# On Human Resource Adaptability in an Agent-Based Virtual Organization*

Costin Bădică[1], Elvira Popescu[1], Grzegorz Frackowiak[2], Maria Ganzha[2], Marcin Paprzycki[2], Michal Szymczak[2], and Myon-Woong Park[3]

[1] University of Craiova, Software Engineering Department
Bvd.Decebal 107, Craiova, RO-200440, Romania
{badica_costin, popescu_elvira}@software.ucv.ro

[2] Systems Research Institute, Polish Academy of Science, Warsaw, Poland
{maria.ganzha,marcin.paprzycki}@ibspan.waw.pl

[3] Korea Institute of Science and Technology, Seoul, Korea
myon@kistmail.kist.re.kr

**Summary.** This paper discusses subsystem responsible for providing human resource adaptation through software-supported training in an agent-based virtual organization. Attention is focused on the requirements, functionalities and components of this subsystem and its interactions with other parts of the system.

## 1 Introduction

In our recent work on agent-based virtual organizations ([3, 4, 12, 13]) we have argued that support for collaborative work in a project-oriented organization must be *adaptable* on a number of levels. Basically, this means that as projects carried within the organization evolve, the behavior of the support system (including usage and access to resources) should evolve as well. Consequently, we have setup the goal of developing a system to meet this requirement.

In our earlier work we have argued that emergent software technologies such as software agents [16] and the Semantic Web [11] should be the base around which the proposed system is conceptualized. In particular: (i) organizational structure consisting of specific "roles" and interactions between them should be represented by software agents and their interactions, and (ii) domain knowledge, resource profiles and resource matching have to be represented using ontologies and semantic reasoning.

Separately, adaptability within the organization was sub-divided into:

- *System adaptability* obtained through: adapting various "structures" within the agent system; and adapting resource profiles.

* Work of G. Frackowiak, M. Ganzha, M. Paprzycki. M. Szymczak and M.-W. Park was partially sponsored by by the KIST-SRI PAS "Agent Technology for Adaptive Information Provisioning" grant.

- *Human resources adaptability* achieved by (e-)learning.

In our previous work we have outlined processes involved when a task/project is introduced into an organization (approached from the point of view of resource management) [12]; in [4] we have approached the proposed system from the point of view of roles played by various entities identified in [12]; while in [13], we have outlined how ontologies are going to be used in the proposed system. This allowed us to conceptualize, in [4], which roles can be played by software agents alone, by human(s), and by human-agent team(s). Separately, focusing on human resources, in [3] we have sketched initial ideas of how e-learning can be introduced into the system to support adaptability.

In this paper we further explore adaptability of human resources, by naturally enhancing the task oriented view of the work with introduction of *training tasks*. Observe that as work carried out by the organization is focused on tasks, this naturally leads to the idea of training as "workplace learning"—i.e. learning taking place in close relation with usual activities performed in the workplace. In this context, we introduce two approaches to training in the system: *reactive*, and *proactive*.

We start the paper by briefly summarizing main features of the system. Then we detail our proposal for achieving human resource adaptability by introducing training tasks covering both reactive and proactive approaches. During this analysis we identify two specialized units that are needed in the system: the *Competence Management Unit* and the *Training Management Unit* and outline their main functionalities.

## 2 System Overview

Our system is conceived as an agent-based virtual organization, which provides adaptive support for project-based collaborative work [3]. Structure of the organization and interactions between participants are represented using software agents and their interactions. Each human participant (member of the organization) has an associated *Personal Agent*—(*PA*). Domain knowledge and resource (human and non-human) profiles are overlaid on top of ontologies [3, 4, 13].

We assumed that work carried out within the organization is project-driven (however, the notion of the project is very broad and includes installation of Cable TV as well as design and implementation of an intranet based information system for a corporation). Based on analysis performed in [4] several components of the system were identified (note that these are "roles," rather than "individuals"):

- *Project Manager* (*PM*) is created whenever a project proposal is submitted to the organization. Its main duties cover: formulation of project requirements, if project is accepted formulation of project schedule, assignment of resources to project activities, supervising project's progress and assuring its completion.
- *Analysis Manager* (*AM*) analyzes project requirements and formulates documents that is used to support the decision if the project should be accepted or not.
- *Organization Provisioning Manager* (*OPM*) is responsible for management of resources of the organization.

- *Resource Procurement Unit* (*RPU*) represents an interface between the organization and the "outside world." Its role is to seek and potentially deliver resources requested by the *OPM*.
- *Task Monitoring Agent* (*TMA*) is responsible for monitoring a given task according to its schedule and informing the *PM* in the case of any problems.
- *Quality of Service Management Unit* (*QoS*) is responsible with quality control of tasks completed by workers.

## 3 Conceptualizing Training Tasks

Following [14], we understand *training* as "acquisition of knowledge, skills, and competencies as a result of the teaching of vocational or practical skills and knowledge that relates to specific useful skills". In the context of our work we refer to training as "workplace learning," i.e. training that is closely related to the needs of improving performance of performing tasks in an organization. Clearly, this type of training can and should be closely related to projects carried out by the organization.

First, note that when conceptualizing a training task, three aspects need to be taken into account: (i) *timing*, i.e. when training should be started (possibly also: when it should end) (ii) *goals*, i.e. what should be goals of each specific training activity and, (iii) *trainees*, i.e. who should be enrolled in a given training task.

In our system we found the timing issue crucial for distinguishing between the *reactive* approach, and the *proactive* approach. Basically, the *reactive approach* may occur in two situations. First, when a new project is introduced in the organization, and consequently defined and represented within the support system. Here, based on the project requirements and available resources, management of the organization may decide to enroll selected human resources into training activities, carried out "within the project" (i.e. within the project budget and time). Second, while the project is running either the *Personal Agent* of an employee or the project manager (*PM*) may decide to enroll an employee or group of employees in ad-hoc training tasksto acquire specialized knowledge increments, to solve specific problems.

The *proactive approach* occurs when the management within the organization, based on current market conditions, history of the interactions between the organization and the external environment, specific regulations, expected projects, etc., decides to enroll selected employees into training tasks. Let us now look into these two approaches in more detail.

## 4 Reactive Approach

### 4.1 First Case—Project Level

The first case of the reactive approach follows the process of introducing a task into the system outlined in [12] and enhances it with activities and roles required for training. Decision to follow this approach is taken at the project level.

When a new project request is submitted to an organization, a new *Project Manager* (*PM*) is created. The first responsibility of the *PM* is to assure that project requirements are carefully analyzed and, on the basis of this analysis, a decision is made to accept or reject the job.

Analysis and suggestion as to what the decision should be are made by the *Analysis Manager* (*AM*). The *AM* produces the *Requirements Analysis Document* containing a detailed specification of the project together with resources required and their associated desired competencies. Some details of this process are given in [12]. Note that the final acceptance/rejection decision is made by the *PM* (i.e. the project *PM*, or an appropriate *PM* on a higher hierarchical position in the organization).

Next, assuming that the job is accepted, differently from [12], *PM* creates an *Abstract Project Schedule* based on the general competencies available in the organization that she is aware of. The initial project schedule is called "abstract" because it contains only refined descriptions of necessary resources from the *Requirements Analysis Document*, i.e. the *Resource Request Descriptions*, rather then the resources themselves. Actual assignment of resources can be fixed later based on matching of required competencies from the *Resource Request Description* with available competencies of existing resources. Note that resources can be either human and non-human. As focus of this paper is on human resource adaptability, here we consider only this situation. For details of the general case please consult [12].

The next step of the process is to transform the *Abstract Project Schedule* into an actual *Project Schedule*. In this process, the *PM* can propose utilization of resources that she knows (see [12]). However, resources that *PM* knows might not be sufficient, i.e. either they are unavailable or they do not exist. In this case the *PM* contacts the *Organization Provisioning Manager* (*OPM*). The *OPM* is the general resource manager of the organization. The process of seeking missing resources, conducted collaboratively by the *PM* and the *OPM* can produce the following results:

- A matching resource exists and is currently available. This is the simplest situation, i.e. the resource will be assigned to the project.
- A matching resource exists in the organization but it is not currently available. Here we have two cases: (i) the *Project Schedule* can be updated by re-scheduling tasks requiring this resource such that he/she will be available (obviously, this is possible only if project constraints, like deadlines, are not violated), and (ii) if the *Project Schedule* cannot be updated to accommodate the available resource then we treat this situation as the one in the next point.
- A matching resource does not exist in the organization. Here there are two cases: (i) an external resource is found by the *Resource Procurement Unit* (in our case someone with adequate skills is hired), or (ii) a decision is made to train an available resource that has some but not all of the required skills for the job.

Note that, additionally to [12], we propose that human resources should be handled in such a way to accommodate adaptability by interleaving work with training. Thus, we provide an overview of the process of training an available human resource that has some but not all of the required skills for the job. Here, note also that training decision depends on the following factors: (i) current level of competence of available resources; (ii) competence increment that represents the gap between available and required competencies for the job; (iii) project constraints. Furthermore, introduction of the training tasks may require also the update of the *Project Schedule* to accommodate the new training activities within the project timing and costs.

### 4.2 Second Case—Individual or Group Level

The second case of reactive training occurs after the project has already started (so either all the necessary resources were found within the organization, or/and training sessions were carried out at the beginning of the project, or/and external resources were hired). Usually decision to apply additional training tasks can be undertaken either at the individual or group (part of the team) level. Note however that such decision may be also made at the project level on the basis of the *PM*'s observations of what is happening within the project while it is running.

We will now illustrate this case in the framework of a software and services company. Let us consider an example of a customer requesting creation of an intranet and a company knowledge portal.

Most often, in the case of IT projects, the decision to start a project and to assign human resources to tasks is taken even if there is no perfect match between the competencies of the available resources and needs of required tasks. As a result human resource adaptability issues may arise during the unfolding of the project (e.g. finding tips on how to overcome the vulnerabilities of the MD5). In this case, programmer informs her *PA* about the missing information that she needs in order to carry on her task. It is the job of the *PA* to provide the human with the needed resource—either non-human (a manual, a tutorial, a book excerpt, etc.) or human (a peer who possesses the needed information and is able to share it). Note that each such resource request from the part of a programmer may and usually does represent an interrupt in his or her current task. However, this is usually tackled locally, without the intervention of the *PM* (unlike the case of organizing training activities, which involves the decision of the *PM* of a given project or another, even higher, level authority within the organization). However such request involves approval of the contacted peer.

In this example, these actions pertain to the "Direct support provided by the *PA* to facilitate the master needs", more specifically "searching for a resource" [4]. In the search of the needed information, the *PA* will query other *PA*s, profiles of which indicate that they represent humans interested in the given subject (and thus store knowledge about pertinent resources); this scenario is similar to well-known cases of collaborative filtering (see, for instance [9]). As pointed out in [3], a non-human resource (e.g. an educational material) is considered appropriate to a learner if there is a correspondence between the following characteristics of the non-human and human resource profiles, as described in their respective ontologies: (i) the prerequisites level and the knowledge level; (ii) the intended purpose and the actual learning goal; (iii) the most appropriate learning style and the recorded cognitive characteristics; (iv) the desirable hardware and software features of the used device and the actual platform available. It is the role of the *PA* to apply ontological matchmaking realizing these criteria (recall that *all* resources are described as ontological instances) and to judge suitability of a particular resource for its user-master. Upon receiving responses, on the basis of implicit and explicit feedback, the *PA* will adjust its "trust values" that it applies to recommendations provided by other *PA*s. This step is needed to rank responses in the case when a large number of them is obtained from peer-*PA*s.

In the case when the needed resource is human (peer help), the *PA* will contact appropriate *PA*s, based on their associated profiles. For example, whether a particular programmer possesses the needed information on MD5 can be easily seen from her profile, since an overlay model on domain ontologies is used to represent human resource profiles [5, 6]. In case there are several peers that possess the required information, a "near-peer-matching" rule can be applied—i.e. directing the user to the peer with a slightly higher knowledge and skills level. This insures a fair distribution of help demands, avoiding the situation that most skilled programmers will be overburdened. The matching will be done by means of negotiations between the *PA*s of programmers and will be based on their profiles and schedules.

To illustrate this approach, let us assume that a senior programmer has the information on MD5 required by a junior programmer. However, if the same information is available from a junior programmer, then the *PA* will assign the help task to the latter. Besides insuring a fair load, this approach could also provide the most efficient training, since the trainer's competence level will closely match the trainee's level.

In case the same help request appears several times from the part of different programmers (and each such request is stored in the project *log*), the *PM* (that analyzes the project *log*) might consider organizing an ad-hoc training on the topic, eventually involving only a subgroup of the team interested in that specific topic, in order to optimize the time spent by the programmer who plays the role of the trainer.

Similar situation takes place when the *Quality of Service* module reports that a task has not been carried correctly by one or more team members (all reports from the *QoS* are also collected in the project *log*). Analyzing the project *log* the *PM* may decide that a just-in-time training is needed for one or more team members to improve their skills and reduce number of incorrectly completed tasks.

Once the ad-hoc training is carried out, each *PA* adjusts the profile of its user.

## 5 Proactive Approach

Consider the situation when a new project request is received and, for various reasons (which might include, among others, that required resources are missing and/or requested competencies are unavailable), the *AM* determines that it should be rejected. Moreover, assuming that situation like this repeats, the management is faced with deciding: (i) to continuously reject similar project proposals; (ii) proactively involve available human resources in training tasks; or (iii) hiring new staff. While situations (ii) and (iii) are instances of human resource adaptation at the organizational level, clearly only option (ii) is within the scope of this paper. Note that other scenarios pertinent to the proactive approach include the organizational management that expects a certain set of projects to materialize within short or mid-range perspective, an expansion or a change in direction of the organization, or more generally long-term and semi-long-term goals and strategies of the organization. Here, the same three possibilities of dealing with availability of human resource competencies arise.

Note that while the *reactive approach* involves mainly decisions at the project level, the *proactive approach* involves mostly decisions at the higher organizational level. These decisions are based on conditions like: recurring competence and expertise needs of incoming project requests, availability of time and financial resources, specific regulations at the national and/or regional level, corporate strategy, etc. Separately, note that granularity of training tasks (and consequently costs, time, and effort) in the reactive approach are expected to be substantially smaller than in the case of proactive approach (i.e. individual and/or small group training, focused training in the reactive approach; larger groups, broader training scope in the proactive approach). For example, proactive training can include continuing professional education, initial training for new employees (e.g. “school to work transition”), coaching and motivational seminars, group/team building activities, etc.

## 6 Competence and Training Management Units

Based on the material presented thus far, as well as on ideas found in related works [10, 15], two specialized units are going to be added to the proposed system (following [12], we use the term *unit* with specific roles for describing these entities):

- *Competence Management Unit* (*CMU*)—responsible for management of competencies;
- *Training Management Unit* (*TMU*)—responsible for management of training activities.

In what follows we outline the main functionalities of these units and their interactions with existing units in the system (see also Figure 1).

### 6.1 Competence Management Unit

The *CMU* is responsible for management of competencies within the organization. Representation of competencies will use a competence ontology described in [1, 7, 10], and associated reasoning mechanisms proposed in [8].

Functionalities of the *CMU* comprise: (i) management of individual competencies of available human resources; this requires the ability to represent, record and update competencies at an individual level; (ii) provisioning of a global view of competencies available at the organizational level; this facility is required for example by the *AM* to be able to asses if the organization has competencies “good enough” to accept a given project; (iii) qualitative and quantitative reasoning about matchings between available and required competencies; this functionality is needed to help decide to hire new staff [2, 8], assign human resources to tasks or enroll human resources into training. Note that the *PM* and the *OPM* will have to interact with the *CMU* during the process of fixing the problem of missing resources. Furthermore, the *CMU* will utilize information from the *Quality of Service* unit that assess work done by individuals and teams (each time a task is completed the *QoS* checks the result). This being the case the *QoS* can provide the *CMU* with information which tasks have been successfully or unsuccessfully completed. This information, in turn can be used to assess which individuals, or teams need extra training (i.e. training needs can be assessed directly on the basis of on-the-job performance).

### 6.2 Training Management Unit

The introduction of the *TMU* is motivated by the need of a specialized unit that is capable of formulating training goals for employees engaged in training activities based on the contextual conditions that resulted in training being requested at various levels within the organization: individual, group, project and organization.

Following [15], the main functionalities of the *TMU* are defined as follows: (i) identification of training goals by analyzing individual, project and business needs, available competencies, and contextual conditions when the training occurs, i.e. reactive (both cases) or proactive approach; (ii) choice of learning objects and selection of a learning strategy.

Note that function (i) requires the interaction with the unit responsible for deciding of the actual assignment of the training task (responsibility of the *PM* or other higher level authority) and with the *CMU* to evaluate the gap between existing and required knowledge. Function (i) requires interaction with *Resource Procurement Unit* (*RPU*) in the case a suitable learning object could not be located at the level of *TMU*. Obviously, work of the *TMU* involves interaction with the actual training unit (structure and functioning of which are out of scope of this paper). However, we can specify that the role of the *TMU* is to provide input specifying: (i) who needs training, (ii) which area needs to be trained, (iii) what training method should be applied, and (iv) when training should take place. The output of the training unit is certification of the the completed training and an assessment of trainee(s), which will be send to the *CMU* (and to appropriate *PA*s) to update profile(s) of trainee(s).

## 7 Concluding Remarks

In this paper we have conceptualized training-related processes involved in human resource adaptability in an agent-based virtual organization. The main accomplishments of this work are: (i) identification of three approaches for introducing training tasks into the proposed system, two approaches based on the idea of reactive training and one approach based on the idea of proactive training; (ii) identification of additional specialized units that are required to be included in the system: *Competence Management Unit* and *Training Management Unit*. Future work will be targeted on providing more details of interactions between units in the system and development and evaluation of a competencies ontology and associated reasoning mechanisms.

## References

1. Biesalski, E., Abecker, A.: Human Resource Management with Ontologies. In: *Wissensmanagement. Professional Knowledge Management, Third Biennial Conference, WM 2005, Kaiserslautern, Germany, 2005, Revised Selected Papers*. LNAI 3782, Springer, 2005, 499–507.
2. Bizer, C. Heese, R., Mochol, M., Oldakowski, R., Tolksdorf, R. and Eckstein, R.: The Impact of Semantic Web Technologies on Job Recruitment Processes. In: *Proc. International Conference Wirtschaftsinformatik (WI 2005), Bamberg, Germany*, 2005.
3. Ganzha, M., Paprzycki, M., Popescu, E., Bădică, C., and Gawinecki, M.: Agent-Based Adaptive Learning Provisioning in a Virtual Organization. In: *Advances in Intelligent Web Mastering. Proc.AWIC'2007*, Fontainebleu, France. *Advances in Soft Computing* 43, 25–40, Springer, 2007
4. Ganzha, M., Paprzycki, M., Gawinecki, M., Szymczak, M., Frackowiak, G., Bădică, C., Popescu, E., Park, M.-W.: Adaptive Information Provisioning in an Agent-Based Virtual Organization—Preliminary Considerations. In: *Proceedings of the SYNASC 2007*, IEEE CS Press, 2007, 235–241.
5. Gawinecki, M., Gordon, M., Paprzycki, M., Vetulani, Z.: Representing Users in a Travel Support System. In: Kwasnicka, H. et. al. (eds): *Proceedings of the ISDA 2005 Conference*, IEEE Press, 2005, 393–398.
6. Gawinecki, M., Gordon, M., Nguyen, N.T., Paprzycki, M., Zygmunt Vetulani, Z.: Ontologically Demarcated Resources in an Agent Based Travel Support System. In: R. K. Katarzyniak (ed.): *Ontologies and Soft Methods in Knowledge Management*, Advanced Knowledge International, Adelaide, Australia, 2005, 219–240.
7. HR-XML Consortium. `http://www.hr-xml.org/`.
8. Mochol, M., Wache, H., and Nixon, L.: Improving the Accuracy of Job Search with Semantic Techniques. In: *Business Information Systems*. LNCS 4439, Springer, 2007, 301–313.
9. M. Montaner, B. López, and J. L. de la Rosa, A taxonomy of recommender agents on the Internet, *Artif. Intell. Rev.*, 19(4), 2003, 285–330.
10. Schmidt, A., Kunzmann, C.: Towards a Human Resource Development Ontology for Combining Competence Management and Technology-Enhanced Workplace Learning. In: *OTM Workshops (2). On the Move to Meaningful Internet Systems 2006: OTM 2006 Workshops*. LNCS 4278, Springer, 2006, 1078–1087.
11. Semantic Web. `http://www.w3.org/2001/sw/`.
12. Szymczak, M., Frackowiak, G., Ganzha, M., Gawinecki, M., Paprzycki, M., Park M.-W.: Resource Management in an Agent-based Virtual Organization—Introducing a Task Into the System. In: *Proceedings of the MaSeB Workshop*, IEEE CS Press, Los Alamitos, CA, 2007, 458–462
13. Szymczak, M., Frackowiak, G., Ganzha, M., Gawinecki, M., Paprzycki, M., Park M.-W.: Adaptive Information Provisioning in an Agent-Based Virtual Organization—Ontologies in the System, In: Proceedings of the AMSTA-KES Conference, to appear.
14. Training. `http://en.wikipedia.org/wiki/Training`.
15. Tzelepis, S., Stephanides, G.: A conceptual model for developing a personalized adaptive elearning system in a business environment. In: *Current Developments in Technology-Assisted Education*, Formatex Publishing House, vol.III, 2006, 2003–2006.
16. Wooldridge, M.: *An Introduction to MultiAgent Systems*, John Wiley & Sons, 2002.

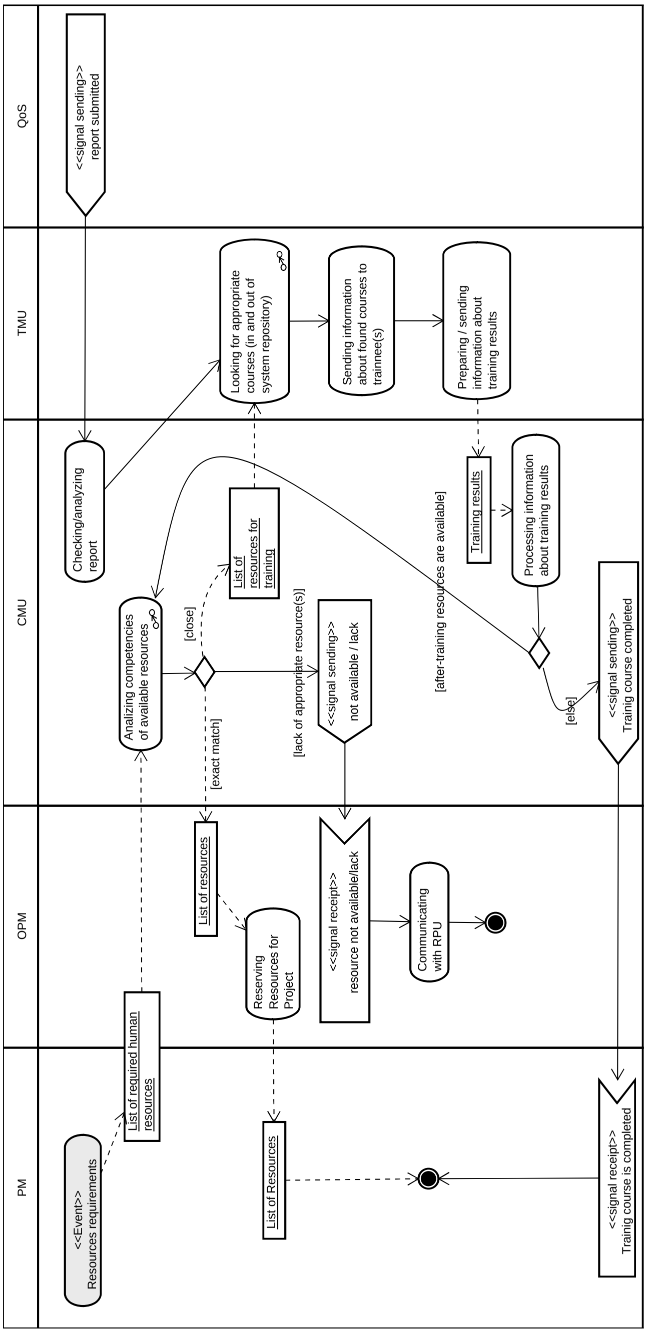
PM
OPM
CMU
TMU
QoS
<<Event>> Resources requirements
List of required human resources
Analizing competencies of available resources
Checking/analyzing report
<<signal sending>> report submitted
[close]
[exact match]
List of resources
List of resources for training
Looking for appropriate courses (in and out of system repository)
Reserving Resources for Project
List of Resources
[lack of appropriate resource(s)]
<<signal receipt>> resource not available/lack
<<signal sending>> not available / lack
Sending information about found courses to trainnee(s)
Communicating with RPU
[after-training resources are available]
Training results
Preparing / sending information about training results
Processing information about training results
[else]
<<signal receipt>> Trainig course is completed
<<signal sending>> Trainig course completed

**Fig. 1.** Interactions of *CMU* and *TMU* with other units in the system