for example, many infectious diseases have been discovered, including SARS, MERS, Ebola, chikungunya, avian flu, swine flu, Zika and, most recently, COVID-19. As a result, in order to support the diagnosis of such new diseases, the system should be able to assign DRAs to new diseases and allow them to find their right position in its holarchy. Figure 64 illustrates the system's behavior in integrating the DRA acting for COVID-19. This DRA is first introduced to the system as a member of the highest and most inclusive DSA (Figure 64-a). At this stage this DRA calculates its Euclidean distance to the rest of the members of its super-holon and finds the closest member and then sends a membership request to this agent. As the DRA acting for COVID-19 is not an outlier in this super-holon, the membership request will be accepted and the DRA will become a member of this DSA (Figure 64-b). The same approach is followed by the agent until no further downward movements are possible, i.e., the new super-holon has no DSA members or the membership request from the DRA is rejected by all DSA members that have received a request. In this case the DRA acting for COVID-19 ends up in a super-holon that includes influenza and common cold (Figure 64-c). This result matches the diseases that are considered for differential diagnosis of illness in patients under investigation for the COVID-19, which are given in [224]. The diagnostic algorithm introduced in this reference included, “immediately upon sample receipt, a rapid molecular test for the most common respiratory pathogens in order to obtain a fast differential diagnosis” [224, p. 2]. Among the diseases that are considered by the simulated HMDS, the mentioned pathogens are the cause of bronchitis, common cold, different types of influenza, and different types of pneumonia, that may also cause pulmonary edema. Figure 65 demonstrates the output of the simulated system in diagnosing a COVID-19 case (signs and symptoms of this disease are given as input). Two super-holons have been activated in response to the given input and the ordered DDx list includes COVID-19, Bronchitis, Pneumonia, common cold, influenza, and pulmonary edema, which clearly match the pathogens mentioned in [224]. It should be noted that the system may produce different outputs by integrating more diseases and of course through learning. However, as the system follows the logic behind the DDx process these all should contribute to its improvement.

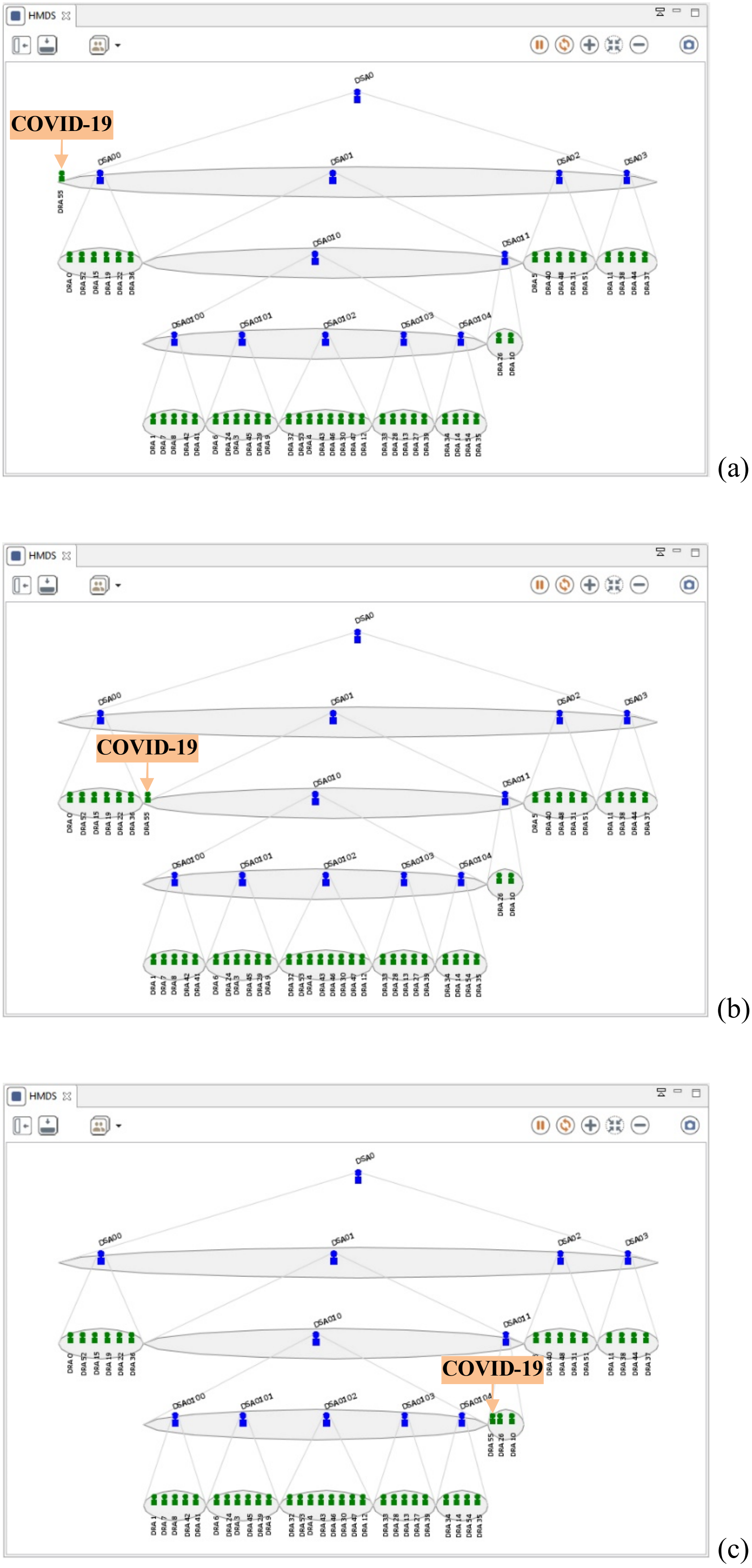
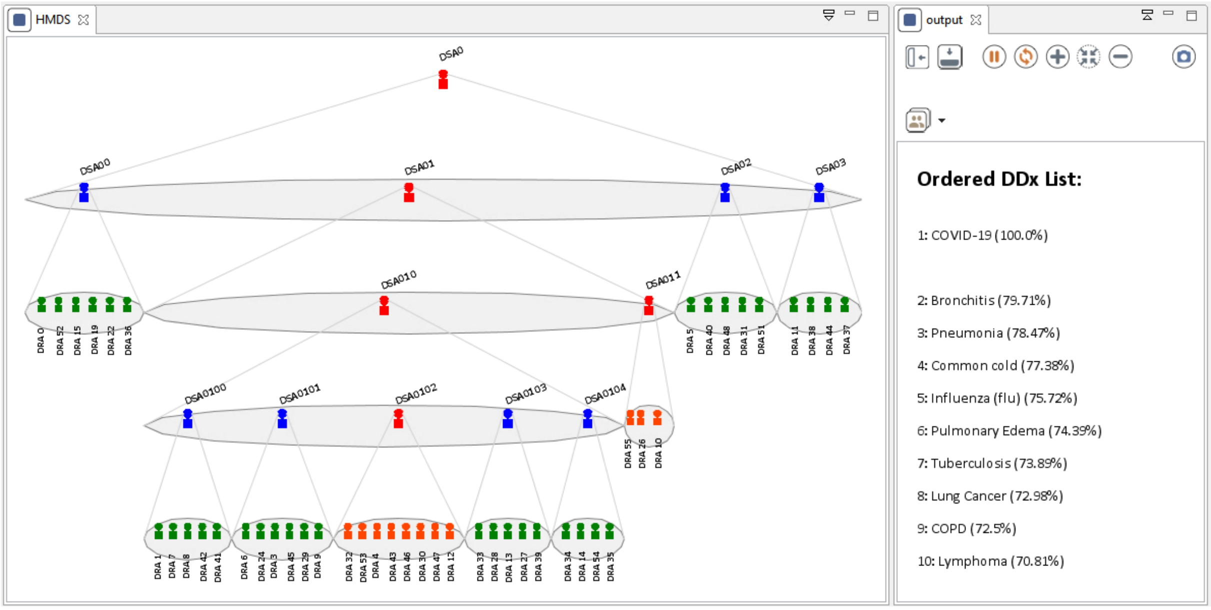


Figure 64. System's behavior in integrating the DRA acting for COVID-19



Ordered DDx List:

1: COVID-19 (100.0%)

2: Bronchitis (79.71%)

3: Pneumonia (78.47%)

4: Common cold (77.38%)

5: Influenza (flu) (75.72%)

6: Pulmonary Edema (74.39%)

7: Tuberculosis (73.89%)

8: Lung Cancer (72.98%)

9: COPD (72.5%)

10: Lymphoma (70.81%)

Figure 65. System's output in diagnosing a COVID-19 case

Chapter 9
CONCLUSION AND FUTURE WORKS

"Our imagination is the only limit to what we can hope to have in the future."

- Charles F. Kettering (1876 - 1958)

9.1 Conclusions

This study has focused on medical diagnosis systems and has discussed that even though such systems aim to support the medical diagnosis in health care institutions such as hospitals, they do not thoroughly cover the differential diagnosis directed history and physical examination and as a result cannot be well integrated into the clinical workflow of these institutions. This concludes the need for a system that can perform DDx and as a result conduct the H&P process. Such a system (1) reduces diagnostic errors by providing immediate second opinions even on signs and symptoms to be checked in the H&P, (2) guides and facilitates filling out the H&P

form at the same time, (3) may help to tackle physician shortages by guiding nurses in preparing the H&P reports that are then to be controlled by physicians, (4) can be added as a software component to available MDSs and provide them with the required comprehensive inputs, which allows these systems to fit into the clinical workflow and promotes the wider use of them, and (5) can offer attractive side benefits, e.g., helping us to broaden our knowledge on diseases, providing a means of more timely detection of outbreaks, and so on.

A careful study of the DDx domain in this research has suggested that this domain meets the characteristics of the holonic domain. In short, the differential diagnostic problem can be recursively broken down into sub-problems by weighting the likelihood of the presence of possible diseases. These subproblems may induce different abstraction levels and can be of different granularities. According to the nature of DDx, the problem solvers are collaborative and those dealing with similar diseases need to communicate more, which is to be conducted in a timely manner. As a result, DDx can be implemented by a holonic MAS.

This work proposed a well-designed HMAS that according to the system assessments can successfully guide DDx and eventually conduct the H&P process. This system improves the state of the art of the MDSs by addressing their critical shortcoming, i.e. the lack of implementation of the ability to guide the user in providing the system with the all-encompassing input, which is the key to a flawless diagnosis. To organize and allow this HMAS to reorganize itself based on the environment it is dealing with, different machine learning techniques have been suggested to be applied to the system. Machine learning has also been used in this project to allow the system to improve its medical knowledge using the data it collects through its continuing interactions with the environment. Accordingly, the development of the Holonic Medical Diagnosis System, which is capable of performing DDx, is the practical contribution of this work. Moreover, the introduction of the machine learning techniques that are used to adapt the functionality of the system can be considered as the conceptual/theoretical contribution of this research. In fact, it should be noted that the proposed techniques can be applied to other HMASs that adopt a similar approach for problem solving to the one followed in this study.

To put it concisely, due to the complexity of the DDx and as neither the number of levels nor the number of the groups in each level of the holarchy are predefined, clustering (unsupervised learning) has been used to build up the initial holarchy of the system. To this end, a simple and effective method for automatically determining the input parameter of the used clustering method has been proposed. Moreover, since the medical knowledge demonstrates a steady upward growth, and diagnosis is also very much affected by the geographical regions, the system needs to adapt and improve its behavior, i.e., update its medical knowledge based on the new instances and improve the holarchy according to the experience and the feedback. In the HMDS, holon identifiers are updated applying the exponential smoothing (supervised learning), and in order to support the self-organization of the holarchy the Holonic-Q-learning (reinforcement learning) has been proposed.

The self-organization problem of the holarchy has been expressed as a sequential decision-making problem and modelled using a Markov chain. As discussed, sequential decision-making problems that can be modeled as Markov decision processes can be solved using methods that combine dynamic programming and reinforcement learning. Depending on the problem and the available decision makers, i.e., the agents, such RL algorithms may be designed for single-agent systems or multi-agent systems that either consists of agents with individual goals and decision-making capabilities, which are influenced by other agent's decisions, or behave as a swarm of agents that collaboratively follow a single objective.

Many studies have been conducted in this area; however, focusing on available swarm RL algorithms provides a clear view of the areas that still need attention. Most of the studies in this area are concentrating on homogeneous swarms and to date, systems introduced as heterogeneous swarms merely include very few, i.e., two or three homogeneous sub-swarms, which either according to their capabilities address specific sub-problem of the general problem or exhibit different behaviors to reduce the risk of bias.

In this study a novel approach has been introduced that allows individuals with higher heterogeneity rates, which are even addressing different problems, to behave as a swarm when solving shared sub-problems. In fact, the affinity between two agents that indicates the compatibility of agents to work together towards solving a specific sub-problem is used to design a heterogeneous swarm RL algorithm that allows heterogeneous swarms to solve sequential decision-making problems consisting of sub-problems that should be addressed by different sub-groups of its members. As a result, the affinity-based heterogeneous swarm RL essentially allows the agents that are not identical but are capable of collaboration to exhibit swarm behavior providing them with the means of sharing their knowledge and eventually dealing with problems that match their specialties. This learning method essentially allows such agents to collect information from a larger swarm and to be able to make better decisions using this broader knowledge.

It should be noted that even though the experiments have shown that the affinity-based heterogeneous swarm RL method is able to increase the performance of the heterogeneous agents solving SDMPs, this method clearly has its own limitations and is solely applicable when subgroups of agents with significant lesser extent of heterogeneity are extractable and are in addition to that sufficiently populated to be able to exhibit meaningful swarm behavior.

Regarding the suggested RL method it should also be noted that very few studies have been done on the machine learning methods that can be applied to HMASs and the research in this area is still at a very nascent stage. Notable examples of the suggested methods are [225] and [226], but none of them are applicable to the HMDS. Even though using these methods learning in each level can be influenced by the learning results, i.e., the behavior, of the holons on the other levels (inter-holonic learning data), these methods do not aim to improve the inter-holonic connections, i.e., memberships. In contrast, the approach presented in this study in particular aims to refine the holarchy, as the position of agents in this system will greatly define their behavior. In other words, available approaches use machine learning to improve the decision-making abilities of the system, however, in HMDS decision-making process has been kept as simple as possible and generally the position of agents is decisive for their success. Accordingly, the machine learning method should target the agent positions.

Regarding the similarity of the suggested problem-solving approach to the existing methods it is worth noting that the idea of using a swarm of experts with similar specialties is indeed very similar to the ensemble learning and particularly one of its commonly used algorithms, the mixture of experts. According to [227], “ensemble learning is the process by which multiple models, such as classifiers and experts, are strategically generated and combined to solve a computational intelligence problem”.

The mixture of experts [228] is one of the algorithms commonly used for ensemble learning that generates several experts whose outputs are combined through a generalized linear rule. A hierarchical mixture of experts can also be further combined if the output is conditional on multiple levels of probabilistic gating functions [229]. This may resemble the use of HetSRL method for the self-organization of a holarchy. However, according to the mentioned definitions, even though the use of experts with similar specialties in the affinity-based HetSRL can be regarded as an application of ensemble learning there is a small but distinctive difference between these two approaches. In ensemble learning experts are strategically generated differently to obtain better results than could be obtained from any of the constituent experts alone. Heterogeneity, however, is a feature of the agents of the environment in affinity-based HetSRL and the idea is to provide those agents that can collaborate with each other with the means to share their knowledge and eventually deal with problems that match their specialties.

Additionally, since applying the affinity-based HetSRL method the problem will be modeled using a Markov chain and the heterogeneous agents can be regarded as multi-labeled instances, it may be concluded that the problem could also have been modeled and solved using classifier chains, which is a machine learning method for problem transformation in multi-label classification [230].

The classifier chains method essentially builds a deterministic high order Markov chain model to capture the interdependencies between labels in multi-label classification problems [231]. However, classification cannot be used to build the Markov chain in the general problem environment that is considered for the HetSRL method or even to structure the holarchy for the Holonic-Q-Learning method, as neither the number of levels nor the number of the groups on each level are predefined. Moreover, the goal here is not to cluster the problem solvers but to build a system that can not only act independently in performing complex diagnosis tasks, but also learn and adapt itself to the changes.

It should also be noted that the classifier chains method is a supervised learning approach, for which as mentioned a set of examples with paired input and desired output should be provided. However, in the general problem scenario considered in this work it is assumed that the examples of desired input/output pairs are not given, and that the algorithm is able to estimate the optimal actions only by interacting with a dynamic environment. In such conditions, reinforcement learning methods that can exhibit flexibility and be applied to open, dynamic, and complex environments are to be used. Considering the self-organization problem in a holarchy, it is obvious that only a reinforcement learning method can assure that the structure is adapting to the changing environment and that classifier chains method cannot be used to organize a dynamic holarchy.

9.2 Future work

This study suggested that in order to practically integrate the available medical diagnosis systems into the workflow of the health care institutions such systems should be able to cover and guide the history and physical examination. The Holonic Medical Diagnosis System introduced in this work aimed to address the key process of the history and physical examination, i.e., the differential diagnosis. To this end, the focus of this study has been on the differential diagnosis domain, its characteristics, and its implementation using the HMAS technology. Indeed, the history and physical examination intends to collect the relevant information for disease diagnosis and eventually generate a differential diagnosis list. However, it should be noted that this information is gathered by questioning and checking the signs and symptoms that are categorized under different sections of the history and physical examination. As a result, in order to implement a system that fully covers the history and physical examination according to its actual clinical structure, the final system should not just be able to determine the relevant signs and symptoms to be checked, but it should be able to mention them in the right section of the history and physical examination. It is obvious that this categorization does not imply that another approach rather than the one introduced in this thesis should be used to implement the history and physical examination but suggests further design details and adjustments in the system design phase.

Moreover, as mentioned in the introduction chapter several functions were excluded from this thesis, which can be considered as future works (see Table 2). The system introduced in this thesis does not intend to support natural language processing and assumes that for the signs and symptoms claimed by the patients the user of the system is able to find the equivalent terms, which are recognizable by the system. However, the automation of this process can add a valuable capability to the system and should be regarded as one of the important steps to be taken before delivering the system to the final user.

The system can use its data in order to suggest the possibility of new diseases or outbreaks, and then provide supplementary information on the matter. However, this function is also not covered in this project. This option can be very helpful as it may lead to an early detection of possible outbreaks. The importance of this capability is obvious as the World Health Organization has warned in its 2007 report that infectious diseases are emerging at a rate that has not been seen before. Since the 1970s, about 40 infectious diseases have been discovered, including SARS, MERS, Ebola, chikungunya, avian flu, swine flu, Zika and, most recently, COVID-19. As a result, adding this option to the system is of vital importance.

Furthermore, as mentioned, providing the diagnoses, the current system is capable of suggesting the relevant drugs and treatments. However, within the scope of the present project it does neither suggest patient specific treatment nor discover negative drug-drug interactions, especially in case of multiple diseases. This capability, however, is very critical as exposure to inappropriate medication can lead to serious irrecoverable effects.

Regarding the system's medical data, as mentioned the relevant information has been gathered from the available disease/symptom databases and the system is capable of updating this data through learning. However, the current system does not include knowledge extraction from natural language sources and considers this as a feature that may be added to the system in future works. This option may be very interesting as it can update the system knowledge according to the latest medical papers and eventually increase the reliability of the system. Nevertheless, as the main focus of this project has been on the implementation of differential diagnosis this option is also suggested to be considered as future work.

In the end, regarding the heterogeneous swarm reinforcement learning method introduced in this thesis, it should be noted that despite the promising results of this machine learning method in dealing with heterogeneous swarms, as most of the studies in the field of swarm intelligence is concentrated on homogeneous swarms, it is highly recommended to address this knowledge gap thoroughly and increase the understanding of the constraints and the effective factors that should be taken into consideration when dealing with heterogeneous swarms. Furthermore, it should be noted that the heterogeneous swarm reinforcement learning method introduced in this study aims to address the self-organization problem in the holonic multi-agent systems. As not much research has been conducted in the field of machine learning for this group of multi-agent systems, this area requires more attention in order to allow the holonic MASs to exhibit their true potential.

References

[1] "MedTerms Medical Dictionary," MedicineNet , [Online]. Available: https://www.medicinenet.com/medterms-medical-dictionary/article.htm . [Accessed 14 12 2018].

[2] "World Health Organization," [Online]. Available: http://www.who.int/en/.

[3] J. C. Segen, Concise Dictionary of Modern Medicine, McGraw-Hill, 2002.

[4] D. J. Bell, "Sine qua non," [Online]. Available: https://radiopaedia.org/articles/sine-qua-non. [Accessed 05 11 2019].

[5] Merriam-Webster, "Differential Diagnosis," [Online]. Available: https://www.merriam-webster.com/dictionary/differential%20diagnosis. [Accessed 15 June 2017].

[6] J. Maude, "Differential diagnosis: the key to reducing diagnosis error, measuring diagnosis and a mechanism to reduce healthcare costs," *Diagnosis,* vol. 1, no. 1, pp. 107-109, 2014.

[7] Z. Akbari and R. Unland, "Automated Determination of the Input Parameter of the DBSCAN Based on Outlier Detection," in *Artificial Intelligence Applications and Innovations, IFIP Advances in Information and Communication Technology, Vol. 475, pp. 280-291*, 2016.

[8] Z. Akbari and R. Unland, "A Holonic Multi-Agent System Approach to Differential Diagnosis," *Multiagent System Technologies: 15th German Conference, MATES 2017,* vol. LNCS 10413, pp. 272 - 290, 2017.

[9] Z. Akbari and R. Unland, "A Holonic Multi-agent Based Diagnostic Decision Support System for Computer-Aided History and Physical Examination," in *Advances in*

Practical Applications of Agents, Multi-Agent Systems, and Complexity: The PAAMS Collection (PAAMS 2018), Lecture Notes in Computer Science, vol 10978, Springer, Cham, 2018.

[10] Z. Akbari and R. Unland, "Towards Computer-Aided Differential Diagnosis: A Holonic Multi-agent Based Medical Diagnosis System," in *Distributed Computing and Artificial Intelligence, Special Sessions, 15th International Conference. DCAI 2018*, 2019.

[11] Z. Akbari and R. Unland, "A Novel Heterogeneous Swarm Reinforcement Learning Method for Sequential Decision Making Problems," *Machine Learning and Knowledge Extraction,* vol. 1, no. 2, pp. 590-610, 2019.

[12] G. Serugendo, M.-P. Gleizes and A. Karageorgos, "Self-Organization in Multi-Agent Systems," *The Knowledge Engineering Review,* vol. 20, no. 2, pp. 165-189, 2005.

[13] M. Ester, H.-P. Kriegel, J. Sander and X. Xu, "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise," in *the 2nd Int. Conf. on Knowledge Discoverey and Data Mining*, 1996.

[14] R. Elio, J. Hoover, I. Nikolaidis, M. Salavatipour, L. Stewart and K. Wong, "About Computing Science Research Methodology," 2011. [Online]. Available: https://webdocs.cs.ualberta.ca/~c603/readings/research-methods.pdf. [Accessed 05 11 2018].

[15] "Diseases and Conditions," Mayo Clinic, [Online]. Available: https://www.mayoclinic.org/diseases-conditions. [Accessed 08 11 2018].

[16] "Health A-Z," WebMD, [Online]. Available: https://www.webmd.com/a-to-z-guides/health-topics. [Accessed 08 11 2018].

[17] "Medscape," WebMD, [Online]. Available: https://emedicine.medscape.com/. [Accessed 28 02 2019].

[18] "Disease Database," Medical Object Oriented Software Enterprises Ltd, [Online]. Available: http://www.diseasesdatabase.com/. [Accessed 08 11 2018].

[19] "eH&P™ custom History & Physical Exam™," [Online]. Available: http://www.scymed.com/en/smnxab/smnxabch.htm. [Accessed 26 10 2018].

[20] "ICD-10 Version:2016," [Online]. Available: http://apps.who.int/classifications/icd10/browse/2016/en.

[21] "GAMA Platform," [Online]. Available: https://gama-platform.github.io/. [Accessed 09 11 2018].

[22] P. Densen, "Challenges and Opportunities Facing Medical Education," *American Clinical and Climatological Association,* vol. 122, p. 48–58, 2011 .

[23] E. S. e. Berner, Clinical Decision Support Systems: Theory and Practice, Springer, 2016.

[24] Berner and E. S, " Clinical decision support systems: State of the Art," AHRQ Publication No. 09-0069-EF. Rockville, Maryland: Agency for Healthcare Research and Quality, June 2009.

[25] M. Alther and C. K. Reddy, "Chapter 19: Clinical Decision Support Systems," in *Healthcare Data Analytics*, Chapman and Hall/CRC 2015, 2015, pp. 625-656.

[26] W. R. Hersh, "The Electronic Medical Record: Promises and Problems," *Journal of the American Society for Information Science,* vol. 46, no. 10, pp. 772-776, 1995.

[27] J. Thompson, "A Practical Guide to Clinical Medicine," UCSD School of Medicine and VA Medical Center, San Diego, California, [Online]. Available: https://meded.ucsd.edu/clinicalmed/write.html. [Accessed 22 09 2020].

[28] J. F. Styron and P. J. Evans, "The evolution of office notes and the electronic medical record: The CAPS note," *Cleveland Clinic Journal of Medicine,* vol. 83, no. 7, pp. 542-544, July 2016.

[29] J. Shoolin, L. Ozeran, C. Hamann and W. 2. Bria, "Association of Medical Directors of Information Systems consensus on inpatient electronic health record documentation," *Applied Clinical Informatics,* vol. 4, no. 2, p. 293–303, 2013.

[30] J. Hahn, J. Bernstein, R. McKenzie, B. King and C. Longhurst, "Rapid implementation of inpatient electronic physician documentation at an academic hospital," *Appl Clin Inform,* vol. 3, p. 175–185, 2012.

[31] L. Jacobs, "Interview with Lawrence Weed, MD—the father of the problem-oriented medical record looks ahead," *The Permanente Journal. Kaiser Permanente,* vol. 13, no. 3, p. 84–89, 2009.

[32] T. Gunter and N. Terry, "The emergence of national electronic health record architectures in the United States and Australia: models, costs, and questions," *Journal of Medical Internet Research,* vol. 7, no. 1, March 2005.

[33] B. Chaudhry, J. Wang, S. Wu, M. Maglione, W. Mojica, E. Roth, S. C. Morton and P. G. Shekelle, "Systematic Review: Impact of Health Information Technology on Quality, Efficiency, and Costs of Medical Care," *Annals of Internal Medicine,* vol. 144, p. 742–752, 2006.

[34] "AmericanEHR," [Online]. Available: http://www.americanehr.com/Home.aspx. [Accessed 25 10 2018].

[35] "Epic," [Online]. Available: https://www.epic.com/software#Clinicals. [Accessed 25 10 2018].

[36] "Cerner Ambulatory," [Online]. Available: https://www.cerner.com/solutions/ambulatory. [Accessed 25 10 2018].

[37] "Smart Medical Apps - H&P," Smart Medical Apps, [Online]. Available: https://play.google.com/store/apps/details?id=com.smartmedicalapps.checklist&hl=en . [Accessed 23 09 2020].

[38] "Clinicals – History & Physical," Medical Gear, [Online]. Available: https://play.google.com/store/apps/details?id=com.smartddx.clinicals&hl=en. [Accessed 23 09 2020].

[39] "History & Physical Exam pc," Börm Bruckmeier Publishing LLC, [Online]. Available: https://play.google.com/store/apps/details?id=com.bbi.History_and_Physical_Exam_apocketcards. [Accessed 23 09 2020].

[40] "IBM Watson Health," [Online]. Available: https://www.ibm.com/watson/health/. [Accessed 23 01 2018].

[41] H. Fisher, A. Tomlinson, P. Ramnarayan and J. Britto, "Isabel: support with clinical decision making," *Pediatric Nursing,15(7),* pp. 34-35, 2003.

[42] P. Ramnarayan, G. Kulkarni, T. A. and J. Britto, "ISABEL: a novel Internet-delivered clinical decision support system.," in *Current perspectives in healthcare computing, pp. 245-256*, 2004.

[43] N. Riches, M. Panagioti, R. Alam, S. Cheraghi-Sohi, S. Campbell, A. Esmail and P. Bower, "The effectiveness of electronic differential diagnoses (DDX) generators: a systematic review and meta-analysis," *PloS one, 11(3), e0148991,* 2016.

[44] M. Yuan, "Watson and healthcare: How natural language processing and semantic search could revolutionize clinical decision support," IBM developerWorks, [Online]. Available: https://www.ibm.com/developerworks/library/os-ind-watson/, last accessed 15.06.2017.. [Accessed 15 06 2017].

[45] M. Graber and A. Mathew, "Performance of a web-based clinical diagnosis support system for internists," *Journal of general internal medicine,* vol. 23, no. 1, pp. 37-40, 2008.

[46] I. Husain, "Why IBM's artificial intelligence "Watson" could not replace a physician," 21 02 2011. [Online]. Available: https://www.imedicalapps.com/2011/02/ibm-watson-replace-physician-artificial-intelligence/. [Accessed 21 01 2018].

[47] M. F. Collen, "Origins of Medical Informatics," *West J Med., 145(6),* pp. 778-785, 1986.

[48] M. F. Collen and H. W. E., "Development of Medical Information Systems (MISs)," in *In The History of Medical Informatics in the United States*, Springer-Verlag London, 2015.

[49] E. H. Shortliffe, B. G. Buchanan and F. E. A., "Knowledge Engineering for Medical Decision Making: A Review of Computer-Based Clinical Decision Aids," in *Proceedings of the IEEE, Vol. 67, NO. 9*, 1979.

[50] "Open Clinical," 2017. [Online]. Available: http://www.openclinical.org/dss.html.

[51] R. A. Miller and A. Geissbuhler, "Clinical Diagnostic Decision Support Systems – An Overview," in *Clinical Decision Support Systems: Theory and Practice*, Springer, 1998.

[52] Y. Shoham, "Agent-Oriented Programming (Technical Report STAN-CS-90-1335)," Stanford University, Computer Science Department, 1990.

[53] B. L. Iantovics, "Agent-Based Medical Diagnosis Systems," *Computing and Informatics, Vol. 27,* p. 593–625, 2008.

[54] C. Klüver, J. Klüver and R. Unland, "A Medical Diagnosis System based on MAS Technology and Neural Network," *Business Process, Services, Computing and Intelligent Service Management,* pp. 179 - 191, 2009.

[55] S. Chao and F. Wong, "Multi-Agent Learning Paradigm for Medical Data Mining Diagnostic Workbench," *Data Mining and Multi-agent Integration,* pp. 177-186, 2009.

[56] H. E. Pople Jr., "Presentation of the INTERNIST System," in *Proceedings of the AIM Workshop, New Brunswick, N.J.: Rutgers University*, 1976.

[57] S. Weiss, A Sysstem for Model-Based Computer-Aided Diagnosis and Therapy, Ph.D. Thesis, Computers in Biomedicine, Department of Computer Science, Rutgers University, CBM-TR-27-Thesis, 1974.

[58] S. G. Pauker, G. A. Gorry, J. P. Kassirer and W. B. Schwartz, "Towards the Simulation of Clinical Cognition: Taking a Present Illness by Computer," *Amer. J. Med.,* vol. 60, 1976.

[59] E. H. Shortliffe, "Computer-Based Medical Consultations: MYCIN," *Elsevier, New York,* 1976.

[60] J. S. Aikins, J. C. Kunz, E. H. Shortliffe and R. J. Fallat, "PUFF: an expert system for interpretation of pulmonary function data," *Computers and biomedical research,* vol. 16, no. 3, pp. 199-208, 1983.

[61] R. Miller, F. Masarie and J. Myers, "Quick medical reference (QMR) for diagnostic assistance," *MD Comput,* vol. 3, no. 5, pp. 34-48, 1986.

[62] P. H. Winston and K. A. Prendergast, "CADUCEUS: An Experimental Expert System for Medical Diagnosis," *The AI Business: Commercial Uses of Artificial Intelligence,* pp. 67-80, 1986.

[63] G. Barnett, J. Cimino, J. Hupp and E. Hoffer, "DXplain. An evolving diagnostic decision-support system," *JAMA,* vol. 258, no. 1, pp. 67-74, 1987.

[64] M. Lincoln, C. Turner, P. Haug and e. al., "Iliad training enhances medical students' diagnostic skills," *J Med Syst,* vol. 15, no. 1, pp. 93-110, 1991.

[65] R. P. De Cresce and M. S. Lifshitz, "PAPNET™ Cytological Screening System," *Laboratory Medicine,* vol. 22, no. 4, p. 276–280, 1 April 1991.

[66] W. Baxt, "Use of an arteficial neural network for the diagnosis of myocardial infarction," *Ann Intern Med,* vol. 115, no. 11, pp. 843-848, 1991.

[67] G. D. Tourassi and J. C. Floyd, "Artificial neural networks for single photon emission computed tomography. A study of cold lesion detection and localization," *Investigative radiology,* vol. 28, no. 8, pp. 671-677, 1993.

[68] G. Tourassi, C. Floyd, H. Sostman and R. Coleman, "Artificial neural network for diagnosis of acute pulmonary embolism: effect of case and observer selection," *Radiology,* vol. 194, no. 3, pp. 889-893, 1995.

[69] D. Fogel, E. 3. Wasson and E. Boughton, "Evolving neural networks for detecting breast cancer," *Cancer Letters,* vol. 96, no. 1, pp. 49-53, 1995.

[70] A. Costa, J. Cabestany, J. Moreno and M. Calvet, "Neuroserum: An artificial neural net-based diagnostic aid tool for serum electrophoresis," in *Third international conference on neural networks and expert systems in medicine and healthcare*, 1998.

[71] "A Super-Gross Diagnostic Tool That Could Save Your Life," WIRED, [Online]. Available: https://www.wired.com/2015/04/visualdx/. [Accessed 03 11 2018].

[72] I. M. Tleyjeh, H. Nada and L. M. Baddour, "VisualDX: Decision-support software for the diagnosis and management of dermatologic disorders," *Clinical Infectious Diseases,* vol. 43, no. 9, p. 1177–1184, 2006.

[73] K. Higuchi, K. Sato, H. Makuuchi, F. A. S. Takamoto and H. Takeda, "Automated diagnosis of heart disease in patients with heart murmurs: application of a neural network technique," *Journal of medical engineering & technology,* vol. 30, no. 2, pp. 61-68, 2006.

[74] N. Barakat, A. P. Bradley and M. N. H. Barakat, "Intelligible support vector machines for diagnosis of diabetes mellitus," *IEEE transactions on information technology in biomedicine,* vol. 14, no. 4, pp. 1114-1120, 2010.

[75] E. Elveren and N. Yumuşak, "Tuberculosis disease diagnosis using artificial neural network trained with genetic algorithm," *Journal of medical systems,* vol. 35, no. 3, pp. 329-332, 2011.

[76] S. Duraipandian, W. Zheng, J. Ng, J. J. Low, A. Ilancheran and Z. Huang, "In vivo diagnosis of cervical precancer using Raman spectroscopy and genetic algorithm techniques," *Analyst,* vol. 136, no. 20, pp. 4328-4336, 2011.

[77] D. C. Barbosa, D. B. Roupar, J. C. Ramos, A. C. Tavares and C. S. Lima, "Automatic small bowel tumor diagnosis by using multi-scale wavelet-based analysis in wireless capsule endoscopy images," *Biomedical engineering online,* vol. 11, no. 1, 2012.

[78] O. Y. Atkov, S. G. Gorokhova, A. G. Sboev, E. V. Generozov, E. V. Muraseyeva, S. Y. Moroshkina and N. N. Cherniy, "Coronary heart disease diagnosis by artificial neural networks including genetic polymorphisms and clinical parameters," *Journal of cardiology,* vol. 59, no. 2, pp. 190-194, 2012.

[79] M. Saxena, "IBM Watson Progress and 2013 Roadmap," 23 02 2013. [Online]. Available: https://www.slideshare.net/manojsaxena2/ibm-watson-progress-and-roadmap-saxena/7-Watson_Healthcare_Products_1H_2013. [Accessed 2018 01 23].

[80] A. Yahiaoui, O. Er and N. Yumusak, "A new method of automatic recognition for tuberculosis disease diagnosis using support vector machines," *Biomedical Research,* vol. 28, no. 9, 2017.

[81] L. Miner, P. Bolding, J. Hilbe, M. Goldstein, T. Hill, R. Nisbet, N. Walton and G. Miner, Practical Predictive Analytics and Decisioning Systems for Medicine: Informatics Accuracy and Cost-Effectiveness for Healthcare Administration and Delivery Including Medical Research, Academic Press, 2014.

[82] K. Papik, B. Molnar, R. Schaefer, Z. Dombovari, Z. Tulassay and J. Feher, "Application of neural networks in medicine-a review," *1998,* vol. 4, no. 3, pp. MT538-MT546, Medical Science Monitor.

[83] Q. K. Al-Shayea, "Artificial neural networks in medical diagnosis," *International Journal of Computer Science Issues,* vol. 8, no. 2, pp. 150-154, 2011.

[84] F. Amato, L. Alberto, E. M. Peña-Méndez, P. Vaňhara and A. H. J. Hampl, "Artificial neural networks in medical diagnosis," *Journal of Applied Biomedicine,* vol. 11, pp. 47-58, 2013.

[85] A. Ghaheri, S. Shoar, M. Naderan and S. S. Hoseini, "The applications of genetic algorithms in medicine," *Oman medical journal,* vol. 30, no. 6, pp. 406-416, 2015.

[86] M. Ulieru and R. Unland, "A Stigmergic Approach to Medical Diagnosis," in *Proc. of the 2nd Int. Workshop on Multi-Agent Systems for Medicine and Computational Biology, pp. 87-103* , 2006.

[87] E. S. Berner, Clinical Decision Support Systems: Theory and Practice, Springer Science & Business Media, 2007, p. 270.

[88] T. L. R. Medicine, "Opening the black box of machine learning," *The Lancent Respiratory Medicine,* vol. 6, no. 11, 2018.

[89] H. Salem, G. Attiya and N. El-Fishawy, "A Survey of Multi-Agent based Intelligent Decision Support System for Medical Classification Problems," *International Journal of Computer Applications,* vol. 123, no. 10 , 2015.

[90] L. Lhotska, V. Marik and T. Vlcek, "Medical applications of enhanced rule-based expert systems," *International journal of medical informatics,* vol. 63, no. 1-2, pp. 61-75, 2001.

[91] B. Hayes-Roth, "A Blackboard Architecture for Control.," *Artificial Intelligence,* vol. 26, no. 3, pp. 251-320, July 1985.

[92] B. Buchanan and E. Shortliffe, Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project, Addison-Wesley, Reading (Mass.), 1984.

[93] S. Z. Zaidi, S. S. Abidi and S. Manickam, "Leveraging intelligent agents for knowledge discovery from heterogeneous healthcare data repositories," *Studies in health technology and informatics,* vol. 90, pp. 335-340, 2002.

[94] C. Arus, B. Celda, S. Dasmahaptra and D. Dupplaw, "On the Design of a Web-Based Decision Support System for Brain Tumour Diagnosis Using Distributed Agents," in *International Conference, Web Intelligence and Intelligent Agent Technology Workshops, IEEE/WIC/ACM*, 2006.

[95] B. L. Iantovics, "A novel diagnosis system specialized in difficult medical diagnosis problems solving," in *Emergent Properties in Natural and Artificial Dynamical Systems* , 2006.

[96] R. M. A. Mateo, L. F. Cervantes, H. K. Yang and J. Lee, "Mobile agents using data mining for diagnosis support in ubiquitous healthcare," in *KES International Symposium on Agent and Multi-Agent Systems: Technologies and Applications*, 2007.

[97] B. Iantovics, "Hybrid expert system agents.," in *Proc. of the Int. Conf. European Integration between Tradition and Modernity, Petru Maior University Press, Tg. Mureş*, 2007.

[98] O. Kazar, Z. Sahnoun and L. Frecon, "Multi-agents system for medical diagnosis," in *International Conference on Intelligent System and Knowledge Engineering*, 2008.

[99] A. Koestler, The Gost in the Machine, Hutchinson, 1967.

[100] C. Gerber, J. H. Siekmann and G. Vierke, Holonic Multi-Agent Systems, Technical Report DFKI-RR-99-03, German Research Centre for Artificial Intelligence, 1999.

[101] R. Unland, "A Holonic Multi-Agent System for Robust, Flexible, and Reliable Medical Diagnosis," in *Meersman, R., Tari, Z., (Eds.), OTM-WS 2003, LNCS 2889*, 2003.

[102] M. Ulieru, "Internet-Enabled Soft Computing Holarchies for e-Health Applications-Soft Computing Enhancing the Internet and the Internet Enhancing Soft Computing," *Enhancing the Power of the Internet,* pp. 131-165, 2004.

[103] G. Moise, P. G. Moise and P. S. Moise, "Toward Holons-Based Architecture for Medical Systems," in *In IEEE/ACM International Workshop on Software Engineering in Healthcare Systems (SEHS)*, 2018.

[104] IBM, "IBM WATSON," [Online]. Available: http://ibmwatson237.weebly.com/. [Accessed 25 09 2020].

[105] "IBM Watson supercomputer," TechTarget, [Online]. Available: https://searchenterpriseai.techtarget.com/definition/IBM-Watson-supercomputer. [Accessed 01 11 2018].

[106] B. Upbin, "IBM's Watson Gets Its First Piece Of Business In Healthcare," Forbes, 08 02 2013. [Online]. Available: https://www.forbes.com/sites/bruceupbin/2013/02/08/ibms-watson-gets-its-first-piece-of-business-in-healthcare/#7926335a5402. [Accessed 01 11 2018].

[107] M. Miliard, "Watson heads to medical school," Healthcare IT news, 30 10 2012. [Online]. Available: https://www.healthcareitnews.com/news/cleveland-clinic-ibm-send-watson-medical-school. [Accessed 01 11 2018].

[108] N. Leske, "Doctors seek help on cancer treatment from IBM supercomputer," Reuters, 08 02 2013. [Online]. Available: https://in.reuters.com/article/ibm-watson-cancer/doctors-seek-help-on-cancer-treatment-from-ibm-supercomputer-idINDEE9170G120130208. [Accessed 01 11 2018].

[109] M. Herper, "MD Anderson Benches IBM Watson In Setback For Artificial Intelligence In Medicine," Forbes, 19 02 2017. [Online]. Available: https://www.forbes.com/sites/matthewherper/2017/02/19/md-anderson-benches-ibm-watson-in-setback-for-artificial-intelligence-in-medicine/#7d4e6da63774. [Accessed 01 11 2018].

[110] "The DeepQA Research Team," IBM, [Online]. Available: https://researcher.watson.ibm.com/researcher/view_group_subpage.php?id=2159. [Accessed 01 11 2018].

[111] Autonomy Corporation plc., "Autonomy Technology Overview (Autonomy Whitepaper)".

[112] "Isabel Products," Isabel, [Online]. Available: https://www.isabelhealthcare.com/products?hsCtaTracking=7e5a6ff1-1381-4d02-984c-924e476167ae%257C9627ae5d-a0d7-46ad-964b-098b1a59088f. [Accessed 01 11 2018].

[113] D. Bauman, "Isabel Differential Diagnosis Tool Achieves 98% Accuracy in New Study," Isabel, [Online]. Available: https://www.prweb.com/releases/2017isabelaidemonstrates/03impressiveaccuracy/prweb14198903.htm. [Accessed 26 09 2020].

[114] H. L. Ioachim and M. L. Jeffrey, "Chapter 57: Hodgkin Lymphoma: Classical," in *Ioachim's Lymph Node Pathology*, Lippincott Williams & Wilkins, 2009, pp. 306-324.

[115] S. N. Adler, D. Adler-Klein and D. B. Gasbarra, A pocket manual of differential diagnosis, Lippincott Williams & Wilkins, 2008.

[116] R. D. Collins, Differential diagnosis in primary care, Lippincott Williams & Wilkins, 2008.

[117] D. Ponka and M. Kirlew, "Top 10 differential diagnosis in primary care," 2006. [Online]. Available: https://ruor.uottawa.ca/handle/10393/12856. [Accessed 14 11 2018].

[118] S. A. Wahls, "Causes and Evaluation of Chronic Dyspnea," *American Family Physician,* vol. 86, no. 2, pp. 173-180, 2012.

[119] P. Bogg, G. Beydoun and G. Low, "When to Use a Multi-Agent System?," in *PRIMA 2008*, 2008.

[120] M. Wooldridge, An Introduction to Multi Agent Systems, John Wiley & Sons, 2009.

[121] T. De Wolf and T. Holvoet, "Towards a full life-cycle methodology for engineering decentralise multi-agent systems," in *Fourth International Workshop on Agent-Oriented Methodologies*, San Diego, 2005.

[122] M. Wooldridge, "Intelligent Agents," in *Multiagent Systems*, The MIT Press, 1999.

[123] M. Wooldridge, An Introduction to MultiAgent Systems, Wiley, 2002, p. 366.

[124] L. Padgham and M. Winikoff, Developing Intelligent Agent Systems: A Practical Guide, John Wiley & Sons, 2005, p. 240 .

[125] R. Unland, "Software Agent Systems," in *Industrial Agents: Emerging Applications of Software Agents in Industry*, Elsevier Publ. Comp., 2015, pp. 3-22.

[126] H. Tianfield, J. Tian and X. Yao, "On the Architectures of Complex Multi-Agent Systems," in *The 2003 IEEE/WIC International Conference on Web Intelligence / Intelligent Agent Technology, The Workshop on "Knowledge Grid and Grid Intelligence"*, Halifax, Canada, 2003.

[127] A. Koestler, "The Gost in the Machine," Hutchinson, 1967.

[128] S. A. Rodriguez, From Analysis to Design of Holonic Multi-Agent Systems: a Framework, Methodological Guidelines and Applications, PhD Thesis, University of Technology of Belfort-Montbéliard, 2005.

[129] R. Unland, "Industrial Agents," in *Industrial Agents: Emerging Applications of Software Agents in Industry*, Elsevier Publ. Comp., 2015, pp. 23-44.

[130] V. Hilaire, A. Koukam, J. P. Gruer and J.-P. Müller, "Formal specification and prototyping of multi-agent systems," in *In Omicini, A., Tolksdorf, R., and Zambonelli,*

F., editors, Engineering Societies in the Agents' World, number 1972 in Lecture Notes in Artificial Intelligence, Springer Verlag, 2000.

[131] "System Design," Techopedia, [Online]. Available: https://www.techopedia.com/definition/29998/system-design. [Accessed 27 09 2020].

[132] R. Unland and M. Ulieru, "Swarm Intelligence and the Holonic Paradigm: a Promising Symbiosis for a Medical Diagnostic System," in *Proc. of the 9th Int. Conf. on Knowledge-Based Intelligent Information and Engineering Systems, pp. 154-160*, 2005.

[133] O. Shehory, K. Sycara, P. Chalasani and S. Jha, "Agent cloning: an approach to agnt mobility and resource allocation," *IEEE Communications Magazine 36(7),* pp. 58 - 67, 1998.

[134] D. Ye, Self-organization in multi-agent systems: theory and application, Doctor of Philosophy thesis, School of Computer Science and Software Engineering, University of Wollongong, 2013.

[135] A. Schwaiger and B. Stahmer, "Probabilistic Holons for Efficient Agent-Based Data Mining and Simulation," in *Second International Conference on Industrial Applications of Holonic and Multi-Agent Systems, HoloMAS 2005, LNAI 3593, pp. 50-63*, 2005.

[136] D. D. Corkill, "Blackboard Systems," in *AI Experts 6(9)*, 1991.

[137] D. D. Corkill, "Collaborating Software: Blackboard and Multi-Agent Systems & the Future," in *Proceedings of the International Lisp Conference*, 2003.

[138] P. Mathieu, J.-C. Routier and Y. Secq, "Principles for Dynamic Multi-agent Organizations," in *Proceedings of the 5th Pacific Rim International Workshop on Multi Agents: Intelligent Agents and Multi-Agent Systems, pp. 109-122*, 2002.

[139] O. Shehory, "Spawning Information Agents on the Web," *Intelligent Information Agents,* pp. 412-430, 1999.

[140] S. Kamboj and K. S. Decker, "Organizational self-design in worth-oriented domains," in *Handbook of Research on Multi-Agent Systems: Semantics and Dynamics of Organizational Models*, IGI Global, 2009, pp. 541-567.

[141] W. S. Richardson, M. C. Wilson, G. H. Guyatt and e. al, "Users' Guides to the Medical Literature: XV. How to use an article about disease probability for differential diagnosis," *JAMA,* vol. 281 , no. 13, p. 1214–1219, 1999.

[142] D. Altkorn, "Chapter 1-7: Rank the Differential Diagnosis," in *Symptom to Diagnosis: An Evidence-Based Guide*, 4 ed., McGraw-Hill Education, 2019, p. 612.

[143] K. Black, Business Statistics: For Contemporary Decision Making (7th Edition), John Wiley and Sons, Inc. , 2011.

[144] T. Mitchell, Machine Learning, McGraw Hill, 1997.

[145] S. Russell and P. Norvig, Artificial Intelligence: A Modern Approach (2nd ed.), Prentice Hall, 2002.

[146] A. Hertzmann and D. Fleet, Machine Learning and Data Mining Lecture Notes, CSC 411/D11, Computer Science Department, University of Toronto, 2012.

[147] S. Shalev-Shwartz and S. Ben-David, Understanding Machine Learning: From Theory to Algorithm, Cambridge University Press, 2014.

[148] S. Ventura and J. M. Luna, Pattern Mining with Evolotionary Algorithms, Springer International Publishing, 2016.

[149] K. Murphy, Machine Learning: A Probabilistic Perspective, The MIT Press, 2012.

[150] L. Rokach and O. Maimon, "Chapter 15: Clustering Methods," in *The Data Mining and Knowledge Discovery Handbook*, Springer Science + Business Media, Inc., 2005, pp. 321-352.

[151] P. Berkhin, "Survey of Clustering Data Mining Techniques, Technical Report," Accrue Software, San Jose, CA, 2002.

[152] J. Han and M. Kamber, "Data Mining Concepts and Techniques," San Francisco, CA, Morgan Kaufmann Publishers, 2001, pp. 335-391.

[153] B. S. Everitt, S. Landau, M. Leese and D. Stahl, Cluster Analysis (5th ed.), John Wiley & Sons, Ltd. , 2011.

[154] N. Beckmann, H.-P. Kriegel, R. Schneider and B. Seeger, "The R*-tree: An Efficient and Robust Access Method for Points and Rectangles," in *Proc. ACM SIGMOD Int. Conf. on Man-agement of Data*, Atlantic City, NJ, 1990.

[155] H. Darong and W. Peng, "Grid-based DBSCAN Algorithm with Referential Parameters," in *Proc. International Conference on Applied Physics and Industrial Engineering (ICAPIE-2012), Physics Procedia*, 2012.

[156] A. Smiti and Z. Elouedi, "DBSCAN-GM: An improved clustering method based on Gaussian Means and DBSCAN techniques," in *16th International Conference on Intelligent Engineer-ing Systems (INES)*, 2012.

[157] A. Karami and R. Johansson, "Choosing DBSCAN Parameters Automatically using Differential Evolution," *International Journal of Computer Applications,* vol. 91, no. 7, pp. 1-11, 2014.

[158] R. L. Ott and M. T. Longnecker, An Introduction to Statistical Methods and Data Analysis (7th Edition), Cengage Learning, 2015.

[159] G. S. Maddala, ""Outliers"," in *Introduction to Econometrics (2nd ed.)*, New York, MacMillan, 1992, p. 88–96.

[160] D. S. Shafer and Z. Zhang, Introductory Statistics, v. 1.0, Flatworld Knowledge, 2012 .

[161] F. L. Coolidge, Statistics: A Gentle Introduction, SAGE Publications, Inc., 2012, p. 458.

[162] B. Amidon, T. A. Ferryman and S. Cooley, "Data Outlier Detection using the Chebyshev Theorem," in *IEEE Aerospace Conference Proceedings*, 2005.

[163] "NIST/SEMATECH e-Handbook of Statistical Methods,," January 2018. [Online]. Available: http://www.itl.nist.gov/div898/handbook/. [Accessed 28 02 2019].

[164] R. Bellman, "A Markovian Decision Process," *Journal of Mathematics and Mechanics,* vol. 6, no. 5, pp. 679-684, 1957.

[165] M. L. Puterman, Markov Decision Processes: Diecrete Stochastic Dynamic, New York: John Wiley & Sons, Inc., 1994.

[166] H. P. van Hasselt, Insights in Reonforcment Learning, Wöhrmann Print Service, 2011.

[167] M. L. Littman, *Algorithms for Sequential Decision Making (PhD Thesis),* Providence, Rhode Island: Department of Computer Science, Brown University, 1996.

[168] P. Tandon, S. Lam, B. Shih, T. Mehta, A. Mitev and Z. Ong, Quantum Robotics: A Primer on Current Science and Future Perspectives, Morgan & Claypool Publichers, 2017.

[169] R. S. Sutton and A. G. Barto, Reinforcement Learning: An Introduction, MIT Press, 1998.

[170] A. Nuzzolo and A. Comi, "Real-time Modelling of Normative Travle Strategies on Unreliable Dynamic Transit Networks: A Framework Analysis," in *Modelling Intelligent Multi-Modal Transit Systems*, A. Nuzzolo and W. Lam, Eds., CRC Press, 2017, p. 338.

[171] D. Poole and A. Mackworth, Artificial Intelligence: Foundations of Computational Agents, Cambridge University Press, 2010.

[172] T. M. Mitchell, "Chapter 13: Reinforcement Learning," in *Machine Learning*, McGraw-Hill Science/Engineering/Math, 1997.

[173] P. Vrancx, Decentralised Reinforcement Learning in Markov Games, PhD Thesis, Vrije Universiteit Brussel, 2010.

[174] M. Coggen, "Exploration and Exploitation in Reinforcement Learning," CRA-W DMP Project at McGrill University, 2004.

[175] A. G. Barto, R. S. Sutton and C. W. Anderson, "Neuronlike adaptive elements that can solve difficult learning control problems," in *IEEE Transactions on Systems, Man, and Cybernetics, SMC-13:834-846*, 1983.

[176] R. S. Sutton, "Learning to predict by the methods of temporal differences," *Machine Learning,* vol. 3, pp. 9-44, 1988.

[177] C. J. C. H. Watkins, Learning from Delayed Rewards. Ph.D. thesis, Cambridge University, 1989.

[178] A. Schwartz, "A reinforcement learning method for maximizing undiscounted rewards," in *In Proceedings of the Tenth International Conference on Machine Learning, pages 298-305*, 1993.

[179] R. G. A. and N. M, "On-Line Q-Learning Using Connectionist Systems," Cambridge University Engineering Department, 1994.

[180] M. A. Wiering and H. van Hasselt, "Two novel on-policy reinforcement learning algortihms based on TD(λ)-methods," in *Approximate Dynamic Programming and Reinforcement Learning, ADPRL 2007*, 2007.

[181] m. Hoffman and A. Jasra, "Trans-dimensional MCMC for Bayesian Policy Learning," *Neural Information Processing Systems,* vol. 20, pp. 1-8, 2008.

[182] L. Buşoniu, R. Babuška and B. D. Schutter, "Multi-agent reinforcement learning: An overview," *Innovations in multi-agent systems and applications-1, Springer Berlin Heidelberg,* pp. 183-221, 2010.

[183] K. Tuyls and G. Weiss, "Multiagent Learning: Basics, Challenges, and Prospects," *AI Magazine,* vol. 33, no. 3, pp. 41-52, 2012.

[184] L. M. Gambardella and M. Dorigo, "Ant-Q: A Reinforcement Learning approach to the traveling salesmn problem," in *Proceedings of ML-95, 12th International Conference on Machine Learning*, 1995.

[185] N. Monekosso and a. P. Remagnino., "Phe-Q: A pheromone based Q-learning," in *Australian Joint Conference on Artificial Intelligence: AI 2001, LNAI 2256*, 2001.

[186] H. Iima, Y. Kuroe and S. Matsuda, "Swarm reinforcement learning method based on ant colony optimization," in *2010 IEEE International Conference on Systems Man and Cybernetics (SMC)*, 2010.

[187] G. Beni and J. Wang, "Swarm Intelligence in Cellular Robotic Systems," *Robots and Biological Systems: Towards a New Bionics?,* pp. 703-712, 1989.

[188] M. Dorigo, D. Floreano, L. M. Gambardella, F. Mondada, S. Nolfi, T. Baaboura, M. Birattari and e. al., "Swarmanoid: a novel concept for the study of heterogeneous robotic swarms," in *IEEE Robotics & Automation Magazine*, 2013.

[189] "Unanimous AI," [Online]. Available: https://unanimous.ai/. [Accessed 28 02 2019].

[190] M. Dorigo and M. Birattari, *Swarm intelligence,* vol. 2(9):1462, Scholarpedia, 2007.

[191] M. A. Montes de Oca, J. Pena, T. Stützle, C. Pinciroli and M. Dorigo, "Heterogeneous Particle Swarm Optimizers," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC)*, 2009.

[192] K. Szwaykowska, L. M.-y.-T. Romero and I. B. Schwartz, "Collective motions of heterogeneous swarms," in *IEEE Transactions on Automation Science and Engineering*, 2015.

[193] M. Kumar, D. P. Garg and V. Kumar, "Segregation of heterogeneous units in a swarm of robotic agents," in *IEEE Transactions on Automatic Control*, 2010.

[194] E. Ferante, "A Control Architecture for a Heterogenous Swarm of Robots," 2009.

[195] C. Pinciroli, R. O'Grady, A. L. Christensen and M. Dorigo, "Coordinating heterogeneous swarms through minimal communication among homogeneous sub-swarms," in *In International Conference on Swarm Intelligence*, 2010.

[196] A. P. Engelbrecht, "Heterogeneous particle swarm optimization," in *International Conference on Swarm Intelligence*, 2010.

[197] X. Ma and H. Sayama, "Hierarchical heterogeneous particle swarm optimization: algorithms and evaluations," *International Journal of Parallel Emergent and Distributed Systems,* vol. 31, no. 5, pp. 504-516, 2015.

[198] M. Hong, J. J. Jung and M. Lee, "Social Affinity-Based Group Recommender System," in *International Conference on Context-Aware Systems and Applications*, 2015.

[199] M. Hong, J. J. Jung and D. Camacho, "GRSAT: a novel method on group recommendation by social affinity and trustworthiness," *Cybernetics and Systems,* vol. 48, no. 3, pp. 140-161, 2017.

[200] "APA Dictionary of Psychology," [Online]. Available: https://dictionary.apa.org. [Accessed 10 03 2019].

[201] C. Hill and E. O'Hara O'Connor, "A Cognitive Theory of Trust," *Wash. U. L. Rev.,* vol. 84, no. 7, pp. 1717-1795, 2006.

[202] K. Chatterjee, R. Majumdar and T. A. Henzinger, "Markov decision processes with multiple objectives," in *Annual Symposium on Theoretical Aspects of Computer Science* , 2006.

[203] D. M. Roijers, P. Vamplew, S. Whiteson and R. Dazeley, "A survey of multi-objective sequential decision-making," *Journal of Artificial Intelligence Research,* vol. 48, pp. 67-113, 2013.

[204] D. J. Lizotte and E. B. Laber, "Multi-Objective Markov Decision Processes for Data-Driven Decision Support," *Journal of Machine Learning Research,* vol. 17, no. 1, pp. 7378-7405, 2016.

[205] S. Dehuri, A. K. Jagadev and M. Panda, Multi-objective Swarm Intelligence: Theoretical Advances and Applications, Springer, 2015.

[206] A. Ghane-Kanafi and E. Khorram, "A new scalarization method for finding the efficient frontier innon-convex multi-objective problems," *Applied Mathematical Modelling,* vol. 39, no. 23-24, pp. 7483-7498, 2015.

[207] C. Watkins and P. Dayan, "Technical Note: Q-Learning," *Machine Learning,* vol. 8, p. 279–292, 1992.

[208] F. S. Melo, "Convergence of Q-learning: A simple proof," Institute Of Systems and Robotics, Tech. Rep (2001): 1-4., 2001.

[209] T. Jaakkola, M. I. Jordan and S. P. Singh, "On the Convergence of Stochastic Iterative Dynamic Programming Algorithms," Massachusetts Institute of Technology, Artificial Intelligence Laboratory, A.I. Memo No. 1441, 1993.

[210] T. Jaakkola, M. I. Jordan and S. P. Singh, "On the convergence of stochastic iterative dynamic programming algorithms," *Neural computation,* vol. 6, no. 6, pp. 1185-1201, 1994.

[211] H. Kushner and D. Clark, Stochastic Approximation Methods for Constrained and Unconstrained Systems, Berlin, Germany: Springer-Verlag, 1978.

[212] D. Dewey, "Reinforcement Learning and the Reward Engineering Principle," in *2014 AAAI Spring Symposium Series*, 2014.

[213] M. Feraud and S. Galland, "First Comparison of SARL to Other Agent-Programming Languages and Frameworks," *Procedia Computer Science ,* vol. 109 , no. C, pp. 1080-1085, 2017.

[214] "Jason," [Online]. Available: http://jason.sourceforge.net/wp/. [Accessed 10 03 2019].

[215] "Jade: JAVA Agent DEvelopment Framework," [Online]. Available: http://jade.tilab.com/. [Accessed 10 03 2019].

[216] "NetLogo," [Online]. Available: https://ccl.northwestern.edu/netlogo/. [Accessed 10 03 2019].

[217] Janus Platform, "Janus Agent and Holonic Platform," [Online]. Available: http://www.sarl.io/runtime/janus/. [Accessed 10 03 2019].

[218] "GAMA Platform - Tutorials," [Online]. Available: https://gama-platform.github.io/wiki/Home. [Accessed 25 03 2019].

[219] A. Drogoul, D. Vanbergue and T. Meurisse, "Multi-Agent Based Simulation: Where are the Agents?," in *Multi-Agent-Based Simulation II. MABS 2002*, Berlin, Heidelberg, 2003.

[220] "Mayo Clinic," [Online]. Available: https://www.mayoclinic.org/. [Accessed 23 01 2018].

[221] "University of North Carolina - School of Medicine, History and Physical Examination (H&P) Examples," [Online]. Available: https://www.med.unc.edu/medclerk/education/grading/history-and-physical-examination-h-p-examples. [Accessed 24 January 2018].

[222] J. Cochrane, "Metastatic Lung Cancer to the Common Bile Duct Presenting as Obstructive Jaundice," *Journal of Hepatology and Gastrointestinal Disorders,* vol. 2, no. 121, 2016.

[223] O. Lemaire, C. Paul and L. Zabraniecki, "Distal Madelung-Launois-Bensaude disease: an unusual differential diagnosis of acromalic arthritis," *Clinical and Experimental Rheumatology,* vol. 26, pp. 351-353, 2008.

[224] L. Bordi, E. Nicastri, L. Scorzolini, A. Di Caro, M. R. Capobianchi, C. Castilletti, E. Lalle and o. b. o. I. C.-1. s. g. a. C. Centers, "Differential diagnosis of illness in patients under investigation for the novel coronavirus (SARS-CoV-2), Italy, February 2020," *Eurosurveillance,* vol. 25, no. 8, pp. 2-5, 2020.

[225] V. Hilarie, A. Koukam and S. Rodrigue, "An adaptive agent architecture for holonic multiagent system," in *ACM Trans on Autonomous Adaptive Systems 3, no. 1*, 2008.

[226] M. Abdoos, N. Mozayani and A. L. Bazzan, "Towards reinforcement learning for holonic multi-agent systems," *Intell. Data Anal. ,* vol. 19, no. 2, pp. 211-232, 2015.

[227] R. Polikar, "Ensemble learning," 2009. [Online]. Available: http://www.scholarpedia.org/article/Ensemble_learning. [Accessed 07 04 2019].

[228] R. A. Jacobs, M. I. Jordan, S. J. Nowlan and G. E. Hinton, "Adaptive mixtures of local experts," *Neural Computation,* vol. 3, pp. 79-87, 1991.

[229] M. I. Jordan and R. A. Jacobs, "Hierarchical mixtures of experts and the EM algorithm," *Neural Computation,* vol. 6, no. 2, pp. 181-214, 1994.

[230] J. Read, B. Pfahringer, G. Holmes and E. Frank, "Classifier Chains for Multi-label Classification," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 2009.

[231] W. Liu and I. Tsang, "On the optimality of classifier chain for multi-label classification," *Advances in Neural Information Processing Systems,* pp. 712-720, 2015.

[232] M. P. Huget, "Agent Uml Notation for Multiagent System Design," *IEEE Internet Computing,* vol. 8, no. 4, pp. 63-71, 2004.

[233] "The Unified Modeling Language," [Online]. Available: http://www.omg.org/spec/UML/2.5/. [Accessed 2017].

[234] "UML Class and Object Diagrams Overview," [Online]. Available: http://www.uml-diagrams.org/class-reference.html. [Accessed 2017].

[235] "UML Activity Diagrams Reference," [Online]. Available: http://www.uml-diagrams.org/activity-diagrams-reference.html. [Accessed 2017].

[236] C. Byrne, D. Sherry, L. Mercincavage, D. Johnston, E. Pan and G. Schiff, "Advancing Clinical Decision Support," 2011.

[237] S. Pantilat, ""What is a Hospitalist?". The Hospitalist. The Society of Hospital Medicine.," February 2006. [Online]. Available: https://www.the-hospitalist.org/hospitalist/article/123072/what-hospitalist. [Accessed 25 10 2018].

[238] "Cognitive Computing," TechTarget, [Online]. Available: https://searchenterpriseai.techtarget.com/definition/cognitive-computing. [Accessed 27 10 2018].

[239] "Disease Eradication," The History of Vaccines, [Online]. Available: https://www.historyofvaccines.org/content/articles/disease-eradication. [Accessed 24 01 2019].

[240] M. Dorigo, Optimization, Learning and Natural Algorithms, PhD thesis, Politecnico di Milano, Italy, 1992.

[241] R. B. Myerson, Game Theory: Analysis of Conflict, Harvard University Press, 1991.

[242] "American Psychological Association," [Online]. Available: https://www.apa.org/about/index. [Accessed 10 03 2019].

[243] "UMMISCO," [Online]. Available: http://www.ummisco.fr/. [Accessed 01 04 2019].

Appendix A: AUML Notation Reference

This appendix provides a brief introduction to those parts of the AUML notation which are used in this project, i.e., the class diagram and the goal diagram. The AUML notation is a UML profile dedicated to agents that exploits the UML extension capabilities.

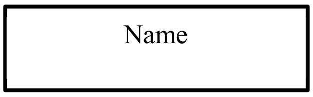
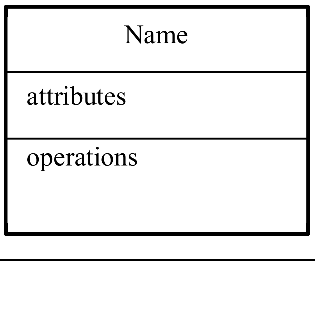
Class diagrams typically drive object system design and cannot express all the richness of agent behaviours. As a result, the AUML community has defined an agent shell as a UML classifier and lets designers fill this shell with building blocks that represent specific features such as actions, events, or protocols [232]. The next step was to define and integrate goals, plans, and actions within the agent shell. For this purpose, the AUML has defined the AUML goal diagram, which is the adapted UML activity diagram for agent goals, plans, and actions (for more information please refer to [232].

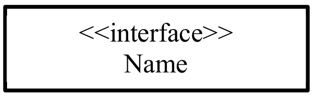
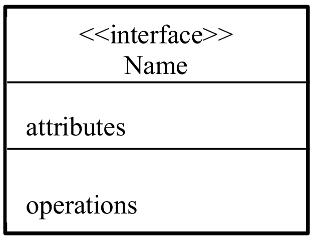
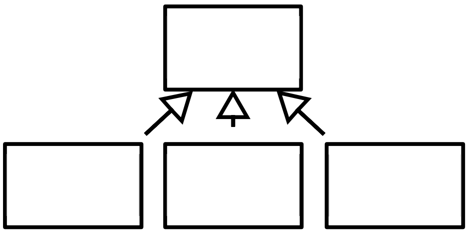
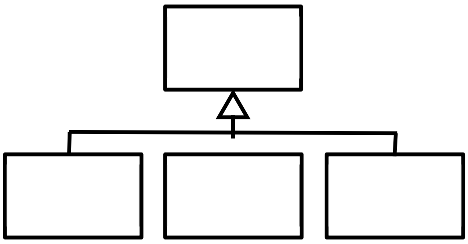
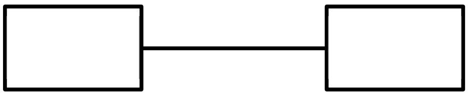
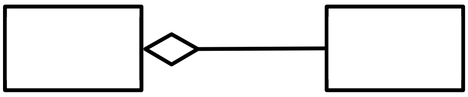
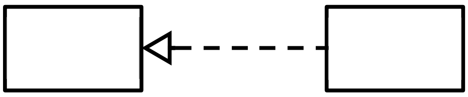
This section includes a UML notation reference for the class and activity diagrams presented in this work. The current version of the Unified Modeling Language is UML 2.5, released in June 2015 [233].

UML Class Diagram Notations

The notations and descriptions presented here are according to [234].

Table 23. UML Class Diagram Notations

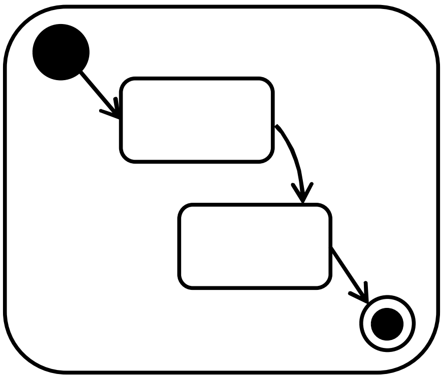
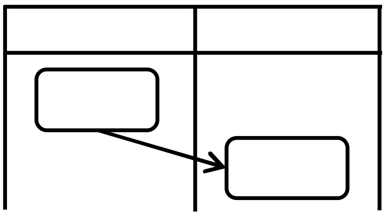
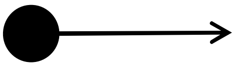
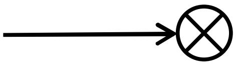
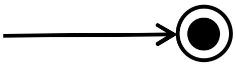
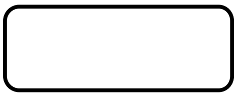
Notation	Description
	A class is a classifier which describes a set of objects that share the same features, constraints, and semantics (meaning). A class is shown as a solid-outline rectangle containing the class name.
	When class is shown with three compartments, the middle compartment holds a list of attributes and the bottom compartment holds a list of operations. Attributes and operations should be left justified in plain face, with the first letter of the names in lower case.

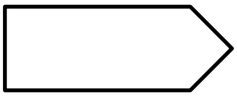
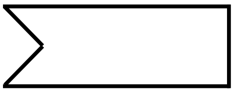
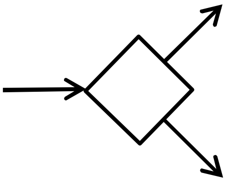
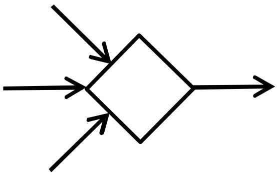
	An interface is a classifier that represent a declaration of a set of public features and obligations that together constitute a coherent service. An interface specifies a contract; any instance of a classifier that realizes the interface shall fulfill that contract. An interface may be shown using a rectangle symbol with the keyword «interface» preceding the name.
	An Interface may be designated using the default notation for Classifier with the keyword ≪interface≫.
	A Generalization is shown as a line with a hollow triangle as an arrowhead between the symbols representing the involved classifiers. The arrowhead points to the symbol representing the general classifier. This notation is referred to as the "separate target style."
	Multiple Generalization relationships that reference the same general classifier can also be connected in the "shared target style."
	Association is a relationship between classifiers. Binary association relates two typed instances. It is normally rendered as a solid line connecting two classifiers, or a solid line connecting a single classifier to itself (the two ends are distinct).
	Aggregation (aka shared aggregation) is shown as binary association decorated with a hollow diamond as a terminal adornment at the aggregate end of the association line.
	Interface realization dependency is denoted with interface realization arrow. The classifier at the tail of the arrow implements the interface at the head of the arrow.

UML Activity Diagram Notations

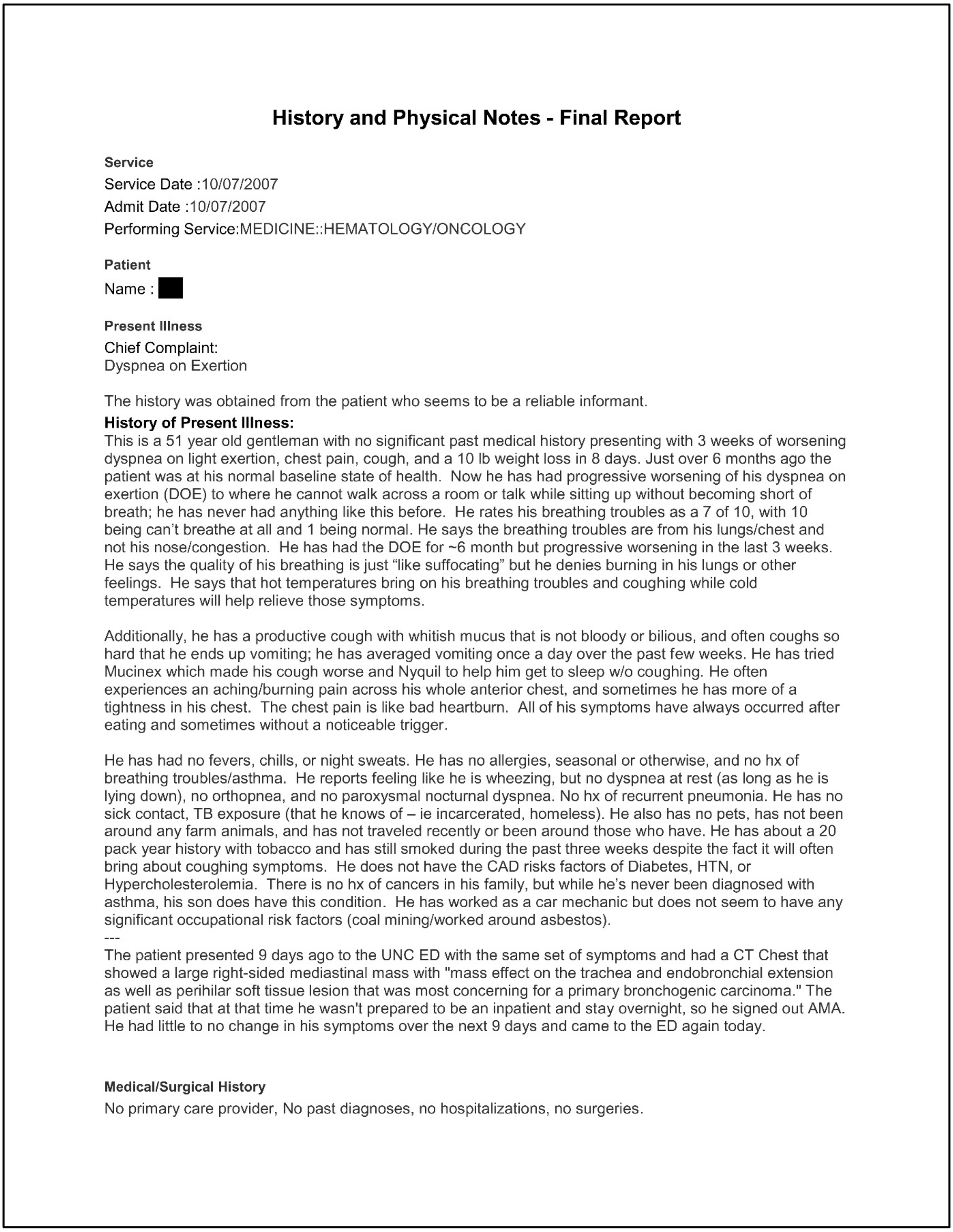
The notations and descriptions presented here are according to [235].

Table 24. UML Activity Diagram Notations

Notation	**Description**
	Activity is parameterized behavior represented as coordinated flow of actions. Activity could be rendered as round-cornered rectangle with activity name in the upper left corner and nodes and edges of the activity inside.
	An activity partition is activity group for actions that have some common characteristic. Activity partition may be shown using a swimlane notation - with two, usually parallel lines, either horizontal or vertical, and a name labeling the partition in a box at one end.
	Initial node is a control node at which flow starts when the activity is invoked. Activity may have more than one initial node. Initial nodes are shown as a small solid circle.
	Flow final node is a control final node that terminates a flow. The notation for flow final node is small circle with X inside.
	Activity final node is a control final node that stops all flows in an activity. Activity final nodes are shown as a solid circle with a hollow circle inside. It can be thought of as a goal notated as "bull's eye," or target.
	Actions are notated as round-cornered rectangles. The name of the action or other description of it may appear in the symbol.

	Send signal action is an invocation action that creates a signal from its inputs, and transmits it to the specified target object, where it may cause the firing of a state machine transition or the execution of an activity. Send signal action is notated as convex pentagon. Note, that the name of the action corresponds to the name of signal class it sends. Target object is not specified with this notation.
	Accept event action is notated with a concave pentagon. If an accept event action has no incoming edges, then the action starts when the containing activity or structured node does, whichever most immediately contains the action. In addition, an accept event action with no incoming edges remains enabled after it accepts an event. It does not terminate after accepting an event and outputting a value but continues to wait for other events. An action whose trigger is a signal event is informally called accept signal action. It corresponds to send signal action.
	Decision node is a control node that accepts tokens on one or two incoming edges and selects one outgoing edge from one or more outgoing flows. The notation for a decision node is a diamond-shaped symbol.
	Merge node is a control node that brings together multiple incoming alternate flows to accept single outgoing flow. There is no joining of tokens. Merge should not be used to synchronize concurrent flows. The notation for a merge node is a diamond-shaped symbol with two or more edges entering it and a single activity edge leaving it.

Appendix B: History and Physical Examination Example



History and Physical Notes - Final Report

Service

Service Date :10/07/2007

Admit Date :10/07/2007

Performing Service:MEDICINE::HEMATOLOGY/ONCOLOGY

Patient

Name : [redacted]

Present Illness

Chief Complaint:
Dyspnea on Exertion

The history was obtained from the patient who seems to be a reliable informant.

History of Present Illness:
This is a 51 year old gentleman with no significant past medical history presenting with 3 weeks of worsening dyspnea on light exertion, chest pain, cough, and a 10 lb weight loss in 8 days. Just over 6 months ago the patient was at his normal baseline state of health. Now he has had progressive worsening of his dyspnea on exertion (DOE) to where he cannot walk across a room or talk while sitting up without becoming short of breath; he has never had anything like this before. He rates his breathing troubles as a 7 of 10, with 10 being can't breathe at all and 1 being normal. He says the breathing troubles are from his lungs/chest and not his nose/congestion. He has had the DOE for ~6 month but progressive worsening in the last 3 weeks. He says the quality of his breathing is just "like suffocating" but he denies burning in his lungs or other feelings. He says that hot temperatures bring on his breathing troubles and coughing while cold temperatures will help relieve those symptoms.

Additionally, he has a productive cough with whitish mucus that is not bloody or bilious, and often coughs so hard that he ends up vomiting; he has averaged vomiting once a day over the past few weeks. He has tried Mucinex which made his cough worse and Nyquil to help him get to sleep w/o coughing. He often experiences an aching/burning pain across his whole anterior chest, and sometimes he has more of a tightness in his chest. The chest pain is like bad heartburn. All of his symptoms have always occurred after eating and sometimes without a noticeable trigger.

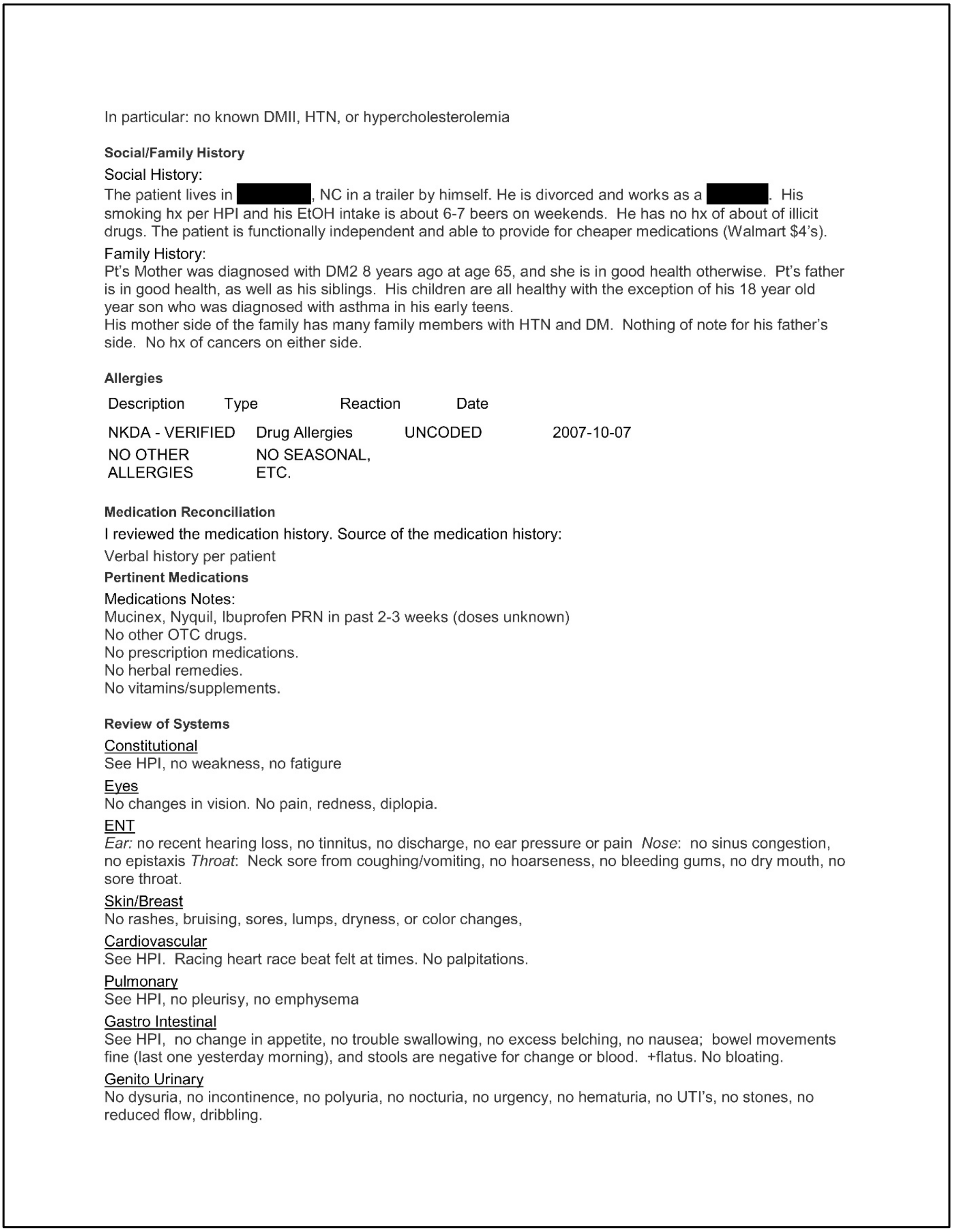
He has had no fevers, chills, or night sweats. He has no allergies, seasonal or otherwise, and no hx of breathing troubles/asthma. He reports feeling like he is wheezing, but no dyspnea at rest (as long as he is lying down), no orthopnea, and no paroxysmal nocturnal dyspnea. No hx of recurrent pneumonia. He has no sick contact, TB exposure (that he knows of – ie incarcerated, homeless). He also has no pets, has not been around any farm animals, and has not traveled recently or been around those who have. He has about a 20 pack year history with tobacco and has still smoked during the past three weeks despite the fact it will often bring about coughing symptoms. He does not have the CAD risks factors of Diabetes, HTN, or Hypercholesterolemia. There is no hx of cancers in his family, but while he's never been diagnosed with asthma, his son does have this condition. He has worked as a car mechanic but does not seem to have any significant occupational risk factors (coal mining/worked around asbestos).

The patient presented 9 days ago to the UNC ED with the same set of symptoms and had a CT Chest that showed a large right-sided mediastinal mass with "mass effect on the trachea and endobronchial extension as well as perihilar soft tissue lesion that was most concerning for a primary bronchogenic carcinoma." The patient said that at that time he wasn't prepared to be an inpatient and stay overnight, so he signed out AMA. He had little to no change in his symptoms over the next 9 days and came to the ED again today.

Medical/Surgical History

No primary care provider, No past diagnoses, no hospitalizations, no surgeries.

Figure 66. H&P example – page 1



In particular: no known DMII, HTN, or hypercholesterolemia

Social/Family History

Social History:

The patient lives in [redacted], NC in a trailer by himself. He is divorced and works as a [redacted]. His smoking hx per HPI and his EtOH intake is about 6-7 beers on weekends. He has no hx of about of illicit drugs. The patient is functionally independent and able to provide for cheaper medications (Walmart $4's).

Family History:

Pt's Mother was diagnosed with DM2 8 years ago at age 65, and she is in good health otherwise. Pt's father is in good health, as well as his siblings. His children are all healthy with the exception of his 18 year old year son who was diagnosed with asthma in his early teens.

His mother side of the family has many family members with HTN and DM. Nothing of note for his father's side. No hx of cancers on either side.

Allergies

Description	Type	Reaction	Date
NKDA - VERIFIED	Drug Allergies	UNCODED	2007-10-07
NO OTHER ALLERGIES	NO SEASONAL, ETC.		

Medication Reconciliation

I reviewed the medication history. Source of the medication history:

Verbal history per patient

Pertinent Medications

Medications Notes:

Mucinex, Nyquil, Ibuprofen PRN in past 2-3 weeks (doses unknown)
No other OTC drugs.
No prescription medications.
No herbal remedies.
No vitamins/supplements.

Review of Systems

Constitutional

See HPI, no weakness, no fatigure

Eyes

No changes in vision. No pain, redness, diplopia.

ENT

Ear: no recent hearing loss, no tinnitus, no discharge, no ear pressure or pain *Nose*: no sinus congestion, no epistaxis *Throat*: Neck sore from coughing/vomiting, no hoarseness, no bleeding gums, no dry mouth, no sore throat.

Skin/Breast

No rashes, bruising, sores, lumps, dryness, or color changes,

Cardiovascular

See HPI. Racing heart race beat felt at times. No palpitations.

Pulmonary

See HPI, no pleurisy, no emphysema

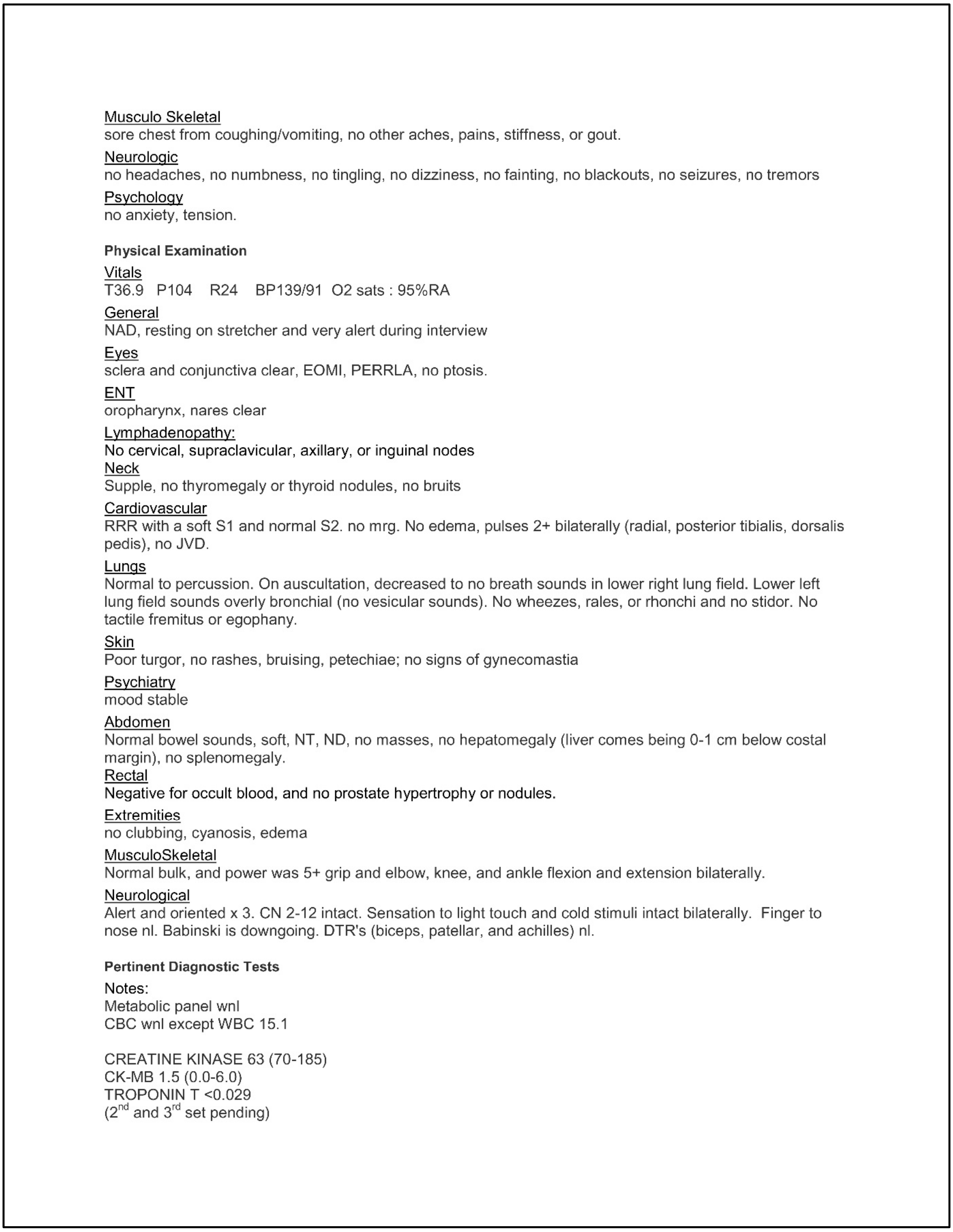
Gastro Intestinal

See HPI, no change in appetite, no trouble swallowing, no excess belching, no nausea; bowel movements fine (last one yesterday morning), and stools are negative for change or blood. +flatus. No bloating.

Genito Urinary

No dysuria, no incontinence, no polyuria, no nocturia, no urgency, no hematuria, no UTI's, no stones, no reduced flow, dribbling.

Figure 67. H&P example – page 2



Musculo Skeletal
sore chest from coughing/vomiting, no other aches, pains, stiffness, or gout.

Neurologic
no headaches, no numbness, no tingling, no dizziness, no fainting, no blackouts, no seizures, no tremors

Psychology
no anxiety, tension.

Physical Examination

Vitals
T36.9 P104 R24 BP139/91 O2 sats : 95%RA

General
NAD, resting on stretcher and very alert during interview

Eyes
sclera and conjunctiva clear, EOMI, PERRLA, no ptosis.

ENT
oropharynx, nares clear

Lymphadenopathy:
No cervical, supraclavicular, axillary, or inguinal nodes

Neck
Supple, no thyromegaly or thyroid nodules, no bruits

Cardiovascular
RRR with a soft S1 and normal S2. no mrg. No edema, pulses 2+ bilaterally (radial, posterior tibialis, dorsalis pedis), no JVD.

Lungs
Normal to percussion. On auscultation, decreased to no breath sounds in lower right lung field. Lower left lung field sounds overly bronchial (no vesicular sounds). No wheezes, rales, or rhonchi and no stidor. No tactile fremitus or egophany.

Skin
Poor turgor, no rashes, bruising, petechiae; no signs of gynecomastia

Psychiatry
mood stable

Abdomen
Normal bowel sounds, soft, NT, ND, no masses, no hepatomegaly (liver comes being 0-1 cm below costal margin), no splenomegaly.

Rectal
Negative for occult blood, and no prostate hypertrophy or nodules.

Extremities
no clubbing, cyanosis, edema

MusculoSkeletal
Normal bulk, and power was 5+ grip and elbow, knee, and ankle flexion and extension bilaterally.

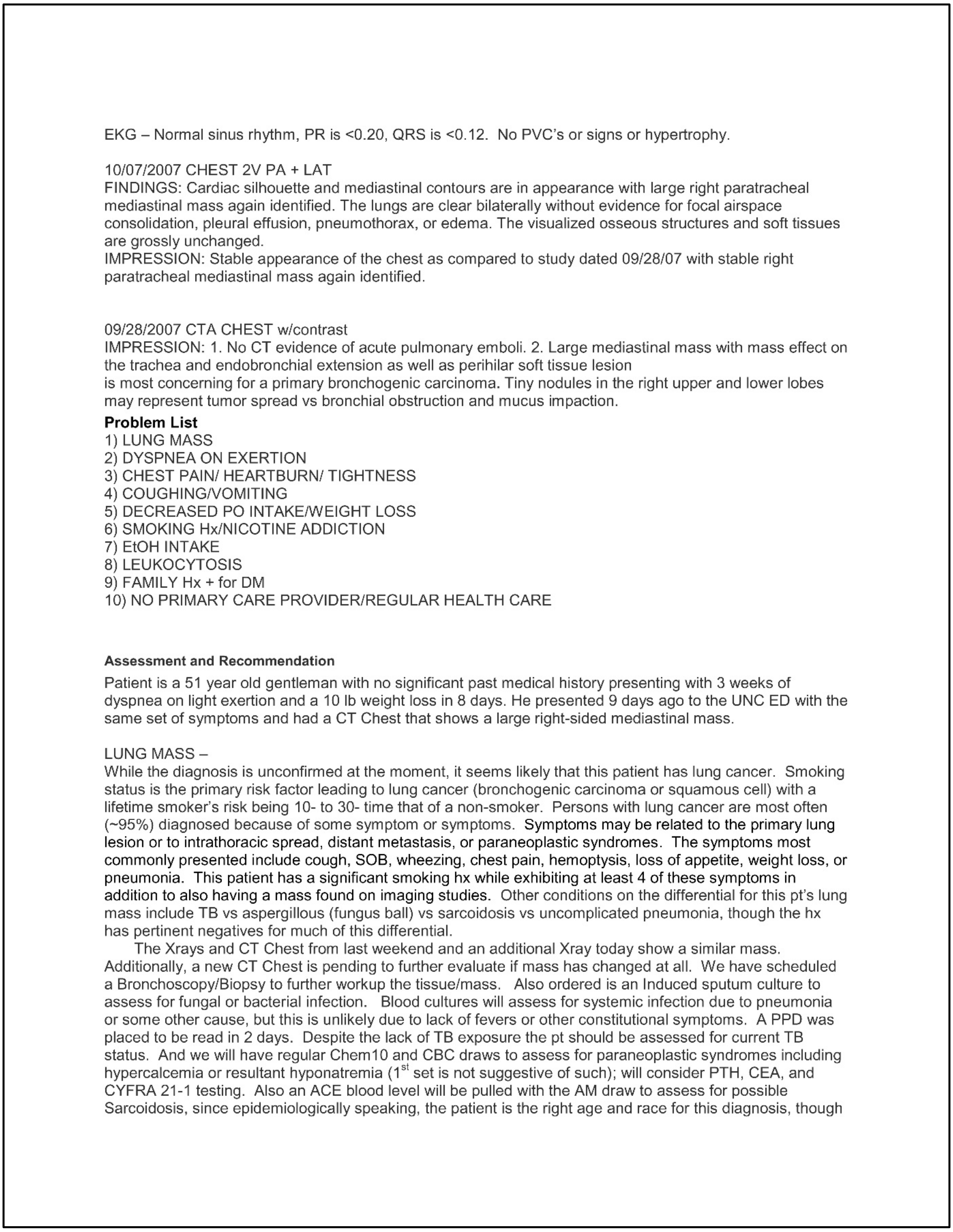
Neurological
Alert and oriented x 3. CN 2-12 intact. Sensation to light touch and cold stimuli intact bilaterally. Finger to nose nl. Babinski is downgoing. DTR's (biceps, patellar, and achilles) nl.

Pertinent Diagnostic Tests

Notes:
Metabolic panel wnl
CBC wnl except WBC 15.1

CREATINE KINASE 63 (70-185)
CK-MB 1.5 (0.0-6.0)
TROPONIN T <0.029
(2nd and 3rd set pending)

Figure 68. H&P example – page 3



EKG – Normal sinus rhythm, PR is <0.20, QRS is <0.12. No PVC's or signs or hypertrophy.

10/07/2007 CHEST 2V PA + LAT
FINDINGS: Cardiac silhouette and mediastinal contours are in appearance with large right paratracheal mediastinal mass again identified. The lungs are clear bilaterally without evidence for focal airspace consolidation, pleural effusion, pneumothorax, or edema. The visualized osseous structures and soft tissues are grossly unchanged.
IMPRESSION: Stable appearance of the chest as compared to study dated 09/28/07 with stable right paratracheal mediastinal mass again identified.

09/28/2007 CTA CHEST w/contrast
IMPRESSION: 1. No CT evidence of acute pulmonary emboli. 2. Large mediastinal mass with mass effect on the trachea and endobronchial extension as well as perihilar soft tissue lesion
is most concerning for a primary bronchogenic carcinoma. Tiny nodules in the right upper and lower lobes may represent tumor spread vs bronchial obstruction and mucus impaction.

Problem List

1) LUNG MASS
2) DYSPNEA ON EXERTION
3) CHEST PAIN/ HEARTBURN/ TIGHTNESS
4) COUGHING/VOMITING
5) DECREASED PO INTAKE/WEIGHT LOSS
6) SMOKING Hx/NICOTINE ADDICTION
7) EtOH INTAKE
8) LEUKOCYTOSIS
9) FAMILY Hx + for DM
10) NO PRIMARY CARE PROVIDER/REGULAR HEALTH CARE

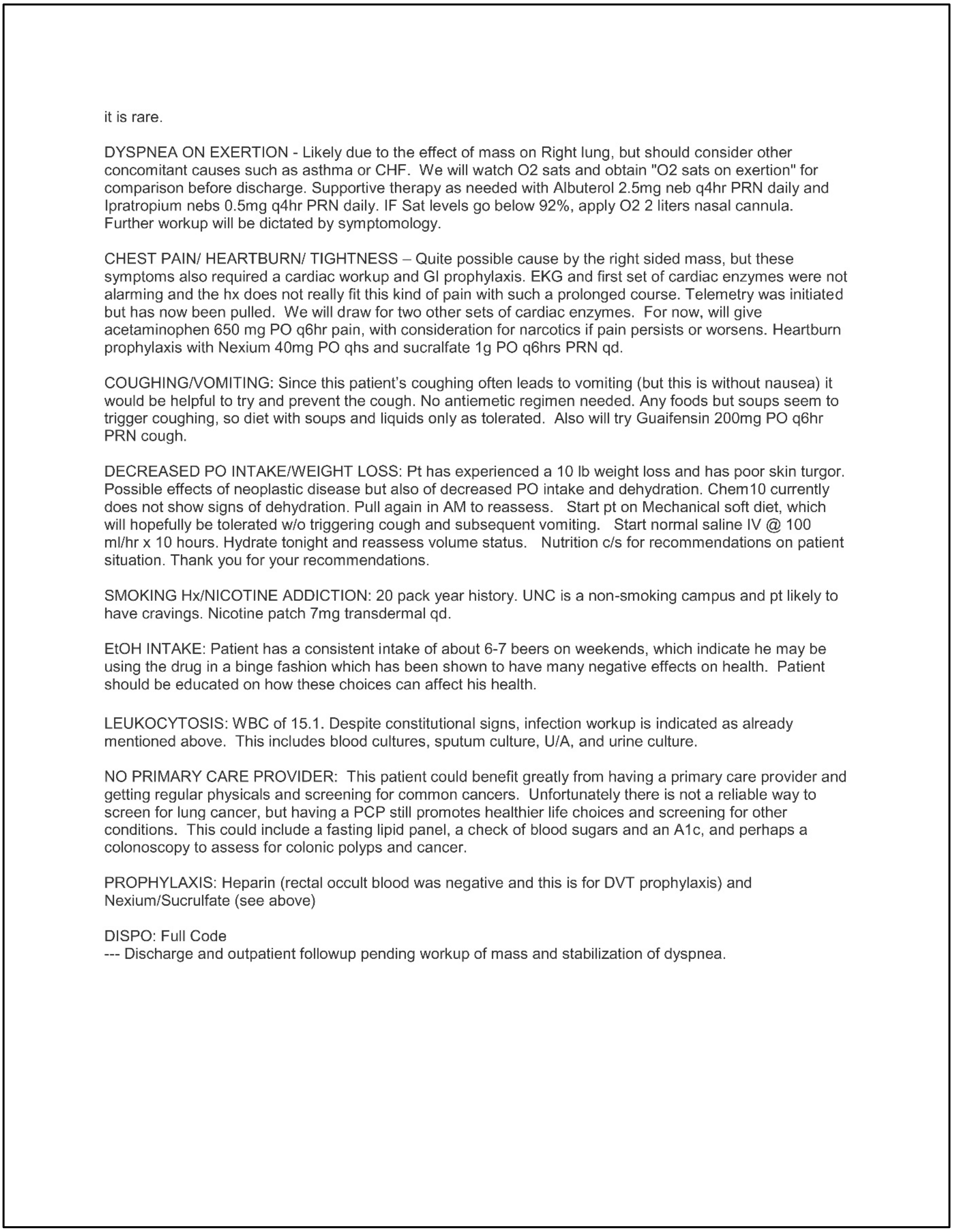
Assessment and Recommendation

Patient is a 51 year old gentleman with no significant past medical history presenting with 3 weeks of dyspnea on light exertion and a 10 lb weight loss in 8 days. He presented 9 days ago to the UNC ED with the same set of symptoms and had a CT Chest that shows a large right-sided mediastinal mass.

LUNG MASS –
While the diagnosis is unconfirmed at the moment, it seems likely that this patient has lung cancer. Smoking status is the primary risk factor leading to lung cancer (bronchogenic carcinoma or squamous cell) with a lifetime smoker's risk being 10- to 30- time that of a non-smoker. Persons with lung cancer are most often (~95%) diagnosed because of some symptom or symptoms. Symptoms may be related to the primary lung lesion or to intrathoracic spread, distant metastasis, or paraneoplastic syndromes. The symptoms most commonly presented include cough, SOB, wheezing, chest pain, hemoptysis, loss of appetite, weight loss, or pneumonia. This patient has a significant smoking hx while exhibiting at least 4 of these symptoms in addition to also having a mass found on imaging studies. Other conditions on the differential for this pt's lung mass include TB vs aspergillous (fungus ball) vs sarcoidosis vs uncomplicated pneumonia, though the hx has pertinent negatives for much of this differential.

The Xrays and CT Chest from last weekend and an additional Xray today show a similar mass. Additionally, a new CT Chest is pending to further evaluate if mass has changed at all. We have scheduled a Bronchoscopy/Biopsy to further workup the tissue/mass. Also ordered is an Induced sputum culture to assess for fungal or bacterial infection. Blood cultures will assess for systemic infection due to pneumonia or some other cause, but this is unlikely due to lack of fevers or other constitutional symptoms. A PPD was placed to be read in 2 days. Despite the lack of TB exposure the pt should be assessed for current TB status. And we will have regular Chem10 and CBC draws to assess for paraneoplastic syndromes including hypercalcemia or resultant hyponatremia (1st set is not suggestive of such); will consider PTH, CEA, and CYFRA 21-1 testing. Also an ACE blood level will be pulled with the AM draw to assess for possible Sarcoidosis, since epidemiologically speaking, the patient is the right age and race for this diagnosis, though

Figure 69. H&P example – page 4



it is rare.

DYSPNEA ON EXERTION - Likely due to the effect of mass on Right lung, but should consider other concomitant causes such as asthma or CHF. We will watch O2 sats and obtain "O2 sats on exertion" for comparison before discharge. Supportive therapy as needed with Albuterol 2.5mg neb q4hr PRN daily and Ipratropium nebs 0.5mg q4hr PRN daily. IF Sat levels go below 92%, apply O2 2 liters nasal cannula. Further workup will be dictated by symptomology.

CHEST PAIN/ HEARTBURN/ TIGHTNESS – Quite possible cause by the right sided mass, but these symptoms also required a cardiac workup and GI prophylaxis. EKG and first set of cardiac enzymes were not alarming and the hx does not really fit this kind of pain with such a prolonged course. Telemetry was initiated but has now been pulled. We will draw for two other sets of cardiac enzymes. For now, will give acetaminophen 650 mg PO q6hr pain, with consideration for narcotics if pain persists or worsens. Heartburn prophylaxis with Nexium 40mg PO qhs and sucralfate 1g PO q6hrs PRN qd.

COUGHING/VOMITING: Since this patient's coughing often leads to vomiting (but this is without nausea) it would be helpful to try and prevent the cough. No antiemetic regimen needed. Any foods but soups seem to trigger coughing, so diet with soups and liquids only as tolerated. Also will try Guaifensin 200mg PO q6hr PRN cough.

DECREASED PO INTAKE/WEIGHT LOSS: Pt has experienced a 10 lb weight loss and has poor skin turgor. Possible effects of neoplastic disease but also of decreased PO intake and dehydration. Chem10 currently does not show signs of dehydration. Pull again in AM to reassess. Start pt on Mechanical soft diet, which will hopefully be tolerated w/o triggering cough and subsequent vomiting. Start normal saline IV @ 100 ml/hr x 10 hours. Hydrate tonight and reassess volume status. Nutrition c/s for recommendations on patient situation. Thank you for your recommendations.

SMOKING Hx/NICOTINE ADDICTION: 20 pack year history. UNC is a non-smoking campus and pt likely to have cravings. Nicotine patch 7mg transdermal qd.

EtOH INTAKE: Patient has a consistent intake of about 6-7 beers on weekends, which indicate he may be using the drug in a binge fashion which has been shown to have many negative effects on health. Patient should be educated on how these choices can affect his health.

LEUKOCYTOSIS: WBC of 15.1. Despite constitutional signs, infection workup is indicated as already mentioned above. This includes blood cultures, sputum culture, U/A, and urine culture.

NO PRIMARY CARE PROVIDER: This patient could benefit greatly from having a primary care provider and getting regular physicals and screening for common cancers. Unfortunately there is not a reliable way to screen for lung cancer, but having a PCP still promotes healthier life choices and screening for other conditions. This could include a fasting lipid panel, a check of blood sugars and an A1c, and perhaps a colonoscopy to assess for colonic polyps and cancer.

PROPHYLAXIS: Heparin (rectal occult blood was negative and this is for DVT prophylaxis) and Nexium/Sucrulfate (see above)

DISPO: Full Code
--- Discharge and outpatient followup pending workup of mass and stabilization of dyspnea.

Figure 70. H&P example – page 5