Universitatea
Transilvania
din Brașov

INTERDISCIPLINARY DOCTORAL SCHOOL

Faculty of Electrical Engineering and Computer Science

Anata-Flavia IONESCU

VIRTUAL ORGANIZATIONS AS FACILITATORS OF THE COLLABORATION BETWEEN ACADEMIA AND THE ECONOMIC ENVIRONMENT

ABSTRACT

PhD Supervisors

Prof. dr. Dorin-Mircea POPOVICI

Prof. dr. Raluca-Ileana VERNIC

BRAȘOV, 2022

Universitatea Transilvania din Brașov

To ....

Reviewing board of the PhD thesis

Appointed by order of the Rector of Universitatea Transilvania din Brașov

Nr. .... from ....

PRESIDENT: Prof. dr. Mihai IVANOVICI, Universitatea Transilvania din Brașov

PHD SUPERVISORS:

Prof. dr. Dorin-Mircea POPOVICI, Universitatea Transilvania din Brașov, Universitatea Ovidius din Constanța

Prof. dr. Raluca-Ileana VERNIC, Universitatea Ovidius din Constanța

OFFICIAL REVIEWERS:

Prof. dr. Luca-Dan ȘERBĂNAȚI, Universitatea Politehnică București

Prof. dr. Dorian GORGAN, Universitatea Tehnică din Cluj-Napoca

Prof. dr. Sorin MORARU, Universitatea Transilvania din Brașov

Date, time and place of the public defense of the PhD thesis: ...., time ...., room ....

Any remarks or observations on the content of the thesis will be sent electronically, in due time, to anata.ionescu@unitbv.ro.

At the same time, we invite you to take part in the public defense of the PhD thesis.

Thank you.

CONTENT

List of Abbreviations

ABM	Agent-Based Modeling
AI	Artificial Intelligence
CH	Competence Holon
CPU	Central Processing Unit
HMAS	Holonic Multiagent System
ISP	Instructor Selection Problem
MAS	Multiagent System
MOEA	Multiobjective Evolutionary Algorithm
MOP	Multiobjective Optimization Problem
PSP	Partner Selection Problem
SH	Scheduling Holon
UML	Unified Modeling Language
VE	Virtual Enterprise
VO	Virtual Organization

Chapter 1

Introduction

Nowadays, it is difficult even for tertiary education to keep up with the increasingly specialized labor market requirements. Solutions such as on-the-job training may be adopted, but these approaches have high associated risks (e.g., material damage) [6] and may lead to a temporary decrease in workgroup performance [2].

A notable current trend in virtual learning is personalization of assessment [see, e.g. 13], learning path [e.g. 5], and even learning content [e.g. 4], driven by artificial intelligence (AI) techniques such as decision trees, genetic algorithms, fuzzy logic, Bayesian networks, neural networks, or hidden Markov models [1]. Yet, heavy automation of teaching/training has been regarded with reservations, and many virtual learning researchers shifted their focus to facilitating real-time collaboration between trainees and tutors via technologies such as virtual classrooms or whiteboarding [9]. Furthermore, UNESCO policies discourage the replacement of human teachers with AI, as negative effects on learner agency, motivation, memory, etc. have been highlighted. The need to redefine teacher roles as users of AI techniques in their didactic activity is emphasized instead [22].

These arguments led us to the conclusion that, instead of having AI substitute instructors, a more efficient approach would be to use AI to connect them to learners, assembling optimized instructor teams to deliver on-demand training for a specific learner request. Such an approach can be implemented as a Virtual Organization (VO).

VOs are temporary alliances of geographically dispersed organizations initiated with the purpose of serving a specific goal [3]). The idea of using a VO for teaching dates back more than two decades ago (see [16]), but implementations are still scarce, and usually focus on a particular curriculum. To our knowledge, extant training VO research has yet to address VO structure optimization in response to ad-hoc requests, i.e., solving the Partner Selection Problem (PSP). PSP is a special case of multi-objective optimization problem (MOP). We assume a typical PSP scenario: the initiator of a project organizes an auction for the n activities the initiator is not willing/able to perform in-house, seeking to optimize the combination of bids. Table 1.1 summarizes the notations we use for PSP throughout the thesis.

Table 1.1: Notations.

Notation	Definition
n	number of project activities
A	set of project activities
P	set of edges indicating project activity precedence constraints
m_i	number of partner bids placed for project activity $i \in A$
C_{ij}	cost of partner bid $j \in \{1, \ldots, m_i\}$ for project activity $i \in A$
Q_{ij}	quality rating of partner bid $j \in \{1, \ldots, m_i\}$ for project activity $i \in A$
D_{ij}	duration of bid $j \in \{1, \ldots, m_i\}$ for project activity $i \in A$
TC	total cost of project
AQ	average project quality
TD	total project duration
$StAv_{ij}$	availability window start time of partner bid $j \in \{1, \ldots, m_i\}$ for project activity $i \in A$
$EndAv_{ij}$	availability window end time of partner bid $j \in \{1, \ldots, m_i\}$ for project activity $i \in A$
ST_i	scheduled start time of project activity $i \in A$
R	set of all resources required by project goals
RR_{ir}	amount of resource $r \in R$ required by project activity $i \in A$
RC_r	capacity of resource $r \in R$

We will consider the following formulation of multiobjective PSP:

$$\text{minimize } TC = \sum_{i=1}^{n} \sum_{j=1}^{m_i} \beta_i^{(j)} C_{ij} \tag{1.1}$$

$$\text{maximize } AQ = \frac{\sum_{i=1}^{n} \sum_{j=1}^{m_i} \beta_i^{(j)} Q_{ij}}{n} \tag{1.2}$$

$$\text{minimize } TD = \max_{\substack{i \in A,\\ j \in \{1,\ldots,m_i\}}} \{ST_i + \beta_i^{(j)} D_{ij}\} \tag{1.3}$$

$$\text{subject to: } \sum_{j=1}^{m_i} \beta_i^{(j)} = 1, \forall i \in A \tag{1.4}$$

$$ST_i + \beta_i^{(j)} D_{ij} \leq Deadline, \quad \forall i \in A, j \in \{1, \ldots, m_i\} \tag{1.5}$$

$$ST_i + \beta_i^{(j)} D_{ij} \leq ST_k, \quad \forall i \in A, k \in A \text{ s.t. } (i,k) \in P,\ j \in \{1, \ldots, m_i\} \tag{1.6}$$

$$\sum_{i=1}^{n} \sum_{j=1}^{m_i} \beta_i^{(j)} C_{ij} \leq Budget \tag{1.7}$$

$$ST_i \geq StAv_{ij},\ \forall i \in A, \quad j \in \{1, \ldots, m_i\} \text{ s.t. } \beta_i^{(j)} = 1 \tag{1.8}$$

$$ST_i + D_{ij} \leq EndAv_{ij},\ \forall i \in A, \quad j \in \{1, \ldots, m_i\} \text{ s.t. } \beta_i^{(j)} = 1 \tag{1.9}$$

$$\textstyle\sum_{i=1}^{n} a_i^t RR_{ir} \leq RC_r,\ \forall r \in R, \quad t \in \{1, \ldots, Deadline\} \tag{1.10}$$

where

$$\beta_i^{(j)} = \begin{cases} 1, & \text{if partner bid } j \text{ is chosen for project activity } i \\ 0, & \text{otherwise} \end{cases}$$

$$a_i^t = \begin{cases} 1, & \text{if project activity } i \text{ is in progress at time } t \\ 0, & \text{otherwise} \end{cases}$$

In the formulation above, Equations (1.1)–(1.3) define the criteria/objectives. Equation (1.1) minimizes the total project cost. Equation (1.2) maximizes the average quality of the selected partner bids. Equation (1.3) minimizes project duration. Constraint (1.4) ensures that only one bid is selected for each activity in the project. Constraint (1.5) states that no activity end time should exceed the imposed project deadline. Constraint (1.6) enforces the time precedence relationships between project activities. Constraint (1.7) guarantees the total project cost does not exceed the allocated budget. Constraints (1.8) and (1.9) impose that activities take place in the time windows imposed by the selected bids for the project activities in question. Constraint (1.10) ensures that the required amount of any resource in use at any time during the project does not exceed the given capacity of the resource.

Many approaches have been proposed for solving MOPs (see [11]) and PSP. Currently, one of the approaches deemed most suitable are global search (population-based) heuristics [14]. We describe our proposed adaptation of a recent meta-heuristic, an algorithm we named Multiobjective Symbiotic Organisms Search for Scheduling (MOSOSS), in section 4.1.

We further propose an architecture for a holonic multiagent system (MAS) supporting the full lifecycle of a training virtual organization (VO) [3]. Our choice of modeling the system as a MAS was motivated by VO characteristics such as decentralized nature, goal-orientation, auction-based formation, decisions based on communication and information flow, negotiation processes, etc. [12], [20] As Peña et al. [19] advocate the use of holons as fundamental elements underlying computing science theory, we modeled both the market and the created VOs as holons, with the latter being subholons of the former. To our knowledge, this is the first research work that approaches collaborative teaching (not collaborative learning) by implementing a training VO as a holonic MAS (HMAS) [7].

Chapter 2

Aim and Objectives

In this thesis, we aim to propose an architecture for a platform capable of supporting the entire lifecycle of a training VO. For this purpose, we dedicate a special attention to PSP, which is a key problem to be solved in two of the four phases of the VO lifecycle: formation and reconfiguration. Implementing the concurrent management of multiple PSP solving processes is an important objective of our work. Furthermore, we also address VO operation and dissolution, by designing coordination and a rating system.

Our approach is to start from a rather general PSP formulation, and develop a generic PSP solver first. The advantage of such an approach is that it can be applied to a larger set of problems, not only to the problems stated in this work. Desirably, though, the general solver should be easily adaptable to more specific problems, and this adaptability is also part of our aim. As such, we chose to develop an heuristic algorithm which does not require any specific type of objective functions or a specific geometry of the Pareto front.

Moreover, the concurrent management of training VO requests is part of our aim. First, we address simultaneous request management by creating, for each individual request, a dedicated holonic software agent in charge of handling the auction and other negotiation mechanisms in its inner context. Second, after all bids are collected and the Pareto set over the set of bid combinations can be computed, the Pareto set may contain more than one (sub)optimal solution. This means additional support for the human decision making process is needed. We aim to address this by creating a separate negotiation space for each potential training VO the learner may choose, allowing concurrent negotiation.

We further capitalize on the holonic framework to design the other phases of the training VO lifecycle. During VO operation, our aim is to support instructor collaboration for personalized curriculum planning by modeling instructors as superholons of instructors in charge of the prerequisites they need. To close the cycle and facilitate subsequent VO formation processes, we also address quality of service ratings of the instructors upon VO dissolution. Additionally, we aim to design the platform such that instructors who become members of one or more training VOs also maintain their initial competitive relationships in the education market environment they are part of. The main design principle we propose apply for this purpose is modeling training VOs as subholons of the education market.

We therefore derive the following specific objectives:

1. O1 — proposing and implementing a novel approach to solving PSP with task scheduling under time, budget, activity precedence and resource capacity constraints
2. O2 — implementing and testing the proposed approach on random PSP instances, simulating an auction for VO formation
3. O3 — designing and implementing a HMAS architecture for a training VO environment for the concurrent management of training requests through the configuration and support for VO operation, reconfiguration and dissolution
4. O4 — conducting an experiment to compare the effectiveness of three different organization structures of the HMAS during the training activity scheduling process.

The remainder of the thesis is organized as follows. The methodology we use to achieve the objectives is described in Chapter 3. The corresponding results are reported in Chapter 4. An extensive discussion of these results is presented in Chapter 5. Chapter 6 concludes the thesis.

Chapter 3

Research Methodology

We developed the approach targeted by objective O1 based on a summary of the theorized and/or empirically demonstrated advantages and disadvantages of extant PSP approaches. Our proposed approach, detailed in section 4.1, is an adaptation of one of the recent heuristics whose effectiveness in solving problems similar to the PSP we focus on was empirically demonstrated in multiple studies.

To achieve objective O2, we implemented the proposed algorithm and competing algorithms in Java, as an extension of the jMetal framework, version 5 [1][18]. To test the algorithms against each other, we simulated VO formation scenarios by randomly generating PSP instances. The algorithm we used for generating random PSP instances is presented in Appendix A. We then conducted numerical experiments comparing the competing algorithms with respect to a set of performance metrics for Pareto optimality. These experiments, as well as experiments for objective O4, were conducted on an Acer computer with 16 GB 1600 MHz DDR3 L Memory and a 2.6 GHz Intel Core i5 processor running Windows 10 Pro operating system.

In pursuit of objective O3, we first designed an agent-based training VO environment using UML use case diagrams, sequence diagrams and class diagrams. The design is described in the next section.

3.1 Preceptor: A Proposed Holonic Multiagent System Architecture for Training Virtual Organization Environments

In this section, we formulate a specific virtual learning problem, we frame it as a special case of PSP and propose a solution based on a holonic multiagent system approach, then we describe its design.

[1]available on https://jmetal.github.io/jMetal

3.1.1 Problem Statement

Suppose a person who seeks specific training (e.g., an applicant for a job) requires training for one or more competences that he/she does not master. The person (which we will refer to as **Learner**) seeks an optimal virtual service provider, given a set of training providers (which we will henceforth refer to as **Instructors**) who activate on a market (which we will refer to as **Education Market**), to cover all the Learner's specific training needs. The Learner establishes two criteria for optimality: cost and quality, and emits a tuple $VORequest \in \mathbb{N} \times 2^C \times \mathbb{R}_0^+ \times \mathbb{N}$, where C denotes the set of all competences listed on the market. The tuple contains the Learner's identification number, the required competences to be trained ($A \subseteq C$), the budget and the deadline. Additional time constraints come into question because some of the required competences may be prerequisites for others, which means that training for those competences should be finished before starting training for the competences that depend on them.

Throughout the rest of the thesis, we will refer to the problem described above as the **Instructor Selection Problem (ISP)**. ISP can be modeled as a bi-objective PSP with the following two objectives to be simultaneously optimized: (1) total cost, to be minimized; (2) average quality of the training services composing the VO, to be maximized. The solutions are subject to a deadline and a budget constraint, are composed of a number of n competence training activities for which an Instructor bid should be chosen, and the following bid characteristics are considered as decision variables: costs, quality ratings, availability window start times and end times, and duration of the training service. Only one resource is required by all training activities: time (number of hours a day the Learner is willing to allocate to the training activities in the VO). Prerequisite relationships between competences induce the precedence constraints in the general PSP. Therefore, the problem is a special case of the PSP defined earlier in the introduction, obtained by dropping the time criterion described by Equation 1.3 and defining the set of resources R as containing one single resource.

The second problem we address is one we call **Prerequisite Strategy Problem**. We assume that each competence $c \in C$ is characterized by a set of strategies that may be applied in its training. Training strategies may differ in terms of content selection and/or sequencing, as well as in terms of the selection of training methods. Another assumption we make is that training a competence c_1 as a prerequisite of another competence c_2 may require a different approach (i.e., strategy) than training the same competence per se, or as a prerequisite of c_3, etc.

Let $P_c \subseteq C \setminus \{c\}$ denote the set of prerequisites for competence $c \in C$ and S_p denote the set of training strategies for prerequisite $p \in P_c$. Then, $\forall c \in C,\ p \in P_c$ we define a success function $success_{c,p} : S_p \rightarrow [0, 1]$, which associates each strategy for training p as a prerequisite of c with a success rating. The higher the rating, the more successful the strategy. These success ratings may be results of various aggregations methods applied to evaluations provided by experts in pedagogy, but this is beyond the scope of this thesis.

3.1.2 Proposed Solution

To solve ISP and the Prerequisite Strategy Problem, we propose an architecture for a VO environment we called Preceptor. The proposed architecture supports two user categories: Learners and Instructors, who use the system according to the use case diagram in Figure 3.1. The proposed VO environment is a holonic implementation of the Education Market, supporting on-demand creation of training VOs as subholons. Preceptor may be used to establish the Learner's personalized learning path using our proposed adaptation of the Breadth first search [17] for prerequisites of requested competences (the pseudocode is listed in Appendix B) and it supports all phases of the VO lifecycle: formation, operation, reconfiguration, and dissolution [3].

After establishing the set A of competences that need training, the Learner may emit a training VO Request containing A on the Education Market. Figure 3.2 illustrates how Preceptor manages two concurrent VO Requests at the same time. Let us assume that VO Request 1 specified a set of competences, A_1, containing three competences, while the set of competences A_2 specified in VO Request 2 contained four competences. Then, each Potential VO Negotiator created for VO Request 1 should in turn create three Competence Holons (CHs), whereas each Potential VO Negotiator for VO Request 2 should create four CHs. A Scheduling Holon (SH) is also created by each Potential VO Negotiator to schedule training activities.

Once the Learner chooses a VO, the VO Request Manager will proceed to VO configuration. A configured VO has the Learner as its head holon. The Instructors will also be organized in a holarchy according to the competence holarchy. The sequence diagram in Figure 3.3 illustrates the entire process of VO formation and initial configuration.

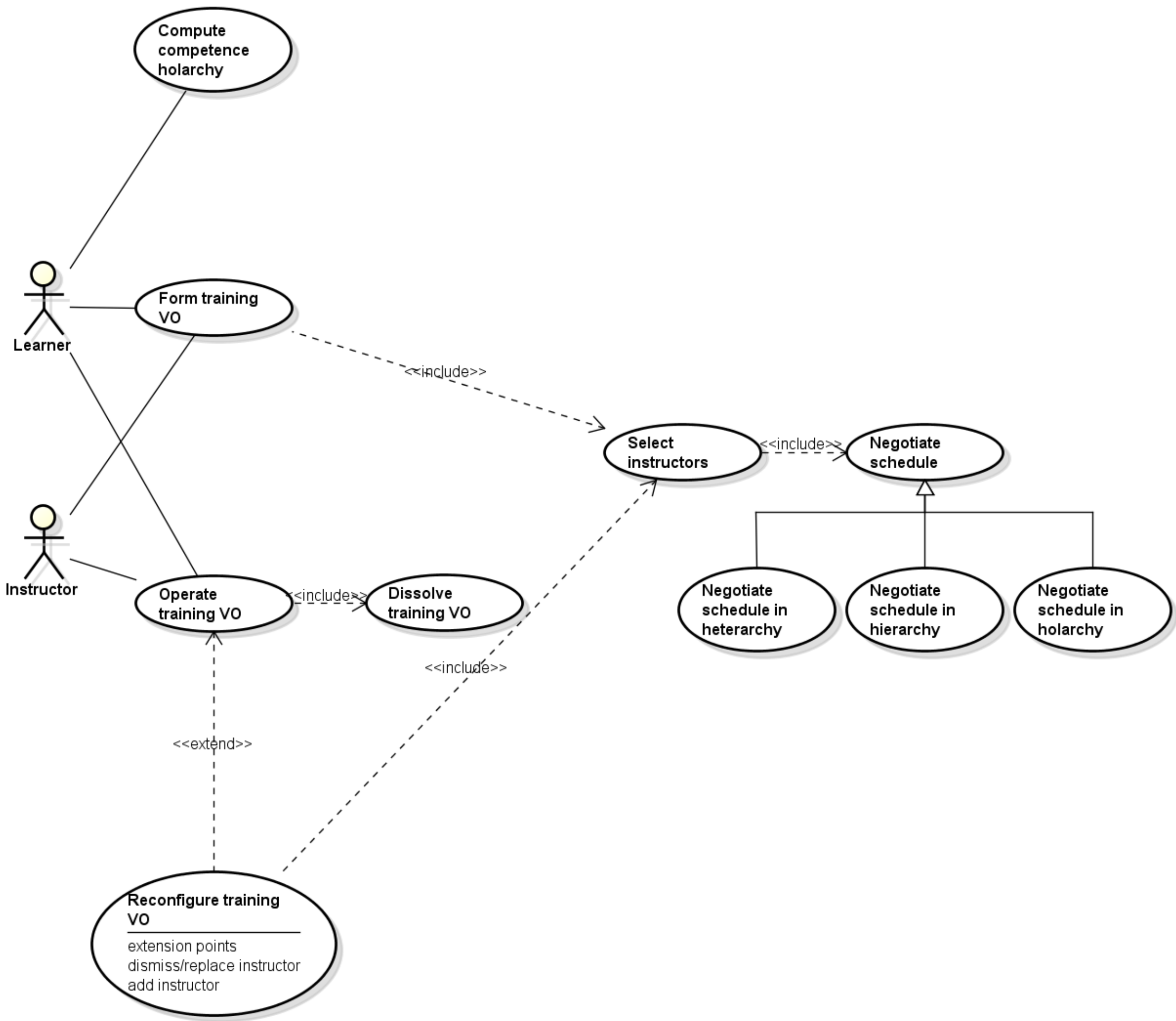

Figure 3.1: Preceptor Use-Case Diagram

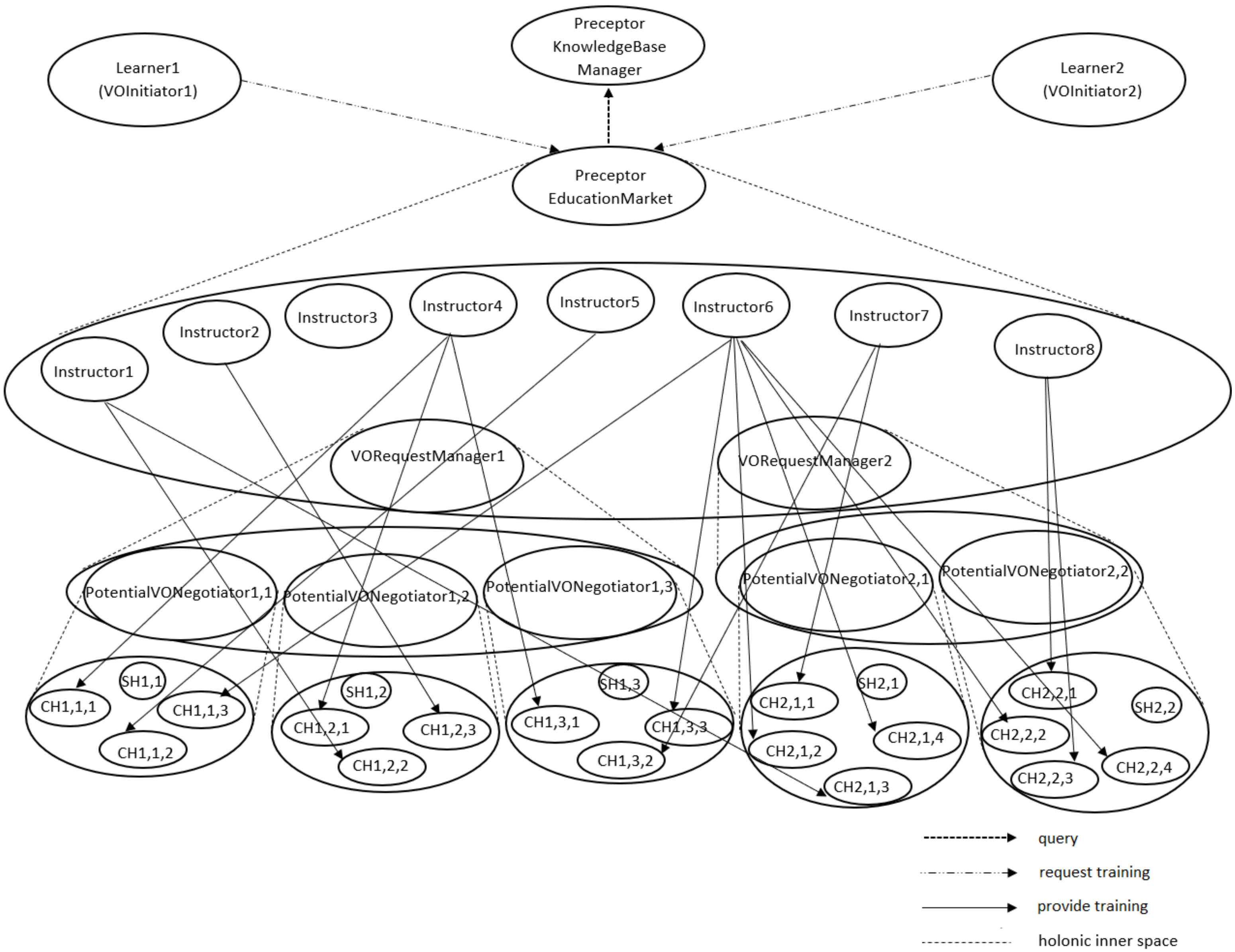

Figure 3.2: Overview of the Preceptor Holonic Organization for VO Formation

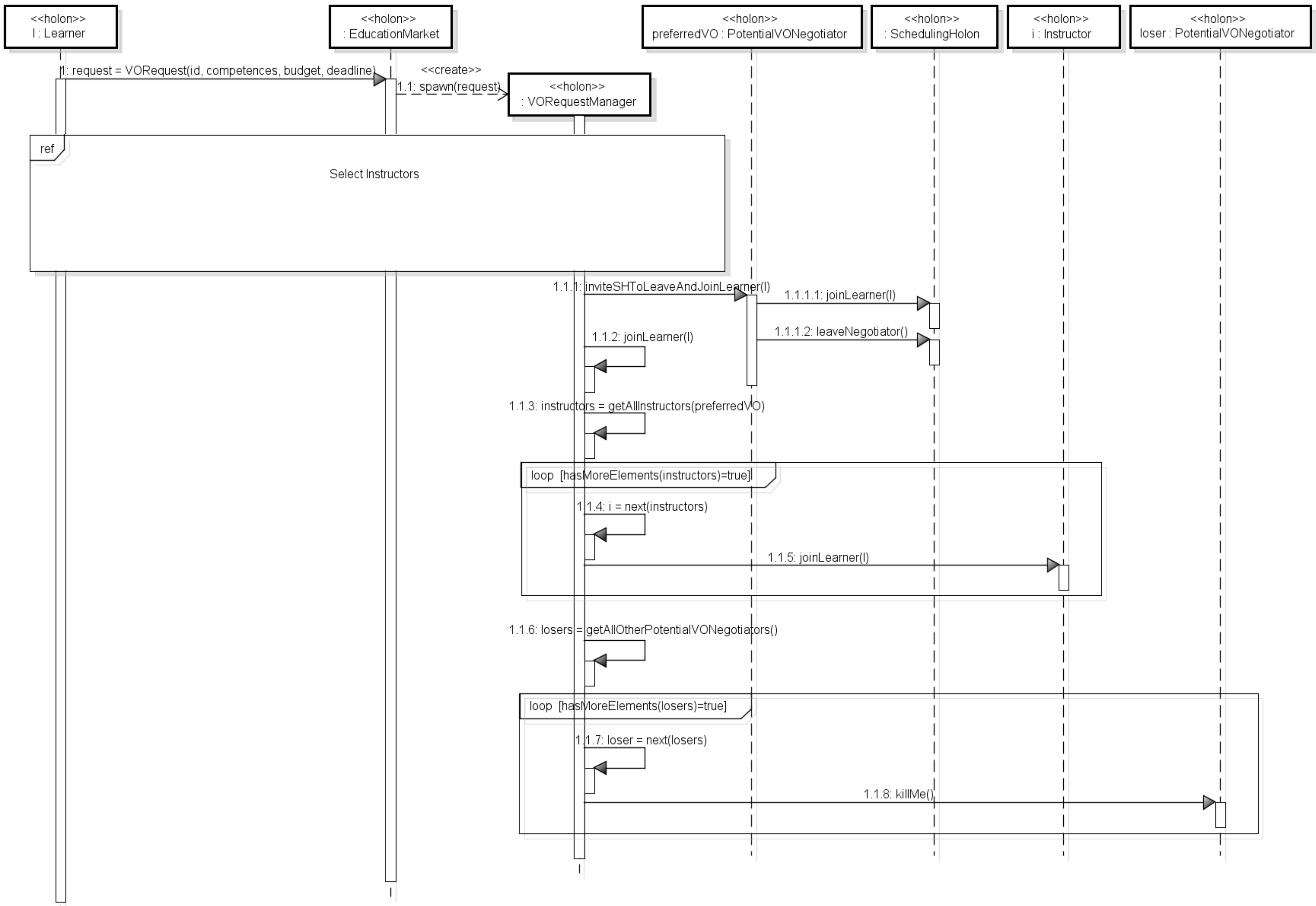

Figure 3.3: Sequence Diagram Describing VO Formation

Determining or approximating the optimal team of instructors who can cover the set A of competences is an ISP that should be solved in order to reach the configured VO state. To this purpose, the VO Request Manager will organize an auction for the requested competences. Each Instructor on the Education Market will place one or more bids for a competence in the VO request if and only if the Instructor is skilled to provide training for the competence in question. A bid is a tuple of the form:

$$bid = (InstructorID, c, StAv, EndAv, D, cost, quality) \tag{3.1}$$

where

$InstructorID \in \mathbb{N}$ is the identifier of the instructor
$c \in A$ is a requested competence the instructor provides training for
$StAv$ and $EndAv$ denote the availability window start time and end time, respectively
$D,\ cost,$ and $quality$ denote the duration, cost, and quality rating of the training service provided by $InstructorID$ for c, respectively.

After closing the auction, the VO Request Manager may proceed to compute the solutions for the VO Request. Let $B_i,\ i \in \{1, \ldots, n\}$ denote the set of bids for competence training activity i. We define a solution s to the VO Request as a possible combination of bids, one for each competence. Therefore, $s \in B_1 \times \ldots \times B_n$.

The VO Request Manager then evaluates all possible solutions. Each solution s is evaluated in terms of two objective functions: total cost and average quality. As the multiobjective nature of this problem warrants the identification of trade-off solutions, a multiobjective optimization approach must be adopted. For instance, the MOSOSS algorithm can be used to output an approximation of the Pareto optimal set of alternative combinations given the set of all possible solutions. For each solution in the Pareto set, the VO Request Manager spawns a Potential VO Negotiator. For the purposes of reaching an agreement on a schedule, each Potential VO Negotiator spawns one SH for scheduling and one CH for each competence training activity. The behavior of the SH in response to a CH scheduling request is described by the sequence diagram in Figure 3.4. Scheduling attempts of the SH are based on an adaptation of the scheduling algorithm proposed in Ionescu and Vernic [14] (see the activity diagram in Figure 3.5).

In its initial state in the scheduling process, the CH is not scheduled. In parallel, it attempts to compute its schedule by emitting a schedule request to the SH and listens for end time notifications from CHs in charge of its prerequisites, as well as for complete feasible schedule notifications from the Potential VO Negotiator. After the SH replies a proposed start time, the CH will update its earliest start time and check its availability. If available, the CH will broadcast the computed activity end time in its default context. Its schedule state will also be updated so as to indicate successful scheduling. A failure message will be broadcast otherwise. When all CHs are scheduled, the SH will check the feasibility of the schedule (i.e., $\forall c_1 \in A,\ \forall c_2 \in A \cap P_{c_1},\ startTime_{c_2} + duration_{c_2} \leq startTime_{c_1}$, where $startTime_c$ and $duration_c$ denote the start time and duration of the training activity for competence $c \in A$, respectively).

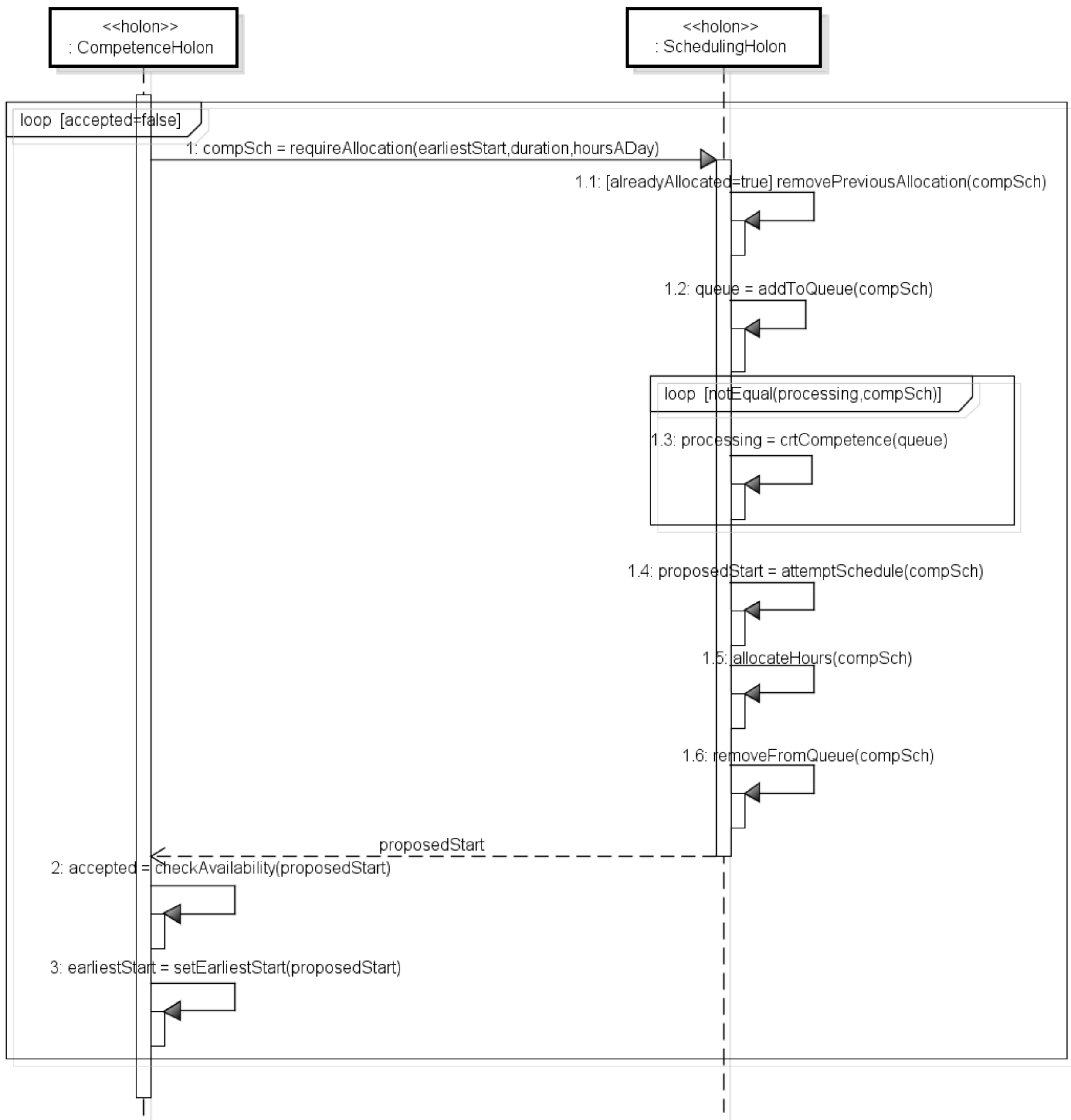

Figure 3.4: Sequence Diagram Describing the Behavior of the Scheduling Holon

When a SH reaches a complete and feasible schedule, it emits the schedule to the Potential VO Negotiator, which in turn emits it to the VO Request Manager. In case the Potential VO Negotiator is informed by the SH that no feasible schedule can be generated for the potential VO, the Potential VO Negotiator

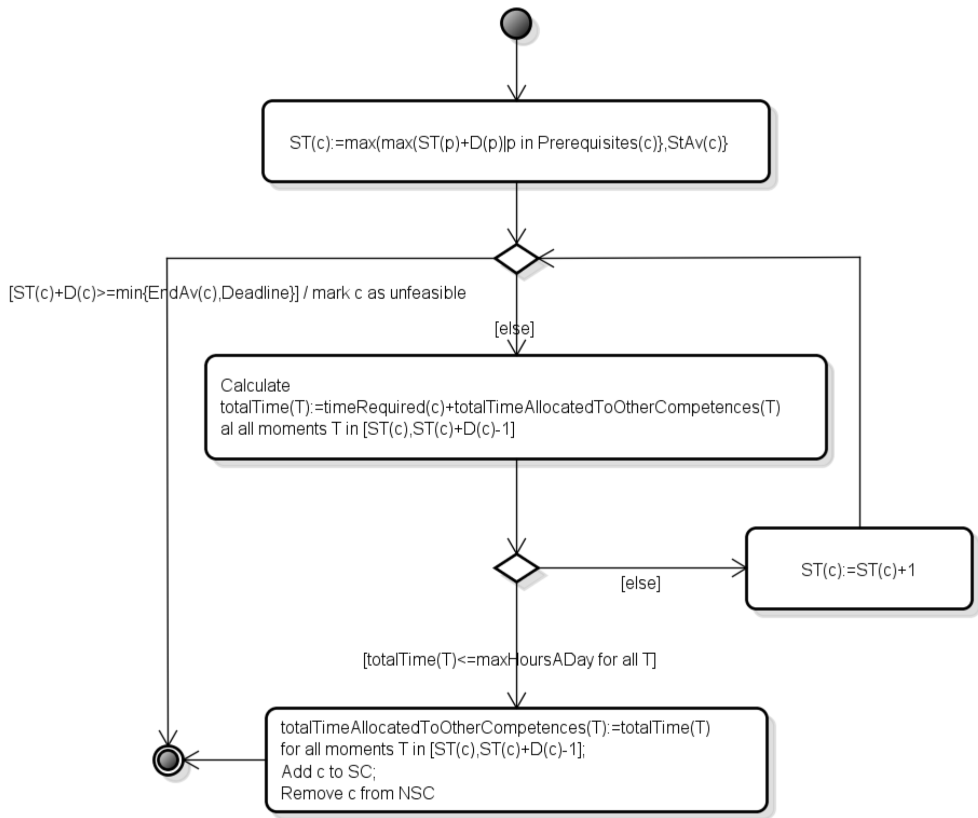

Figure 3.5: Activity Diagram Describing a Schedule Attempt of the Scheduling Holon

sends a failure message to the VO Request Manager.

Finally, after collecting all schedules (and failure messages, if that is the case) from Potential VO Negotiators, the VO Request Manager retains the potential VOs that are feasible in terms of schedule and asks the Learner to choose the one he/she prefers. A detailed view of the instructor selection use case is offered in Figure 3.6.

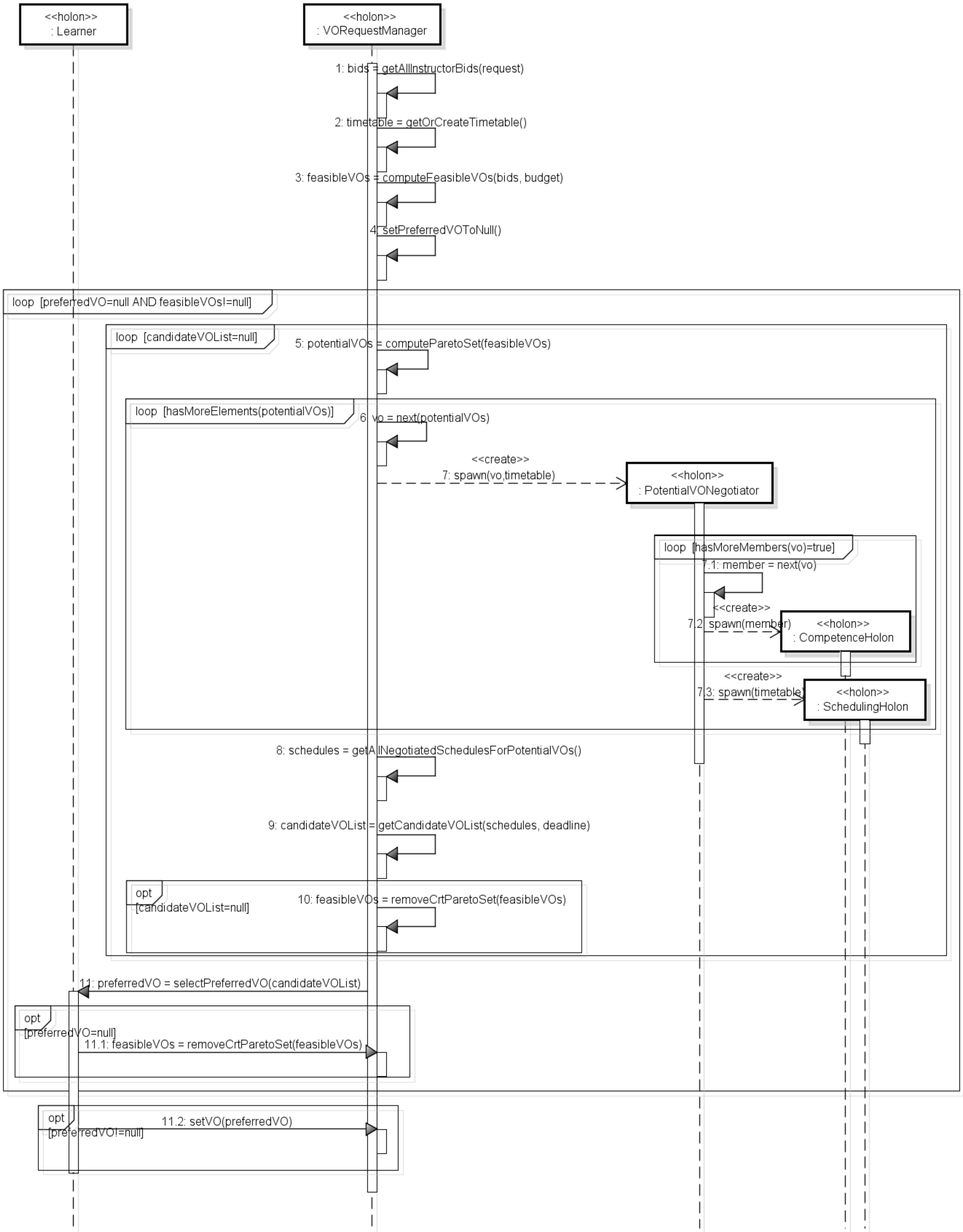

Figure 3.6: Sequence Diagram Describing how Preceptor Handles Instructor Selection

For the chosen potential VO to become an operational training VO, each Instructor agent that placed a winning bid for the VO should be notified by the VO Request Manager of its individual schedule, the address of the Learner agent, and the addresses of its immediate superholons and subholons. The Instructor agents are multi-part holons [8]). First of all, they preserve their status as subholons of the Education Market. This allows them to keep listening for VO Requests that might be of interest given their training skills. Second, by joining the inner context of the Learner, they become subholons of the Learner. Third, they aggregate zero or more superholons and zero or more subholons among the other Instructors in the training VO. These latter holonic relationships are dictated by the prerequisite relationships between competences. Each Instructor agent, for each of the competences it provides training for in the VO, coordinates the training activity of the Instructor agents that train prerequisites (if any) of the competence in question. This is the core mechanism of delivering personalized curriculum planning to the Learner.

A competence may be a prerequisite for more than one other competence in the VO. This means the Instructor agent that provides training for competence $c \in A$ will be coordinated by all Instructor agents in the VO that train any other competence for which c is a prerequisite. Each of these superholons may indicate another preferred strategy, because whenever $p \in P_{c_1} \cap P_{c_2},\ c_1, c_2 \in A, c_1 \neq c_2$, it is possible that $success_{c_1,p} \neq success_{c_2,p}$.

Our proposed solution is the following: each of the superholons should transmit its success rating function to its subholons (in charge of training prerequisites). A subholon that provides training for prerequisite p will then select the strategy $s_0 \in S_p$ that maximizes $\sum_{c \in \{c_1 \in A | p \in P_{c_1}\}} success_{c,p}(s)$.

As shown in Figure 3.7, each Instructor agent will perform the training activity (or activities) it was selected for in the VO—i.e., corresponding to its winning bid(s). A reconfiguration request is an extension point for the training VO operation use case. Contracts with subholons may be renegotiated if the Learner decides.

If the request is one of dismissal or replacement, the VO Request Manager will ask the Learner to rate the training service provided for the given competence. Upon receipt of a Learner reply containing the rating, the VO Request Manager will update the knowledge base with the rating for the provided competence training service in the form of a tuple $(Instructor, competence, crt\_rating)$. This tuple will be sent to the Knowledge Base Manager agent, which keeps track of all ratings for all Instructor-competence pairs. After the Knowledge Base Manager agent receives the update message from the VO Request Manager, the new rating will be calculated using Equation 3.2b:

$$rating_0 = \infty \tag{3.2a}$$

$$rating_{n+1} = \frac{n * rating_n + crt\_rating}{n+1},\ \forall n \geq 0 \tag{3.2b}$$

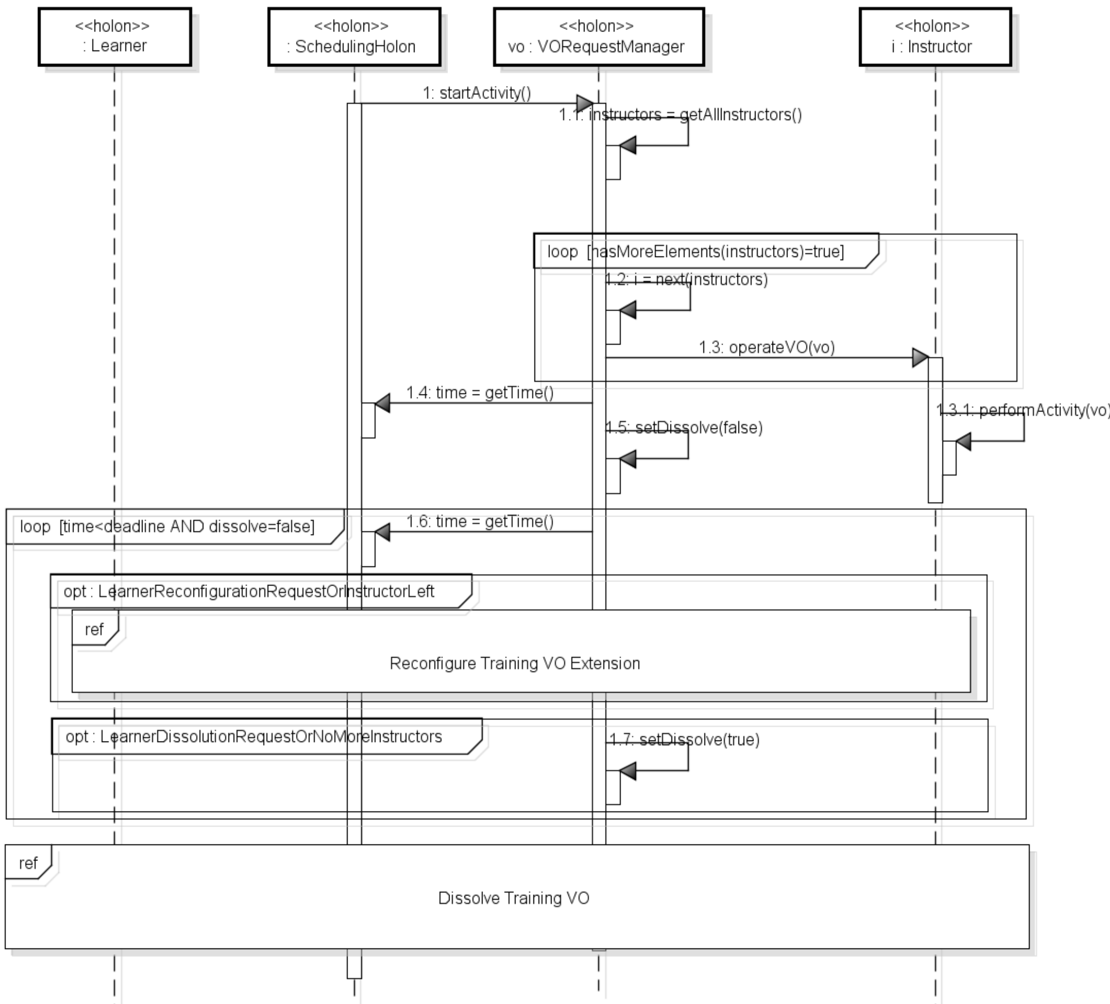

Figure 3.7: Sequence Diagram Describing VO Operation

If the Learner's reconfiguration request was either one of replacement or one of addition, the original VO Request will be updated by replacing the initial set of competences (A) with the competence in the reconfiguration request. The system will re-enter the Instructor Selection use case with this updated VO Request (see Figure 3.8).

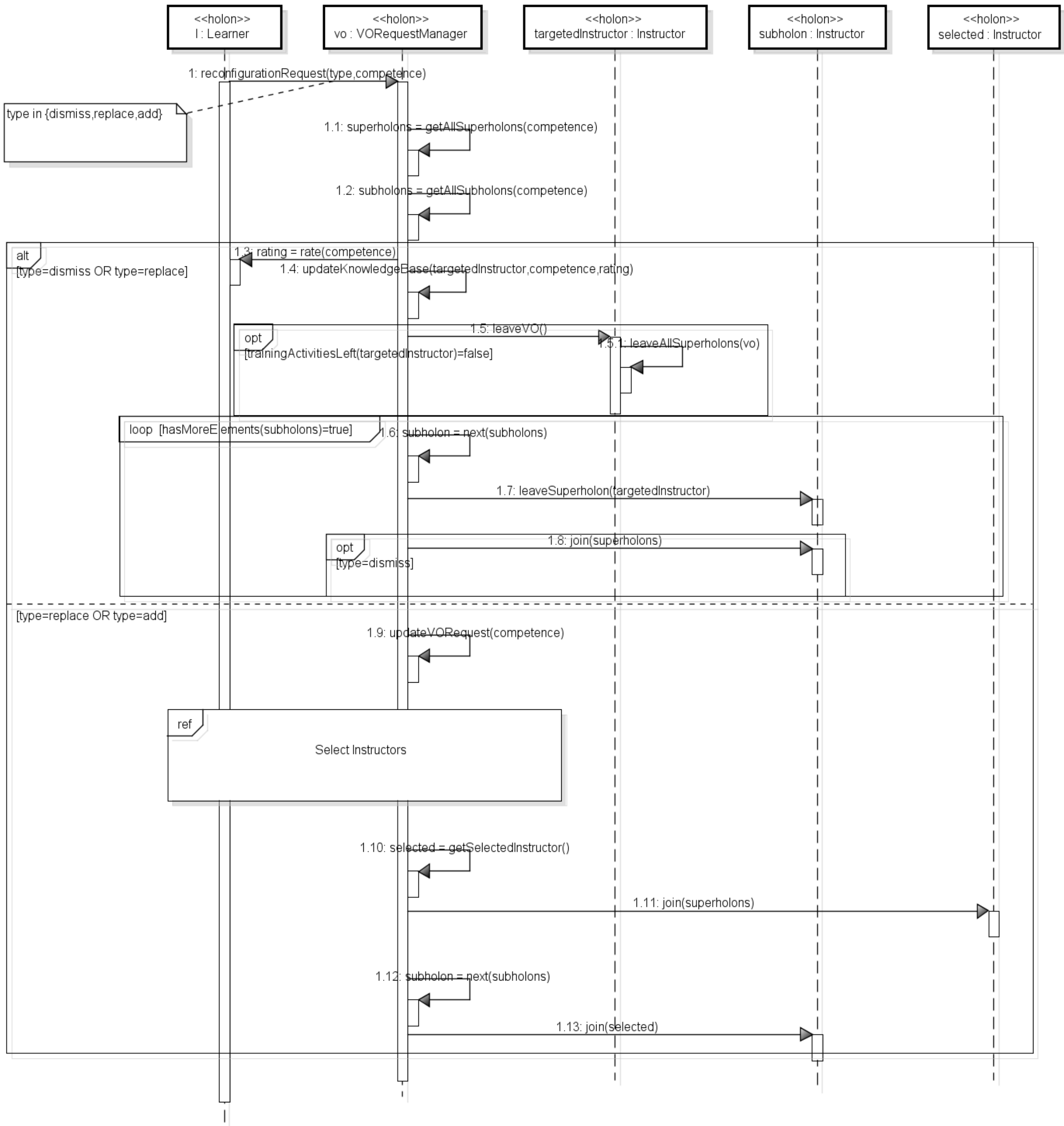

Figure 3.8: Sequence Diagram Describing the Processes Involved in the Reconfiguration Extension of the VO Operation Use Case

Whenever the Learner explicitly requests dissolution, or the training VO is left without Instructors, or all training activities have finished, the training VO dissolves (see Figure 3.9).

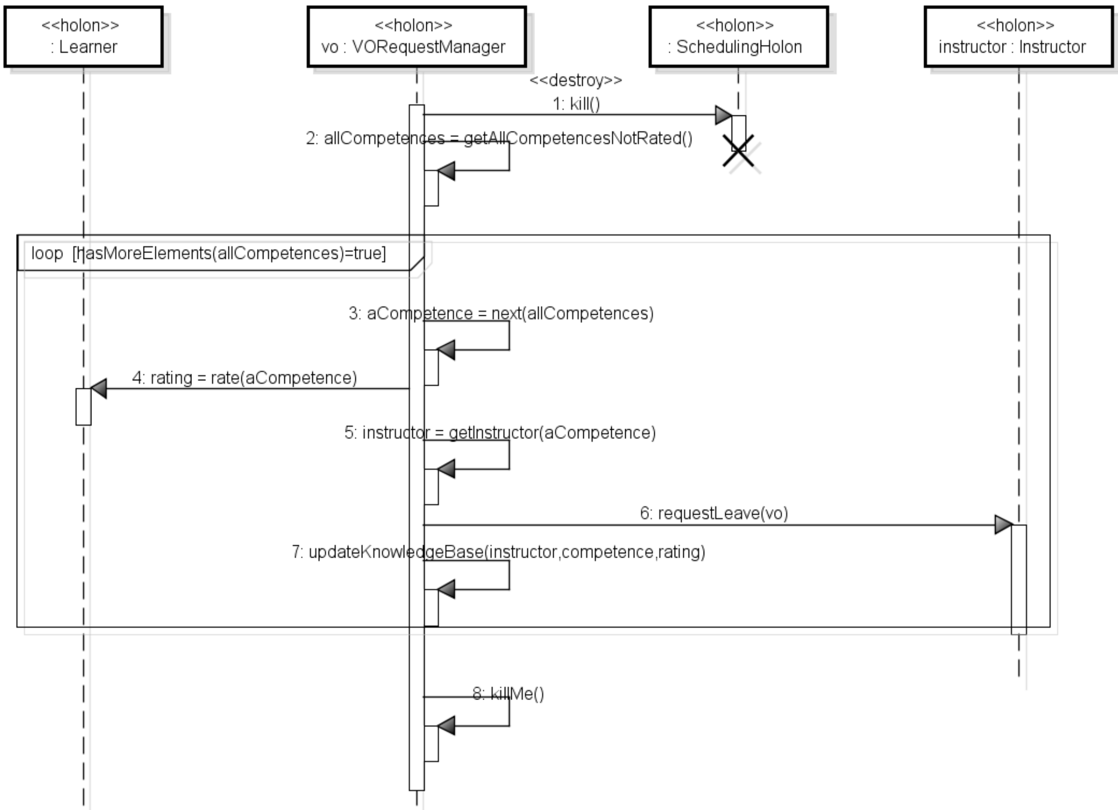

Figure 3.9: Sequence Diagram Describing Training VO Dissolution

We implemented the designed training VO environment as a HMAS platform coded in the SARL programming language, version 12 [21]. Our choice was motivated by the native support for holonic multi-agent systems offered by SARL [7]. As a runtime environment for the HMAS we implemented in SARL, we used the recommended Janus platform [8]. Janus was developed with the aim of being suitable for large, modular applications. Its design is based on the CRIO metamodel, adopting the principle of separation of roles from the holons playing them.

3.2 Case Study

Finally, to achieve objective O4, we tested the Preceptor architecture using a case study described as follows. A Learner requests training for one competence: Behavior Modeling and Simulation in Virtual Environments. The Learner and Preceptor interact in order to establish the entire list of competences the

Learner actually needs training for. The corresponding competence training VO that should be configured should therefore have a holonic structure such as the one represented in Figure 3.10.

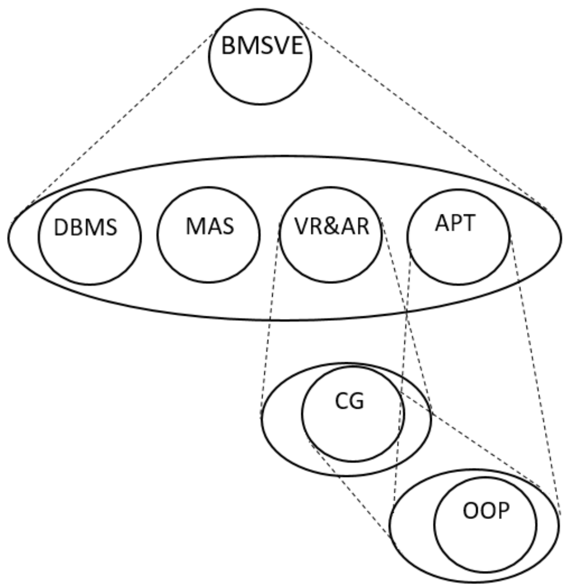

Figure 3.10: Case Study: Competence Holarchy

We tested the proposed organization structure of Preceptor during scheduling ("heterarchy"/decentralized) against two alternative organization structures ("hierarchy"/centralized and "holarchy"/holonic). The three organization structures differ with respect to the behavior of the Potential VO Negotiator, of the SH and of the CHs. In the hierarchical organization structure, scheduling is the exclusive responsibility of the Potential VO Negotiator (see Figure 3.11). In the holarchy structure each CH becomes a super-holon of all CHs in charge of any of its immediate prerequisites. As illustrated in the sequence diagram in Figure 3.12, the CH and the SH will interact by sending proposals until they reach an agreement.

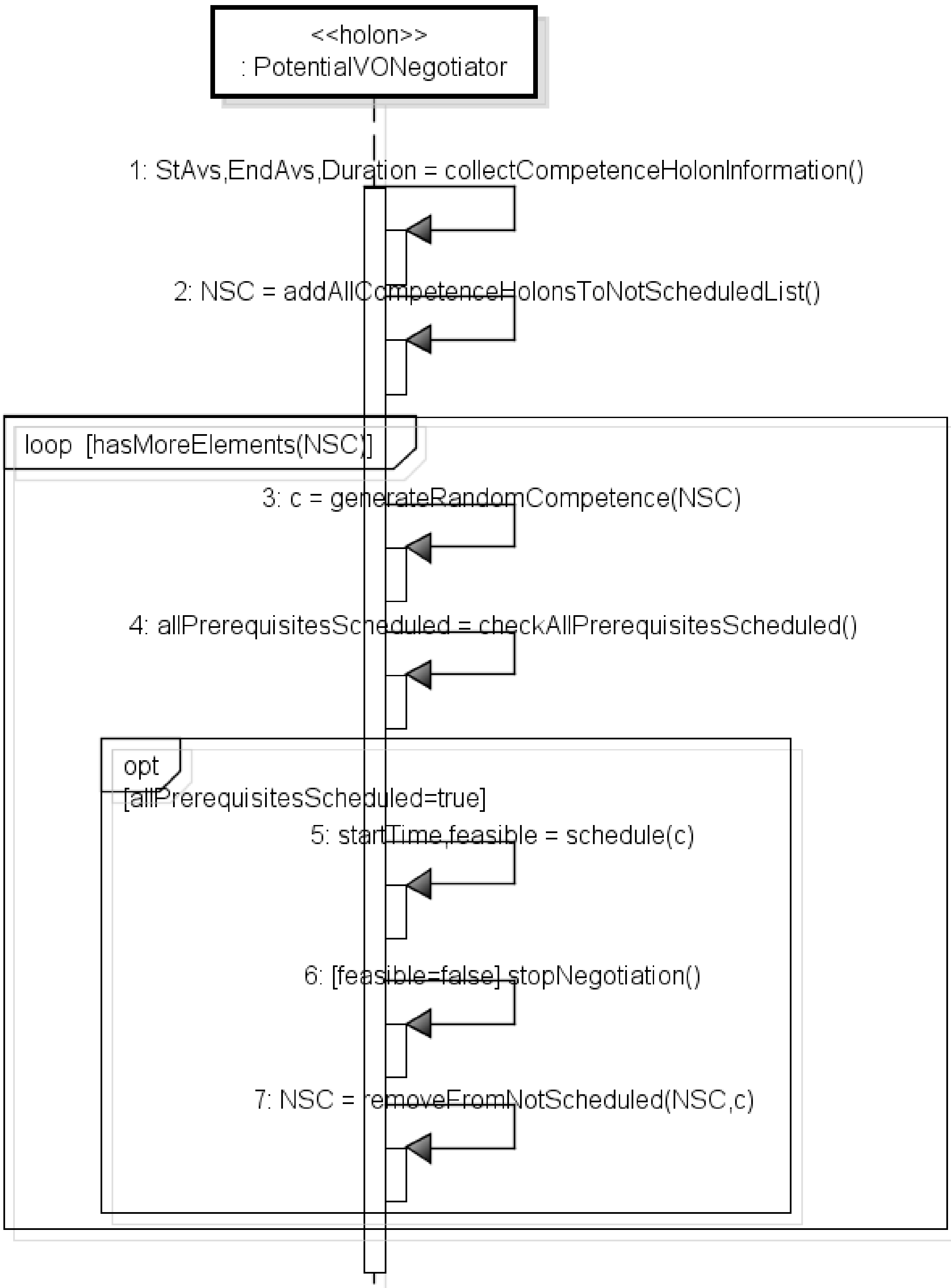

Figure 3.11: Sequence Diagram Describing the Scheduling Process in the Hierarchy Organization Structure

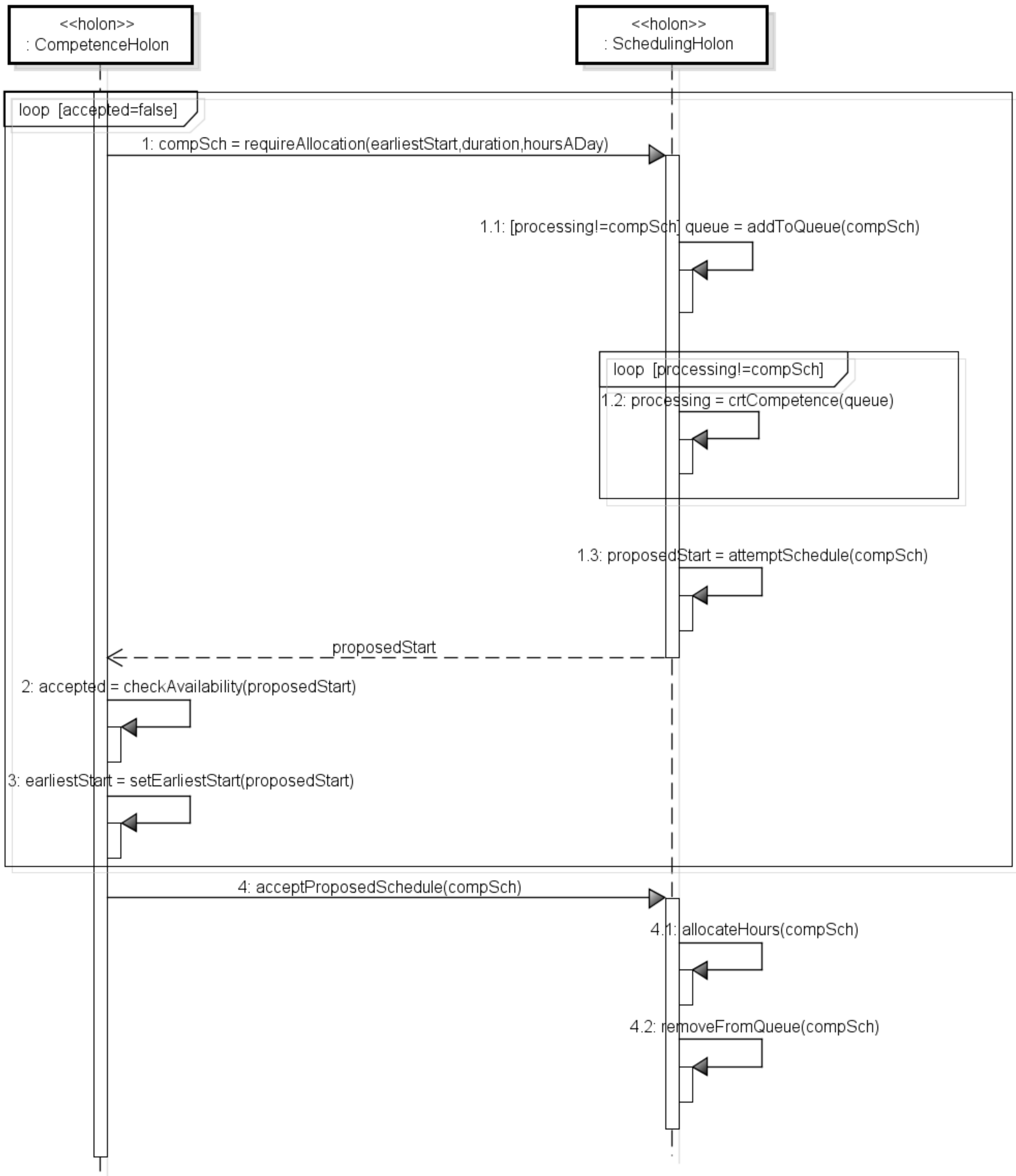

Figure 3.12: Sequence Diagram Describing the Scheduling Process in the Holarchy Organization Structure

3.2.1 Hypotheses

We hypothesize that the three organization structures will differ significantly with respect to their computational efficiency, considering three performance measures: (1) elapsed time (in milliseconds); (2) CPU usage (the percentage of CPU clock ticks used by Preceptor processes out of the total number of ticks); and (3) memory usage (the percentage of memory used by Preceptor processes). For each performance measure, we are interested in two values: the total value for the whole configuration process, and the value for the scheduling process only.

We expect the execution time for the holarchy structure to be longer, because the CHs also form a holarchy, thus inducing delays in communication. Therefore, we hypothesize:

Hypothesis 1. Total configuration time (a) and scheduling time (b) will be significantly longer for the holarchy organization structure as compared to heterarchy and hierarchy.

In terms of CPU usage, however, we expect the holonic organization to be superior at least to the heterarchical structure, because holons wait for all information a decision depends on and only communicate the final decision to other holons. We thus hypothesize:

Hypothesis 2. Total CPU usage (a) and CPU usage during the scheduling process (b) will be significantly lower for the holarchy organization structure as compared to heterarchy and hierarchy.

In terms of average memory usage, we expect the hierarchy organization to produce the highest values, because all data structures are stored in memory simultaneously until the VO negotiator finishes the scheduling process. As such, we hypothesize:

Hypothesis 3. Average memory usage during configuration (a) and during the scheduling process (b) will be significantly higher for the hierarchy organization structures as compared to heterarchy and holarchy.

3.2.2 Experimental Settings

To create experimental settings which can easily be replicated, we randomly generated 8 solutions for the problem. Let S denote the set of 8 solutions. We can generate the first 50 combinations of 5 solutions from S and assume that each combination is the Pareto set of an unknown set of possible bid combinations including the bids that compose the solutions in S. The set of randomly generated solutions (S) and the 50 Pareto sets are listed in Appendix C. Each of the 50 generated Pareto sets was a case in the experiments we conducted to compare the three different organization structures. The frequency with which CHs checked prerequisite end time notifications in the heterarchy and holarchy structures was set to once every 100 milliseconds. Experimental results are reported in section 4.2.

Chapter 4

Results

In this chapter, we first present our novel method for solving PSP under time and resource constraints (in section 4.1), which can be easily adapted to special cases of PSP such as finding the optimal combination of instructors for a given training request. Then, in section 4.2 we report on the results we obtained after using the Preceptor architecture for the training VO request case study described in section 3.2.

4.1 Proposed Approach to the Partner Selection Problem: The Multiobjective Symbiotic Organisms Search for Scheduling (MOSOSS)

Two major innovations that differentiate MOSOSS from established algorithms used for solving similar problems are its reliance on specially-designed symbiotic operators (mutualism and commensalism) for combinatorial optimization and its treatment of unfeasible solutions (MOSOSS operates with partially scheduled solutions).

Solution encoding was the following:

$$\underbrace{bid_1 bid_2 \ldots bid_n}_{\text{component bids}} \underbrace{ST_1 ST_2 \ldots ST_n}_{\text{scheduled start times}} \tag{4.1}$$

Whenever scheduling an activity is not feasible due to time constraint violations, the activity is scheduled such that its end time exceeds the deadline by 1 day. Thus, for every organism, at least a partial schedule is produced, having the chance of being repaired using symbiotic evolutionary operators during subsequent iterations of the algorithm, instead of applying an additional repair method. The symbiotic operators (mutualism and commensalism) are specially designed to take advantage of each other's composition. They both use the same exploitation mechanism, presented in Algorithm 1.

Algorithm 1 Exploit($solution_i$, $solution_j$)

Require: $solution_i = ((bid_1^{(i)}, \ldots, bid_n^{(i)}), (ST_1^{(i)}, \ldots, ST_n^{(i)}))$;
$solution_j = ((bid_1^{(j)}, \ldots, bid_n^{(j)}), (ST_1^{(j)}, \ldots, ST_n^{(j)}))$.
Ensure: $solution_i' = ((bid_1^{(i')}, \ldots, bid_n^{(i')}), (ST_1^{(i')}, \ldots, ST_n^{(i')}))$, a solution obtained by replacing bids in $solution_i$ with corresponding bids in $solution_j$.

```
solution'_i ← solution_i
replace ← False
If solution_i has a pending activity, replace bid in i' with bid in j:
if ∃k s.t. Pending_i = k then
    bid_k^(i') ← bid_k^(j)
    evaluate(solution'_i)
    if solution'_i ≻ solution_i then
        replace ← True
    end if
    if replace = False then
        for k ∈ {1, ..., n} do
            if ST_k^(i) = −1 or Pending_i = k then
                bid_k^(i') ← bid_k^(j)
                evaluate(solution'_i)
                if solution'_i ≻ solution_i then
                    replace ← True
                end if
            end if
        end for
    end if
else
    Reset schedule of solution'_i:
    ST_k^(i') ← −1, ∀k ∈ {1, ..., n}
    Pending_i' ← NULL
    Generate a random number of activities to be replaced, rand ∈ {1, ..., n}.
    Replace bids for rand random activities with corresponding bids from solution_j :
    for k ∈ {1, ..., rand} do
        Generate random activity act_k ∈ {1, ..., n}
        if bid_{act_k}^(i) ≠ bid_{act_k}^(j) then
            bid_{act_k}^(i') ← bid_{act_k}^(j)
        end if
    end for
    {totalCost'_i, totalDuration'_i, avgQuality'_i, c'_i} ← Evaluate(solution'_i)
end if
return solution'_i
```

We conducted numerical experiments (reported in [14]) comparing MOSOSS with the established MOSOS and NSGA-II with respect to the following six performance indicators for Pareto optimality: additive ϵ, GD, IGD, IGD+, HV and Δ. Results supported the superiority of the proposed MOSOSS over the competing algorithms for all performance metrics and for all randomly generated PSP test instances. The test instances were 15, 5 instances with 3 activities, 5 of them with 4 activities, and other 5 with 5 activities. We also demonstrated the greater convergence speed of MOSOSS, which covered more than 90% of the Pareto set after much fewer iterations than it took the competing algorithms to find any solutions. The interested reader is referred to [14] for additional details.

4.2 Case Study: On-Demand Response to a Training Virtual Organization Request using the Preceptor Architecture

In this section, we report the results of the case study described in section 3.2.

Descriptive statistics for the six quality indicators are reported numerically in Table 4.1 and visually in the figure from Appendix D.

We were interested in the existence of differences between organization structures with respect to performance measures. First, we needed to test the effect of our factor (the organization structure, with three possible values: heterarchy, hierarchy, and holarchy) on each of our dependent variables (the performance measures).

We used the R rstatix package [15] to apply the Friedman test and the post-hoc tests. A significant result of the Friedman test would imply that at least two of the three organization structures differ with respect to the dependent variable in question. As reported in Table 4.2, Friedman test results indicated a significant difference ($p < .05$) between organization structures only in terms of time and CPU usage. Average memory usage was not found to differ significantly across the three organization structures. Hypothesis 3 was therefore not supported.

The next step was to perform pairwise comparisons in order to establish which specific organization structures differed significantly with respect to the dependent variables on which the organization structure was proven to display a significant effect. We therefore conducted post-hoc pairwise comparisons using the Nemenyi-Wilcoxon-Wilcox all-pairs test from rstatix [15]. Results are reported in Tables 4.3–4.4.

As the tables show, test results indicate significant differences ($p < .05$) between all pairs for both total and scheduling time. As expected, with respect to elapsed time, the hierarchical organization outperformed heterarchy, which, in turn, was faster than holarchy. Hypothesis 1 was therefore fully supported.

However, interpreting the results presented in Table 4.4, both holarchy and heterarchy outperformed hierarchy with respect to total CPU usage ($p < .05$). Hypothesis 2a was thus partially supported. The dif-

Table 4.1: Descriptive Statistics for the Performance Measures

Performance Measure	Organization Structure	Mean	SD	Median	IQR
Time (total)	Heterarchy	1701	95.2	1710	110.0
	Hierarchy	1135	88.3	1114	104
	Holarchy	2000	145.0	2002	101
Time (scheduling)	Heterarchy	743	78.9	705	106
	Hierarchy	153	91.5	150	97.8
	Holarchy	1002	142.0	996	99
CPU usage (total)	Heterarchy	11.8	5.76	11	3.75
	Hierarchy	13.4	6.67	12	4.75
	Holarchy	11.6	4.64	11	3
CPU usage (scheduling)	Heterarchy	17.7	7.67	17	7.75
	Hierarchy	19.0	18.2	14.5	14.8
	Holarchy	11.6	3.97	11	6
Memory usage (total)	Heterarchy	21.5	6.00	22.0	7.42
	Hierarchy	23.1	6.82	22.8	8.30
	Holarchy	25.1	8.64	25.2	12.0
Memory usage (scheduling)	Heterarchy	27.7	8.04	27.4	8.88
	Hierarchy	32.8	11.6	32.8	12.4
	Holarchy	29.0	9.59	30.1	11.2

SD = Standard Deviation; IQR = Interquartile Range

Table 4.2: Friedman Test Results

Performance Measure	F	p
Time (total)	100	1.93e-22
Time (scheduling)	100	1.93e-22
CPU usage (total)	12.3	0.00217
CPU usage (scheduling)	12.5	0.00190
Memory usage (total)	4.68	0.0963
Memory usage (scheduling)	3.88	0.144

ference between holarchy and heterarchy was not significant. In terms of CPU usage for the scheduling process, only the difference between holarchy and heterarchy was significant, with holarchy displaying a more efficient CPU usage. Hypothesis 2b was thus also partially supported.

Table 4.3: Pairwise Comparisons between Organization Structures: Time

	Total time			Scheduling time		
Organization Structure	Heterarchy	Hierarchy	Holarchy	Heterarchy	Hierarchy	Holarchy
Heterarchy	-			-		
Hierarchy	1.7e-06 ▲	-		1.7e-06 ▲	-	
Holarchy	1.7e-06 ▽	2.4e-14 ▽	-	1.7e-06 ▽	2.4e-14 ▽	-

Notes. Values reported in the table are the p-values for the Nemenyi-Wilcoxon-Wilcox test.
▲ = organization structure on row outperforms organization structure on column;
▽ = organization structure on row is outperformed by organization structure on column;
– = the difference between organization structure on row and organization structure on column is not significant.

Table 4.4: Pairwise Comparisons between Organization Structures: CPU Usage

	Total CPU Usage			Scheduling CPU Usage		
Organization Structure	Heterarchy	Hierarchy	Holarchy	Heterarchy	Hierarchy	Holarchy
Heterarchy	-			-		
Hierarchy	0.0141 ▽	-		0.2456 –	-	
Holarchy	0.9661 –	0.0065 ▲	-	0.0014 ▲	0.1386 –	-

Notes. Values reported in the table are the p-values for the Nemenyi-Wilcoxon-Wilcox test.
▲ = organization structure on row outperforms organization structure on column;
▽ = organization structure on row is outperformed by organization structure on column;
– = the difference between organization structure on row and organization structure on column is not significant.

Chapter 5

Discussion

In this thesis, we proposed an approach to offering individualized training via a strategic HMAS, followed by suggestions for the design and implementation of a system supporting such functionalities and numerical experiments comparing three different architectures (organization structures) for schedule negotiation.

Our theoretical arguments and results suggest that an entirely holonic organization pattern during VO formation not only allows for a more dynamic VO environment, but it is also more efficient in terms of CPU usage. However, it is more expensive in terms of computational time than a heterarchical or hierarchical organization structure. Experimental results suggest that, though decentralized organization structures (holarchy and heterarchy) perform better with respect to CPU usage, the hierarchy organization structure has an important advantage—that of being faster. Nevertheless, one of the main disadvantages of the hierarchical structure is its relative lack of flexibility. If subholons have at most the right to veto, an agreement with all subholons may be hard to reach. If VO formation and reconfiguration are entirely centralized, without accepting interrupts from subholons, the Potential VO Negotiator produces rigid configurations and schedules.

5.1 Utility

Our proposed architecture addresses on-demand personalized curriculum planning. Specifically, it offers support for novices in three major areas: identifying their training needs in terms of competences, finding an optimal combination of instructors for those competences that require training, and generating a personalized learning path through the holonic coordination of instructors.

The holonic structure of the platform creates a balance between centralization and decentralization by allowing holons at different levels of the holarchy to manage their subholons. Modeling VOs as subholons of the Education Market offers the advantage of allowing simultaneous collaborative relations (as trainers for competences in the same request) and competitive relations (as bidders for different requests) between instructors.

5.2 Original Contributions

The most important original contributions of the present thesis can be summarized as follows:

- proposing a framework for on-demand collaborative teaching. We reported on designing and implementing a solution to the growing demand for specialized training which also addresses UNESCO recommendations and recent critical reviews of the role of AI in education, in that it avoids substituting human teachers with AI.
- proposing an adaptation of a novel heuristic by completely redesigning MOSOS operators so as to better serve the purpose of solving PSP as a combinatorial MOP with task scheduling under time, budget, resource and task precedence constraints. We also provide an innovative approach to task scheduling using global search heuristics, by evolving partially scheduled solutions.
- bridging the gap between VOs and VO environments, by proposing a holonic organization in which VOs are subholons of the VO environment ("market"). The applicability of this idea may be extended beyond the scope of this thesis (training), to hypothetically any VO.
- proposing, testing and comparing competing approaches to scheduling negotiation during ISP. Based on our case study, we highlighted advantages and disadvantages of different organization structures during these processes.

5.3 Limitations

In the present thesis, we confined ourselves to providing a proof of concept for the purpose of demonstrating part of the functionality the Preceptor architecture is meant to support. Several important limitations that should be addressed in a future research agenda are listed in the following.

First, the architecture has yet to be validated on real-world scenarios, and stakeholder satisfaction with the quality of service needs to be measured. Additionally, more experiments comparing competing organization structures during VO formation need to be conducted to analyze the impact of parameters such as frequency of listening/monitoring tasks and fine-tune them based on results of sensitivity analyses. Second, more alternative structures for different phases of the VO lifecycle are worth being explored in future works. Third, a more advanced rating system for training quality should be considered, such as collaboration ratings.

5.4 Future Research Directions

Based on the aforementioned limitations, as well as on the architecture's potential of being extended, the following directions for a future research agenda can be identified.

First, the behavior of Preceptor during the VO operation phase could be refined in more concrete implementations of teacher collaboration. In order for them to be effective, these implementations should be based on an analysis of the needs, opinions and preferences of instructors. For example, data collection efforts could target members of European Universities[1], or other instruction providers offering learner-centered, flexible curriculum design and/or transdisciplinary goal-driven team formation, such as the EU Business School[2], or UNIR[3].

Second, given the great emphasis that has constantly been placed in the VO literature on agility, it would be especially interesting to distinguish between various reconfiguration triggers, such as changes in the required competences or changes in the schedule imposed by the Learner. The optimal behaviors of holons involved in the reconfiguration process may differ for different triggers, so each of these scenarios warrants a more detailed analysis.

Third, drawing on the Market of Resources model, future extensions of Preceptor may narrow down the search for potential VO partners by solving ISP in a two-stage search. First, in the passive stage, Preceptor could create a list of training offers that are relevant for the required competences and satisfy the constraints imposed by the Learner. Second, in the active stage, it could notify the Instructor agents in the list of the training request and allow the Learner to negotiate with them.

Fouth, developing the Preceptor architecture serves as preliminary step to what can be referred to as E-learning as a Service (EaaS) [10]. Instructor offers are services that could be described and discovered using the semantic web. Furthermore, Preceptor lends itself to being extended to a web-based platform supporting three user categories: (1) Employers, who need to recruit applicants for a certain job and use the platform to post the job announcement; (2) Instructors, who could use the platform to post training offers; (3) Learners, who could subscribe to the platform to read job announcements posted by employers and post training requests for the required competences they lack. All user interactions with the platform should be performed via a web interface allowing contributions to the knowledge base from all user categories. For example, employers could contribute by describing new competences they require and by specifying prerequisite relationships between these competences. A competence ontology may be collaboratively developed in this manner.

[1] see the European Universities Initiative: https://ec.europa.eu/education/education-in-the-eu/european-education-area/european-universities-initiative_en

[2] https://www.euruni.edu/

[3] https://en.unir.net

5.5 Dissemination of Research

Reviews and original research findings presented in this thesis were also reported in the following list of publications:

5.5.1 Journal Articles Indexed in the Web of Science

A.-F. Ionescu and R. Vernic, "MOSOSS: An adapted multi-objective symbiotic organisms search for scheduling," Soft Computing, vol. 25, no. 14, pp. 9591–9607, 2021. WOS:000641235700007

5.5.2 Articles/Book Chapters Indexed in the Web of Science

A.-F. Ionescu, "Methods and algorithms for creating and reconfiguring virtual organizations," Decision Making in Social Sciences: Between Traditions and Innovations, Springer, 2020, pp. 49–63. WOS:000640236300003

5.5.3 Other Chapters in Edited Books

A.-F. Ionescu, "Multi-objective evolutionary algorithms: Decomposition versus indicator-based approach," in Algorithms as a Basis of Modern Applied Mathematics, Springer, 2021, pp. 69–85.

A.-F. Ionescu and D.-M. Popovici, "Applications of multi-agent systems in social sciences: Virtual enterprises as an example," in Models and Theories in Social Systems, Springer, 2019, pp. 311–325.

5.5.4 International Conference Proceedings Indexed in the Web of Science

A.-F. Ionescu, "Designing virtual learning systems: Current trends and evaluation," in Proceedings of the 14th International Conference on Virtual Learning (ICVL), 2019, pp. 303–308. WOS:000506084800044

A.-F. Ionescu, "E-learning as a Service: Benefits of the semantic web and SOA for virtual learning," in Proceedings of the 14th International Conference on Virtual Learning (ICVL), 2019, pp. 401–407. WOS:000506084800059

A.-F. Ionescu and D. Sburlan, "AdABI: An adaptive assessment system based on Bayesian inference," in Proceedings of the 15th International Scientific Conference on eLearning and Software for Education (eLSE)—New Technologies and Redesigning Learning Spaces, 2019, pp. 288–295. WOS:000473322400039

5.5.5 Oral Presentations at International Conferences

A.-F. Ionescu and D.-M. Popovici, "Preceptor: A Proposed Architecture for an On-Demand Virtual Learning Platform", paper accepted for presentation at the 23rd International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, SYNASC 2021.

Chapter 6

Conclusions

To conclude, all objectives of the thesis have been met. In order to achieve objectives O1 and O2, we reported on conducting a literature review of methods and techniques for solving PSP. For achieving O3, we provided a problem statement and detailed the design of our proposed software solution. To reach O4, we described a case study we used to test the developed solution, and we reported the corresponding results—our proposed algorithm for solving PSP and its validation, as well as experimental tests of the efficiency of the developed VO environment.

All in all, we can mention several important original contributions of the research presented in this thesis:

- designing and implementing a solution to the growing demand for specialized training: an environment for on-demand collaborative teaching.
- explicitly framing VO formation/reconfiguration as special cases of PSP, and integrating the PSP and multiobjective optimization literatures in the VO framework
- proposing an adaptation of a novel heuristic (MOSOS) for solving PSP with task scheduling under time, budget, resource and task precedence constraints.
- bridging the gap between VOs and VO environments, by proposing a holonic organization in which VOs are subholons of the VO environment ("market").
- proposing, testing and comparing competing approaches to scheduling negotiation during ISP.

The Preceptor architecture may serve as a basis for the development of a web platform with the purpose of facilitating personalized curriculum planning and on-demand job training. Both learners and instructors could benefit from the use of the platform, and we argued that future extensions may also serve employers in search of job applicants. As such, the platform has the potential to connect instructors and learners to job demands and thus increase employment rate.

Bibliography

[1] K. Almohammadi, H. Hagras, D. Alghazzawi, and G. Aldabbagh, “A survey of artificial intelligence techniques employed for adaptive educational systems within e-learning platforms,” Journal of Artificial Intelligence and Soft Computing Research, vol. 7, no. 1, pp. 47–64, 2017.

[2] V. Buş, “Relația dintre siguranța psihologică, învățarea în echipă şi performanță: O perspectivă longitudinală,” in “Horia D. Pitariu” National Conference of Industrial and Organizational Psychology, Cluj-Napoca, 2016.

[3] L. M. Camarinha-Matos and H. Afsarmanesh, “Virtual enterprise modeling and support infrastructures: Applying multi-agent system approaches,” in ECCAI advanced course on artificial intelligence, Springer, 2001, pp. 335–364.

[4] N. Capuano and D. Toti, “Experimentation of a smart learning system for law based on knowledge discovery and cognitive computing,” Computers in Human Behavior, vol. 92, pp. 459–467, 2019.

[5] V. Caputi and A. Garrido, “Student-oriented planning of e-learning contents for Moodle,” Journal of Network and Computer Applications, vol. 53, pp. 115–127, 2015.

[6] C. Cooper and I. Rothmann, Organizational and work psychology: Topics in applied psychology. Routledge, 2013.

[7] M. Cossentino, N. Gaud, S. Galland, V. Hilaire, and A. Koukam, “A holonic metamodel for agent-oriented analysis and design,” in International Conference on Industrial Applications of Holonic and Multi-Agent Systems, Springer, 2007, pp. 237–246.

[8] S. Galland, S. Rodriguez, and N. Gaud, “Run-time environment for the SARL agent-programming language: The example of the Janus platform,” Future Generation Computer Systems, vol. 107, pp. 1105–1115, 2020.

[9] A.-F. Ionescu, “Designing virtual learning systems: Current trends and evaluation,” in Proceedings of the 14th International Conference on Virtual Learning (ICVL), 2019, pp. 303–308.

[10] A.-F. Ionescu, “E-learning as a Service: Benefits of the semantic web and SOA for virtual learning,” in Proceedings of the 14th International Conference on Virtual Learning (ICVL), 2019, pp. 401–407.

[11] A.-F. Ionescu, “Multi-objective evolutionary algorithms: Decomposition versus indicator-based approach,” in Algorithms as a Basis of Modern Applied Mathematics, Springer, 2021, pp. 69–85.

[12] A.-F. Ionescu and D.-M. Popovici, “Applications of multi-agent systems in social sciences: Virtual enterprises as an example,” in Models and Theories in Social Systems, Springer, 2019, pp. 311–325.

[13] A.-F. Ionescu and D. Sburlan, “AdABI: An adaptive assessment system based on Bayesian inference,” in 15th International Scientific Conference on eLearning and Software for Education (eLSE)—New Technologies and Redesigning Learning Spaces, 2019, pp. 288–295.

[14] A.-F. Ionescu and R. Vernic, “MOSOSS: An adapted multi-objective symbiotic organisms search for scheduling,” Soft Computing, vol. 25, no. 14, pp. 9591–9607, 2021.

[15] A. Kassambara, “Rstatix: Pipe-friendly framework for basic statistical tests,” R package version 0.6. 0, 2020.

[16] H. Mayr, “Teaching software engineering by means of a virtual enterprise,” in Proceedings Tenth Conference on Software Engineering Education and Training, IEEE, 1997, pp. 176–184.

[17] E. F. Moore, “The shortest path through a maze,” in Proc. Int. Symp. Switching Theory, 1959, 1959, pp. 285–292.

[18] A. J. Nebro, J. J. Durillo, and M. Vergne, “Redesigning the jMetal multi-objective optimization framework,” in Proceedings of the companion publication of the 2015 annual conference on genetic and evolutionary computation, 2015, pp. 1093–1100.

[19] F. D. de la Peña, J. A. Lara, D. Lizcano, M. A. Martínez, and J. Pazos, “A new approach to computing using informons and holons: Towards a theory of computing science,” Foundations of Science, vol. 25, no. 4, pp. 1173–1201, 2020.

[20] N. Protogeros, Agent and web service technologies in virtual enterprises. IGI Global, 2007.

[21] S. Rodriguez, N. Gaud, and S. Galland, “SARL: A general-purpose agent-oriented programming language,” in 2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT), IEEE, vol. 3, 2014, pp. 103–110.

[22] UNESCO, Beijing consensus on artificial intelligence and education: Outcome document of the International Conference on Artificial Intelligence and Education ”Plan education in the AI era: Lead the leap”, 2019.

Abstract

The present thesis focuses on on-demand personalized virtual learning, in response to a specific training request emitted by a learner. The thesis is structured as follows:

In Chapter 1, we provide an introduction to virtual learning and a critical view of the limitations or gaps in the current developments. A special focus is dedicated to Virtual Organizations as a potential solution to virtual learning.

In Chapter 2, we state the aim of the thesis—to design, implement and test a market-like environment for ensuring on-demand individualized training to a learner via a virtual organization (VO)—and derive specific objectives.

Chapter 3 presents the research methodology we used in our studies. First, we provide the justification for our design choices in developing our proposed heuristic for solving multiobjective PSP, then we frame the instructor selection problem as a special case of PSP and introduce the proposed Preceptor architecture for a holonic market-like Virtual Organization environment supporting the whole lifecycle of a training Virtual Organization. Finally, using a case study, we provide a test of the architecture and a comparison of three competing organization structures that may be used during the schedule negotiation process.

In Chapter 4 we first present our newly-developed algorithm for generic PSP instances with task scheduling under time, budget, task precedence and resource constraints, the Multiobjective Symbiotic Organisms Search for Scheduling (MOSOSS). We then analyze the case study, comparing three different organization structures of the training VOs in the process of VO formation/reconfiguration —heterarchy, hierarchy, and holarchy. Results of the Friedman test and the Nemenyi-Wilcoxon-Wilcox all-pairs test revealed that, while a hierarchical organization structure may lead to a shorter execution time as compared to decentralized organization structures, an entirely holonic organization structure may be more effective with respect to CPU usage.

A discussion of our propositions and results, as well as of limitations and future research directions, is presented in Chapter 5. The original contributions of the reported research are summarized in Chapter 6, which concludes the thesis.