

Validation of an Agent and Ontology-based Information Technology Assessment System

Czarnecki Adam, Orłowski Cezary, Ziółkowski Artur

Gdańsk University of Technology, Gdańsk, Poland

Cybernetics and Systems: An International Journal, Volume 41, Issue 1, 2010, p. 62-74, DOI:10.1080/01969720903408805

Abstract. The aim of this paper is to present a new method of the validation of an Agent and Ontology-based Information Technology Assessment System. In the introduction part of the paper, the characteristics of the proposed multi-agent system are presented. Next, some important details regarding the ontology functionality of this system are described and an approach to its verification process is proposed. The approach employs semantic description of system variables and implements a knowledge representation model in Ontology Web Language (OWL). The paper finishes with an illustrative case study for the presented methodology.

Keywords: agent systems, enterprise architecture, information technology assessment, ontology engineering

1 INTRODUCTION

Information Technology (IT) development stimulates the intense growth of various IT-based tools which support essential organizational processes. The enormous range of such tools and technologies calls for new management approaches, models and techniques when they are designed, developed and then implemented for the activities that are supposed to be supported. The project-driven approach to tasks connected with software development is currently becoming one of the most promising approaches studied and researched for this area. This new approach involves a variety of technologies which are integrated to support project tasks. The biggest challenge of this approach is how to bring into line proper technologies that can perform the required tasks and project management processes so they can work altogether as an efficient integration. This paper tries to address the above challenge by proposing a system that might be the required solution to the problem.

The multi-agent system (Altmann & F. Gruber 2001) supported by ontologies, which is introduced in this paper, is a tool which serves to help project managers in their choice of a proper method and technology of project management. The above-mentioned system for information technology assessment is based on intelligent units designed for acquiring and processing knowledge (Galant & Tubrycy 2001). The role of these units is performed by agents which are the basic components of multi-agent systems and, at the same time, are identified as independent components able to take action and perform tasks on behalf of the user. The idea of using agents for information technology assessment results directly from their

immanent features (particularly considering Semantic Web, where the function of intelligent agents seems to be essential) (Jennings 2001).

By definition, the function of agents is to acquire and process information; therefore, their significance for the assessment system might be seen in two aspects. The first aspect is to acquire information from the environment (supplying the system with data), the second is to generate an assessment and transfer it to the user (or customer). In order to avoid subjectivity, the assessment should be generated by independent agents. The assessment is based on the attached knowledge-base (facts), free from any preconceptions held by the user or expert.

For the purpose of the effective and reliable acquisition and processing of data, information and knowledge, it is recommended that the system makes use of ontologies which are defined as a clear and formal conceptualization of a given field (T. Gruber 1993). Literature identifies the application of ontologies (Noy & McGuinness 2001): to share a common understanding of the structure of information among people or software agents; to enable the reuse of domain knowledge; to make domain assumptions explicit; to separate domain knowledge from operational knowledge (data and meta-data); to analyze domain knowledge (including inference).

2 AGENT SYSTEM ARCHITECTURE

The agent system model (Angryk et al. 2002) results from a number of iterations which led to the evolution of the model in the course of the research. Successive work resulted in the final version of the information technology assessment model, which supports IT project management.

In the course of work on the model, the following model components have been accepted (Fig. 1):

- An expert system (based on a collection of facts and rules which provide inference based on fuzzy rules).
- Ontologies responsible for the formal correctness of information transmitted in the system.
- Other intelligent modules (an artificial neuron network, genetic algorithms, etc.) which are to support the inference process during assessment generation.

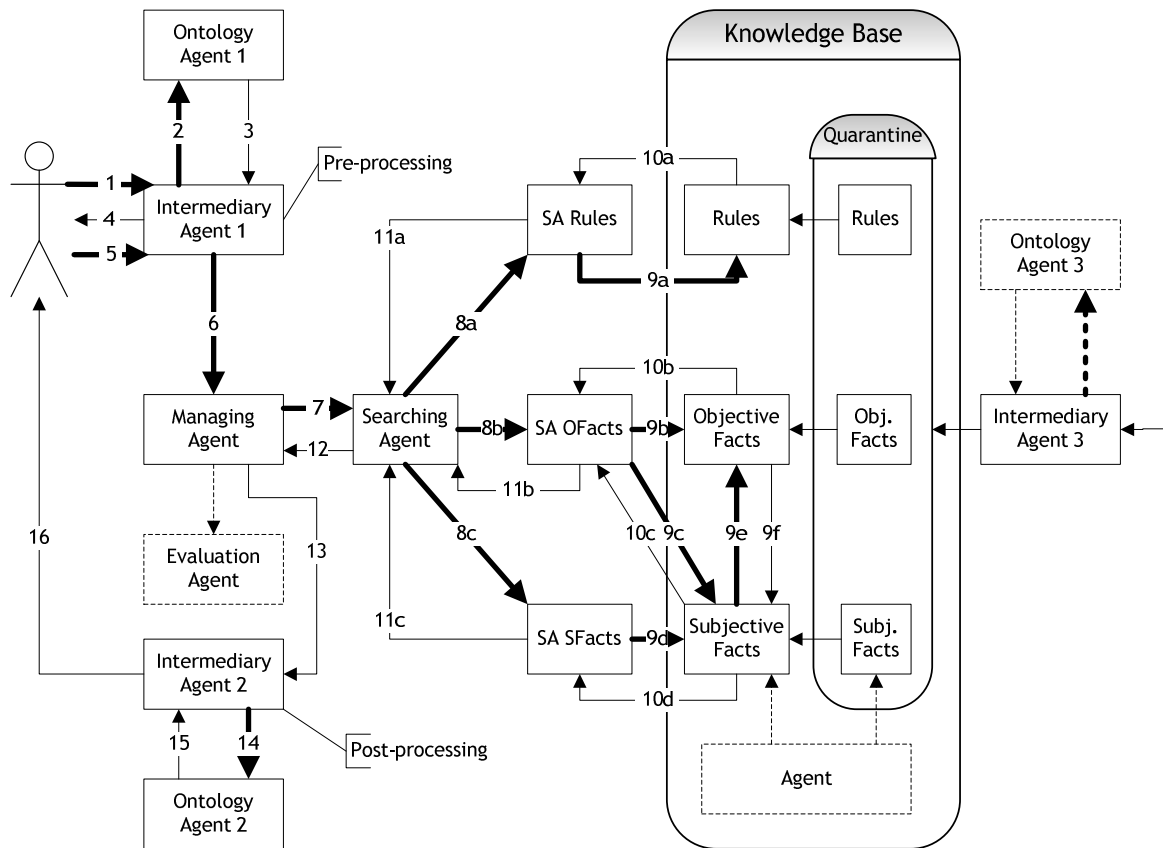


Fig. 1. Architecture of an IT assessment agent system.

A User makes a request concerning a specific domain (arrow no. 1—see Fig. 1). The request is handled by the Intermediary Agent 1 (with a Graphic User Interface [GUI] role). The Ontology Agent 1 is then used to provide the proper domain ontology (arrows 2 and 3). The questionnaire form is presented to the User by the Intermediary Agent 1 (4). The User inputs a question (arrow 5), which is further transferred to the Managing Agent (6). This agent sends the User's question to the Searching Agent (7). Depending on the question provided, different parts of the knowledge base may need to be utilized in this process. Thus, dedicated searching agents can be run—for rule bases (8a) or fact bases (8bc). The searching process progresses (9) and the results are collected (10) and passed to the Searching Agent (11). The answer is transferred to the Managing Agent, who sends it to the Intermediary Agent 2 (12). The answer must be provided to the User in a human-readable form, which is why ontology is used to serve the appropriate notions and semantics (14, 15). After these post-processing activities are completed, the answer is displayed to the User (arrow 16) (Orłowski & Ziółkowski 2008).

3 TASKS OF AN ONTOLOGY-SUPPORTED AGENT SYSTEM

The following step specifies the main functional features of the constructed model in order to adapt it to the needs of IT project management assessment. This approach has provided a prototype of a functional system for information technology assessment. All possible types of agents were defined, the agents were allocated proper tasks and the information flow among agents was calculated. This stage of the model's development resulted in a determination of the following system functions: acquiring information from the user (customer), searching the resources,

inference based on the existing resources, management of system agents. The above-specified functions have their impact on ontology use cases (Fig. 2).

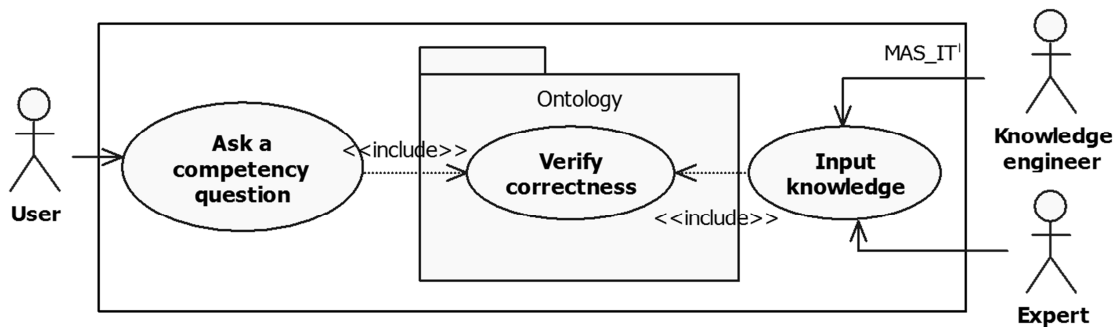


Fig.2. Ontology used in the system.

As shown in Figure 2, ontology functions are not directly available to the User. The reason is the role of ontology as an element of the mechanism which controls the correctness of data, information and knowledge that flow as the input and output of the system. Ontology is used by agents in order to: enter the User's query; return the answer to the User; supply the system with knowledge from an Expert or a Knowledge engineer

The above-mentioned applications of ontology might be extended when the system is widened by pre-processing mechanisms and when an artificial neuron network—aside from the expert system—is put into inference methods.

Once the functions and their application were defined, the mode of information flow in a given structure, i.e. a system of messages exchanged by the agents taking part in the assessment process, could be worked on. This approach led to the next stage—connecting the agents with each other and organizing the information flow. At this stage, the existence of ontology and the expert system turned out to be fundamental as the system needs to stay complete, i.e. it must provide a closed (within the model) cycle of information flow (information provided by ontology), but must also be credible and precise (which might be granted by inference derived from a rule-based expert system). Therefore, the agents have become subjects of the system, which manages both information flow and the whole assessment system.

4 SYSTEM VERIFICATION BASED ON A CASE OF ENTERPRISE ARCHITECTURE DEVELOPMENT

In this article, the process of enterprise architecture development has been used as a case to verify the assessment agent system. This approach, however, is possible only if enterprise architecture development is treated as a particular type of information technology project. For the sake of the case analyzed in this article, it is necessary to assume that the agent system is a tool for the assessment of information technology used for supporting enterprise architecture development. Before the verification of the agent system can be presented in this paper, the subject of enterprise architecture development should be introduced, along with several input parameters for the assessment system.

Enterprise architecture forms a cohesive platform for the needs of an organization and includes the following levels: business, application, data and technology. Thus, it is a base for creating corporate governance through the integration of existing IT systems. Such an approach to enterprise architecture shows that this concept

changes the performance of the organization (into a more flexible one) both on the operational and the strategic level. Components of information technology and business and their relationship, together with the influence of IT governance on this relationship form a new and consistent image of the organization.

The approach proposed in the assessment model assumes that the success of an information technology project is determined by three factors: (1) project complexity, (2) organization maturity and (3) customer maturity. The first factor—the complexity level of a project—can also be seen as a difficulty level and as a potentially larger failure risk. Organization maturity is indicated by the maturity level of its processes according to Capability Maturity Model Integration (CMMI) standards (where an organization can have one of the following levels: 1. Initial, 2. Managed, 3. Defined, 4. Quantitatively Managed, 5. Optimizing). Customer maturity is defined as a customer's information technology awareness connected with the project (the following cases can be specified: inappropriate and matched customer, appropriate and matched customer, inappropriate and mismatched customer and, finally, appropriate and mismatched customer).

4.1 ONTOLOGY STRUCTURE FOR THE PROJECT COMPLEXITY ASSESSMENT CASE

On account of the scope of terminology, ontology is generally subdivided into **upper ontology** (a.k.a. top-level ontology, or foundation ontology) and **domain ontology**. This article describes the use of domain ontology.

The whole system, as mentioned earlier, belongs to the field of information technology, in particular, considering parameters which determine the choice of methods and tools for IT project management. For the purpose of the current phase of the research, it has been assumed that an element of this field of knowledge will be sectioned off to cover the topic of project complexity assessment. Therefore, the ontology structure prepared for verification of the model will focus on categorizing and establishing a hierarchy of terms related to project complexity, developing and management areas, and documenting and enterprise architecture within a scope needed for the functioning of the whole agent system.

4.1.1 Semantic description of variables

The taxonomy of terms (sometimes abbreviated with respect to clarity) is shown in Figure 3. The taxonomy includes all the terms explained above in the section dedicated to ontology field.

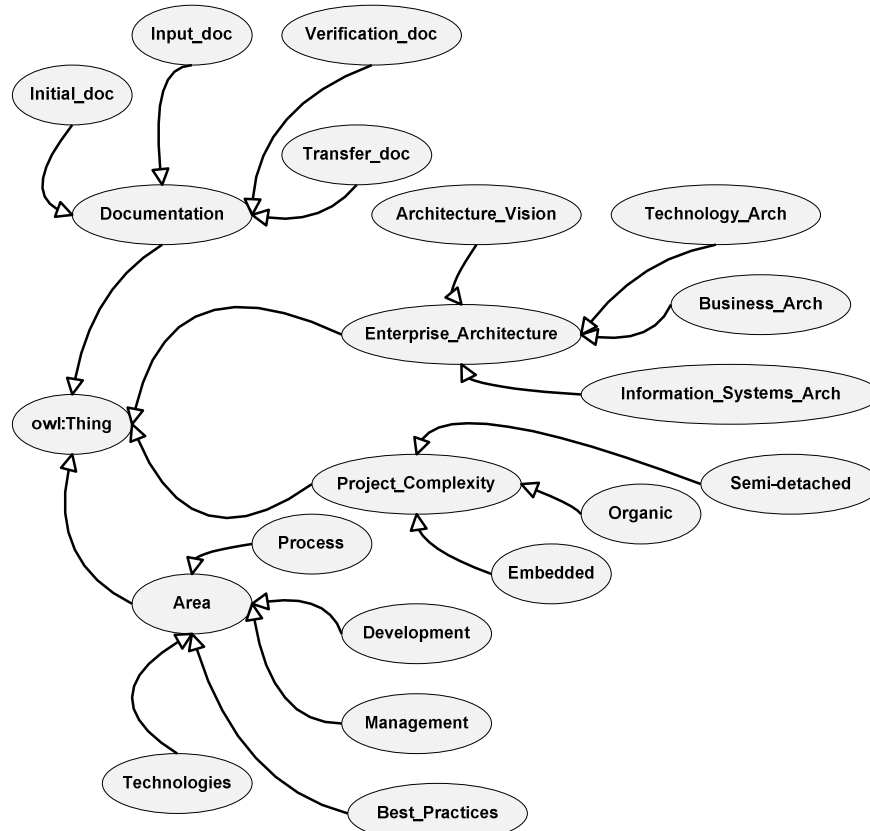


Fig.3. Taxonomy of ontology terms.

The class 'owl: Thing' is the superclass characteristic for ontologies created in the OWL language. The arcs define the class-subclass relationship and the arrows point to superclass. They link certain variables to specific terms related to the range of values of these variables. Since the three main variables are located on the same level of hierarchy with a parameter which depends on those variables, a new relationship has been created and named 'is_a_variable'. The relationship is pointed from the terms 'Area', 'Documentation' and 'Enterprise_Architecture' to the class 'Project_Complexity'. Semantics, which indicates input and output terms for the project complexity assessment model, is a key factor as variable values defined by those terms are used in the main inference modules, i.e. in the expert system, the artificial neuron network and also in the pre-processing where data may be directed to the tool of statistics analysis.

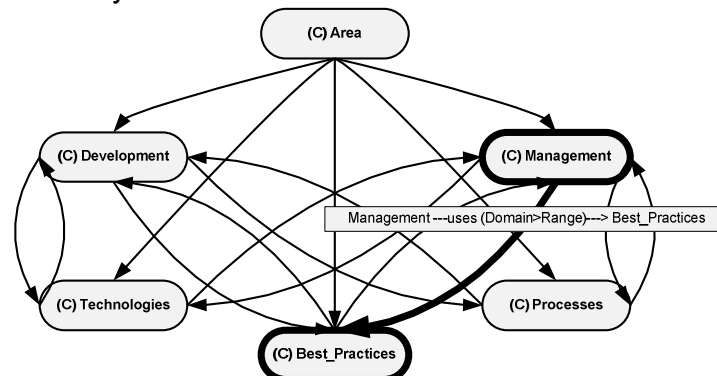


Fig.4. Semantics of the development and management areas variable.

Semantics has additionally been used for both variables that describe 'Development' and 'Management' areas. Given that the 'Development' and

'Management' areas are divided into more detailed subareas, i.e. 'Technologies', 'Best_Practices' and 'Processes', those three terms have been defined, once only, as equal to the 'Development' and 'Management' of the 'Area' variable subclass. Then the property 'is_used_in' has been created which links the three subareas to each of the main areas. Moreover, it was decided to form the inverse property called 'uses'. The semantic relations are shown in Figure 4.

4.1.2 Variable values and restrictions

The research on project complexity as an environment for the verification of ontology use was preceded by modelling a technical system for the measurement and prediction of BTEX substance concentration in the atmosphere. The chemical and meteorological data entered into the system were constant values of real numbers falling into a certain range which resulted from certain characteristics of the measured quantity (e.g. percentage values of concentration could only take on values between 0 and 100), or from long-term observation (e.g. air temperature never reached -40°C). In view of the project complexity variables, the character of measured values should also be taken into consideration.

Each of the variables considered during project complexity assessment—including the hard value which describes this complexity—should be measured on a percentage scale. Their values should depend on the fulfilment level of the requirements. The set of values would be established by real numbers [between 0 - 100]. Perhaps for particular variables the lower limit of permitted values would be greater than 0, e.g. 5%, as—for example—it is rare for development technologies in an IT project to be assessed at merely 0%. The number values of variables would form the input for the fuzzy expert system where they would undergo fuzzification, whereas the set of function values describing project complexity would be applied while creating the hard result from the defuzzification operation of this system.

Since the percentage degree of the completion of a requirement is mentioned, the question of an assessment model should arise: a model which would enable a number value to be assigned to a given fuzzy state of a variable. At the present stage of research, no such model has yet been created; thus, another simplification of system functioning has been made where each of the variables takes a binary value: 0 or 1, to denote parameter completion.

In the course of the research, certain correlations between variables have been noticed. For instance, it is assumed that using particular management procedures influences project documentation. Moreover, the progress in project completion designated by passing from one phase of enterprise architecture development to another should also trigger an increase in the subsequent classes of documentation. These relationships between variables may cause difficulties in reaching a detailed model. Yet a number of data need to be presented in order to assess the impact of this phenomenon.

4.1.3 Formal notation of ontology created in the OWL language

As mentioned earlier, ontology is created in the OWL language with the Protégé tool. It appears to be a standard tool for creating ontology, along with XML notation.

For a given field, the definition of a term in a form of class looks as follows:

```
<rdf:Description rdf:about="#Project_Complexity">  
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>  
</rdf:Description>
```

If a defined class has a superclass, the OWL/XML notation looks as follows:

```
<rdf:Description rdf:about="#Semi-detached">  
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>  
  <rdfs:subClassOf rdf:resource="#Project_Complexity"/>  
</rdf:Description>
```

Besides classes, ontology may include functions (relationships) also called nodes. The definition of the relationship between 'uses' and its inverse 'is_used_in' is specified below:

```
<rdf:Description rdf:about="#uses">  
  <rdfs:domain rdf:nodeID="A15"/>  
  <rdfs:range rdf:nodeID="A0"/>  
  <owl:inverseOf rdf:resource="#is_used_in"/>  
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>  
</rdf:Description>
```

Ontology created in the OWL language is based on triple expressions which consist of an object, property and value. In the ontology described in this article 'Project_Complexity' is defined as the object, 'rdf:type' is defined as the property and 'owl#Class' as the value. This structure allows for inference mechanisms additionally limited by a chosen OWL sublanguage, i.e. OWL Full, OWL DL or OWL Lite.

4.2 AGENT SYSTEM DEVELOPMENT ON THE LEVEL OF ENTERPRISE ARCHITECTURE

The main objective of the agent system model described in this article is to determine the method and technology of the completion of an IT project of a particular degree of complexity with a given degree of organization maturity and customer maturity. It must be noted that the three input parameters take on the nature of qualitative data, which means that loading these data into the assessment system in a linguistic form is impossible. The data should first undergo quantification, i.e. a process of transformation into numeric data.

What is more, it should be mentioned that the current stage of research does not ensure enough data for the two last input parameters, i.e. organization maturity and customer maturity. These data are successively being gathered; however, they cannot be discussed here in detail due to their incompleteness (for the time being). Therefore, the first variable, project complexity, shall be discussed.

On the basis of analyses, it can be stated that the project complexity of enterprise architecture development consists of the following variables: areas, architectures and documents, as shown in Figure 5:

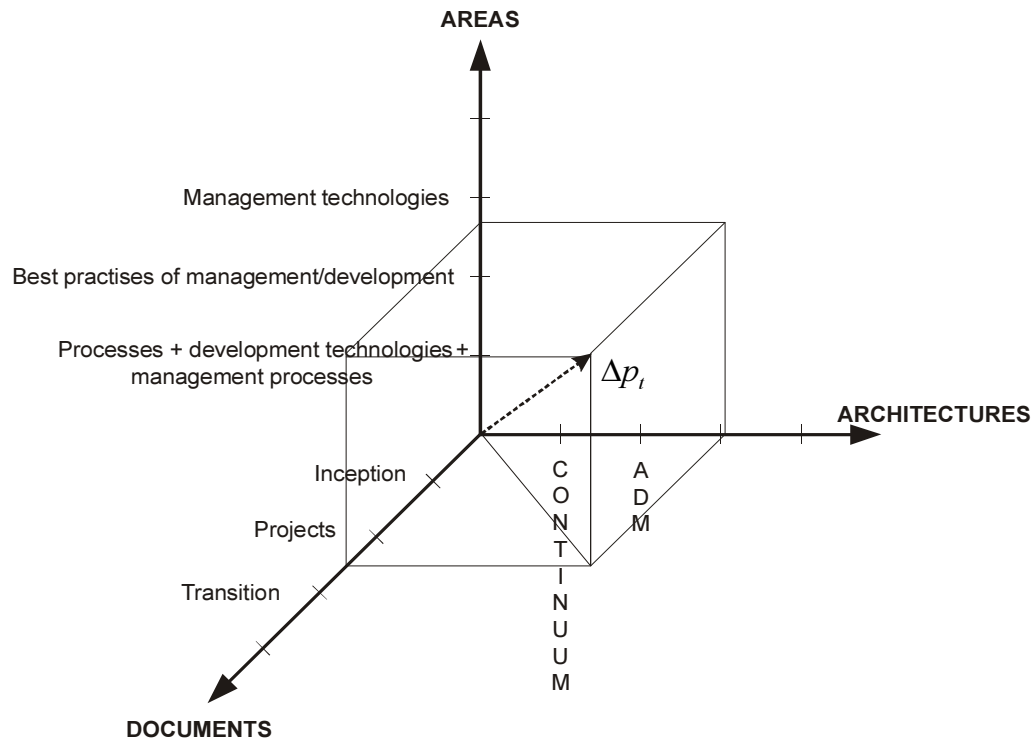


Fig. 5. Arrangement of the proposed parameters which define project complexity.

It should also be indicated that within the 'Architectures' axis, the architectures gradate from 'vision', through 'business and system architecture', to 'technology architecture'. Vision is used for defining the scope of the project and resembles the classic management planning function. Business architecture is the general concept of the IT system that is to be created. At the stage of systems architecture, a logic model of the system is made and technology architecture gives a model which illustrates the physical completion of the architecture.

Another axis presents 'Areas' of enterprise architecture, which influence the process complexity of architecture completion. The areas classify the levels of methods and tools used for management and development processes of the enterprise architecture project.

The third axis shows the status of the presence and increment of documents. The status provides information about the stage of the project and about its management quality.

The above-mentioned parameters presented on each of the three axes show the degree of enterprise architecture development complexity considered as a type of IT project. Therefore, it should be assumed that in the process of defining those parameters, it is possible to find whether the project has a high development risk. Hence, defining the degree of project complexity is a primary variable for the agent system, which allows the selection of appropriate technologies.

The model of parameters which define project complexity indicates the fact that the risk of project complexity and therefore, its failure is determined by the distance from point 0 of the coordinate system: the bigger the distance, the smaller the risk. In accordance with what has been mentioned earlier, the authors of this model have gathered detailed data connected only with project complexity, and these data form the basis for analysis in the agent system. The remaining values of input variables (customer maturity and organization maturity) are still being collected. After having

completed the data it will be possible to implement an assessment model with an intelligent agent environment. The mechanism of this model is shown below.

4.3 FORMAL DESCRIPTION OF THE AGENT SYSTEM AND INPUT DATA

Before the description of the operation of the agent system is formalized, its main constituents should be presented. According to what was described at the beginning of this article, the basic functionalities of the system are represented by the main types of agents. Moreover, it should be mentioned that the original assessment model developed in the Visual Basic for Applications (VBA) environment did not provide agents with the proper level of intelligence and autonomy. The following description proposes a method of providing agents with an apposite level of intelligence through ascribing each of the agent's individual knowledge bases so that the agent is able to undertake a given set of operations independently.

4.3.1 Mediator agent

The task of the mediator agent is to acquire information from the customer and identify the nature of the query (whether the customer needs to find the information—the search mode; or the customer expects decision support through the proper analysis of input data—calculation mode). The mediator agent first has to recognize the customer's expectation regarding the system, and then transfer this information to the centre, i.e. to the management agent. Thus, the following examples of rules for the mediator agent:

IF "query field" is not null THEN "query"=1 and "agent status"=1 (1—action, 0—standby)
IF "query field" is not null and "agent status"=1 THEN "operation"=1 (1—download and upload data) and "operation"=2 (2—check nature of query)
IF "nature of query"=2 THEN "send message about calculation"
IF "operation"=1 THEN "send message to instruct management agent"

4.3.2 Management agent

The management agent makes decisions concerning proper use of resources, i.e. calling one of elements of the system (the neuron network, the expert system) and sending messages about data acquired by one of the agents.

Example of rules for the management agent:

IF "nature of query"=1 THEN "activate search agent"
IF "nature of query"=2 THEN "check data types"
IF "data type"=1 THEN "activate expert system" and "send search agent to expert system"
IF "data type"=2 THEN "activate neuron network" and "send search agent for results"
IF "data type"=3 THEN "use statistical model"
Data types: 1—hard with fuzzification indication, 2—large amount of data with no classic format, 3—repeatable

These rules shall be used for further implementation of the assessment model with the environment typical for agent solutions. It needs to be noticed that some rules are in the form of messages sent within the system. Hence, for these functionalities of the agents, the IT assessment model should regard project complexity as its input variable. An example of a defining level of a variable is shown below, considering the above-specified parameters of description, i.e. areas, documents and architectures.

4.3.3 Example of formal notation for the Areas parameter

For the aggregation of input levels of complexity, a vectorial-matrix description of the project has been proposed. Project complexity has been described by means of

constituents which represent development and management technologies, development and management processes and best practices of the two areas.

$$\Delta p_t = \begin{bmatrix} \Delta p_{t_1} \\ \Delta m_{t_1} \\ \Delta d_{p_t} \\ \Delta m_{p_t} \\ \Delta b_{pp_t} \\ \Delta b_{pz_t} \end{bmatrix} \quad (1)$$

wherein:

Δd_{t_1} —the initial assessment variable for development technology knowledge, $\Delta d_{t_1} \in J_{[5,100]}$

Δm_{t_1} —the initial assessment variable for management technology knowledge, $\Delta m_{t_1} \in J_{[5,100]}$

Δd_{p_t} —the initial assessment variable for development processes, $\Delta d_{p_t} \in J_{[5,100]}$

Δm_{p_t} —the initial assessment variable for management processes, $\Delta m_{p_t} \in J_{[5,100]}$

Δb_{dp_t} —the initial assessment variable for best development practices, $\Delta b_{dp_t} \in J_{[5,100]}$

Δb_{mp_t} —the initial assessment variable for best management practices, $\Delta b_{mp_t} \in J_{[5,100]}$

A similar rule can be applied to formalize the remaining two parameters of the description: architectures and documents. Formalization of project complexity is the first step leading to the full assessment of project management technology. Customer maturity and organization maturity shall be formalized in the same way.

The need for optimal choice of the assessment model must also be stated. Currently, a solution which could use fuzzy modelling appears to be the most efficient one. It has also been planned to extend the module responsible for the statistical analysis as well as modelling using data verification. As a result, the developed system—as a whole—should be aimed at achieving high-quality results.

5 CONCLUSIONS

The above description, connected with the verification of the model, formalization of knowledge-based ontologies and specification of the system agents, indicates that the system is able to support decisions connected with the methodology of IT project management.

Further works on ontology creation should be conducted parallel to developing the problem of measuring variables which are used for project complexity assessment. Specifically, the problem of describing the mode of giving a percentage rank to degrees of requirement satisfaction, and the analysis of variable relationships and their potential consequences should be addressed. Both the ideas of expanding the application of ontologies in the system for data pre-processing or of separating a functional module responsible for building confidence in data should also be analyzed, with ontology as a constituent.

As far as agents are concerned, it should be stressed that they preserve their functionalities in the case of the input data connected with enterprise architecture development and they are prepared to operate on these data. It was observed that the agent structure accepted earlier and verified for the technical system data (where complexity and data structure is known from the beginning) can be applied for the data taken from a social-technical system.

6 BIBLIOGRAPHY

- Altmann, J. & Gruber, F., 2001. Using mobile agents in real world: A survey and evaluation of agent platforms. In *Proceedings of the Second International Workshop on "Infrastructure for MAS and Scalable MAS"*. Montreal.
- Angryk, R. et al., 2002. Travel Support System: An Agent Based Fram work. In *Proceeding of the International Conference on Internet Computing*. Las Vegas: CSREA Press.
- Galant, V. & Tubrycy, J., 2001. *Inteligentny agent programowy.*, Wrocław: Prace Naukowe AE.
- Gruber, T., 1993. A translation approach to portable ontology specifications. *Knowledge acquisition*, pp.199–220.
- Jennings, N., 2001. An agent-based approach for building complex software systems. *Communications of the ACM*, pp.35–41.
- Noy, F.N. & McGuinness, D.L., 2001. *Ontology Development 101: A Guide to Creating Your First Ontology*. Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880, Stanford.
- Orłowski, C. & Ziółkowski, A., 2008. Evolution of the Concept of an Agent System for Information Technology Evaluation. In A. Grzech et al., eds. *Information Systems Architecture and Technology. Information Systems and Computer Communication Networks*. Wrocław: Wrocław University of Technology, pp. 25–36.

For citation:

Please use the reference above

Link to the post-print file:

<http://www.tandfonline.com/doi/abs/10.1080/01969720903408805>