# Adaptive Information Provisioning in an Agent-Based Virtual Organization — Preliminary Considerations *

Maria Ganzha, Marcin Paprzycki, Maciej Gawinecki, Michal Szymczak, Grzegorz Frackowiak
Systems Research Institute, Polish Academy of Science, Warsaw, Poland
{maria.ganzha, marcin.paprzycki, maciej.gawinecki}@ibspan.waw.pl

Costin Bădică, Elvira Popescu
Software Engineering Department, University of Craiova, Romania
{badica_costin}@software.ucv.ro

Myon-Woong Park
Korea Institute of Science and Technology, Seoul, Korea
myon@kist.re.kr

## Abstract

*In this note we consider design of an information provisioning subsystem for an agent-based virtual organization. Flexible delivery of information is based on matching of ontologically demarcated resource profiles, work contexts, and domain specific knowledge.*

## 1. Introduction

Let us consider an organization in which technology is used to support collaboration between research teams. Here, support has to go beyond document versioning or automatic flow of resources. What needs to be taken into account is, among others:

1. *representation of domain specific knowledge* (area where an organization is engaged)—to provide context for management of resources of each existing project (e.g. to specify semantic relations between projects, or establishing a specific "place" of a resource within the domain knowledge, which allows for resource indexing and clustering)

2. *representation of structure and interactions within a project* (and possibly between related projects)—to route resources based on project needs and responsibilities of team members (and sharing resources between projects)

3. *representation of resource profiles* (situated within the domain knowledge and the structure of the project, or multiple projects—when a resource is associated with more than one)—specifies among others: (a) place of a resource within an organization, (b) team members' *interests*, *needs* and *skills*, and (c) what to do with new/incoming resources

4. *adaptability of the system*—as time passes: (1) domain of interest of the project may expand, contract or shift (potentially affecting also its relationships with other projects); (2) team membership, as well as functional interrelationships between team members can change; finally, (3) their interests, needs and skills may evolve.

These concerns can be generalized beyond the collaborative work scenario. Let us assume that we utilize a notion of a *virtual organization* (*VO*) and within it define roles and interactions [17, 1, 2, 10, 11, 3]. In the *VO* its members need to access resources (human and non-human) to finalize their individual tasks (to complete assigned projects). Access to resources should be adaptive (change with assigned tasks, and evolve as tasks evolve) and personalized (team members require

*This work was partially sponsored by the KIST-SRI PAS "Agent Technology for Adaptive Information Provisioning" grant.

access to different resources depending on their roles in the project and the organization). Developing a system that would facilitate this is the aim of our project. In our earlier work [15] we have outlined processes involved when a task/project is introduced into an organization (approached from the point of view of resource management). The aim of this note is to view the system from the perspective of agents and their interactions.

## 2. Proposed system

It is claimed that the best approach to resource representation and personalized information delivery is ontological demarcation [4]. There, representation and management of resource flow can be achieved in a two-step process. First, roles of participants are specified, and second, the real-world organization is represented as an agent-based *VO*, where each person is represented by its *Personal Agent* (*PA*), which plays different roles in different situations. Additionally, auxiliary agents facilitate functioning of the system. Since such system is ontology-driven, human resources (represented by their *PA*s) as well as other resources (e.g. books) are demarcated using both domain ontologies (specifying the area of interest of a given organization ) and the organizational ontology. The remaining resources are described using only domain ontologies. In both cases an overlay model is used to specify profiles of resources (see [5, 12, 14, 13, 8]). Let us stress that ontologies, overlay-based profiles and agent systems are naturally adaptable. Ontologies can be easily modified, while adaptation of overlay-based profiles involves changes in weight of individual features. Finally, changes in the *VO* result in modifications of patterns of agent interactions ([9]). These considerations underline our approach to building an environment for supporting context aware personalized resource provisioning:

1. *Domain knowledge* is ontologically represented.
2. *Organizational structure* is based on an ontology of an organization and consists of interacting agents.
3. *Resource profiles* are represent using an overlay on top of organizational and domain ontologies.
4. *Resource matching* utilizes ontological reasoning involving resource profiles.
5. *System adaptability* is obtained through: (a) adapting structure of the agent system; (b) adapting resource profiles, (c) adjusting system ontologies.
6. *Human resources adaptability* is achieved through (e-)learning.

In Figure 1 we represent the proposed system through its use case diagram. This diagram was used in [15] to describe processes involved in introducing a task/project into the system (and this reference should be consulted for additional details). Let us assume that a task is proposed to the system. To handle it, a *Project Manager*(*PM*) is "created." It orders the *Analysis Manager* (*AM*), to analyze the proposed project and create report required to decide whether to accept the job. If the job is accepted the *PM* creates a *Project Schedule* (based on analysis of available and needed resources). We assume that every *PM* has knowledge about some resources in the *VO*. As a result of such analysis, available resources are reserved (*Resource Reservation* document). If the project requires additional resources the *PM* contacts the *Organization Provisioning Manager* (*OPM*) and requests them. The *OPM* has a knowledge about all resources in the *VO* and can either find them within the organization or ask the *Resource Procurement Unit* (*RPU*) to provide them from the "outside world." After the *Project Schedule* is created *PM* divides specified tasks among the group of workers (*PA*s) which she leads and for each task it instantiates a *Task Monitoring Agent* (*TMA*), in order to monitor its progress. Tasks completion is evaluated by a task-specific *Quality of Service* (*QoS*) module in order to assure quality of work.

## 3. Agents in the system

Previous section described a generic business process within a *VO*. Within it we have distinguished *roles* that are played by basic entities: (1) *PM*, (2) *AM*, (3) *RPU*, (4) *OPM*, (5) *TMA*, and (6) *QoS*. However, in our approach the *VO*, which blends human and non-human entities, is instantiated as an agent system. Specifically, (a) some roles specified in the use case diagram could be fulfilled by a software agent, (b) some roles are likely to be played by a human(s), while (c) some roles are likely to be completed by a team consisting of software agents and humans. Note, however that while specifics may depend on the particular organization (e.g. its size and/or domain of operation), abstract processes and interactions described above remain unchanged. Obviously, our goal is to utilize autonomous software agents as often as possible. Therefore, we have identified situations that are expected to require human intervention.

1. requirements analysis assessment
2. accepting a person as a *PM* of a project
3. changes in customer requirements
4. *OPM* cannot find required resource within the *VO*

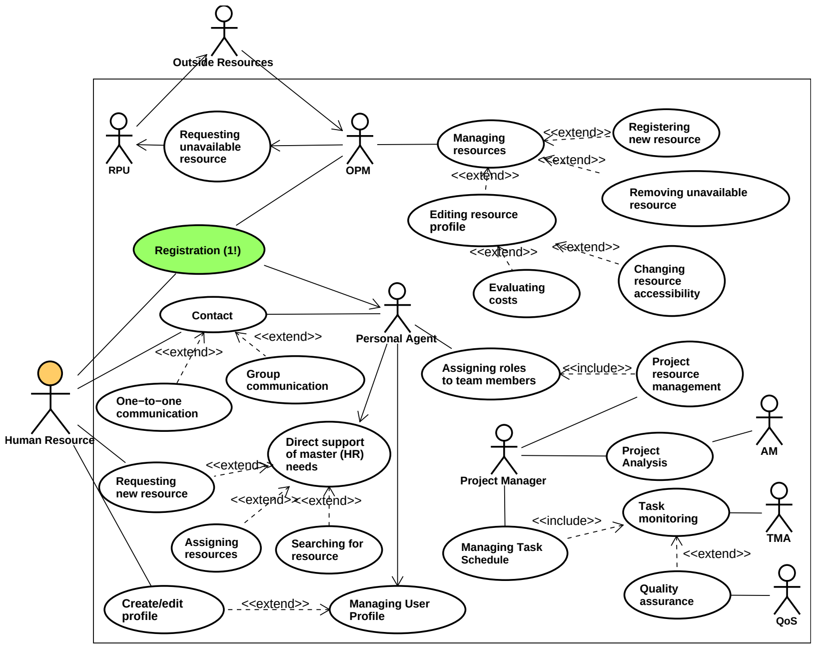


**Figure 1. Use case of the system**

5. accepting *Resource Reservation* document
6. accepting *Project Schedule*
7. final task acceptance

Identification of these situations allows us to establish where sole utilization of software agents is not possible. Now, we can look into each role in more detail.

Every human is represented by its *Personal Agent* (*PA*). Upon joining the system the *PA* registers with the *OPM* (the central resource manager; a one-time action). The *PA* represents its user in the *VO* and serves as her interface to the system. First, when a user needs a resource she can either look for it, or forward this request to the *PA* (which is what we are interested in). Note that in addition to searching for a given resource the *PA* is autonomously searching for supplementary resources that the user may need. The latter searches are based on earlier requests as well as the user profile. For instance user who asked about information about Jena software and the RDQL query language will receive needed information. However, later the *PA* may provide her with information about the SPARQL language; a result of queries that combine earlier requests and knowledge that the project involves persisting and querying ontologically demarcated content. The *PA* may also suggest an appropriate learning module (see [7] for more details). In the use case these actions are dubbed "direct support of master (HR) needs." The second major function of the *PA* is communication with other entities in the system. In the use case we have recognized two basic types of communication: (a) one-to-one, and (b) group (which involves all forms of one-to-all and all-to-one communication, such as mailing lists, RSS feeds, etc.). Note that this is also the channel through which the user is informed that he was assigned a specific task. Finally, we have associated the *PA* with user profile management. However, the user profile can be either managed by the *PA* or (parts of it that the user is authorized to adjust) directly modified by the user (in addition to other resources in the system specifically authorized to do so). Finally, the *PA* can undertake other roles (alone or as a part of the team).

The basic role of the *Personal Agent* is to support a "worker" responsible for completing a task (according to the schedule). In this case the user produces resources (e.g software artifacts or rear view mirrors for a Ford Explorer). These resources have profiles that are used, among others, to specify "what to do with them" (e.g. software delivered to the client, or mirrors send to a specific Ford plant to be installed). Profiles of produced resources are assigned by the *PA*. The worker (her *PA*) reports status of tasks to its supervisors: *Task Manager* that controls its progress and *Project Manager*.

Typically, the *PM* is a human (supported by her *PA*). Obviously, functions like: dedicating proposed project to be analyzed (by the *Analysis Manager*), rudimentary management of the task schedule, etc.

can be performed autonomously (by the *PA* with the role *PM*), while deciding task acceptance is likely to require an action of a human. Note, however that even acceptance of a task may be dedicated to an agent; e.g. if the task is installing a satellite TV antenna, and the requester lives within a specific area, and the installation team and hardware are available, the decision can be left to an agent. This is clearly not the case, when a company is to undertake a project that has a price tag of 1,500,000 euro. Transformation of the *PA* into the *PM* involves loading appropriate management modules, tailored to a given organization and denoted in its ontology (see, [16] for more details).

The *Organization Provisioning Manager* (*OPM*) has access to information about all resources within the *VO*. Recall that the *PM* may not have full access to such information. Furthermore, in a hierarchical *VO* *PM*'s access to information about resources varies (the "higher" in the hierarchy the *PM* is, the "broader" is the knowledge). *OPM*s main role is to provide resources to entities that request them (when they are authorized to make such a request). It may also provide information about the cost and availability of the requested resource. When the *OPM* does not find the resource within the *VO*, it requests that the *Resource Procurement Unit* finds it (outside of the *VO*). As a separate role, the *OPM* takes care of incoming resources. For instance, when a new book is received by the library its profile is instantiated by the *OPM*. Next, the *OPM* utilizes profile matching to distribute information about the book to appropriate *PA*s. Similarly, the *OPM* takes care of selected repositories and various active channels (e.g. RSS feeds). Finally, the *OPM* removes from the system information about resources that are no longer available, e.g. persons that left the *VO*. To fulfill these (and possibly other; see Section 4) roles, each one of them is "serviced" by a separate entity. In some cases these roles are fulfilled by agents (e.g. wrapping and distributing news feeds), while in other (e.g. hiring a person) they are facilitated by agent-human pairs.

The *Resource Procurement Unit* (*RPU*) is an intelligent interface between the organization and the "outside world." Its role is to seek and potentially deliver resources requested by the *OPM*. When sought resources involve payment for their use, results of the search can be within the budget (and delivered to the *OPM*), or outside of the budget and then the *OPM* is informed by the *RPU* and this information is forwarded to the *PM* to determine the further course of action.

Usually, the *Task Monitoring Agent* (*TMA*) can be instantiated as an autonomous agent. When created, it receives a task schedule and proceeds to monitor its execution. On the basis of milestones specified in the schedule it controls timeliness of their delivery (obviously, it can be also remind users about upcoming deadlines). If a given deliverable is not send in time its role is to inform the *PM*.

Finally, the *Quality Service Manager* (*QoS*) is responsible for testing deliverables. For a very simple task, this could be an agent (e.g. when an Internet connection is installed, it can interact with the user to establish that it works correctly). However, in most cases it will involve multiple agents and humans.

## 4. Application

Let us now discuss how the proposed system can be adapted to support a personalized information provisioning scenario, which involves support for the "duty travel" at a research institute located in East Asia. Here workers use an intranet system to apply for "business travel" and to submit a trip report. Our aim is to provide users with additional functionalities thought off as important by the authorities. First, if the employee travels to a given city, it is desirable that she visited also other institutions (research institutes, universities or companies), or persons that her institute has contacts with and that are in the area. The main reasons here is that cost of round trip air travel to most destinations is much higher—in a relative sense—than local costs of stay extended by a few days. Second, it is believed that a recommender that could facilitate advice where to stay and eat could also be of value. Note that specific needs of researchers from East Asia may differ from these of their European colleagues (consider for instance Malesian researchers who are confronted with typical Finnish food). In addition to these examples of personalized information delivery the system is expected to feature functions that will help researchers in all phases of duty trip participation; from the preparation of the initial application until filing the final report. In Figure 2 we present the activity diagram of processes involved in duty trip support. The initial interaction starts from the user logging to the duty trip support module and requesting that a new trip is initialized. This results in a new *travel resource* being initialized. This resource is an "almost empty" instance of the *institutional travel ontology*. The only information that it contains is about the employee that initiated the process. This step is followed with preparation of the duty travel application document. Here the "conversation" between the *Personal Agent* and the user has two goals (and is facilitated through a sequence of templates). First, to collect information necessary to complete an application. Second, if the user is interested in this (she may not be, as she may have time only to attend

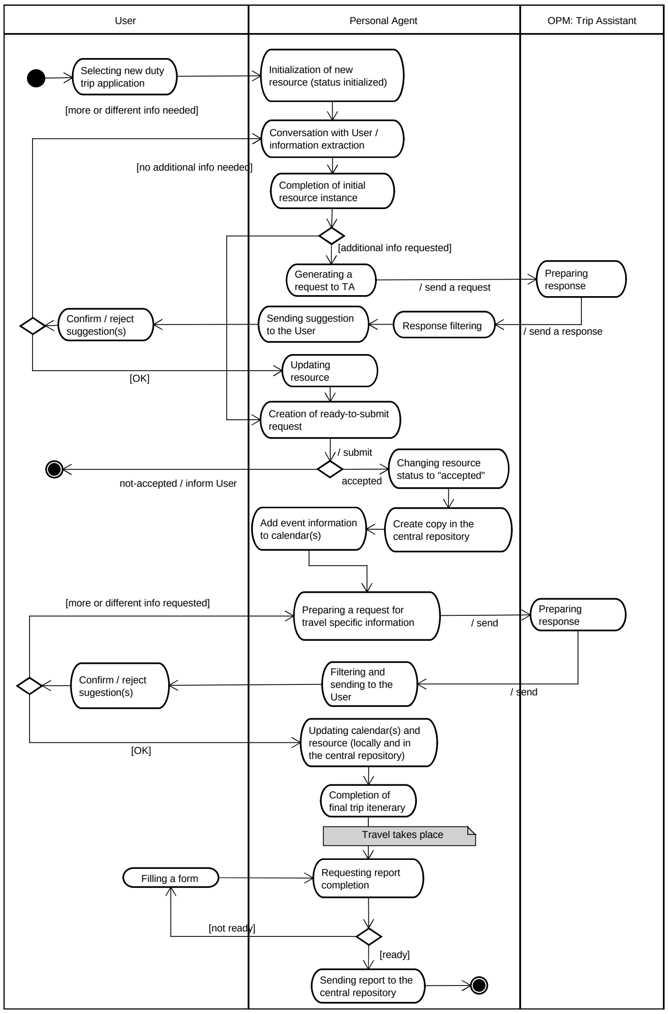


**Figure 2. Activity Diagram of the system**

a conference), to advise what other institutions and/or individuals can be visited in the area. All collected information is stored within the resource resulting in: (a) the resource containing all accepted information, and (b) ready to submit application (formatted according to the standards of the organization). Note that information about possible additional places to visit is based on queries of the central repository (an archive of all resources containing information about employees past trips); e.g. find all trips to Amsterdam that involved other goals than conference participation and happened within last 2 years. However, all suggestions are filtered by the *PA* according to the user profile (see [6]).

The request is submitted to an appropriate manager and can either be rejected (which ends the process) or accepted. Trip acceptance results in status of the resource changed into “approved” and its copy created in the central repository. This is done because: (1) the institution may want to advise its employees that a given person will be in a given location at a certain time—and thus it is necessary to keep data in the central repository; (2) a copy of the resource should be kept locally so that user can work on the trip report even if she is off-line—when connecting to the net, her *PA* will upload the updated resource to the central server. Finally, various calendars: personal, group and central are fed information about the upcoming trip, allowing a number of institutional activities to take into account presence / absence of a given employee at work.

The second phase of the process consists of completing the trip itinerary. Here the recommendations concerning travel details are provided to the user. This is an interactive process utilizing travel ontology developed in [9] and recommending techniques discussed in

[6]. Note that in the proposed system we have an easy access to explicit and implicit feedback, as pertinent “impressions” are a part of the final travel report. Since all travel resources are ontologically demarcated, their utilization within the proposed system is relatively simple. However, due to space considerations we have to omit further details. Results of this phase are: (1) final travel itinerary, (2) update of the travel resource, and (3) possibly a calendar update.

The last part of the process is the work on completion of the travel report. Here, again, the user is supported through a sequence of templates and checks that assure that all required information is included. This is the role of the *PA* that facilitates this process. After the report is completed it is submitted in an appropriate authority (*PM*), while the resource is archived in the “system knowledge base” for future uses (suggested above).

Observe that the duty travel support fits within our vision of the *VO*, as the proposed functions have to be supported in it (there exist a large number of organizations that support and require duty travel of their employees). Thus, observe that the *PA* supports its user and is responsible for facilitating all interactions (dialogs) and querying the *Travel Assistant*. A the same time, actions of the *Travel Assistant* are enclosed within the roles of the *OPM* (the *Travel Assistant* is assumed to be a “part” of the *OPM*). From the description follows that the *PA* plays also roles of the *TMA* (when pushing the user to complete the report) and the *QoS* (when it assures that all necessary fields of the report are completed). Finally, *PM*(s) have been omitted from the description, but exist as authorities that approve the travel and collect the final report.

## 5. Concluding remarks

In this paper we have presented our approach to developing and agent-based virtual organization in which personalized information delivery is facilitated through ontological demarcation and semantic reasoning. We have focused our attention on describing identified roles and their realization by agents and humans. A duty travel support system scenario was used to describe how the system will work and how the proposed generic concepts can be used in practice.

## References


[1] Barnatt, C.: Office Space, Cyberspace and Virtual Organization, "‘Journal of General Management"’, summer 1995, 20(4), 78–92.

[2] Bleeker, S.E., The Virtual Organization, in: Leading Organizations, ed.: Hickman, G.R. Sage Thousand Oaks (CA) 1998.

[3] Dunbar, R., Virtual Organizing, in: International Encyclopedia of Business and Management, ed.: Warner, M., Thomson Learning, London 2001, 6709–6717.

[4] Fensel, D., Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce, Springer, 2001

[5] Fink, J., Kobsa, A.: User Modeling for Personalized City Tours, *Artif. Intell. Rev.*, Vol. 18, No. 1, Kluwer Academic Publishers (2002) 33–74.

[6] Ganzha, M., Gawinecki, M., Paprzycki, M., Gąsiorowski, R., Pisarek S., Hyska, W.: Utilizing Semantic Web and Software Agents in a Travel Support System, In: A.F. Salam, J. Stevens (eds) Semantic Web Technologies and eBusiness: Virtual Organization and Business Process Automation, Idea Publ. Group, Hershey, USA, 2006, pp. 325–359

[7] Ganzha, M., Paprzycki, M., Popescu, E., Badica, C., Gawinecki, M.: Agent-Based Adaptive Learning Provisioning in a Virtual Organization. In: K.M. Wegrzyn-Wolska and P. Szczepaniak (eds.), Advances in Intelligent Web Mastering, Springer, Berlin, 2007, pp. 112–117.

[8] Gawinecki, M., Gordon, M., Paprzycki, M., Vetulani, Z.: Representing Users in a Travel Support System. In: Kwanicka, H. et. al. (eds): *Proceedings of the ISDA 2005 Conference*, IEEE Press (2005) 393–398.

[9] 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.

[10] Greer, J., McCalla, G.: Student modeling: the key to individualized knowledgebased instruction. *NATO ASI Series F*, Vol. 125, Springer-Verlag (1993).

[11] Goldman, S.L., Nagel, R.N., Preiss, K., Agile Competitors and Virtual Organizations, Van Nostrand Reinhold, New York 1995.

[12] Kobsa, A., Koenemann, J., Pohl, W.: Personalised hypermedia presentation techniques for improving online customer relationships. In: *Knowl. Eng. Rev.* 16 (2001) 111-155.

[13] Montaner, M., López, B., de la Rosa, J. L.: A Taxonomy of Recommender Agents on the Internet. In: *Artif. Intell. Rev.* 19 (2003) 285–330.

[14] Rich, E.A.: User modeling via stereotypes. In: *Cognitive Science* 3 (1979) 329–354.

[15] 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, *Proc. of the IAT Conference, IEEE CS Press*, Los Alamitos, 2007, in press.

[16] Tu, M.T., Griffel, F., Merz, M., Lamersdorf, W.: A Plug-in Architecture Providing Dynamic Negotiation Capabilities for Mobile Agents. In: Kurt Rothermel, F. Hohl (Eds.): *Proc. MA’98: Mobile Agents*, Stuttgart, Germany. LNCS 1477, Springer, pp. 222–236, 1999.

[17] Warner, M., Witzel, M.: Zarzadzanie organizacja wirtualna, Oficyna Ekonomiczna 2005.