Vadim Ermolayev, Heinrich C. Mayr, Mykola Nikitchenko,
Aleksander Spivakovsky, Mikhail Zavileysky
and Grygoriy Zholtkevych (Eds.)

ICT in Education, Research and Industrial Applications: Integration, Harmonization and Knowledge Transfer

Proceedings of the 7-th International Conference ICTERI 2011

Kherson, Ukraine
May, 2011

Preface

It is our pleasure to bring you the volume of the selected papers of the seventh edition of the International Conference on Information and Communication Technologies (ICT) in Education, Research, and Industrial Applications: Integration, Harmonization, and Knowledge Transfer (ICTERI 2011) held at Kherson, Ukraine on May 5-7, 2011.

ICTERI 2011 is concerned with several interrelated topics that proved to be vibrant for the research and industrial communities judging by the number of submissions we have received this year:

- ICT infrastructures, integration and interoperability
- Machine Intelligence, knowledge engineering (KE), and knowledge management (KM) for ICT
- Cooperation between academia and industry in ICT
- Methodological and didactical aspects of teaching ICT and using ICT in education

This year we gave a particular encouragement to the submissions that facilitate bringing together academics and practitioners interested in the technological advances and business applications of Information and Communication Technologies and Infrastructures. Therefore, the call was primarily targeted to real world applications; highlighting the benefits of research results or experience for industry and services, in addition to academic world. Papers describing advanced prototypes, systems, tools and techniques and general survey papers indicating future directions were also encouraged.

We were lucky to attract quite a substantial number of submissions – a total of 128 – which broadly fell down into the four genres: full research papers (39), short research papers (61), discussion or problem analysis papers (15), and the papers on industrial experience or case study (13) – evenly covering the themes of the conference scope. Out of those we have selected 11 top quality and most interesting papers to be published in our proceedings. The acceptance rate was therefore 8.87 percent. In addition to those selected publications we included the papers of our invited speakers.

The invited paper by our keynote speaker Prof. Costin Bădică on dynamic negotiations in multi-agent systems for disaster management opens the proceedings. It is followed by the abstracts of the invited talks by Prof. Heinrich C. Mayr on integrated university information systems and information system strategy, Prof. Alexander Letichevsky on insertion modeling and its applications, and Prof. Abdel-Badeeh M. Salem on intelligent technologies and methodologies for medical knowledge engineering.

The selected contributions cover the issues of:

- **Foundations, Intelligence and Methodologies**. This theme is represented by full research papers on: a gnoseology-based approach to foundations of informatics; pre-automata as mathematical models of event flow recognizers; insertion modeling system and constraint programming; simulation of expanded iterated prisoner's dilemma, and a discussion paper entitled “Is Your Ontology a Burden or

a Gem? – Towards Xtreme Ontology Engineering" dealing with the methodological challenges in knowledge engineering for industries.

- **ICT infrastructures, integration and interoperability**. This theme is covered by a full research paper on design and implementation of a quality management system for electronic training information resources and short papers dealing with: optimization criteria for task assignment in cluster and wide-area computing; a lightweight approach to contact data synchronization in mobile social networks; and conceptual design and technology choices for building a virtual laboratory of distance learning.
- **Methodological and didactical aspects**. This theme is introduced by the case study report on the choices of topics and didactical support in a Software Engineering class and a short research paper on the influence of the labor market on the formation of competencies of the future IT specialists.

The conference would not have been possible without the support of many people. First of all we would like to thank the members of our Program Committee for providing timely and thorough reviews, and also for being cooperative in doing additional review work. We are also very grateful to all the authors who submitted papers to ICTERI and thus demonstrated their interest in the research problems within our scope. We would like also to thank the local organizers of the conference whose devotion and efficiency made ICTERI a very comfortable and effective scientific forum.

May, 2011

Vadim Ermolayev
Heinrich C. Mayr
Mykola Nikitchenko
Aleksander Spivakovsky
Mikhail Zavileysky
Grygoriy Zholtkevych

Organization

Program Co-chairs

Vadim Ermolayev, *Zaporozhye National University, Ukraine*
Heinrich C. Mayr, *Alpen-Adria-Universität Klagenfurt, Klagenfurt, Austria*
Mykola Nikitchenko, Taras *Shevchenko National University of Kyiv, Ukraine*
Aleksander Spivakovsky, *Kherson State University, Ukraine*
Mikhail Zavileysky, *DataArt Solutions Inc., Russian Federation*
Grygoriy Zholtkevych, *V.N.Karazin Kharkiv National University, Ukraine*

Program Committee

Dmytro Bui, Taras *Shevchenko National University of Kyiv, Ukraine*
Andrey Bulat, *Kherson State University, Ukraine*
Jose Manuel, Gomez Perez, *Intelligent Software Components (iSOCO) S.A., Spain*
Viktor Gorokh, *Kharkiv National Pedagogical University, Ukraine*
Giancarlo Guizzardi, *Federal University of Espirito Santo (UFES), Brazil*
Natalya Keberle, *Zaporozhye National University, Ukraine*
Valery Kravets, *National Technical University "Kharkiv Polytechnic Institute", Ukraine*
Gennadi Kravtsov, *Kherson State University, Ukraine*
Vladislav Kruglik, *Kherson State University, Ukraine*
Natalya Kushnir, *Kherson State University, Ukraine*
Valentin Lazurik, *V.N.Karazin Kharkiv National University, Ukraine*
Alexander Letichevsky, *V.M. Glushkov Institute of Cybernetics, Ukraine*
Mikhail Lvov, *Kherson State University, Ukraine*
Natalia Morse, *National University of Life and Environmental Sciences, Ukraine*
Natalia Osipova, *Kherson State University, Ukraine*
Vladimir Peschanenko, *Kherson State University, Ukraine*
Sergey Rakov, *Kharkiv National Pedagogical University, Ukraine*
Abdel-Badeeh Salem, *Ain Shams University Abbasia, Egypt*
Ludmila Shishko, *Kherson State University, Ukraine*
Oleksandr Sokolov, *National Aerospace University, Ukraine*
Vagan Terziyan, *Juvaskyla University, Finland*
Nikolay Tkachuk, *National Technical University "Kharkiv Polytechnic Institute", Ukraine*
Yuri Trius, *Cherkasy State Technological University, Ukraine*
Helmut Veith, *Vienna University of Technology, Austria*
Maryna Vladimirova, *V.N.Karazin Kharkiv National University, Ukraine*
Paul Warren, *British Telecom, UK*
Hannes Werthner, *Vienna University of Technology, Austria*
Irina Zaretskaya, *V.N.Karazin Kharkiv National University, Ukraine*

Additional Reviewers

Maxim Davidovsky, *Zaporozhye National University, Ukraine*
Julia Neidhardt, *Vienna University of Technology, Austria*
Vladimir Shekhovtsov, *National Technical University "Kharkiv Polytechnic Institute", Ukraine*

Table of Contents

Invited Papers and Talks

Dynamic Negotiations in Multi-Agent Systems

Costin Bădică, Mihnea Scafeş, Sorin Ilie, Amelia Bădică, and Alex Muscar

University of Craiova, Bvd.Decebal 107, 200440, Craiova, Romania
{cbadica,mscafes,silie,amuscar}@software.ro, ameliabd@yahoo.com

Abstract. Collaboration workflows for human experts involved in distributed problem solving and acting in dynamic environments require advanced self-configuration mechanisms for optimal selection of service providers. Such dynamic environments, specific for example to problems like environmental management, disaster and crisis management, and high-risk project management, are characterized by continuously changing situations, occurrence of unexpected events, as well as high variability in resources' availability. We propose a solution for self-configuration based on a framework for the development of flexible cooperative multi-issue one-to-many service negotiations. The framework allows two levels of dynamic configuration of negotiations. First, the negotiation protocol can by dynamically selected depending on the profiles of available service providers. Second, negotiation subject as well as the parameters of the negotiation protocol, can be dynamically set depending on the current requirements of the service requester. In this paper we illustrate the features of our framework for dynamic negotiations using an example of service contracting in the field of disaster management. We also provide initial experimental results concerning the effect of the dynamic selection of negotiation protocol onto the quality of the negotiation outcome as well as onto the communication complexity of interactions incurred during negotiations.

Keywords: multi-agent system, service negotiation, contract net

1 Introduction

The increasing complexity of real-world problems demands special support for distributed collaborative problem solving. Multi-agent systems (MAS) are a special class of distributed systems that combine interaction and coordination with distribution of computation to improve performance of problem solving processes. MAS were successfully applied for solving problems that require distributed reasoning, decentralization and collaboration [19]. An example is a collaboration system for helping human experts and population to deal with disasters (see the FP7 DIADEM project[1] that targets crisis management in the context of chemical incidents in industrial and urban areas).

A workflow is represented as a structured set of coordinated tasks and information exchanges defined to carry out a well-defined business process in (possibly multiple

[1] DIADEM Distributed information acquisition and decision making for environmental management: http://www.ist-diadem.eu/.

networked) organizations [18]. Collaborative problem solving in heterogenous, unpredictable, and dynamic environments can be achieved by providing an increased flexibility in workflow formation, that goes far beyond the classic approach of static and centralized workflow definitions. Examples based on empirical analysis of real experiences can be given in the area of organizational adaptation to disasters [9]. Such infrastructure support can be provided by service-oriented MAS, like for example the Dynamic Process Integration Framework (DPIF) [13]. In such frameworks, communication links between local processes in agents are facilitated using service discovery: whenever an agent supplying some service (we will call this agent the *manager*) in a workflow requires data provided by some other service (we will call this agent the *contractor*), a communication link must be established between them. A similar problem occurs in optimal resource allocation for workflow scheduling [10, 6], as well as in dynamic role binding in MAS organizations [11].

The self-configuration provided by service discovery and matching can be improved by enhancing it with a finer level of control based on one-to-many service negotiation ([5], [12], [14], [20]). In our work we proposed a novel negotiation model that follows the conceptual framework of one-to-many service negotiation [2]. This model allows the dynamic configuration of negotiation protocol and negotiation parameters based on the requirements of each negotiation instance. More specifically, the model provides two levels of dynamic configuration of negotiations. First, the negotiation protocol can by dynamically selected by the manager depending on the profiles of available service providers. Second, negotiation subject as well as the parameters of the negotiation protocol, can be dynamically set depending on the current requirements of the service requester. Both features are considered in this paper.

In this paper we illustrate the features of our framework for dynamic negotiations using an example of service contracting in the field of disaster management. We also provide initial experimental results concerning the effect of the dynamic selection of negotiation protocol onto the quality of the negotiation outcome as well as onto the communication complexity of interactions incurred during negotiations.

The paper is organized as follows. We start in Section 2 by introducing the components of our framework. In Section 3 we introduce a sample example concerning a typical activity in a situation assessment problem encountered in chemical incident management. In Section 4 we provide an initial experimental analysis of the effect of negotiation protocol selection onto the negotiation outcome as well as onto the communication complexity of negotiation interactions. In Section 5 we cover related works in the area of MAS for business process management and negotiation for service contracting and resource allocation in flexible workflows. In Section 6 we present our conclusions and point to future works.

2 Background

Negotiation is a process that describes the interaction between one or more participants that must agree on a subject by exchanging proposals about this subject [8]. Negotiation about a service that one or more participants agree to provide to other participants is called *service negotiation*. We have developed a conceptual framework for service ne-

gotiation that addresses protocols, subjects and decision components. Our framework supports generic one-to-many negotiations and it defines two roles: manager and contractor [17, 12]. The manager is the agent that requests a service and thus initiates the negotiation. The contractor is the agent that is able to provide the service requested by the manager. For a more complete review of the conceptual framework, please see [2]. A brief description of the design and implementation is given in [16].

Currently we have configured our framework with three negotiation protocols that we have found useful in disaster and environment management problems. These protocols are:

1. *Direct Task Assignment*. Using this protocol there is actually no negotiation happening between the manager and the contractors. The manager simply picks up one or more contractors and assigns them the task[2]. Although simple, *Direct Task Assignment – DTA* negotiation protocol is actually very useful to model subordination relationships, that are common for task assignments to team members that operate in a disaster environment. For example in a disaster management information system, an *Incident Commander* can dispatch tasks to team members operating in the area affected by the disaster.
2. *Contract Net*. Proposed by [17], *Contract Net – CNET*[3] is probably one of the most influential negotiation protocols utilized in distributed collaborative problem solving, with many extensions currently available (see for example [20]). *CNET* is essentially a one step protocol: the manager announces a task to the contractors in the announcement stage, each contractor proposes or refuses to submit a bid in the bidding stage, and finally the manager decides to award the task to at least one of the contractors in the awarding phase. It can be easily noticed that the *DTA* protocol is in fact a simplified *CNET* without the announcement and the bidding stages.
3. *Iterated Contract Net*. *CNET* can be extended to multiple steps thus obtaining *Iterated Contract Net – ICNET*[4]. This protocol is very useful in situations when a single step is not sufficient for the manager to select a contractor for awarding the task. This may happen for example if the requirements set for the task announced by the manager are too restrictive and thus neither of the contractors is able to meet them in a satisfactory way. Another situation is when contractors are operating in a constrained environment that does not allows them to bid. For example, considering a disaster management scenario, one of the stakeholders (for example a chemical expert) might be caught in an important meeting that forbids him or her to answer to incoming phone calls. However, repeated calls, probably originating from repeated task announcements in a negotiation caused by the occurrence of a very important event, can enable him to pickup his phone and answer, i.e. to bid.

A set of generic negotiation steps are defined: (i) negotiation subject identification and negotiation announcement (initiation of negotiation), (ii) bidding, i.e. making pro-

[2] *DTA* has similarities with *FIPA Request* interaction protocol, see `http://www.fipa.org/specs/fipa00026/`, excepting that the contractor cannot refuse the task.

[3] *CNET* is standardized by Foundation for Intelligent Physical Agents, see `http://www.fipa.org/specs/fipa00029/`.

[4] *ICNET* is standardized by Foundation for Intelligent Physical Agents, see `http://www.fipa.org/specs/fipa00030/`

posals and counter-proposals, (iii) deciding whether an agreement or a conflict was reached, and (iv) termination.

Negotiation subject comprises the service description and a subset of the service parameters that are important decision factors during negotiation (i.e. their current values are taken into consideration when selecting the appropriate service providers). During negotiation, these parameters are considered *negotiation issues*.

Negotiation issues are described by name, data type, and monotonicity. The name uniquely identifies the issue in a negotiation subject. The data type describes the type of the value that the issue is allowed to take (e.g. number, string, geographical location, date/time, etc.). The monotonicity specifies whether the manager prefers higher values to lower values of this issue: (i) INCREASING if the agent prefers high utility values of the issue and (ii) DECREASING if the agent prefers low utility values of this issue.

Service negotiations in disaster management are *cooperative*. Cooperativity stems from the fact that the overall goal of participants is the optimization of the response for situation assessment. Negotiations for a certain service provision are carried out only with agents that can provide the required service (i.e. that possess the domain knowledge or physical capabilities that are needed to provide the service). Provider agents will usually accept to offer their services if they are currently able to do so (i.e. if they posses all the necessary resources). During a negotiation: (i) the manager is the leading decision factor seeking to optimize the assignment of the negotiation task to the contractor(s); (ii) the contractor(s) make their best proposals for the manager, taking into account their current duties and availability, thus preserving their autonomy according to the principles of professional bureaucracy [1].

Service parameters can be classified into 4 classes:

(i) DYNAMIC that specifies that the issue value is not fixed by the manager, i.e. the contractor can propose different values for the issue;
(ii) FIXED that specifies that the issue value is fixed by the manager, i.e. if the contractor proposes a different value for the issue then the corresponding local utility of the issue is zero;
(iii) CONDITION that specifies that the issue value is fixed by the manager, but if the contractor proposes a different value for the issue then the total utility of the proposal is zero; normally a contractor that cannot meet the issue value requested by the manager should decide to not bid because the utility of her bid will be zero;
(iv) TRIVIAL that means that the issue is not taken into account in the computation of the bid utility, although it can be set in the request of the manager and consequently, it can be taken into account in the negotiation by the contractor and help her to make a more informed decision if and what to bid.

Negotiation participants playing either the manager or contractor roles use utility functions to quantify their preferences over proposals. In our framework the manager uses a weighted additive utility function to evaluate proposals and to select the service provider (we omit the discussion of contractor utility functions, as they are not relevant for this paper; for details on contractors' utilities please consult reference [2]). Each negotiation issue i has a weight $w_i \in [0, 1]$ and a partial utility function f_i. Note that weights are normalized i.e. $\sum_i w_i = 1$. Intuitively, for the manager the weight of an

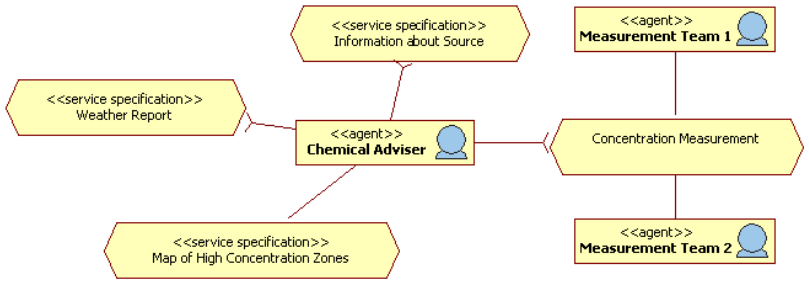


Fig. 1. AML agent and services diagram for the sample negotiation scenario.

issue represents the relative importance of that issue in the set of all issues associated to a negotiation subject.

The partial utility of an issue i maps the issue domain D_i to a value in the interval $[0, 1]$, i.e. $f_i : D_i \rightarrow [0, 1]$. The function f_i depends on the domain of the issue. For example, a possibility to define the partial utility function of a real valued issue with $D_i = [x_{min}, x_{max}]$ is as follows:

$$f_i(x) = \frac{|x - x^*|}{|x_{max} - x_{min}|}$$

Here, x^* is the reference value assigned by the manager to the issue i (that represents the optimal value from the manager point of view) and $|x - y|$ is the distance between x and y (note that the distance depends on the data type of the negotiation issue). Note that a negotiation issue for which the partial utility is defined as a distance from the reference value has always a DECREASING monotonicity. Let I be the set of negotiation issues partitioned into sets $I^{\uparrow}$ and $I^{\downarrow}$ of issues with INCREASING and DECREASING monotonicity. The utility function of a proposal $x = (x_i)_{i \in I}$ is computed as follows:

$$u_m(x) = \sum_{i \in I^{\uparrow}} w_i * f_i(x_i) + \sum_{i \in I^{\downarrow}} w_i * (1 - f_i(x_i))$$

3 Sample Scenario

We illustrate our approach by using an example derived from a real world use case investigated in the FP7 DIADEM project. For the sake of clarity but without the loss of generality we assume a significantly simplified scenario that is illustrated in Figure 1 utilizing the Agent Modeling Notation – AML [4]. In a chemical incident at a refinery a chemical starts leaking and forms a toxic plume spreading over a populated area. The impact of the resulting fumes is assessed through a service composition involving collaboration of human experts. During this incident health complaints are reported. Consequently, the *Incident Commander* dispatches a chemical expert that holds expertise in estimating the gas concentration in the affected area. This expert is denoted as

Table 1. Negotiable issues and manager request for *Concentration Management* service

Issue	Location	Quality	Deadline
Reference value	*loc*	100	11:47 AM
Weight	1	2	3
Data type	REGION	INTEGER	TIME
Boundary	n/a	100	100
Negotiable	FIXED	DYNAMIC	DYNAMIC

Table 2. Contractors' bids.

Issue	Location	Quality	Deadline
Bid Value (1)	*loc*	70	11:58 AM
Bid Value (2)	*loc*	100	12:12 PM

Chemical Adviser and, among other things, she guides fire fighter *Measurement Team* agents which can measure gas concentrations at specific locations in order to provide feedback for a more accurate estimation of the critical area. This interaction between *Chemical Adviser* and *Measurement Team* agents involves negotiation to determine the optimal providers of appropriate measurements (for more details see [3]).

We can observe from Figure 1 that *Chemical Adviser* provides the service *Map of High Concentration Zones* and this provisioning requires the contracting of *Concentration Measurement*, *Weather Report*, as well as *Information about Source* services. In this scenario *Chemical Adviser* agent plays the negotiation role of manager looking for a provider for the service *Concentration Measurement*.

The optimal selection of the service provider takes into account: the location where the measurement must be performed, the quality of the measurement, and the duration for performing the measurement. Additionally we assume that the measurement quality is given as a percentage and that the maximum time frame for performing the measurement is 100 minutes. The description of the negotiation issue, together with the manager proposal are given in Table 1. Weights of negotiation issues were normalized as follows:

$$w_{Location} = 1/6, \;\; w_{Quality} = 2/6, \;\; w_{Deadline} = 3/6$$

Let us assume that there are two *Measurement Team* agents in the system and each of them decides to bid with an offer for providing the service *Concentration Measurement*. Their bids are shown in Table 2.

The utility of the bid of *Measurement Team 1* is computed as follows:

$$\begin{aligned} u_{Location_1} &= 1/6 \times (1 - 0/1) &&= 0.166 \\ u_{Quality_1} &= 2/6 \times (1 - 30/100) &&= 0.233 \\ u_{Deadline_1} &= 3/6 \times (1 - 11/100) &&= 0.445 \end{aligned}$$

$$u_{MT_1} = u_{Location_1} + u_{Quality_1} + u_{Deadline_1} = 0.844$$

The utility of the bid of *Measurement Team 2* is computed as follows:

$$\begin{aligned} u_{Location_2} &= 1/6 \times (1 - 0/1) &&= 0.166 \\ u_{Quality_2} &= 2/6 \times (1 - 0/100) &&= 0.333 \\ u_{Deadline_2} &= 3/6 \times (1 - 25/100) &&= 0.375 \\ u_{MT_2} &= u_{Location_2} + u_{Quality_2} + u_{Deadline_2} &&= 0.874 \end{aligned}$$

Chemical Adviser agent uses these equations to compute the utilities of each bid received from *Measurement Team* agents. Then *Chemical Adviser* applies a strategy that allows it to either immediately select the winning bid or to decide if to continue the negotiation using a new iteration. Let us assume that *Chemical Adviser* applies a strategy that considers acceptable only those bids that pass a given threshold. If none is above the threshold then *Chemical Adviser* can perform a second iteration either by relaxing the conditions of the call for proposals (for example by decreasing the required quality of the measurements or by extending the deadline for performing the measurements) or by decrementing the threshold, thus giving a chance to the *Measurement Team* agents to update their bids. If at least one bid is considered acceptable then *Chemical Adviser* can decide to accept one or more *Measurement Team* agents to contract the *Concentration Measurement* service. Assuming a threshold of 0.85, according to this algorithm *Chemical Adviser* will select *Measurement Team 2* after the first iteration.

4 Experimental Results

In this section we discuss the results of our initial experiments with service negotiations utilizing the different negotiation protocols currently supported by our framework: *DTA*, *CNET*, and *ICNET*. We first propose a model for the experimental analysis of one-to-many negotiations and then we provide experimental results that cover negotiation outcome and communication complexity of negotiation interactions.

4.1 Modeling Assumptions

When faced with a service negotiation problem, an agent playing the manager role will have to decide what negotiation protocol to use. In our case, the manager may decide to use one of the *DTA*, *CNET* or *ICNET* negotiation protocols that were discussed in Section 2. This choice will affect important factors like quality of the negotiation outcome as well as communication complexity incurred during the negotiation interaction [15] that overall impact the efficacy of the collaboration process for resolving the incident.

In what follows we propose a simple experimental setting aimed at analyzing the impact of the strategy employed by the manager agent to choose a service negotiation protocol on the quality of the negotiation outcome as well as on the communication complexity of negotiation interactions. The experiment is focused on evaluating a single negotiation rather than a complex collaboration workflow comprising more negotiations (we are aware of this limitation and we plan to address it in the near future).

Let us assume that a manager agent M is negotiating for contracting a service from a contractor agent that is member of a set of n contractors $C_1, \ldots, C_n$. Simplifying things we assume that each contractor C_i can offer a utility value u_i to the manager such that $u_i \in [u_i^{min}, u_i^{max}] \subseteq [0, 1]$. The interval $[u_i^{min}, u_i^{max}]$ is part of the profile of the contractor. For example, if the utility intervals of contractors C_i and C_j have an empty intersection and $u_i^{min} < u_j^{min}$ it means that C_j will always appear as more profitable than C_i for the manager. However, if the intersection of the intervals is nonempty it means that sometimes C_i can also be more profitable than C_j for the manager.

Moreover, each negotiation will always involve requirements that are set by the manager and must be met by the contractor in order to be allowed to bid for a contract. These requirements depend on the negotiation issues and on the constraints on the negotiation issues that are set by the manager in the request for service. For example, considering the *Concentration Measurement* service, *Chemical Adviser* may require *Measurement Team* agents to bid if and only if they are equipped with certain specialized measurement devices like for example *drager tubes*[5]. We assume that for each contractor C_i there is a probability of $c_i \in [0, 1]$ that she will be able to satisfy the requirements set by the manager's request. The probability p_i is also part of contractor's C_i profile.

Contractors are usually critical resources that do not always have the possibility to bid for a service contract. Let us take for example the *Chemical Adviser* that can be caught in an important meeting while health complaints start being reported. In such a situation the *Incident Commander* must assign the task *Map of Critical Zones* to an expert in the chemistry of gases, i.e. the *Chemical Adviser*. Then the *Incident Commander* must adopt an iterative negotiation protocol to allow the *Chemical Adviser* to bid. Let us assume that in our experimental setting each contractor C_i is busy with a probability $b_i \in [0.1]$. Nevertheless, we assume that even when she is busy, if C_i is able to meet the requirements set by the manager then she will find time to bid in the second iteration.

Let us also assume that the manager M will choose to utilize one of the DTA, $CNET$ or $ICNET$ (two iterations version) negotiation protocols with the probabilities $p, q, r \in [0, 1]$ such that $p + q + r = 1$. The triple (p, q, r) is part of the manager's profile. Depending on the utilized protocol the negotiation will incur a certain communication cost estimated as the number of messages exchanged between the manager and the contractor during the negotiation. Moreover the quality of the negotiation outcome will be estimated as the utility perceived by the manager for the contracted service.

If the manager is utilizing the DTA negotiation protocol then she will randomly assign the task to one of the contractors. However, a contractor that does not meet the requirements will not be able to provide the service, so she will have to report failure. In this case the manager will randomly select another contractor and so on, until a suitable contractor is found (we assume this is always the case). Note however that this trial-and-error process performed by the manager affects the outcome of the negotiation by decrementing her perceived utility. More precisely, if the successful contractor C_i that could perform the task was selected in the k-th trial then the utility perceived by the

[5] They are tubes that contain chemicals that react to a specific compound to form a stain or color when passed through the gas.

manager will be $u_i \times (1 - (k-1)/n)$ rather than u_i. Moreover, the communication cost associated to this negotiation interaction consists of $2 * k$ message exchanges.

If the manager is utilizing the $CNET$ negotiation protocol then she will select the contractor C_i that provides her the highest utility u_i from those contractors that met the requirements of the call for proposals and were not busy (i.e. they were able to bid). The communication cost consumed for a busy contractor consists of 2 message exchanges, while for a not busy contractor (it doesn't matter if she could met or not the requirements, according to the $CNET$ negotiation protocol [17] she either proposed or refused to bid) the communication cost consists of 3 message exchanges.

If the manager is utilizing the $ICNET$ protocol we assume that she will always perform two negotiation iterations. We simply the negotiation by assuming that contractors do not change their between iterations. This assumption is not as restrictive as it might look, because some contractors are busy and can bid only in the second iteration. Moreover, we also assume that a busy contractor that meets the requirements of the call for proposals will always find time to bid during the second negotiation iteration. The manager decides the contractor to whom to award the task after the second iteration. The communication cost for a busy contractor consists in 4 message exchanges, while for a non-busy contractor it consists of 5 message exchanges.

4.2 Experimental Results and Discussions

We created a simulation experiment assuming the model introduced in Section 4.1. In the simulation we considered one manager and n contractors. During a simulation, the given manager and contractors' profiles are set and a large number of negotiation instances are run. The manager is characterized by her profile (p, q, r) that defines her strategy for dynamic selection of the negotiation protocol for each negotiation instance. Each contractor C_i is characterized by her profile defined as a triple $([u_i^{min}, u_i^{max}], c_i, b_i)$. The contractors' profiles are given as input to the simulation algorithm. The values of the utilities u_i are randomly selected for each negotiation instance assuming uniform distributions. Moreover, the probabilities c_i and b_i are utilized to determine the status of a contractor (as satisfying or not satisfying the requirements of the manager proposal and busy or not busy) for each negotiation instance.

The goal of the simulation was to observe and analyze the quality of the negotiation outcome as well as the incurred communication cost for different profiles of the manager agent. Therefore we ran the simulation for various manager profiles by sampling the probability space (p, q, r) accordingly. Assuming a sampling rate of $1/m$ (where m is a given natural number) we developed a simulation program that evaluates the average utility perceived by the manager, as well as the average communication cost incurred during negotiations for the following large set of manager profiles $\{(i/m, j/m, 1 - (i + j)/m) | 0 \leq i, j \leq m,\ 0 \leq i + j \leq m\}$.

In our initial experiment we considered a simple negotiation case with 1 manager and 3 contractors. We ran 2000 negotiations and we recorded the negotiation outcome as well as the number of exchanged messages. We took $m = 20$ in the sampling rate of the probabilities that define the manager profile. The contractor profiles were set as follows: $C_1 = ([0.2, 0.3], 0.8, 0.2)$, $C_2 = ([0.5, 0.6], 0.5, 0.5)$, and $C_3 = ([0.8, 0.9], 0.2, 0.8)$. These values show that the manager will always prefer C_2 and C_3 to C_1. Note that

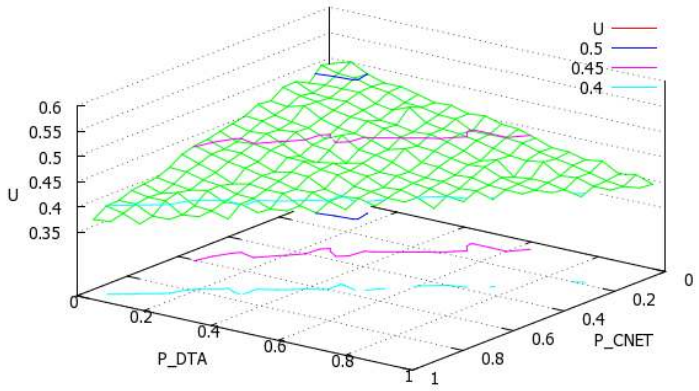


Fig. 2. Manager utility vs. probabilities for selecting DTA and $CNET$ protocols.

failed negotiations (i.e. negotiations that failed to allocate a contractor for providing the service) are possible given this set of contractor profiles, and consequently they were filtered out during our experiments. Note however that their number was very small (approximately 1% from the total number of negotiations), so it was considered not relevant for the results of the experiment.

If the manager is using the DTA protocol then C_1 will perform the task whenever she is able to meet the manager's requirements and (i) she is either selected the first or (ii) she is selected after C_2 and C_3 but both C_2 and C_3 could not meet the manager's requirements and reported failure (note that as C_1 meets the requirements requested by the manager with a higher probability than C_2 and C_3, she will perform quite often the task according to the DTA protocol). The conditions for C_2 or C_3 to perform the task are analogous with the conditions for C_1. Note that the status of being "busy" is not taken into account by the manager when she is playing the DTA negotiation protocol, i.e. if she selects a certain contractor then the task will be assigned to her in any case. Nevertheless, if the assigned contractor cannot finalize the task successfully then she will report failure and consequently the manager will retry the operation of service contracting by assigning the task to another contractor.

If the manager is using the $CNET$ protocol the contractors that are set to "busy" are not taken into account by the manager, as they cannot bid. although they receive the call for proposals from the manager. For example, even if C_3 can offer a high utility to the manager, she has a high probability of being busy, so she will not be able to bid in many $CNET$ negotiations. So in this case C_2 can C_1 can win the negotiation more often. Moreover, C_2 is also busy quite often, giving a chance to award the task to C_1.

Finally, when the manager is using the $ICNET$ protocol, even if a contractor is "busy" and cannot bid in the first negotiation iteration she will still be able to bid in the second iteration. This interaction pattern allows for example to contractor C_3 to bid and consequently to increase the utility perceived by the manager.

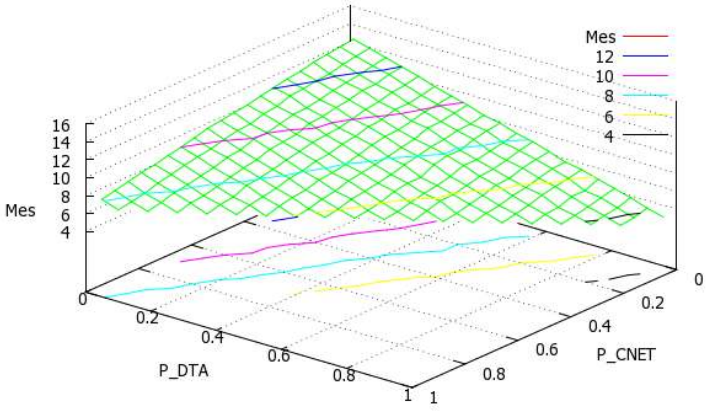


Fig. 3. Number of exchanged messages vs. probabilities for selecting *DTA* and *CNET* protocols.

The experimental results are shown graphically in Figures 2 and 3. These figures present the plots of the average utility perceived by the manager as well as of the average number of exchanged messages per negotiation as functions of the probabilities p and q (shown as P_DTA and P_CNET on the figures) of the manager to choose between the *DTA* and *CNET* negotiation protocols. Note that the dependencies on the probability r for choosing the *ICNET* protocol trivially follow as $p + q + r = 1$.

The first observation is that the highest utility (slightly above 0.5) is perceived by the manager when she always uses *ICNET* negotiation protocol. However, this strategy also brings her the highest overhead in terms of communication complexity, i.e. slightly above 14 messages on average per negotiation. This clearly shows the tradeoff that exists between the optimality of the solution and the communication complexity that occurs in negotiation interactions when the negotiation protocol can be dynamically selected by the initiator of the negotiation.

Secondly, we can observe that the difference in terms of manager's perceived utility between the *DTA* and *CNET* protocols is small (at least in this experiment). For example, if *ICNET* is not utilized, i.e. $r = 0$ or equivalently $p + q = 1$, we can easily observe that the average utility perceived by the manager is slightly variable around 0.4. This can be explained by the fact that what is gained by the bidding stage that is present in *CNET* (and absent in *DTA*) is actually lost by the fact that a "busy" contractor cannot bid, while *DTA* can use him or her for awarding the task. In such a situation the optimal strategy of the manager will depend on his knowledge of the probabilities of availability to bid of the contractors. i.e. the less are they busy the higher will be the manager's perceived utility. Note however that communication complexity is clearly higher for *CNET* (about 10 messages per negotiation if $p = 0$ and $q = 1$) than for *DTA* (about 4 messages per negotiation if $p = 1$ and $q = 0$).

Finally, note that the manager utility u_m for a manager profile (p, q, r) can be decomposed as "expected utility" based on the utilities that would have been obtained if the

manager had been used either DTA, $CNET$ or $ICNET$ protocols only, i.e. $u_m(p.q.r) = p \times u_m^{DTA} + q \times u_m^{CNET} + r \times u_m^{ICNET} = p \times u_m(1,0,0) + q \times u_m(0,1,0) + r \times u_m(0,0,1)$. This decomposition of u_m explains the planar shape of the surface representing function u_m in Figure 2. This observation extends also to the planar shape of the surface representing Mes function in Figure 3.

5 Related Works

ADEPT – *Advanced Decision Environment for Process Tasks* is probably one of the first business process management systems that proposed the utilization of intelligent software agents to cope with inter-organizational collaborative aspects of business processes including: multiple and geographically dispersed organizations, autonomous management of resources, highly dynamic and unpredictable nature of business processes, decentralized control, mixtures of human activities and automated tasks [7]. ADEPT introduced many concepts that we found useful including loosely coupling of agent tasks and services by means of service matchmaking and negotiation and usage of the notion of agency (with peer-to-peer and hierarchical relationships in organizations) that we found similar with our communities. However ADEPT was an early work and could not benefit on the recent developments including principled approaches of agent-oriented methodologies and technological advancements in software agent platforms. Additionally we did not find in ADEPT the concept of ad-hoc community that we introduced to model teams of agents that act together towards resolving a given incident.

Recently, a service negotiation framework has been proposed in paper [12]. That framework was applied to Web service negotiations between insurance companies and car repair companies. The negotiation is implemented as an interaction between Web services and the participants' preferences are represented using an ontology. The framework utilizes *ICNET* protocol for negotiation, with the message exchanges implemented as Web service method invocations. In this approach, the negotiation protocol(s) used are fixed, unlike in our approach where we do not constrain them to fixed interaction protocols. Rather, we define a set of generic negotiation steps that are likely to be followed by many negotiation protocols. Our framework is generic in the sense that it allows creation and integration of different negotiation protocols. The approach of combining generic negotiation with components for representation of negotiation participants' deal space, utilities, and resources allows us to design and evaluate different negotiation protocols useful for service negotiation in disaster management. Also, unlike [12] where the authors utilize a fixed set of subject issues specially tailored for their scenario, we give the possibility to define new subject issues with their properties that best suit the application in hand. Additionally, contractor utilities have a different semantics inspired by task allocation problems ([14]) that is more appropriate for the collaborative rather than competitive context for which our system is designed. However, similarly to [12], we use additive weighted utility functions for managers that take into account partial utility functions defined for each negotiation issue.

In paper [5], the authors introduced a new multi-issue negotiation model such that the negotiation subject is characterized by interdependencies between the issues. Similarly to our negotiation framework, this model is applied to cooperative negotiation in

crisis management. The paper discusses in some detail a scenario involving the activity of an emergency call center for victims dispatching taking into account appropriate spaces in the hospitals as well as transport constraints and availabilities. However, there are notable differences between this approach and our approach. The model proposed in [5] is using a mediator with the role of a centralized authority (for example a physician or a higher level authority) that makes proposals to participant agents, and receives their responses that either accept or reject the proposals. Rejections are accompanied by recommendations made to the mediator that enable him to adjust his proposal in the next negotiation step. Participant agents decide to accept or reject the mediator proposals using multi-criteria decision analysis tools. In our approach the problem is different – to determine one or more optimal service providers according to a set of service parameters that are dynamically determined depending on the current conditions in a situation assessment problem.

In paper [10], the authors proposed a new method for generic workflow scheduling that is able to support both human and non-human (i.e. machines, tools, or computers) resource allocation taking into account quantitative measures of the competence and preference of resources for workflow operations. In paper [6] the authors proposed a new method for optimal allocation of resources in complex manufacturing workflows using CNET negotiation. Resource agents are allowed to bid for resource allocation and deallocation for each workflow operation. Resources can dynamically change by updating their characteristics like workload and processing time. A certain cost function is defined for each workflow operation. The authors propose a sound bid evaluation function that allows to find an optimal allocation of resources to minimize the cost of workflow execution. Note however that differently from our work, both papers [10] and [6] assume a static workflow definition, while this assumption cannot be applied for our type of problems where the workflow is dynamically formed during the distributed problem solving process, possibly spanning multiple organizations [3].

In paper [11], the authors consider the problem of dynamic role assignment in agent organizations where agents can dynamically join and/or leave an organization. They propose a new version of CNET called *agent centric contract net* protocol that takes into account agent reputation for reliable agent discovery and dynamic role binding. Although the problem of dynamic role binding is more complex than service contracting that we considered in our approach, agent characteristics like reputation, as well as competence or preference can be also easily incorporated into our model by considering them as negotiation issues that convey a certain utility for the manager agent that is requesting the service.

6 Conclusions

In this paper we presented a framework that allows definition of one-to-many service negotiation protocols in agent-based collaborative processes. This framework supports flexible configuration of multi-issue negotiation subjects, properties of negotiation issues, and utility functions of participant agents. An example describing a sample negotiation scenario was discussed in detail, emphasizing how service negotiation can improve the selection of optimal service providers. We also presented a simple exper-

imental setting and initial experimental results aimed at analyzing the impact of the strategy employed by the manager agent for selecting a service negotiation protocol on the quality of the negotiation outcome as well as on the communication complexity of negotiation interactions.

As future work we plan to expand the experiments in at least two directions: (i) to analyze the impact of the negotiation protocol on the quality of the negotiation outcome as well as on the communication complexity of negotiation interactions depending on several profiles of contractor agents; the results can help the manager to tune his strategy for better selection of the negotiation protocol; (ii) to consider more complex negotiation instances that take into account more negotiation iterations, as well as that contractors might change their bids and managers can change their strategy for accepting contractors' bids during each iteration; (iii) to consider more complex workflows involving at least two interdependent negotiations such that the contracted service might also involve contracting of other required services.

Acknowledgement

The work reported in this paper was carried out as part of the Diadem project: `http://www.ist-diadem.eu`. The Diadem project is funded by the E.U. under the Information and Communication Technologies (ICT) theme of the 7th Framework Programme for R&D, ref. no: 224318.

References

1. Argente, E., Julian, V., and Botti, V.: Multi-Agent System Development Based on Organizations. Electronic Notes in Theoretical Computer Science 150, 55–71 Elsevier (2006)
2. Bădică, C., Scafeş, M.: Conceptual Framework for Design of Service Negotiation in Disaster Management Applications. In: Bai, Q., Fukuta, N. (eds.), Advances in Practical Multi-Agent Systems, Studies in Computational Intelligence 325, 359–375 Springer Verlag (2010)
3. Bădică, C., Ilie, S., Kamermans, M., Pavlin, G., and Scafeş, M.: Using Negotiation for Dynamic Composition of Services in Multi-Organizational Environmental Management. International Symposium on Environmental Software Systems ISESS'2011 (2011) (submitted)
4. Cervenka, R., Trencansky, I.: The Agent Modeling Language – AML, Whitestein Series in Software Agent Technologies and Autonomic Computing (2007)
5. Hemaissia-Jeannin, M., El Fallah Seghrouchni, A., Labreuche, C.: A New Multilateral Multi-issue Negotiation Protocol and its Application to a Crisis Management Problem. Multiagent and Grid Systems 4(1), 103–123 IOS Press (2008)
6. Hsieh, F.-S.: Analysis of contract net in multi-agent systems. Automatica 42(5), 733–740 Elsevier (2006)
7. Jennings, N. R., Norman, T. J., Faratin, P., O'Brien, P., and Odgers, B.: Autonomous Agents For Business Process Management. Applied Artificial Intelligence 14(2), 145–189 Taylor & Francis (2000)
8. Jennings, N.R., Faratin, P., Lomuscio, A.R., Parsons, S., Sierra, C., and Wooldridge, M.: Automated Negotiation: Prospects, Methods and Challenges. Group Decision and Negotiation 10(2), 199–215 Springer (2001)

9. Kreps G.A., Bosworth, S.L.: Organizational Adaptation to Disaster. In: Rodriguez, H., Quarantelli, E.L., and Dynes, R.R. (eds.), Handbook of Disaster Research, 297–315 Springer (2007)
10. Lee, K.M.: Adaptive Resource Scheduling for Workflows Considering Competence and Preference. In: Negoita, M.Gh., Howlett, R.J., and Jain, L.C. (eds.) Knowledge-Based Intelligent Information and Engineering Systems. Lecture Notes in Computer Science 3214, 723–730 Springer (2004)
11. Lee, J., Lee, S.-J., and Chen, H.-M.: Dynamic role binding with Agent-centric Contract Net Protocol in agent organizations. IEEE International Conference on Systems, Man and Cybernetics, SMC 2008, 536–643 (2008)
12. Paurobally, S., Tamma, V., and Wooldridge, M.: A Framework for Web service negotiation. ACM Transactions on Autonomous and Adaptive Systems 2(4), ACM Press (2007)
13. Pavlin, G., Kamermans, M., and Scafeş, M.: Dynamic Process Integration Framework: Toward Efficient Information Processing in Complex Distributed Systems. Informatica 34, 477–490 (2010)
14. Sandholm, T.: An implementation of the contract net protocol based on marginal cost calculations. Proceedings of the 12th International Workshop on Distributed Artificial Intelligence, 295–308 (1993)
15. Scafeş M., and Bădică, C.: Framework for Performance Evaluation of Service Negotiations in Agent Systems. Proc. of the 8th Int.Workshop on Modelling, Simulation, Verification and Validation of Enterprise Information Systems MSVVEIS 2010, 19-29, INSTICC Press (2010)
16. Scafeş, M., Bădică, C., Pavlin, G., and Kamermans, M.: Design and Implementation of a Service Negotiation Framework for Collaborative Disaster Management Applications. Proceedings of the 2nd International Conference on Intelligent Networking and Collaborative Systems INCOS'2010, 519–524 (2010)
17. Smith, R.G.: The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver. IEEE Transactions on Computers 29(12), 1104–1113 IEEE Computer Society (1980)
18. van der Aalst, W., van Hee, K.: Workflow Management: Models, Methods, and Systems. MIT Press (2002)
19. Wooldridge, M.: An Introduction to MultiAgent Systems (2nd ed.), John Wiley & Sons (2009)
20. Yang, J., Li, W.-L., and Hong, C.-Y.: An Improvement to CNCP in Large-Scale Muti-Agent System. Proccedings of the 3rd International Conference on Innovative Computing Information and Control ICICIC'08, 147 (2008)

Integrated University Information Systems and Information System Strategy

Heinrich C. Mayr

Alpen-Adria-Universität Klagenfurt
Universitätsstrasse 65-67, 9020 Klagenfurt, Austria
heinrich.mayr@aau.at

Abstract: The growing autonomy of European universities comes with a more entrepreneurial approach to governance. I.e., universities mutate from subordinated agencies of the respective ministry to independent corporate bodies that are to be managed professionally similar to enterprises of comparable size. This requires a more entrepreneurial way of thinking, decision making and acting from the leading bodies, and as a consequence, leads to a more strategic weight of the supporting information systems: all "University Processes" (i.e. the business processes of the "enterprise" University) have to be integrated and interlinked with decision support systems on the different management levels.

The Alpen-Adria-Universität (AAU) aims at being in the front line concerning IT-support for its students, staff, and University Processes. Thereby, it emphasizes strategically on self organization and self service by researchers, teachers and students. I.e., the classical counter served by administrative clerks is "out", all service processes are tailored to the needs of students and staff and to their respective context in a particular situation, i.e. use case. This includes barrier-free interfaces for persons with disabilities or special needs as well as localized services and the integration of specific infra structures.

The way to an integrated management information system that meets all needs of a flexibly acting university management is still a long one to go. Among others, this is due to the fact that the development of individual university profiles which comes with the increasing autonomy leads to specialization and thus to the opposite of a standardization of the university processes. At the other hand however, research, teaching and vocational training still are the key processes of a university, to which the university organization as well as all other processes have to be aligned with.

Typically, the scenery of information systems, software products and data warehouse systems used in a university is stepwise grown, inhomogeneous and mostly consists of locally developed software as well as of open source and proprietary standard products that are only partly harmonized by standard data exchange interfaces.

The speech outlines the AAU Information System Strategy and sketches the architecture of the AAU information system scenery, the services offered as well as the philosophy behind the realization. In addition, it discusses potential for the transition from project to product development. Finally, emphasis will be put on the AAU full costing approach and system which was developed recently in order to provide maximum cost transparency and comprehensive information to support decision making on all levels.

Insertion Modeling and its Applications

Alexander A. Letichevsky

Glushkov Institute of Cybernetics, Academy of Sciences of Ukraine,
let@cyfra.net

Abstract: The talk describes insertion modeling methodology, its theory, implementation and applications. Insertion modeling is a methodology of model driven distributed system design. It is based on the model of interaction of agents and environments. Both agents and environments are characterized by their behaviors represented as the elements of continuous behavior algebra, a kind of the ACP with approximation relation, but in addition each environment is supplied by an insertion function, which takes the behavior of an agent and the behavior of an environment as arguments and returns a new behavior of this environment. Each agent can be considered as a transformer of environment behaviors and a new kind of equivalence of agents weaker than bisimulation is defined in terms of the algebra of behavior transformations. Arbitrary continuous functions can be used as insertion functions and rewriting logic is used to define computable ones. The theory has applications for studying distributed computations, multi agent systems and semantics of specification languages.

In applications to distributed system design we use Basic Protocol Specification Language (BPSL) for the representation of requirement specifications of distributed systems. The central notion of this language is the notion of basic protocol – a sequencing diagram with pre- and postconditions represented as logic formulas interpreted by environment description. Semantics of BPSL allows concrete and abstract models on different levels of abstraction. Models defined by Basic Protocol Specifications (BPS) can be used for verification of requirement specifications as well as for generation of test cases for testing products, developed on the basis of BPS.

Insertion modeling is supported by the system VRS (Verification of Requirement Specifications), developed for Motorola by Kiev VRS group. The system provides static requirement checking on the base of automatic theorem proving, symbolic and deductive model checking, and generation of traces for testing with different coverage criteria. All tools have been developed on a base of formal semantics of BPSL constructed according to insertion modeling methodology. The VRS has been successfully applied to a number of industrial projects from different domains including Telecommunications, Telematics and real time applications.

Intelligent Technologies and Methodologies for Medical Knowledge Engineering

Abdel-Badeeh M. Salem

Head of Medical Informatics and Knowledge Engineering Research Unit
Faculty of Computer & Information Sciences,
Ain Shams University, Abbassia, Cairo, Egypt
absalem@cis.asu.edu.eg, abmsalem@yahoo.com

Abstract. Medical Intelligent Systems (MISs) are concerned with the construction of intelligent software that performs diagnosis and makes therapy recommendations. Unlike other medical applications based on other programming methods such as purely statistical methods, MISs are based on symbolic models of disease and their relationship to patient factors. Many types of MISs exist today and are applied to different medical tasks, e.g. generation alerts and remainders, diagnosis assistance, and education. In the last years various intelligent technologies and methodologies (ITM) have been proposed by the researchers in order to develop efficient MISs for different tasks. ITM offer robust computational methods for accumulating, changing, and updating knowledge (i.e. knowledge engineering) in intelligent systems. In particular they enable users with learning mechanisms that help to induce knowledge from raw data. ITM provide methods, techniques, and tools that can help solving diagnostic and prognostic problems in a variety of medical domains. ITM are used for the analysis of the importance of clinical parameters and their combinations for prognosis, e.g. prediction of disease progression; the extraction of medical knowledge of outcomes research; therapy planning and support; overall patient management.

This paper presents some of the intelligent methodologies for managing and engineering knowledge in medical knowledge-based systems. Some of the results of the research that has been carried out by the author and his colleagues at the Medical Informatics and Knowledge Engineering Research Unit, Computer Science Department, Faculty of Computer and Information Sciences, Ain Shams University, Cairo, are discussed as well. The paper covers the following topics: (a) knowledge representation techniques from the knowledge engineering point of view; (b) expert systems methodologies, rule-based and case-based reasoning; (c) producing knowledge with intelligent data mining methodology; and (d) ontological engineering approach.

Foundations, Intelligence and Methodologies

Gnoseology-based Approach to Foundations of Informatics

Mykola Nikitchenko

National Taras Shevchenko University of Kyiv
01601, Kyiv, Volodymyrska st, 60
nikitchenko@unicyb.kiev.ua

Abstract. Now it is generally recognized that informatics is to be based on its own solid foundations which should state its self-dependence and provide its self-development. In the paper we propose to use a gnoseology-based approach for developing methodological, conceptual, and formal levels of foundations. At the methodological level we describe a number of general gnoseological principles and a system of philosophical categories specifying the main features of the approach. On the conceptual level we propose to elucidate basic notions of informatics in integrity of their intensional and extensional aspects. Then we use developed notion models to construct their mathematical counterparts at the formal level. Such constructions start with formalization of the notion of data as intensionalized data. The main kinds of such data are presets, sets, and nominats (nominative data). Then we define the notion of abstract computability applicable to the described types of intensionalized data. At last, we construct hierarchy of predicate logics over introduced intensionalized data.

Keywords: foundations of informatics, methodology, intension, extension, intensionalized data, abstract computability, predicate logics.

1 Introduction

Informatics is a relatively young discipline. As such, it borrows its foundations from disciplines it is based on, primarily from mathematics, linguistics, logic, and philosophy. But coming into mature age, informatics is searching for its own foundations, which should state its self-dependence and provide its self-development.

Constructing of foundations is a highly challenging problem. This is caused by the diversity of topics studied in informatics, by variety of approaches to such topics, by vagueness of notions specified for such topics, etc.

In the first approximation we treat foundations of informatics as the systematic study of basic notions (concepts) of informatics taking in the integrity of their essential aspects; as scientific inquiry into the nature of information-related theories, their scopes, properties, and limitations.

Speaking about foundations we cannot avoid discussions about methodological (philosophical) principles of developing such foundations. It means that a certain philosophical position should help to specify methodological principles for

constructing foundations of informatics. We will try to demonstrate that many principles of gnoseology confirm their usefulness for elucidating of basic notions of informatics. Thus, we will advocate that a gnoseology-based approach can help in constructing foundations of informatics.

The aim of this paper is to describe the main principles of gnoseology-based approach and show how it can be used to explicate the basic notion of informatics – the notion of data – in integrity of its intensional and extensional aspects; and to develop fragments of mathematical formalisms, namely, computability theory and mathematical logic, oriented on such conception of data. This aim determines the structure of the paper: first, we describe methodological principles of our approach for developing foundations of informatics; then we specify the notion of intensionalized data and define basic computability theory over such data; and, at last, we describe logics of partial predicates oriented on such data.

2 Methodological Principles of the Approach

We propose to identify three levels of foundations specified by the following principle.

Principle of three-level foundations of informatics: foundations of informatics should be constructed in integrity of methodological (philosophical), conceptual (scientific), and formal (mathematical) levels.

What methodological theory should be used at the first level? There can be several ways to answer this question. The first is ad hoc one which specifies methodological principles only when it is required by some special topics. The second is specialized one. It introduces only methodological principles specific for informatics and uses them as a basis for theory development. And the third tries to integrate general (philosophical) methodological principles with special ones and present them in an explicit form.

Our approach [1] is the third one. We base it on a philosophical system developed by Hegel [2]. Strictly speaking, we use only a gnoseological component of his system. Gnoseology (epistemology) is a theory of cognition, which aims to examine the nature of knowledge, its limits, how it is derived, and how it is to be validated and tested. Thus, our approach is *a gnoseology-based approach*. This is to be supported by *the principle of gnoseology-based theory*: a theory should be developed according to the main principles, laws, and methods of gnoseology. This principle is a weak form of a Hegel's idea of a theory as applied logic. The central part of gnoseology is a system of philosophical categories. These categories can be considered as the most general features, characteristic of things. Examples are: *subject* and *object*; *abstract* and *concrete*; *internal* and *external*; *quality*, *quantity*, and *measure*; *essence* and *phenomenon*; *individual*, *general*, and *particular*; *whole* and *part*; *content* and *form*; *cause* and *effect*; *goal* and *tool*; etc. [2].

Thus, the principle of three-level foundations identifies three types of notions – categories, scientific notions (concepts), and formal notions – that constitute the basis of each level respectively.

Other general methodological principles can be briefly explained by the following considerations. Any object of study has numerous connections (relations, aspects) with other objects, thus, it can be considered as the totality of all its aspects. But we cannot immediately investigate this totality, therefore we start with some aspects (chosen due to abstraction); then we study some other aspects with their relations to the first aspects (thus making concretization). These aspects are considered as essential ones and are chosen according to practical and theoretical criteria. Despite the simplicity and coarseness of the above considerations they lead to the following principles of gnoseology.

Principle of development from abstract to concrete: development is definitely oriented change of the object (notion) from abstract to concrete (from simple to complex, from a lower level to a higher one, from the old to the new).

Triadic principle of development: one of the main schemes of development is specified as a development triad: *thesis – antithesis – synthesis*.

Principle of unity of theory and practice: theory and practice should be considered as influencing each other. This principle substantiates development of informatics notions in praxeological perspective, i.e. this development should be based on analysis of human action in information domain. The praxeological aspect is one of the main philosophical aspects relating categories of subject and object. It lies in one line with ontological, gnoseological, and axiological aspects.

Let us note that the importance of philosophical foundations for information-related disciplines (such as information science) is widely recognized. Different philosophical systems were proposed to use for this purpose, for example, K. Popper's ontology in [3], philosophy of Kuhn and Peirce in [4]. There are also proposals to develop a specific epistemology for information science [5]. A short description of philosophical approaches can be found in [6, 7]. We also advocate the necessity of philosophical studies oriented on informatics. Actually it means that an interdisciplinary approach should be used for developing foundations of informatics.

At the second, conceptual level of foundations, methodological principles are used to develop the basic scientific notions and their interrelations. Such system of basic notions (ontology) presents structure and properties of a domain under investigation. Developing scientific notions (concepts) of informatics, we chose the "closest" categories and "project" them on such concepts. Such projections transfer properties of categories and their relationships from philosophical onto conceptual level.

Mathematical base of informatics is formed by set theory, universal algebra, mathematical logic, and computability theory. Many important and useful results for informatics were obtained on this base. Still, some discrepancies between above mentioned disciplines and problems of informatics can be also admitted. They concern questions of defining data structures on various levels of abstraction, computability of functions over such data, processing of data with incomplete or fuzzy information, construction of logics oriented on information processing, etc. Such discrepancies require additional efforts in modeling problems of information domain with existing mathematical formalisms. Therefore it is reasonable to formulate a problem of developing own, more adequate mathematical foundations for informatics. But what should be the starting point of such development?

Analysis of existing mathematical formalisms shows that they are constructed on a set-theoretic platform. But its main notion – the notion of set – is explicated in

extensional style. This style is supported by the very first axiom of set theory – the extensionality axiom: two sets are equal if they consist of the same elements [8]. N. Bourbaki in his numerous treatises aimed to write a thorough unified account of all mathematics based on extensional set theory. At that period the extensional approach played a positive role permitting to specify formally many properties of mathematical objects. But we can see now more and more facts when a pure extensional orientation becomes restrictive for further development of informatics.

Thus, we propose to add explicit intensional component to notions and construct them in integrity of intensional and extensional aspects. Here the intension of a notion (of a concept) means properties which specify that notion, and the extension means objects which fall under the notion, i.e. have the properties specified by the notion intension. Intension and extension of a notion we consider as projections of categories *general* and *individual* respectively.

We should admit that this proposal is not new. The distinction between intensional and extensional aspects of a notion was known from ancient times. Aristotle in his *Posterior Analytics* already specified this distinction though he did not use explicitly the above terms. Many logicians since that time examined the questions of intension/extension dichotomy. A second wind to these investigations was given by G. Frege with his famous *meaning triangle* and R. Carnap with his *intensional/extensional investigations*. Though the dichotomy under discussion was studied primarily in logic, semiotics, and linguistics, last years it was also investigated in informatics. In its branches related to artificial intelligence, data and knowledge bases, semantic web, etc., the intensional aspects now play an important role. But in formalized (mathematical) theories intensional aspects are still used very restrictively.

The above presented considerations advocate the following principle: a notion should be presented by the triad *notion intension – notion extension – integrity of intension and extension; the intension in this integrity play a leading role* (*the principle of triadic model of a notion*). This principle may be considered as an enhancement of Frege's meaning triangle.

At the formal level of foundations, the notions, constructed at the previous level, are specialized in order to get their reasonable formalization. This formalization should take into account intensional and extensional notion aspects. This level is important for informatics because formal notions provide a basis for automatization of various phases of information processing.

Having specified a three-level structure of foundations and main principles of theory development, we can now consider basic notions of informatics at the conceptual level. As the name shows, the main notion of informatics is the notion of *information*. It can be formalized in various aspects [9, 10], but the most important ones are aspects represented by the philosophical categories of *form* and *content*. This understanding is supported by the etymology of 'information', derived from Latin '*informare*': "to give form to". Forms of information which are relatively independent of its content we call *data*.

Thus, we get the initial part of the well-known "data – information – knowledge – wisdom" hierarchy (DIKW-hierarchy) [11]. These concepts are the building blocks of informatics; and explications should be developed for all these notions. This is a difficult challenge, therefore it is reasonable to start with the notion of data which lies in the base of DIKW-like hierarchies, is relatively simple, and is an appropriate

subject for further formalization. The word 'data' is used in this paper as a plural and as a singular noun.

Among many characteristics of data (see [9] for a detailed discussion) we choose "data as manipulable objects" [12]. This treatment is a projection on conceptual level of the category of *object* taken in the praxeological aspect.

So, our abstract understanding of the notion of data includes three moments only: 1) data are regarded as a form of information content; 2) data are (relatively) independent from information content; 3) data are manipulable objects. These moments demonstrate that we treat data in a very broad sense. The principle of development from abstract to concrete suggests that other data characteristics should be introduced on the later stages of development.

According to the principle of triadic model, we treat data as integrity of their intension and extension, thus we enrich traditional understanding of data with intensional component [13]. Obtained data are called *intensionalized data*. Let us admit that there is an analogy with the notion of *typed data*, but the latter is usually understood in the extensional sense while we aim to emphasize intensional features of our approach.

3 Intensionalized Data

To simplify formalization of the notion of intensionalized data, we start with the most abstract understanding of data as some objects (keeping in memory that on later stages data should be considered as manipulable objects representing information content). The first step in objects explication is classification of intensions which can be prescribed to objects. Such classification can be made with respect to various criteria. Our gnoseology-based approach suggests to develop a classification induced by some categories. As such we choose categories *whole* and *part* which are among the first categories revealing the category of *essence* [2]. Thus, we introduce two different intensions I_W and I_P: object as a whole (unstructured object) and object with parts (structured object) respectively.

Further development of I_W is done in accordance with categories of *abstract* and *concrete*. One of projections (restrictions) of these categories on the conceptual level describes abstract as less informative than concrete. In extreme cases, an object can be regarded as a "black box" (intuitively it means that nothing is "visible", and therefore nothing is known about object) or as a "white box" (everything is "visible" and recognizable). Thus, we articulated new intensions I_{WB} and I_{WW} respectively. An intermediate intension is denoted by I_{WBW} ("black or white box"). The introduced intensions describe the main possibilities to treat object as a whole.

To come to richer intensions we should treat objects as structured (with intension I_P). In this case we get a triad: *whole – part – structure*, where structure is the synthesis of categories *whole* and *part*. Now we will invent properties (intensions) of object structures. The development principle stimulates us to start with simple structures. Simplicity means that all parts of an object are recognized and fixed. In this case each part can be regarded as a whole. Relations within the object are also recognized and fixed. The next question is: what intensions can be prescribed to the

parts and relations? Being the wholes, parts can have intensions of black and/or white boxes. Should it be allowed for relations to have the same three intensions? It is not reasonable to do this at the first stages of data development; therefore we prescribe to relations intensions of white boxes only. The above specification of object structure permits to call it *hard structure*. Thus, we divide the intension I_P into two sub-intensions I_{PH} and I_{PS} specifying objects with hard and soft structures respectively. In this paper we restrict ourselves by studying objects with prescribed intension I_{PH} only which is simpler than I_{PS}.

We continue with I_{PH} classification that is caused by possible relations between object parts. Such relations are classified along the line *tight–loose*. Loose relations mean that parts are not connected with each other (in Hegel's words, are indifferent to each other); tight relations mean that parts are connected. Thus, new intensions I_{PHL} and I_{PHT} are articulated.

Parts of objects with intension I_{PHL} and I_{PHT} are usually called *elements* and *components* respectively. Considering elements as wholes, we can treat them with intensions of black and/or white boxes. Three new intensions which are sub-intensions of I_{PHL} stem from this: I_{PHLB}, I_{PHLW}, and I_{PHLBW}.

It is worth to discuss objects (data) with these intensions in more detail. Objects with intension I_{PHLB} should be regarded as collections of black boxes because they consist of clearly separated elements with unknown interior (content) that have no relations with each other. Such objects we call *presets*. For example, buying several tickets of an instant lottery someone gets a preset because surfaces of the tickets are covered by opaque material making them black boxes. Collections of elements which are white boxes (intension I_{PHLW}) are called *explicit sets*. Collections with intension I_{PHLBW} contain "black" and "white" elements (*mixed presets*). For example, when someone comes at a party he first classifies people as known to him (white elements) or unknown (black elements). Here we should note that richer intensions than I_{PHLB} and I_{PHLBW} can allow extracting additional information from "black" elements (and in this case they become "white" or "gray" elements).

Thus, we have articulate three kinds of objects: presets, sets, and mixed presets. Now just sets serve as a basis for formalization of informatics notions, but it seems reasonable to use also the notions of preset and mixed preset for this purpose. Their importance can be substantiated by the necessity of defining data at various abstraction levels that cannot be adequately captured by the extensional notion of set. Therefore in is not strange that numerous attempts were made for constructing set theory without extensionality (see, for example, [14]). We stop further classifications of objects with intension I_{PHL} (I_{PHL}-objects) and start classifying I_{PHT}-objects.

Components of I_{PHT}-objects are in some way related to each other, so, contrary to the elements of I_{PHL}-objects, the components are not allowed to permute freely with each other within I_{PHT}-objects. Examples are: lists, trees, graphs, arrays, etc. It seems (and it is true) that a lot of useful kinds of I_{PHT}-objects can be considered. This fact poses a question what sub-intension of I_{PHT} should be introduced first? And again we appeal to the principle of development which advises to start with the simplest relation between components. Such a relation connects only two components. In this case these components look as opponents, and the relation should link them, thus making their synthesis. From this follows that it is reasonable to treat the first component as a black box, the second as a white box, and their synthesis as a "dipole"

consisting of the black and white boxes. To make these abstract considerations more concrete we should involve practical observations of activities in the information domain which prompts us that the white box is a *name* of the black box; and their relation is a *naming* (*nominative*) relation. Thus, described objects are presets whose elements are named values. We propose to call such objects *nominats* and denote corresponding intension as I_{ND}. In Slavic languages the term 'nominat' has two different meanings: a naming expression or a value of such expression. Our proposal unites these meanings, because nominat is a unity of names and values. Nominats are also called *flat nominative data*.

Traditionally, notations of functional style are chosen to represent nominats. For example, a nominat with names $v_1, \ldots, v_n$ and values $a_1, \ldots, a_n$ respectively, is denoted by $[v_1 \mapsto a_1, \ldots, v_n \mapsto a_n]$. If values themselves are nominats, then we get the notion of hierarchic nominats (hierarchic nominative data); for example $[v_1 \mapsto [u_1 \mapsto b_1, \ldots, u_k \mapsto b_k], \ldots, v_n \mapsto [t_1 \mapsto c_1, \ldots, t_m \mapsto c_m]]$ is a 2-level nominat.

It is important to admit that nominats can model the majority of data structures used in informatics [1]. For example, a set $\{e_1, \ldots, e_m\}$ can be represented as $[1 \mapsto e_1, \ldots, 1 \mapsto e_m]$, where 1 is a standard name which have different values $e_1, \ldots, e_m$; a tuple $(e_1, \ldots, e_m)$ can be represented as $[1 \mapsto e_1, \ldots, m \mapsto e_m]$ with 1,…, m as standard names; a sequence $<e_1, \ldots, e_m>$ can be represented as $[1 \mapsto e_1, 2 \mapsto [\ \ldots, 2 \mapsto [1 \mapsto e_m, 2 \mapsto \varnothing_n]\ldots]]$, where 1, 2 are standard names and $\varnothing_n$ is the empty nominat.

Summing up, we can conclude that the developed notion of intensionalized data can represent data structures used in informatics, and besides, this representation looks richer and more adequate than traditional set-theoretic representation.

In this section we have concentrated only on classification of object intensions ignoring operations with objects. To come back to the initial treatment of data as manipulable objects, we should describe operations allowed for data with different intensions. Actually it means that we should try to construct basic computability theory for intensionalized data. This will be done in the next section.

4 Computability over Intensionalized Data

We begin with general considerations about traditional computability. Such computability is usually understood as computability of *n*-ary functions defined on integers or strings. It may be called Turing computability. In the light of our investigations traditional computability does not pay much attention to variety of data intensions, because it concentrates on computability over integers (or strings) which have fixed intensions. But in informatics other data structures with different intensions are also used; therefore for these structures a new notion of computability is required [15].

The computability problem is not the only aim of our investigations. Now it is generally recognized that information systems should be developed successively from abstract specifications via more concrete representations up to detailed implementations in chosen programming languages. And it is important to connect computability with stages of system development. We intend to introduce such a

unified notion of computability that can be applied to every stage of system development and can be easily transformed when moving from stage to stage. Such a kind of computability should be applicable to data structures of different abstraction levels and is called *abstract computability*. A partial case of this computability, oriented on intensionalized data, is called *intensionalized computability*. The idea behind it is the following: for data processing it is allowed to use only those operations that conform to their intensions. Thus, intensionalized computability is *intensionally restricted computability*. In fact, such computability is a *relative computability* – relative to data intensions. Usually it is required that data have *finite structures*; a corresponding data intension we denote by I_{PHF}.

There is a difficulty in defining intensionalized computability that is caused by the fact that for finite structured data with intension I_{PHF} we do not have precise definitions of their components and relations between components; thus, precise definition of computability is not possible. Of cause, we can introduce data with precisely described sub-intensions of I_{PHF} (like intensions for tuples, list, trees, etc.) and then define computability for such specific intensionalized data. But in this case we will not get a unified computability theory for intensionalized data. To overcome this difficulty we propose to apply the method of reduction of intensionalized computability to traditional Turing computability. To do this we will first define a special form of finite structured data with a fixed intension, and then reduce data with other intensions to this special form.

Let D be a class of data with intension I_D. Such class is also denoted as $[I_D, D]$, data from this class are called I_D-intensionalized data or simply I_D-data. Assume that we treat data from D as finite structured data. Our intuitive understanding of a such data is the following: any such data d consists of several basic (atomic) components $b_1, ..., b_m$, organised (connected) in a certain way. If there are enumerably many different forms of organisation, each of these data can be represented in the (possibly non-unique) form $(k,<b_1, ..., b_m>)$, where k is the *data code* and the sequence $<b_1, ..., b_m>$ is the *data base*. Data of this form are called *natural data* [1]. More precisely, if B is any class and Nat is the set of natural numbers, then the class of *natural data* over B is the class $Nat(B) = Nat \times B^*$. We use the term 'class' for collections of intensionalized data; term 'set' is used for collections which intensions are sub-intensions of sets. As finite structured data can have different representations, we should introduce multi-valued functions for constructing such representations. Function f is *multi-valued* (non-deterministic) if being applied to the same input data d it can yield different results during different applications to d (i.e., function's graph is not a functional relation). To avoid complex notations with subscripts, to will denoted a class of partial multi-valued functions over D as $D \to\to D$. A multi-valued function is *injective*, if it yields different values on different arguments.

Now we are ready to give the formal definition of a class of intensionalized data with some intension I_D which is a sub-intension of I_{PHF}. A class D is called a class *of finite structured data*, if a class B and a total multi-valued injective mapping *nat*: $D \to\to Nat(B)$ are given. This mapping *nat* is called the *naturalization* mapping. Naturalization mapping is actually an *analysing* mapping: it finds in a data d its components and their interrelations according to the properties of data prescribed by its intension. Dually to *nat* we introduce *denaturalization* mapping *denat* which reconstructs (*synthesizes*) data of class D from natural data. For simplicity's sake we

assume that $denat=nat^{-1}$. Denaturalization mapping is a partial single-valued mapping. Naturalization and denaturalization mapping are also called *concretization* and *abstraction* mappings respectively.

Introduction of naturalization mapping is a crucial moment for defining intensionalized computability. This mapping can be regarded as a formalization of data intension; and this enables us to reduce an intuitive notion of intensionalized computability over D with intension I_D to formally defined *natural computability* over D. The latter is then reduce to a new special computability over $Nat(B)$ that is called *code computability*. To define this type of computability we should recall that in a natural data the code collects all known information about data components. Thus, code computability should be independent of any specific manipulation (processing) operations of the elements of B and can use only information that is explicitly exposed in the natural data. The only explicit information is the data code and the length of the data base. Therefore in code computability the data code plays a major role, while the elements of the data base are treated as black boxes which virtually do not affect the computations. These elements may be only used to form the base of the resulting data. To describe the code of the resulting data and the order in which elements of the initial base are put into the base of resulting data, a special function of type $Nat(B)\to\to Nat(B)$ should be defined. Such a function is called *index-computable*. These considerations lead to the following definition.

A function g: $Nat(B)\to\to Nat(B)$ is called *code-computable* if there exists an index-computable multi-valued function h: $Nat^2\to\to Nat \times Nat^*$ such that for any k, $m\in Nat$, $b_1,..., b_m\in B$, $m\geq 0$, we have $g(k,<b_1,...,b_m>) = (k',<b_{i_1}, ..., b_{i_l}>)$ if and only if $h(k,m) = (k',<i_1,...,i_l>)$, $1\leq i_1\leq m,..., 1\leq i_l \leq m$, $l\geq 0$. If one of the indexes $i_1,...,i_l$ lies outside the interval $[1, m]$, or $h(k,m)$ is undefined, then $g(k,<b_1,...,b_m>)$ is also undefined.

In other words, in order to compute g on $(k,<b_1,...,b_m>)$, we have to compute h on (k,m), generate a certain value $(k',<i_1,...,i_l>)$, and then try to form the value of the function g by selecting the components of the sequence $<b_1,...,b_m>$ pointed to by the indexes $i_1,...,i_l$.

It is clear, that index computability of h: $Nat^2\to\to Nat \times Nat^*$ may be reduced by traditional methods of recursion theory to computability of a certain partial recursive function r:$Nat\to\to Nat$.

We are ready now to give the main definition of this section. A function f: $D\to\to D$ is called *naturally computable* (with respect to given B and *nat*) if there is a code-computable function g: $Nat(B)\to\to Nat(B)$ such that $f = denat \circ g \circ nat$.

The class of all naturally computable functions is denoted by $NatComp(D, B, nat)$.

Thus, intensionalized computability is defined via a sequence of the following reductions: intensionalized computability – natural computability – code computability – index computability – partial recursive computability. Analysing the definitions, we can also conclude, that natural computability as a generalization (relativization) of enumeration computability. In fact, for $B = \varnothing$ code computability is reduced to partial recursive computability on Nat, and natural computability is reduced to enumeration computability (with respect to *nat*). Therefore, the notions of code and natural computability defined above are quite rich.

Having defined the notion of natural computability, we can now construct algebraic representations of complete classes of naturally computable partial multi-valued functions for various kinds of intensionalized data. In this short paper we give without details only few examples. We start with the simplest case.

Let D be a preset with prescribed intension I_{PS} ($=\mathtt{I}_{\mathtt{PHLB}}$). It means that nothing is known about its elements. This treatment can be formalized by the naturalization mapping $\mathrm{nat}[\mathrm{I}_{PS}]$: $D \to\to Nat(D)$ such that $nat[\mathrm{I}_{PS}](d)=(0,<d>)$ for every $d \in D$. To define the complete class of naturally computable functions over $[\mathrm{I}_{PS}, D]$, we have to describe all index-computable function of the type h: $Nat^2 \to\to Nat \times Nat^*$. It is easy to understand that under the naturalization mapping $nat[\mathrm{I}_{PS}]$ we need to know the results of index-computable function only on the element (0,1). On this input data an index-computable function 1) can be undefined, 2) can yield (0, 1), or 3) can make non-deterministic choice between being undefined and yielding (0, 1).

These three cases induce the following functions of type $D \to\to D$: 1) the everywhere undefined function *und*, 2) the identity function *id*, and 3) the non-deterministic function *und-id* such that *und-id*(d) is undefined or is equal to d.

It means that the following result was proved: *the complete class of naturally computable partial multi-valued functions over* I_{PS}*-intensionalized preset D consists of functions und, id, and und-id.*

In other words, the three functions defined above are the only computable function over "black box" intensionalized data.

The next example describes computability over subclass of hierarchic nominats. This subclass $NAD(V,W)$ is called the class of *named data* and is defined inductively on the basis of a finite set of names $V=\{v_1, \ldots, v_{\mathrm{m}}\}$ and a preset of basic values W:

1) If $w \in W$, then $w \in NAD(V,W)$,
2) If $v_1,\ldots,v_{\mathrm{n}}$ are pairwise distinct names from V, $d_1, \ldots, d_{\mathrm{n}}$ are from $NAD(V,W)$, then $[v_1 \mapsto d_1, \ldots, v_{\mathrm{n}} \mapsto d_n]$ belongs to $NAD(V,W)$.

The intension of such data is denoted by I_{NAD}. This understanding of $NAD(V,W)$ can be represented by the naturalization mapping $nat[\mathrm{I}_{NAD}]$: $NAD(V,W) \to\to Nat(W)$ which is defined inductively as follows:

1) if $d \in W$, then $\mathrm{nat}[\mathrm{I}_{NAD}](\mathrm{d})=(\mathrm{c}(0,0),<\mathrm{d}>)$;
2) if $d = [v_{i_1} \mapsto d_1,\ldots, v_{i_n} \mapsto d_n]$, $i_1<\ldots<i_n$, $n \geq 0$,

$nat[\mathrm{I}_{NAD}](d_j)=(k_j,<b_{j1}, \ldots, b_{jl_j}>)$, $1 \leq j \leq n$, then

$nat[\mathrm{I}_{NAD}](d) = (c(1, c(n,c(k'_1,\ldots c(k'_n,0)\ldots))),<b_{11},\ldots,b_{1l_1}, \ldots,b_{n1}, \ldots, b_{nl_n}>)$,

where c: $Nat \times Nat \to Nat$ is the Cantor's pairing function; $k'_j = c(i_j,c(k_j, l_j))$, $1 \leq j \leq n$.

Having defined the naturalization mapping for $[\mathrm{I}_{NAD}, NAD(V,W)]$, we obtain the class $NatComp(D, B, nat[\mathrm{I}_{NAD}])$ of all naturally computable functions over the class $NAD(V,W)$. As the basic functions from this class we choose operations [15] of *naming* $\Rightarrow v$, *denaming* $v \Rightarrow$, and *checking* $v!$ with name $v \in V$ as a parameter; we also use non-deterministic *choice* $\Diamond$ which on d yields d or $\varnothing_{\mathrm{n}}$. The main operations over this class of functions (we call them *compositions*) are *multiplication* $\circ$ (functional composition), *conditional iteration* * (while–do), and *overriding* ∇.

The following theorem which gives an algebraic description of the class $NatComp(D, B, nat[\mathrm{I}_{NAD}])$ can be proved: *the complete class of naturally computable*

partial multi-valued functions over the I_{NAD}*-intensionalized class of named data NAD(V,W) coincides with the class of functions obtained by closure of functions* $\Rightarrow v$, $v\Rightarrow$, $v!$, *and* $\circlearrowleft$ *under compositions* $\circ, *$, *and* $\nabla(v \in V)$.

The last example will describe computability over the I_{SEQ}-intensionalized class *Seq*(*B*) of sequences constructed hierarchically over a preset *B*. The structure *Seq*(*B*) has been investigated in different works. *W*e shall use the notations of [16]. The following functions are introduced: *first*, *tail*, *apndl*, and *is-atom*. Also, we need a composition, called *construction*: $[f,g](d) = \langle f(d), g(d)\rangle$ (*d* belongs to *Seq*(*B*), *f* and *g* are functions over *Seq*(*B*)). The following theorem can be proved: *the complete class of naturally computable partial multi-valued functions over the* I_{SEQ}*-intensionalized class Seq(B) coincides with the class of functions obtained by closure of functions first, tail, apndl, is-atom, and* $\circlearrowleft$ *under compositions* $\circ, *$, *and* $[]$.

Having introduced in this section the notion of intensionalized computability, we actually defined those operations which are allowed to apply to such data. Thus, combining definitions of intensionalized data with definitions of computable functions over such data, we made explication of data as manipulable objects. To reason about intensionalized data, we should develop special logics oriented on such data. This will be done in the next section.

5 Predicate Logics over Intensionalized Data

The main idea of developing logics over intensionalized data consists in defining such logical constructs (connectives, quantifiers, etc.) that conform to the data intensions. It means that these logical constructs (to be semantically explicated as compositions of predicates) should use only such data information that is specified by data intensions.

To make these intuitive considerations more strict, we start with constructing a semantic base for intensionalized logics.

Let $[I_D, D]$ be a class of intensionalized data. A class of partial functions $P = D \rightarrow Bool$ is called a class of *partial predicates* over *D* (*Bool* = {*T*, *F*}). Operations over *P* are called *predicate compositions*. Let us admit that we do not restrict predicates by data intensions. This is necessary in order to have a wider class of models for logics. But data intensions should restrict the class of predicate compositions; from this stems the main problem of logic development: how to define compositions which are intensionally restricted by I_D?

We will define such compositions according to the principle of development from abstract to concrete. Therefore we should start with the most abstract intension I_B ("black box" data). The basic compositions defined in this case are compositions of predicate disjunction $\vee$ and negation $\neg$. We define these compositions in the style of strong Kleene's connectives.

Let *p* and *q* be predicates, *d* be from *D*. Then $(p \vee q)(d)$ is defined and equal to *T* if $p(d)$ or $q(d)$ is defined and equal to *T*; is defined and equal to *F*, if both $p(d)$ or $q(d)$ are defined and equal to *F*; and is undefined in all other cases. The value $\neg p(d)$ is a dual to the value of $p(d)$. Other propositional compositions can be defined analogously.

From these definitions we see that logics over $[I_B, D]$ are just different variants of propositional logics (partiality should be also taken into account). A semantics base of such logics are predicate algebras of the form $<D \to Bool, \vee, \neg >$. Properties of such algebras will specify calculi for our logics.

We will not describe logics over classes of data with intensions I_W and I_{BW}, because the intension I_W specifies only one concrete class of data, and its logic is only a logic of this class (singular logic); logics over I_{BW}-intensionalized data are combinations of propositional and singular logics.

The next data intension we consider here is the intension of (infinite) flat nominats I_{ND} which is a sub-intension of I_B. Data, intensionalized with I_{ND}, have a form $[v_1 \mapsto a_1, v_2 \mapsto a_2, \ldots]$. In traditional logic names $v_1, v_2, \ldots$ are called individual variables, and data from D are called variable assignments, variable valuations, etc. In this case $D = {}^{V}A$, where V is a set of individual variables, A is a class of individual values, and ${}^{V}A$ can be considered as a class of partial function from V to A. Predicates from ${}^{V}A \to Bool$ are called *quasi-ary predicates.* At the level of I_{ND}-data, additionally to propositional compositions, we can define a new composition of *renomination* $R^{v_1,\ldots,v_n}_{x_1,\ldots,x_n}$. Given a predicate p and data d the value of $R^{v_1,\ldots,v_n}_{x_1,\ldots,x_n}(p)(d)$ is equal to the value of $p(d')$, where d' is obtained from d by assigning to variables $v_1, \ldots, v_n$ values of variables $x_1, \ldots, x_n$ respectively. Note, that renomination composition (primarily in syntactical aspects) is widely used in classical logic, lambda-calculus, and specification languages like Z-notation, B, TLA, etc. The obtained logics are called renomination logics (quantifier-free logics). Their semantics base are predicate algebras of the form $<{}^{V}A \to Bool, \vee, \neg, R^{v_1,\ldots,v_n}_{x_1,\ldots,x_n} >$. Properties of such algebras will specify calculi for renomination logics.

To introduce first-order logics we should specify a new data intension I_{NDQ}. This intension allows an exhaustive search within the class A. It permits to introduce quantification compositions $\exists x$ and $\forall x$ (x is a variable). The value of $\exists x p(d)$ is defined and equal to T, if there is d', differing from d only in the value of x, such that $p(d')$ is defined and equal to T; is defined and equal to F, if for all such d' the value $p(d')$ is defined and equal to F; and is undefined in all other cases. The composition of universal quantification is defined in a similar way. The semantics base of first-order logics are predicate algebras of the form $<{}^{V}A \to Bool, \vee, \neg, R^{v_1,\ldots,v_n}_{x_1,\ldots,x_n}, \exists x>$.

To preserve properties of classical first-order logic we should restrict the class ${}^{V}A \to Bool$ of quasi-ary predicates. This restriction stems from the fact that predicates of first-order logic, being defined on some data, are also defined with the same value on all extensions of this data. Such quasi-ary predicates are called *equitone*. We also introduce different variations of equitone predicates such as maxitotal (necessarily defined on maximal data), local-equitone (equitone for finite extensions only), and equicompatible (extensible to equitone predicates). Logics based on equitone and maxitotal equitone predicates are the "closest" generalizations of classical first-order logic that preserve its main properties. These logics are called *neoclassical logics*. For all these logics corresponding calculi were constructed; their *soundness and completeness* were proved. All necessary mathematical details can be found in [17].

The last class of logics considered here are logics over hierarchic nominats with intension I_{NDH}. Corresponding logics will use composite names of the form $x_1.x_2. \ldots .x_n$ as parameters of renomination and quantification compositions. For this case their definitions should be redefined to take into account hierarchic structure of data.

Summing up, we can conclude that the notion of intensionalized data 1) permits to construct new kinds of semantically-based predicate logics oriented on such data (intensionalized logics), and 2) gives possibility to explain origination of classical logics as logics oriented on "black box" data (propositional logics) or on flat nominats (first-order logics).

6 Conclusion

In the paper we tried to advocate an idea that foundations of informatics can be developed using a gnoseology-based approach. This approach specifies methodological, conceptual, and formal levels of foundations. For the methodological level we have described a number of general gnoseological principles and a system of philosophical categories specifying the main features of the approach. For the conceptual level we have proposed to elucidate basic notions of informatics in integrity of their intensional and extensional aspects. To support this idea we have defined the notion of intensionalized data, and have presented its formalization at the mathematical level. Then for such intensionalized data we have constructed basic intensionalized computability theory and several intensionalized logics of partial predicates. Still, these investigations are at the beginning phase, and we plan to continue them in the direction of developing mathematical formalisms for specification of information systems.

References

1. Nikitchenko, N.S.: A Composition-nominative approach to program semantics. Technical Report IT–TR 1998-020, Technical University of Denmark, ISSN 1396–1608 (1998)
2. Hegel, G.W.F.: Hegel's Science of Logic / translated by A.V. Miller; foreword by J. N. Findlay. London: G. Allen & Unwin (1969)
3. Brookes, B.C.: A new paradigm for information science. The Information Scientist, 1, 03, pp. 103–112 (1976)
4. Robinson, L. Karamuftuoglu, M.: The nature of information science: changing models. Information Research, 15, 4, colis717 (2010). Available at http://InformationR. net/ir/15-4/colis717.html
5. Hjorland, B.: Epistemology and the Socio-cognitive Perspective in Information science. Journal of the American Society for Information science and Technology, 53, 4, pp. 257–270 (2002)
6. Bawden, D.: Smoother pebbles and the shoulders of giants: the developing foundations of information science. Journal of Information science 34, 4, pp. 415–426 (2008)
7. Cisek, S.: Information science in the XXI century: the meta-theoretical research. In: Od książki dawnej do biblioteki wirtualnej. Przeobrażenia bibliologii polskiej. Degen, D.; Fedorowicz, M. (eds.), pp. 47–56. Toruń: Wydaw. Naukowe UMK (2009). In Polish.
8. Bourbaki, N.: Theory of Sets. Berlin: Springer-Verlag (2004)

9. Zins, C.: Conceptual approaches for defining data, information and knowledge. Journal of the American Society for Information science and Technology, 58, 4, pp. 479–493 (2007)
10. Formal Theories of Information: From Shannon to Semantic Information Theory and General Concepts of Information. Sommaruga G. (ed): LNCS, vol. 5363, Springer, New York (2009)
11. Ackoff, R. L.: From data to wisdom. Journal of Applied Systems Analysis 15, pp. 3–9 (1989)
12. Hey, J.: The Data, Information, Knowledge, Wisdom Chain: The Metaphorical link, published at Intergovernmental Oceanographic Commission, 2004, 17 pages. Available at http://web.archive.org/web/20070206032947/ioc.unesco.org/oceanteacher/OceanTeacher2/02_InfTchSciCmm/DIKWchain.pdf
13. Nikitchenko, N.S.: Intensional aspects of the notion of program. Problems of Programming, Kiev, 3–4, pp. 5-13 (2001). In Russian.
14. Apostoli, P., Hinnion, R., Kanda, A, Libert, T.: Alternative Set Theories. In: Philosophy of Mathematics, Irvine A.D. (ed.), pp. 461-491, Elsevier (2009)
15. Nikitchenko, N.S.: Abstract Computability of Non-deterministic Programs over Various Data Structures. In: Perspectives of System Information science. LNCS, vol. 2244, pp. 471–484. Berlin: Springer (2001)
16. Backus, J.: Can programming be liberated from the von Neumann style? A functional style and its algebra of programs. Communs. ACM, 21, pp. 613-641(1978)
17. Nikitchenko, M.S., Shkilniak, S.S.: Mathematical logic and theory of algorithms. Publishing house of National Taras Shevchenko Univ. of Kyiv (2008). In Ukrainian.

Pre-automata as Mathematical Models of Event Flows Recognisers

Boris Novikov, Ivan Perepelytsya, and Grygoriy Zholtkevych

V.N. Karazin Kharkiv National University,
4, Svobody Sqr., Kharkiv, 61077, Ukraine
novikov@univer.kharkov.ua, ivanperepelytsya@gmail.com,
zholtkevych@univer.kharkov.ua

Abstract. The new class of recognisers is introduced and studied in the paper. The models are based on the notion of partial action of a free finite generated monoid. Authors called such models by preautomata. Some properties of preautomata were established and proved in the paper. These properties allow to consider the pre-automata as mathematical models of recognizers of event flows in processes of the interaction of software systems.

Introduction

The experience of software development demonstrates that we have no the means for forecasting of progress of software projects (see [1]). So, in 2009, only 32% of the software development projects were successful. At the same time, the percentage of projects that ended with a significant budget overruns and the disruption of a schedule was 44%, and the projects that were interrupted in the form of loss control costs or timelines - 24% of total software development projects. As we can see, the implementation of different methods in software management, the use of increasingly sophisticated technologies in software development, have not led to significant improvement in the quality of software development processes. The reason for the complexity of the development processes of large software systems is the need to provide correct handling for all possible flows of system events. One of the authors of this article in 1990 noted [12]: "it is almost impossible to foresee the sequence of the information processing procedures for complex computer systems, and therefore impossible to plan the flow of control". Rejection of an identification of all possible control flows can provide scalability and flexibility of software product in a process of system design. Breaking down of monolithic architectures leads us to the concept of data-driven systems [11], in particular - to event-driven architecture (EDA) [4].

Modern applications development tools for EDA are based on using standard methodologies such as "Event Dispatcher - Event Listener". This methodology assumes that, each generated event can be listened by a number of handlers However, in case of an interaction of many systems the dispatchers have to listen flows of events, not only single events. The flow of events forms sensible

messages, and in this case, there are no standardized software components even at the level of mathematical models, namely, events listeners are oriented on a recognition of event flows. This work describes one mathematical model of a machine for the event flows recognition. In this mathematical model each event is modelled by the symbol of some alphabet, and messages are modelled by certain words in this alphabet. The pre-automata notion has been introduced. This notion provides a possibility to analyze the event flows to highlight from them reasonable messages that are carried by these flows.

The aim of this paper is to study recognisers which are similar to automata-based recognisers. But we will suggest that a reconiser responses to finite sequences of events. This modification leads to study of partial actions of finite generated free monoids on a set as a recogniser's model.

The notion of a partial action was introduced for groups in [3] and for monoids in [10].

This paper is organised as follows.

In section 1, definitions of the terms are given and basic notation is introduced. Then the key example is considered therein.

In section 2, the relationship between preautomata and automata is studied. The Theorem about Universal Globalisation contains the main result of the section. It substantiates using of pre-automata as models for behaviour of systems in the case of restricted observability of system's states.

In section 3, a class of a languages, which are recognised by a preautomaton, is introduced. We call this class as a class of P-recognisable languages. Then we specify such languages in the terms of right congruences on a free monoid generated by a preautomaton's alphabet.

In section 4, the Eilenberg's Structural Theorem [2, see p. 83] is proved for P-recognisable languages.

In section 5, a capability of preautomata as recognizers is clarified by comparison of the class of P-recognisable languages with other known classes of languages.

In conclusion, the set of problems, which solution gives an answer to question of adequacy using preautomata for modelling behaviour of systems, is formulated.

1 Preliminaries

The notion of a partial action is adopted from [6] as follows.

Definition 1. *Suppose X is an arbitrary set, M is a monoid with unit 1, and $X \times M \dashrightarrow X : (x, m) \mapsto x \cdot m$ is a partial map. The triple $(X, M, \cdot)$ is called a partial M-action on X iff the following conditions are held*

$$x \cdot 1 = x \text{ for all } x \in X; \tag{1}$$

$$\begin{aligned} &\text{if } x \cdot m_1 \text{ and } (x \cdot m_1) \cdot m_2 \text{ are defined then } x \cdot (m_1 m_2) \text{ is defined} \\ &\text{and } (x \cdot m_1) \cdot m_2 = x \cdot (m_1 m_2); \end{aligned} \tag{2}$$

$$\begin{aligned}&\textit{if } x \cdot m_1 \textit{ and } x \cdot (m_1 m_2) \textit{ are defined then } (x \cdot m_1) \cdot m_2 \textit{ is defined}\\ &\textit{and } x \cdot (m_1 m_2) = (x \cdot m_1) \cdot m_2.\end{aligned} \tag{3}$$

We write $x \cdot m \neq \emptyset$ if $x \cdot m$ is defined, and $x \cdot m = \emptyset$ if $x \cdot m$ is undefined.

The case of a finite generated free monoid M will be considered in the article only. Therefore we need to reformulate Definition 1.

Definition 2. *Let Q be a set of states, Σ be a finite alphabet, and suppose a partial Σ^*-action on Q is defined then the triple $(Q, \Sigma, \cdot)$ is called a preautomaton.*

As usial for free monoid Σ^* we denote its unit by ϵ.

Example 1. Some class of examples of preautomata can be built in the following way.
Let X be a set, Q be a subset of X, Σ be a finite alphabet, and suppose a Σ^*-action on X is defined. We can build a partial Σ^*-action on Q with respect to the next formula

$$x \cdot w = \begin{cases} \emptyset, & \text{iff } x \cdot w \notin Q \\ x \cdot w, & \text{iff } x \cdot w \in Q \end{cases}$$

when $x \in Q$ and $w \in \Sigma^*$.
It is easy to prove that conditions 1, 2, and 3 of Definition 1 are held. Hence, $\mathcal{P} = (Q, \Sigma, \cdot)$ is a preautomaton.
We can consider the preautomaton $\mathcal{P}$ as a restriction of a deterministic automaton [5] $\mathcal{A} = (X, \Sigma, \cdot)$ on the set Q.

Example 1 describes a general case. It will be demonstrated in the next section.

The following definition makes it possible to consider the class of all preautomata as a category.

Definition 3. *Suppose $\mathcal{P}_1 = (Q_1, \Sigma, \cdot)$ and $\mathcal{P}_2 = (Q_2, \Sigma, \cdot)$ are preautomata, $\psi\colon Q_1 \longrightarrow Q_2$ is a map. The map ψ is called an equivariant map if for each $x \in Q_1$ and $w \in \Sigma^*$ such that $x \cdot w \neq \emptyset$ the following condition is held:*

$$\psi(x) \cdot w \neq \emptyset \textit{ and } \psi(x \cdot w) = \psi(x) \cdot w.$$

The class of all Σ-preautomata with equivariant maps as morphisms is a category [9]. The proof is trivial. We denote this category by $\Sigma\mathbf{PA}$, and by $\Sigma\mathbf{PA}(\mathcal{P}_1, \mathcal{P}_2)$ we denote a set of morphisms from $\mathcal{P}_1$ to $\mathcal{P}_2$ when $\mathcal{P}_1$ and $\mathcal{P}_2$ are preautomata.

As usual [9], we introduce notions of a monomorphism, an epimorphism, and an isomorphism. Note, that an equivariant map is a monomorphism iff it is injective; in the category $\Sigma\mathbf{PA}$ there are bimorphisms which are not isomorphisms.

Definition 4. *We shall say that a preautomaton $(Q, \Sigma, \cdot)$ is a finite preautomaton iff the set Q is finite.*

The class of all finite Σ-preautomata with equivariant maps form a subcategory of the category $\Sigma\mathbf{PA}$. We denote this subcategory by $\Sigma\mathbf{FPA}$.

2 Universal Globalisation of Preautomata

The aim of this section is to prove that each preautomaton is a restriction of some automaton with same alphabet.

Definition 5. *An automaton $\mathcal{A} = (X, \Sigma, \cdot)$ is called a globalization of a preautomaton $\mathcal{P} = (Q, \Sigma, \cdot)$ if there is a monomorphism $\zeta \in \Sigma\mathbf{PA}(\mathcal{P}, \mathcal{A})$.*

At first, for each preautomaton $\mathcal{P} = (Q, \Sigma, \cdot)$ we build a set $Q_{\mathbf{gl}}$ and an injection $\iota\colon Q \longrightarrow Q_{\mathbf{gl}}$.
Put $\overline{Q} = Q \times \Sigma^*$.
For any $q_1, q_2 \in Q$ and $w_1, w_2 \in \Sigma^*$ we shall write $(q_1, w_1) \vdash (q_2, w_2)$ iff for some $u \in \Sigma^*$ is held the following condition: $w_1 = uw_2$ and $\emptyset \neq q_1 \cdot u = q_2$.
Denote by $\simeq$ the least equivalence on $\overline{Q}$ such that the condition $(q_1, w_1) \vdash (q_2, w_2) \Rightarrow (q_1, w_1) \simeq (q_2, w_2)$ is satisfied.
Now, by definition put $Q_{\mathbf{gl}} = \overline{Q}/\simeq$.
Denote by $[q, w]$ the $\simeq$-class of the $(q, w) \in \overline{Q}$.

Lemma 1. *The triple $\mathcal{P}_{\mathbf{gl}} = (Q_{\mathbf{gl}}, \Sigma, \cdot)$ is an automaton, where the action is defined by the formula $[q, w] \cdot a = [q, wa]$, when $q \in Q$, $w \in \Sigma^*$, and $a \in \Sigma$.*

Proof. One can establish this fact by direct checking of automaton's definition. □

Then, express explicitly the condition $(q_1, w_1) \simeq (q_2, w_2)$, where $q_1, q_2 \in Q$, $w_1, w_2 \in \Sigma^*$.

Definition 6. *Suppose $q \in Q$ and $w \in \Sigma^*$, we shall say that they form a canonical piar iff for each $u, v \in \Sigma^*$ such that $uv = w$ the following condition is held $q \cdot u \neq \emptyset \Rightarrow u = \epsilon$.*

We shall use the notation $q \times w$ if q and w form a canonical piar.

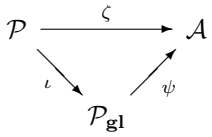
Lemma 2. *Suppose $q_1, q_2 \in Q$ and $w_1, w_2 \in \Sigma^*$ then $(q_1, w_1) \simeq (q_2, w_2)$ iff there exist u_1, u_2, and s in Σ^* such that $w_1 = u_1 s$, $w_2 = u_2 s$, $q_1 \cdot u_1 \neq \emptyset$, $q_2 \cdot u_2 \neq \emptyset$, $q_1 \cdot u_1 = q_2 \cdot u_2$, and for $q' = q_1 \cdot u_1 = q_2 \cdot u_2$ the condition $q' \times s$ is held.*

Proof. Evidently, the conclusion of the Lemma defines some equivalence, which we denote by $\sim$. The assertion $(q_1, w_1) \vdash (q_2, w_2) \Rightarrow (q_1, w_1) \sim (q_2, w_2)$ follows from the definition of $\sim$. Now, one can use the definition of $\simeq$ and check that the condition $(q_1, w_1) \sim (q_2, w_2) \Rightarrow (q_1, w_1) \simeq (q_2, w_2)$ is satisfied. From this assertion and the definition of $\simeq$ it follows that $\sim$ equals $\simeq$. □

Corollary 1. *In each $\simeq$-class there exists an unique canonical pair $(q, w) \in \overline{Q}$.*

By definition, put $\iota(q) = [q, \epsilon]$. Then from the Corallary 1 it follows that the map $\iota\colon Q \longrightarrow Q_{\mathbf{gl}}$ is injective.

Theorem 1 (about Universal Globalisation). *The map $\iota\colon Q \longrightarrow Q_{\mathbf{gl}}$ defines a globalisation $\iota\colon \mathcal{P} \longrightarrow \mathcal{P}_{\mathbf{gl}}$. It satisfies the following condition:*
for any globalisation $\zeta\colon \mathcal{P} \longrightarrow \mathcal{A}$ there is a unique morphism $\psi \in \Sigma\mathbf{PA}(\mathcal{P}_{\mathbf{gl}}, \mathcal{A})$ such that the diagram



is commutative.

Proof. First let us prove that ι is an equivariant map. In fact, suppose $q \cdot w \neq \emptyset$ when $q \in Q$ and $w \in \Sigma^*$. Using Lemma 2, we get

$$\iota(q \cdot w) = [q \cdot w, \epsilon] = [q, w] = [q, \epsilon] \cdot w = \iota(q) \cdot w.$$

Hence, ι is an injective morphism, i.e. a monomorphism, and $\iota\colon \mathcal{P} \longrightarrow \mathcal{P}_{\mathbf{gl}}$ is a globalisation.
Let $[q, w]$ be an element of $Q_{\mathbf{gl}}$. Without loss of generality, we can assume that $q \times w$. By definition, put

$$\psi([q, w]) = \zeta(q) \cdot w.$$

By construction, if q and w as above and $u \in \Sigma^*$ then

$$\psi([q, w] \cdot u) = \psi([q, wu]) = \psi([q \cdot wu_1, u_2]) = \\ \zeta(q \cdot wu_1) \cdot u_2 = \zeta(q) \cdot (wu_1u_2) = ((\zeta(q) \cdot w) \cdot u) = \psi([q, w]) \cdot u$$

when $u = u_1u_2$ and $q \times (wu_1)$.
Thus, $\psi \in \Sigma\mathbf{PA}(\mathcal{P}_{\mathbf{gl}}, \mathcal{A})$.
Finally, let q be an element of Q then we have

$$(\psi \circ \iota)(q) = \psi(\iota(q)) = \psi([q, \epsilon]) = \zeta(q) \cdot \epsilon = \zeta(q)$$

Evidently, ψ is unique. This completes the proof. □

Theorem 1 gives us the positive answer to the question "Is any preautomaton a restriction of some automaton?".

Problem 1. Let $\mathcal{P} = (Q, \Sigma, \cdot)$ be a finte preatomaton. Determine the necessary and sufficient existance conditions of a finite globalisation of $\mathcal{P}$.

3 Preacceptors and P-Recognisable Languages

Parsing of texts is the important class of tasks in computer science. Methods for solving these tasks aregrounded on the automata theory. The main concept in the context is the concept of a recognisable set [2, 5]. In this section we shall connect each preautomaton with some language. The class of such languges will be called as the class of P-recognisable language.

We begin with some notation.

Definition 7. *Let $\mathcal{P} = (Q, \Sigma, \cdot)$ be a finite preautomaton. Suppose some element $q_{in} \in Q$ (the initial state) and some subset $T \subset Q$ (the terminal subset) is marked out then a triple $(\mathcal{P}, q_{in}, T)$ is called a preacceptor.*

We shall denote the preacceptor $(\mathcal{P}, q_{in}, T)$ by $\mathcal{P}(q_{in}, T)$.
By definition, put

$$L[\mathcal{P}(q_{in}, T)] = \{w \in \Sigma^* \mid \emptyset \neq q_{in} \cdot w \in T\}, \tag{4}$$

where $\mathcal{P}(q_{in}, T)$ is a preacceptor.

Definition 8. *Let $\mathcal{P}(q_{in}, T)$ be a preacceptor then we shall say that the language $L[\mathcal{P}(q_{in}, T)]$ is recognised by $\mathcal{P}(q_{in}, T)$.*

Now we can define the class of P-recognisable languages.

Definition 9. *Let L be a language over an alphabet Σ. We shall say that the language L is P-recognisable if there exists some preacceptor such that L is recognised by it.*

Our immediate aim is to find necessary and sufficient conditions for a language be a P-recognisable language. To achieve this aim, we need several definitions.

Recall [8] that an equivalence ρ on Σ^* is called a right congruence iff for any $u, v, w \in \Sigma^*$ from $u \,\rho\, v$ it follows $uw \,\rho\, vw$.

Theorem 2. *Let L be a language over an alphabet Σ. It is P-recognisable iff there exists a right congruence on the monoid Σ^* such that L is equal to some finite union of its classes.*

Proof. Suppose, that $\mathcal{P}(q_{in}, T)$ is a preacceptor that it recognises the language L. Denote by $\mathcal{P} = (Q, \Sigma, \cdot)$ the preautomaton such that $\mathcal{P}(q_{in}, T) = (\mathcal{P}, q_{in}, T)$. Let $\iota \colon \mathcal{P} \longrightarrow \mathcal{P}_{\mathbf{gl}}$ be the universal globalisation of $\mathcal{P}$, $Q_{\mathbf{gl}}$ be a set such that $\mathcal{P}_{\mathbf{gl}} = (Q_{\mathbf{gl}}, \Sigma, \cdot)$. By definition, put $T_{\mathbf{gl}} = \{[q, \epsilon] \in Q_{\mathbf{gl}} \,|\, q \in T\}$ and denote, by $\mathcal{P}_{\mathbf{gl}}([q_{in}, \epsilon], T_{\mathbf{gl}})$ the acceptor $(\mathcal{P}_{\mathbf{gl}}, [q_{in}, \epsilon], T_{\mathbf{gl}})$.
Put $u \,\rho\, v$ iff $[q_{in}, u] = [q_{in}, v]$.
The binary relation ρ on Σ^* is a right congruence. It follows from Lemma 1.
From Corollary 1 it follows that the acceptor $\mathcal{P}_{\mathbf{gl}}([q_{in}, \epsilon], T_{\mathbf{gl}})$ recognises the same language as the preacceptor $\mathcal{P}(q_{in}, T)$. Moreover,

$$[w]_\rho = \{w \in \Sigma^* \mid w = us, \emptyset \neq q_{in} \cdot u \times s, (q_{in}, w) \in [q_{in} \cdot u, s]\}.$$

Hence, $\emptyset \neq q_{in} \cdot w \in T$ iff $(q_{in}, w) \in [q, \epsilon]$ for some $q \in T$.
Summing the reasoning, we get $L = \bigcup\limits_{q \in T} \{w \in \Sigma^* \mid (q_{in}, w) \in [q, \epsilon]\}$, i.e. L is equal to a finite union of ρ-classes.
Conversely, suppose $L = \bigcup\limits_{i=1}^{n} [w_i]_\rho$, where ρ is some right congruence on Σ^*, $w_1, \ldots, w_n \in \Sigma^*$.
By definition, put $Q = \Sigma^*/\rho$, $[u]_\rho \cdot w = [uw]_\rho$.

Evidently, $\mathcal{A} = (\Sigma^*, \Sigma, \cdot)$ is an automaton. Therefore, we can define a preautomaton $\mathcal{P} = (Q, \Sigma, \cdot)$ as the restriction $\mathcal{A}$ on the set Q.
Now, consider the preacceptor $\mathcal{P}([\epsilon]_\rho, T)$, where $T = \{[w_1]_\rho, \ldots, [w_n]_\rho\}$.
If $w \in L$ then $w \,\rho\, w_i$ for some $1 \leq i \leq n$ by assumption, therefore $\emptyset \neq [\epsilon]_\rho \cdot w = [w_i]_\rho \in T$ and w is recognised by $\mathcal{P}([\epsilon]_\rho, T)$.
If w is recognised by $\mathcal{P}([\epsilon]_\rho, T)$ then $\emptyset \neq [\epsilon]_\rho \cdot w \in T$, i.e. $[w]_\rho = [w_i]_\rho$ for some $1 \leq i \leq n$. Hence, $w \in L$.
This completes the proof. □

Corollary 2. *Let L_1 and L_2 be P-recognasible languages over the same alphabet then $L_1 \bigcap L_2$ is a P-recognisable language too.*

Corollary 3. *The class of P-recognisable languges over a single-letter alphabet equals to the class of recognisable languages over the same alphabet.*

4 Structure of P-Recognisable Languages

In this section we shall prove that the structure of P-recognisable languages is similar to the structure of recognisable languages [2].

Lemma 3. *Let L be a P-languages then $L = \bigcup_{i=1}^{n} L_i$, where*

$$L_i \bigcap L_j = \emptyset \text{ for } 1 \leq i \neq j \leq n; \tag{5}$$

$$\text{each } L_i \text{ is recognised by a preacceptor that its terminal subset is an unit set.} \tag{6}$$

Proof. Let $\mathcal{P}(q_{in}, T)$ be a preacceptor that recognises the language L. Suppose $T = \{q_1, \ldots, q_n\}$ then denote by L_i the language recognised by $\mathcal{P}(q_{in}, \{q_i\})$, where $1 \leq i \neq j \leq n$. By construction, properties (5) and (6) are satisfied. □

Let us remember [2, 8]

1. let L be a subset of Σ^*, and u be a word over Σ^* then $u^{-1}L = \{w \in \Sigma^* \mid uw \in L\}$;
2. a language $L \subset \Sigma^*$ is unitary if for any $u_1, u_2 \in L$ it is held $u_1^{-1}L = u_2^{-1}L$;
3. a language $L \subset \Sigma^*$ is a prefix code iff for any $u, v \in \Sigma^*$ such that $u, uv \in L$ it follows $v = \epsilon$.

Note, if L is a prefix code then from $\epsilon \in L$ it follows $L = \{\epsilon\}$.

Lemma 4. *Let L be a language over an alphabet Σ then L is unitary iff L is recognised by a preacceptor such that its terminal subset is a unit set.*

Proof. Let L be a unitary language then there exists an acceptor $\mathcal{A}(q_{in}, \{q_{accept}\})$ which recognises the language L [2]. Denote by $\mathcal{P}(q_{in}, \{q_{accept}\})$ the restriction of $\mathcal{A}(q_{in}, \{q_{accept}\})$ on the set $\{q_{in}, q_{accept}\}$ then the preacceptor $\mathcal{P}(q_{in}, \{q_{accept}\})$

recognises the language L.
Conversely, suppose L is recognised by some preacceptor $\mathcal{P}(q_{in}, \{q_{accept}\})$, and $\mathcal{P}_{gl}([q_{in}, \epsilon], \{[q_{accept}, \epsilon]\})$ is its universal globalisation.
The acceptor $\mathcal{P}_{gl}([q_{in}, \epsilon], \{[q_{accept}, \epsilon]\})$ recognises the language L. Using results of [2, Prop. 1.1], one can get that L is an unitary language. □

Theorem 3 (about Structure of P-recognisable Languages). *Let L be a P-recognisable language then $L = \bigcup_{i=1}^{n} E_i B_i^*$, where E_i, B_i are prefix codes for $i = 1, \dots n$, and $E_i B_i^* \bigcap E_j B_j^* = \emptyset$ for $1 \leq i \neq j \leq n$.*

Proof. Indeed, from Lemma 3 and Lemma 4 follows that $L = \bigcup_{i=1}^{n} L_i$, where each L_i is a unitary language, and $L_i \bigcap L_j = \emptyset$ if $i \neq j$. In [2, Prop. 3.4] it has been proved that any unitary language has the representation EB^*, when E, B are prefix codes. This completes the proof. □

Problem 2. Describe the class of languages with structure as in Theorem 3 which are P-recognisable.

5 Preautomata Recognition Capability

In this section we compare the class of P-recognisable language with other classes of languages [7]: the class of recognisable languages, the class of context free languages, the classes of recursive and recursively enumerable languages.

At first, compare the class of P-recognisable language with the class of recognisable languages.

Proposition 1. *Any recognisable language is P-recognisable.*

Proof. It is trivial. □

Others cases of a comparison are more complicated.

Example 2. As known [7], $E_1 = \{a^n b^n \mid n > 0\} \subset \{a, b\}^*$ is a context free language. It is evident that E_1 is a prefix code. From Lemma 4 it follows that E_1 is P-recognisable.

We need to improve Theorem 2.

Definition 10. *Let L be a language over an alphabet Σ, u, v be words over Σ. We shall use notation $u \, \rho_L \, v$ iff for any $w \in \Sigma^*$ it is satisfied $uw \in L \Leftrightarrow vw \in L$. In this case, we shall call ρ_L a right syntactic congruence induced by L.*

It is evident that ρ_L is a right congruence on Σ^*.

Proposition 2. *Let L be a language over an alphabet Σ. It is P-recognisable iff L is a finite union of ρ_L-classes.*

Proof. It follows from Theorem 2 and properties of right syntactic congruences [8, p. 27]. □

Example 3. Let L_1 be a language that is formed by all palindromes over the alphabet $\{a, b\}$. Note, that L_1 is a context free language [7]. But it is easy to see, that it is not held $a^m \, \rho_{L_1} \, a^n$ for $0 < m < n$. Therefore, $L_1 \supset \bigcup\limits_{n>0} [a^n]_{\rho_{L_1}}$ and L_1 is not P-recognisable.

Example 4. Let E_2 be a language over the alphabet $\{a, b, c\}$. Suppose $E_2 = \{a^n b^n c^n \mid n > 0\}$. It is evident, that E_2 is a prefix code, therefore it is P-recognisable. But well known [7], E_2 is not a context free language.

Example 5. Let L_2 be a language over the alphabet $\{a\}$. Suppose $L_2 = \{a^{n^2} \mid n > 0\}$. Evidently, L_2 is a recursive language. It is easy to see, that L_2 is not P-recognisable.

In contrast to recognisable languages, there exist a P-recognisable language which is not a recursively enumerable language. Unfortunally, our proof is not constructive.

Proposition 3. *There exists a P-recognisable language which is not recursively enumerable.*

Proof. The class of a recursively enumerable languages over some finite alphabet is countable. The cardinality of the class of all prefix codes over some finite alphabet equals to the cardinality of continuum. This completes the proof. □

Next proposition establishes that the class of P-recognisable languages is not closed under operations of a Kleene algebra.

Proposition 4. *Let Σ be a finite alphabet such that its power greater than 1, and $\Sigma\mathfrak{PR}$ be the class of P-recognisable languages over Σ then*

$$\text{there exist } L_1, L_2 \in \Sigma\mathfrak{PR} \text{ such that } L_1 \cup L_2 \notin \Sigma\mathfrak{PR} \tag{7}$$

$$\text{there exist } L_1, L_2 \in \Sigma\mathfrak{PR} \text{ such that } L_1 \cdot L_2 \notin \Sigma\mathfrak{PR} \tag{8}$$

$$\text{there exists } L \in \Sigma\mathfrak{PR} \text{ such that } L^* \notin \Sigma\mathfrak{PR} \tag{9}$$

Proof. To prove (7) put $L_1 = \{a^n \mid n > 0\}$, $L_2 = \{a^n b^n \mid n > 0\}$, and $L = L_1 \cup L_2$. Evidently, $L_1, L_2 \in \Sigma\mathfrak{PR}$ and for any $n > 0$ it is satisfied $[a^n]_{\rho_L} \subset L$. But it is not satisfied $a^m \, \rho_L \, a^n$ for $m \neq n$. From Proposition 2 it follows that $L \notin \Sigma\mathfrak{PR}$.
To prove (8) put $L = L_1 \cdot L_2$. Suppose that $1 < m < n$ then it is not satisfied $a^{m+1}b \, \rho_L \, a^{n+1}b$.
Indeed,

$$(a^{m+1}b) \cdot b^m = a^{m+1}b^{m+1} \notin L$$
$$(a^{n+1}b) \cdot b^m = a^{n+1}b^{m+1} = a^{n-m}a^{m+1}b^{m+1} \in L$$

As above, it is easy to see $L \notin \Sigma\mathfrak{PR}$.
To prove (9) put $L = \{a^n b^n \mid n > 0\} \cup \{a\}$. It is easy to see $[ab]_{\rho_L} = \{a^n b^n \mid n > 0\}$ and $[a]_{\rho_L} = \{a\}$, hence $L \in \Sigma\mathfrak{PR}$. As above, it is not satisfied $a^m \, \rho_{L^*} \, a^n$ for $m \neq n$. But $a^n \in L^*$, therefore $L^* \notin \Sigma\mathfrak{PR}$. □

Conclusion

We have introduced the new class of algebraic objects for systems behaviour modelling. Objects of this class are similar to deterministic finite automata. But presented models permit to describe hidden from observer behaviour of a system.

A model of this class can be obtained by a restriction some automaton on a finite subset of its states. An abstract concept to describe such models have been introduced. We call corresponding abstract objects by preautomata.

Theorem about universal globalisation for preautomata has been proved in the article. The theorem states that any preautomaton can be represented by a restriction of some automaton on a finite subset of its states.

Then we studied recognisers which based on preautomata and the corresponding class of languages.

Languages of this class have been called P-recognisable languages. The theorem about structure of P-recognisable languages have been proved.

Finally, the place of P-recognisable languages was determined among other classes of languages.

References

1. CHAOS Summary 2009. Standish Group, CHAOS Report (2009) http://www1.standishgroup.com/newsroom/chaos˙2009.php
2. Eilenberg, S.: Automata, Languages, and Machines. Volume A. Academic Press, New York and London (1974)
3. Exel, R.: Partial actions of groups and actions of semigroups. Proc. Amer. Math. Soc., **126** (1998) 3481–3494
4. Ferg, S.: Event-Driven Programming: Introduction, Tutorial, History. SourceForge (2006) http://eventdrivenprg.sourceforge.net
5. Holcombe, W. M. L.: Algebraic automata theory. Cambridge University Press, Cambridge (1982)
6. Hollings, C. Partial actions of monoids. Semigroup Forum, **75** (2007) 293–316
7. Hopcroft, G.E., Motwani, R., Ullman, J.D. Introduction to Automata Theory, Languages, and Computation (2nd Edition). Addison Wesley Publishing Co., Boston (2000)
8. Lallemant, G.: Semigroups and Combinatorial Applications. John Wiley & Sons, New York (1979)
9. Mac Lane, S.: Categories for the Working Mathematician. Springer, Berlin (1971)
10. Megrelishvili, M., Schröder, L.: Globalization of confluent partial actions on topological and metric spaces. Topology and its Appl., **145** (2004) 119–145
11. Microsoft Developer Framework. Microsoft Corporation (2010) http://msdn.microsoft.com/en-us/libriary/dd819894(VS.85).aspx
12. Zholtkevych, G: Design Principles of CAD for Engineering of a Reusable Technological Fitment [in Russian]. Central Research Institute of Information, Moscow (1990)

Insertion Modeling System And Constraint Programming

Alexander A. Letichevsky[1], Olexander A.Letychevskyi[1], Vladimir S. Peschanenko[2], Igor O. Blynov[2], Dmitry M. Klionov

[1] Glushkov Institute of Cybernetics, Academy of Sciences of Ukraine,let@cyfra.net,lit@iss.org.ua
[2] Kherson State University, vladimirius@gmail.com, anubis.igor@gmail.com, soulslayermaster@gmail.com

Abstract. The paper relates to practical aspects of insertion modeling. Insertion modeling system is an environment for the development of insertion machines, used to represent insertion models of distributed systems. The architecture of insertion machines and insertion modeling system IMS is presented. Insertion machine for constraint programming is specified as an example, and as a starting point of 'verifiable programming' project.

1 Introduction

Insertion modeling is the approach to modeling complex distributed systems based on the theory of interaction of agents and environments [1–3]. Mathematical foundation of this theory was presented in [4]. During the last decade insertion modeling was applied to the verification of requirements for software systems [5–9].

First time the theory of interaction of agents and environments was proposed as an alternative to well known theories of interaction such as Milner's CCS [10] and *pi*-calculus [11], Hoare's CSP [12], Cardelli's mobile ambients [13] and so on. The idea of decomposition of a system to a composition of environment and agents inserted into this environment implicitly exists in all theories of interaction and for some special case it appears explicitly in the model of mobile ambients.

Another source of ideas for insertion modeling is the search of universal programming paradigms such as Gurevich's ASM [14], Hoare's unified theories of programming [15], rewriting logic of Meseguer [16]. These ideas were taken as a basis for the system of insertion programming [17] developed as the extension of algebraic programming system APS [18]. Now this system initiated the development of insertion modeling system IMS which started in Glushkov Institute of Cybernetics. The development of this system is based on the version of APS enhanced by the former student of the author V.Peschanenko. The first version of IMS and some simple examples of its use are available from [19].

To implement the insertion model in IMS one must develop insertion machine with easily extensible input language, the rules to compute insertion functions

and a program of interpretation and analyzing of insertion models. The architecture, input languages and examples of insertion machines and insertion modeling system are considered in the paper.

2 The Architecture of Insertion Modeling System

Insertion modeling system is an environment for the development of insertion machines and performing experiments with them. The notion of insertion machine was first introduced in [17] and it was used as a tool for programming with some special class of insertion functions. Later this notion was extended for more wide area of applications, different levels of abstraction, and multilevel structures.

Insertion model of a system represent this system as a composition of environment and agents inserted into it. Contrariwise the whole system as an agent can be inserted into another environment. In this case we speak about internal and external environment of a system. Agents inserted into the internal environment of a system themselves can be environments with respect to their internal agents. In this case we speak about multilevel structure of agent or environment and about high level and low level environments.

As usually, insertion function is denoted as $E[u]$ were E is the state of environment and u is the state of an agent (agent in a given state). $E[u]$ is a new environment state after insertion an agent u. So, the expression $E[u[v], F[x, y, z]]$ denotes the state of a two level environment with two agents inserted into it. At the same time E is an external environment of a system $F[x, y, z]$ and F is an internal environment of it. All agents and environments are labeled or attributed transition systems (labeled systems with states labeled by attribute labels [9]). The states of transition systems are considered up to bisimilarity. This means that we should adhere to the following restriction in the definition of states: if $E \sim_B E'$ and $u \sim_B u'$ then $E[u] \sim_B E'[u']$.

The main invariant of bisimilarity is the behavior $\mathrm{beh}(E)$ of transition system in the state E (an oriented tree with edges labeled by actions and nodes labeled by attribute labels). Therefore the restriction above can be written as follows:

$$\mathrm{beh}(E) = \mathrm{beh}(E') \wedge \mathrm{beh}(u) = \mathrm{beh}(u') \Rightarrow \mathrm{beh}(E[u]) = \mathrm{beh}(E'[u'])$$

Behaviors themselves can be considered as states of transition systems. If the states are behaviors then the relation above is valid automatically, because in this case $\mathrm{beh}(E) = E, \mathrm{beh}(u) = u$. Otherwise the correctness of insertion function must be proved in addition to its definition. In any case we shall identify the states with the corresponding behaviors independently from their representation.

To define finite behaviors we use the language of behavior algebra (a kind of process algebra defined in [4]). This algebra has operation of prefixing, nondeterministic choice, termination constants $(\Delta, 0, \bot)$ and approximation relation. For attributed transition systems we introduce the labeling operator for behaviors. To define infinite behaviors we use equations in behavior algebra. Usually these equations have the form of recursive definitions $u_i = F_i(u), i \in I$. Left hand sides

of these definitions can depend on parameters $u_i(x_i) = F_i(u, x), i \in I$. To define the attribute labels we use the set of attributes, symbols taking their values in corresponding data domains. These attributes constitute a part of a state of a system and change their values in time. All attributes are devided to external (observable) and internal (nonobservable). By default the attribute label of a state is the set of values of all observable attributes for this state.

The general architecture of insertion machine is represented on the fig. 1.

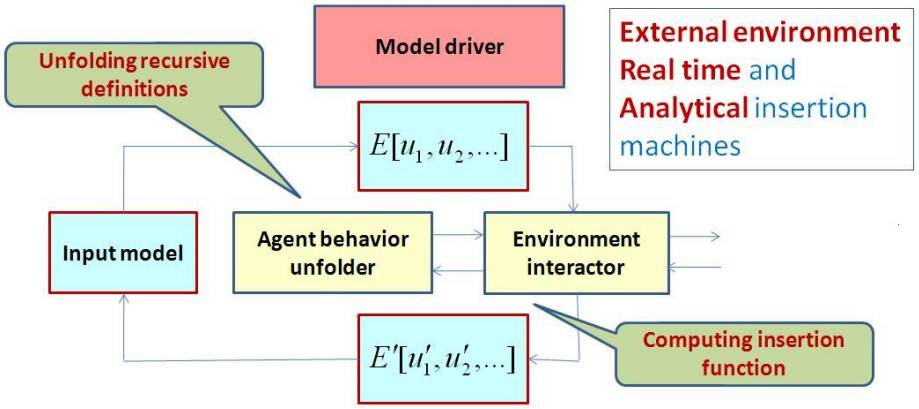


Fig. 1. Architecture of Insertion Machine

The main component of insertion machine is model driver, the component which controls the machine movement along the behavior tree of a model. The state of a model is represented as a text in the input language of insertion machine and is considered as an algebraic expression. The input language include the recursive definitions of agent behaviors, the notation for insertion function, and possibly some compositions for environment states. The state of a system must be reduced to the form $E[u_1, u_2, \ldots]$. This functionality is performed by the module called agent behavior unfolder. To make the movement, the state of environment must be reduced to the normal form

$$\sum_{i \in I} a_i.E_i + \varepsilon$$

where a_i are actions, E_i are environment states, ε is a termination constant. This functionality is performed by the module environment interactor. It computes the insertion function calling if it is necessary the agent behavior unfolder. If the infinite set I of indices in the normal is allowed, then the weak normal form $a.F + G$ is used, where G is arbitrary expression of input language.

Two kinds of insertion machines are considered: *real type* or *interactive* and *analytical* insertion machines. The first ones exist in the real or virtual environment, interacting with it in the real or virtual time. Analytical machines

intended for model analyses, investigation of its properties, solving problems etc. The drivers for two kinds of machines correspondingly are also divided on *interactive* and *analytical drivers*.

Interactive driver after normalizing the state of environment must select exactly one alternative and perform the action specified as a prefix of this alternative. Insertion machine with interactive driver operates as an agent inserted into external environment with insertion function defining the laws of functioning of this environment. External environment, for example, can change a behavior prefix of insertion machine according to their insertion function. Interactive driver can be organized in a rather complex way. If it has criteria of successful functioning in external environment intellectual driver can accumulate the information about its past, develop the models of external environment, improve the algorithms of selecting actions to increase the level of successful functioning. In addition it can have specialized tools for exchange the signals with external environment (for example, perception of visual or acoustical information, space movement etc.).

Analytical insertion machine as opposed to interactive one can consider different variants of making decision about performed actions, returning to choice points (as in logic programming) and consider different paths in the behavior tree of a model. The model of a system can include the model of external environment of this system, and the driver performance depends on the goals of insertion machine. In the general case analytical machine solves the problems by search of states, having the corresponding properties(goal states) or states in which given safety properties are violated. The external environment for insertion machine can be represented by a user who interacts with insertion machine, sets problems, and controls the activity of insertion machine.

Analytical machine enriched by logic and deductive tools can be used for symbolic modeling. The state of symbolic model is represented by means of properties of the values of attributes rather then their concrete values.

General architecture of insertion modeling system is represented on fig. 2. High level model driver provides the interface between the system and external environment including the users of the system. Design tools based on algebraic programming system APS are used for the development of insertion machines and model drivers for different application domains and modeling technologies. Verification tools are used for the verification of insertion machines, proving their properties statically or dynamically. Dynamic verification uses generating symbolic model traces by means of special kinds of analytical model drivers and deductive components.

The repository of insertion machines collects already developed machines and their components which can be used for the development of new machines as their components or templates for starting. Special library of APLAN functions supports the development and design in new projects. The C++ library for IMS supports APLAN compilers and efficient implementation of insertion machines. Deductive system provides the possibility of verification of insertion models.

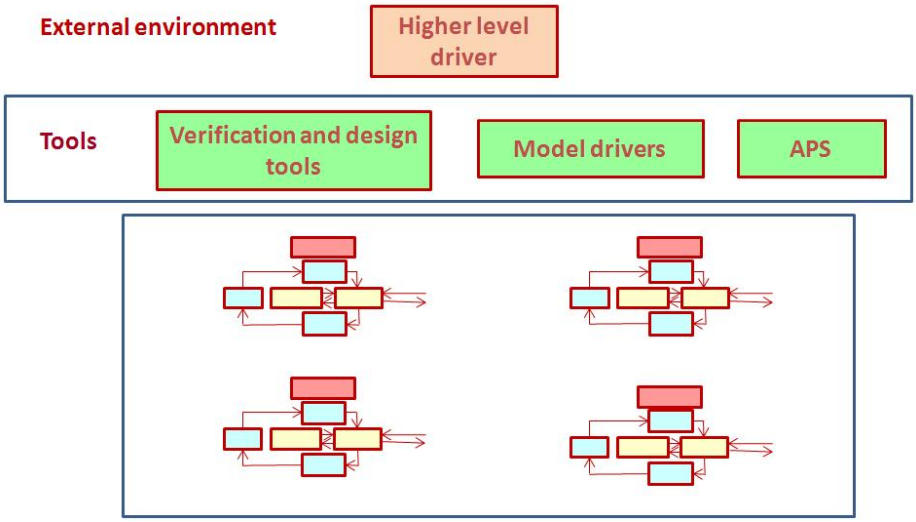


Fig. 2. Architecture of Insertion Modeling System IMS

3 Input Languages of Insertion Machines

Input language of insertion machine is used to describe the properties of a model and its behavior. This description consists of the following parts: environment description, behavior description (including the behavior of environment and the behaviors of agents), and insertion function. The behavior description has the following very simple syntax:

<behavior>::= Delta | bot | 0 | < action > | <action> . <behavior> |
<behavior> + <behavior>|
<environment state>[<list of named agent behaviors separated by ,>]|
<functional expression>

<named agent behavior>::=<agent name>:<behavior>

Therefore, the language of behavior algebra (termination constants, prefixing and nondeterministic choice) is extended by functionals expressions and explicit representation of insertion function. The syntax and semantics of actions, environment states, and functional expressions are defined in the environment description. We shall not consider all possibilities and details of environment description language restricting ourselves by making only some necessary comments.

First of all note, that all main components of behavior algebra language (actions, environment states, and functional expressions) are algebraic or logic expressions of base language (terms and formulas). This language is a multisorted (multitype) first order logic language. The signature of this language is defined in the environment description. Functional and predicate symbols can be interpreted and uninterpreted. Interpreted symbols have fixed domains and

interpretations given by algorithms of computing values or reducing to canonical forms. All uninterpreted symbols have types and their possible interpretations are restricted by definite domains and ranges. Uninterpreted functional symbols are called *attributes*. They represent the changing part of the environment. Attributes of arity 0 are called simple attributes, others are called *functional* ones. Predicates are considered as functions ranging in Boolean type $\{0, 1\}$. If an attribute f has functional type $(\tau_1, \tau_2, \ldots) \to \tau$ then *attribute expressions* $f(t_1, t_2, \ldots)$ are available for all other expressions.

3.1 Examples of Insertion Machines

The simplest insertion machines are machines for parallel and sequential insertion. Insertion function is called sequential if $E[u, v] = E[u; v]$ where ";" means sequential composition of behaviors. Special case of sequential insertion is a strong sequential composition: $E[u] = (E; u)$. This definition assumes that actions of agents and environment are the same and environment is defined by its behaviors. The sequentiality of this composition follows from associativity of sequential composition of behaviors.

Example of insertion machine with strong sequential insertion is represented on fig. 3.

```
Model Sequential
   interactor rs(P,Q,a)(
      Delta[P+Q]=Delta[P]+Delta[Q],
      Delta[a.P]=a.Delta[P],
      Delta[P]=Delta[unfold P],
      Q[P]=(Q;P)
   );
   unfolder rs(x,y)(
      (x;y)=seq(x,y),
      A=a.A+Delta,
      C=c.C+Delta
   );
   initial(C[A]);
   terminal(Delta[Delta])
)
```

Fig. 3. Example of Strong Sequential Insertion

The function **seq** is a function from IMS library that defines the sequential composition of behaviors:

$$(u; v) = \sum_{u \xrightarrow{a} u'} a.(u'; v) + \sum_{u = u + \varepsilon} (\varepsilon; v), (0; v) = 0, (\Delta; v) = v, (\bot; v) = \bot$$

The function **unfold** reduces the behavior expression to normal form $\sum a_i.u_i + \varepsilon$. This insertion machine generates a word $c^n a^m$ with nondeterministically chosen

$m, n \geq 0$ and successfully terminates. We can define as the condition for the goal state the equality $m = n$ and the driver for this machine will terminate on traces $c^n a^n$.

An example of sequential (not strong) insertion is shown on fig. 4.

```
Model Imperative(
  insertion rs(P,Q,H,a,x,y,u,v)(
    E[P+Q]=Delta[P]+Delta[Q],
    E[define_env H.P]=H[P],
    E[(x:=y).P]=assign_proc(E,x,y,P),
    E[check(u,x,y).P]=if(compute_obj(E,u),E[x;P],E[y;P]),
    E[a.P]=a.Delta[P],
    E[P]=E[unfold P]
  )where(
    assign_proc:=proc(E,x,y,P)(E.x⟶compute_obj(E,y);return E[P])
  );
  behaviors rs(P,Q,x,y,z,u)(
    (x;y)=seq(x,y),
    (u→ else Q)=check(u,P,Q),
    while(u,P)=check(u,(P;while(u,P)),Delta),
    for(x,y,z,P)=(x;while(y,(P;z)))
  );
  initial(
    define_env obj(i:Nil,x:10,y:Nil,fact:Nil);
    y:=1;for(i:=1,i ≤ x,i:=i+1,y:=y*i);
    fact:=y
  );
  terminal rs(E)(E[Delta]=1,E=0)
)
```

Fig. 4. Model of Simple Imperative Language

This example is a model of simple imperative language and can be considered as insertion representation of its operational semantics.

Insertion function is called a parallel insertion function if $E[u, v] = E[u \parallel v]$. Special case of parallel composition is a strong parallel insertion:$E[u] = E \parallel u]$. As in the case of strong sequential composition this definition assumes that actions of environment and agents are the same. Example of a model with strong parallel insertion is presented on the fig. 5. Functions **synchr**, **lmrg**, and **delta** from IMS library are used for definition of parallel composition. Their meaning can be define by the following formulas:

$$\mathbf{synchr}(x, y) = \sum_{\substack{x \xrightarrow{a} x' \\ y \xrightarrow{b} y'}} (a \times b).(x', y'),$$

```
Model Parallel
   interactor rs(P,Q,a)(
      Delta[P+Q]=Delta[P]+Delta[Q],
      Delta[a.P]=a.Delta[P],
      Delta[P]=Delta[unfold P],
      Q[P]=(Q ‖ P )
   );
   unfolder rs(x,y,n)(
      (x;y)=seq(x,y),
      x ‖ y = synchr(x,y)+lmrg(x,y)+lmrg(y,x)+delta(x,y),
      x |^ 1=x,
      x |^ 2=synchr(x,x)+lmrg(x,x)+delta(x,x),
      x |^ n= x ‖ (x |^(n-1))),
   );
   initial (Delta[((a;b) ‖ (a;b));a+b ]);
   terminal (Delta[Delta])
)
```

Fig. 5. Example of Strong Parallel Insertion

$$\mathbf{lmrg}(x,y) = \sum_{x \xrightarrow{a} a'} a.\left(x' \parallel y\right), \mathbf{delta}(x,y) = \sum_{\substack{x=x+\varepsilon \\ y=y+\mu}} \varepsilon \parallel \mu$$

3.2 Restrictions on Insertion Functions

The most typical restriction is additivity. Insertion function is called additive if $E[u+v] = E[u]+E[v], (E+F)[u] = E[u]+F[u]$. Another restriction, which allow to reduce the number of considered alternatives when behaviors are analyzed is the commutativity of insertion function: $E[u,v] = E[v,u]$. Especially the parallel insertion is a commutative one. Some additional equations: $0[u] = 0, \Delta[u] = u, \bot[u] = bot$.

The state of environment is called indecomposable if from $E = F[u]$ it follows that $E = F$ and $u = \Delta$. Equality means bisimilarity. The set of all indecomposable states constitutes the kernel of a system. Indecomposable states (if they exist) can be considered as states of environment without inserted agents. For indecomposable states usually the following equations hold: $E[0] = 0, E[\Delta] = E, E[\bot] = \bot$.

In [3] the classification of insertion functions was presented: one-step insertion, head insertion, and look-ahead insertion. Later we shall use insertion functions with the following main rule:

$$\frac{E \xrightarrow{a} E', \alpha : u \xrightarrow{b} \beta : u'}{E[\alpha : u] \xrightarrow{c} E'[\beta : u']}, P(E, a, \alpha, b, \beta, c),$$

where P is a continuous predicate. Continuous means that the value of this predicate depends only on some part of behavior tree in the environment state

E, which has a finite height (prefix of the tree E of finite height). Hereby, this rule refers to a head insertion. The rules for indecomposable environment states and for termination constants should be added to the main rule.

The next rule

$$\frac{E \xrightarrow{a} E', u \xrightarrow{b} \beta : u'}{E\,[u] \xrightarrow{c} E'[u']}, P(E, a, c),$$

is the particular case for the head insertion rule in combination with additivity and parallel insertion or commutativity requirements. Such rule will be named *permitted rule*. It could be interpreted by as follows: agent can execute the action a, and environment permits to execute this action. Predicate E for permitted rule will be named *permitted predicate*.

4 Constraint Programming

Constraint programming is a powerful paradigm for solving combinatorial search problems that draws on a wide range of techniques from artificial intelligence, computer science, databases, programming languages, and operations research. Constraint programming is currently applied with success to many domains, such as scheduling, planning, vehicle routing, configuration, networks, and bioinformatics [24].

The Constraint programming paradigm has some resemblance to traditional Operations Research (OR) approach, in that the general path to a solution is:

1. analyzing the problem to solve, in order to understand clearly which are its parts;
2. determining which conditions(relationships) hold among those parts: these relationships and conditions are key to the solving, for they will be used to model the problem;
3. stating such conditions(relationships) as equations; to achieve this step not only the right variables and relationships must be chosen: as we will see, Constraint programming usually offers a series of different constraint systems, some of which are better suited than others for a given task;
4. setting up these equations and solving them to produce a solution; this is usually transparent to the user, because the language itself has built-in solvers [25].

There are, however, notable differences with OR, mainly in the possibility of selecting different domains of constraints, and in the dynamic, generation of those constraints. This seamless combination of programming and equation solving accounts for some of the unique components of Constraint Programming:

– the use of sound mathematical methods: well-known and proved algorithms are provided as intrinsic, builtin components of Constraint programming languages and tools;
– the provision of means to perform programmed search, especially in Constraint programming (were search is implicit in language itself);

- the possibility of developing modular, hybrid models, when necessary: many Constraint programming systems offer different constraint systems, which can be combined to model the various parts of the problem using the tool more adequate for them;
- the flexibility provided by the programming language used, which allows the programmer to create the equations to be solved dynamically, possibly depending on the input data.

As with any other computational approach, all problems are amenable to be tackled with Constraint programming; notwithstanding, there are some types of problems which can be solved with comparatively little effort using Constraint programming based tools. Those applications share some general characteristics:

- No general, efficient algorithms exist (NP-completeness): specific techniques (heuristics) must be used. These are usually problems with a heavy combinatorial part, and enumerating solutions is often impractical altogether. A fast program using usual programming paradigms is often too hard and complicated to produce, and normally it is so tied to the particular problem that adapting it to a related problem is not easy.
- The problem specification has a dynamic component: it should be easy to change programs rapidly to adapt. This has points in common with the previous item: Constraint programming tools have builtin algorithms which have been tuned to show good behavior in a variety of scenarios, so updating the program to new conditions amounts to changing the setting up of the equations.
- Decision support required: either automatically in the program or in cooperation with the user. Many decisions can be encoded in mathematical formulae, which appear as rules and which are handled by the internal solvers, so (although, of course, not always) there is no need to program explicit decision trees [26].

Among the applications with these characteristics, the following may be cited: planning, scheduling, resource allocation, logistics, circuit design and verification, finite state machines, financial decision making, transportation, spatial databases, etc.

5 Insertion Machine for Constraint Programming

Some example of insertion machines and restrictions for insertion function are in [3, 9, 23]. In this section we try to show how to use insertion modelling for constraint programming [26]. The problems of constraint programming, where a main goal is behavior of the system, is the closest to the insertion modelling. For example, the problem of wolf-goat-cabbage [27] (A farmer wishes to transfer (by boat) a wolf, a goat, and a cabbage from the left bank of a river to the right bank. If left unsupervised, the wolf will eat the goat and the goat will eat

the cabbage, but nothing will happen as long as the farmer is near. Beside the farmer there is only a place for one item in the boat).

Let's consider a formalization of this problem in insertion modelling. Let E be the next enviroment:

$$
\begin{array}{l}
obj(\\
\quad constraints : rs(x, y, z)(\\
\quad\quad obj(Wolf : left, Goat : left, x, Ferryman : right) = 0, \\
\quad\quad obj(Wolf : right, Goat : right, x, Ferryman : left) = 0, \\
\quad\quad obj(x, Goat : left, Cabbage : left, Ferryman : right) = 0, \\
\quad\quad obj(x, Goat : right, Cabbage : right, Ferryman : left) = 0, \\
\quad\quad obj(Wolf : z, x, y, Ferryman : z) = 1, \\
\quad\quad obj(x, Goat : z, y, Ferryman : z) = 1, \\
\quad\quad obj(x, y, Cabbage : z, Ferryman : z) = 1 \\
\quad); \\
\quad initial : obj(\\
\quad\quad Wolf : left, \\
\quad\quad Goat : left, \\
\quad\quad Cabbage : left, \\
\quad\quad Ferryman : left \\
\quad) \\
)
\end{array}
$$

where *initial* is initial state where all creatures are in left bank of the river, *constraits* are constraint equations with right part of 0 define non possible cases (0 is the neutral element of non-deterministic choice + of insertion modeling) and with 1 if ferryman could transport thous creatures in that case. These both values are covered by two different states of ferry: 1 is just before ferry and 0 - after.

The coresponded input data could be defined in the following way:

$(ferry\ Wolf \parallel ferry\ Goat \parallel ferry\ Cabbage).assetion\_constraints$

So, the transition relation of the system is defined in fig. 6.

$$
\begin{array}{l}
\varphi\,[p] \xrightarrow{ferry\ x} \varphi_i\,[p]\,, constraints(\varphi) = 1 \\
\varphi\,[p] \xrightarrow{ferry\ x} 0, constraints(\varphi_i) = 0 \vee \neg constraints(\varphi) = 1 \\
\varphi\,[p] \xrightarrow{assetion\_constraints} \varphi\,[p]\,, \neg constraints(\varphi) = 0 \\
\varphi\,[p] \xrightarrow{assetion\_constraints} 0, constraints(\varphi) = 0
\end{array}
$$

Fig. 6. Relations of System's Transitions

where φ_i is a new environment state without acception of constration equation.

Insertion modelling system has found 1 goal trace - all creatures are in other coast and 10 visited traces - those traces cover all possible behaviors of such system.

Typically IMS generated trace is defined by user. It could look like sequance of actions or environment states etc. For this example, to simplify the view of the traces we propose to use one uninterpreted action *transport*:

$transport(\neg(c = \varphi_i.Ferryman), Ferryman, x)$,

where operation . returns state of the ferryman and $\neg$ returns other coast.

So, goal state trace has the next view:

$$
\begin{array}{l}
init \\
transport(right, Ferryman, She_goat) \\
transport(left, Ferryman, Nil) \\
transport(right, Ferryman, Wolf) \\
transport(left, Ferryman, She_goat) \\
transport(right, Ferryman, Cabbage) \\
transport(left, Ferryman, Nil) \\
transport(right, Ferryman, She_goat)
\end{array}
$$

Fig. 7. Example of Goal State Trace

where *init* is the initial state and Nil means that ferryman is ferried along.

In general case, insertion machine for constraint programming should use:

1. *assetion_constraints* agents action in agent behaviour and initial state.
2. Environment description should have non empty section constraints.

6 Conclusion

The main concepts of insertion modeling system has been considered in the present paper. The system was successfully used for the development of prototypes of the tools for industrial VRS (Verification of Requirement Specification) system and research projects in Glushkov Institute of Cybernetics. Now it is used for the development of program verification tool and 'verifiable programming' project, and for constraint programming. The system continues its enhancement and new features are added while developing new projects.

The far goal in the developing of IMS consists of getting of sufficiently rich cognitive architecture to its basis, which could be used in the artificial intelligence research.

References

1. D.R. Gilbert, A.A. Letichevsky: A universal interpreter for nondeterministic concurrent programming languages.In M. Gabbrielli (eds.), Fifth Compulog network area meeting on language design and semantic analysis methods (1996).
2. A. Letichevsky and D. Gilbert: A general theory of action languages.Cybernetics and System Analyses, vol. 1, 16–36 (1998).

3. A. Letichevsky and D. Gilbert: A Model for Interaction of Agents and Environments. In D. Bert, C. Choppy, P. Moses, (eds.). Recent Trends in Algebraic Development Techniques. LNCS, vol. 1827, pp.311–328. Springer (1999).
4. A. Letichevsky: Algebra of behavior transformations and its applications. In V.B.Kudryavtsev and I.G.Rosenberg (eds). Structural theory of Automata, Semigroups, and Universal Algebra, NATO Science Series II. Mathematics, Physics and Chemistry, vol. 207, pp. 241–272. Springer (2005).
5. S. Baranov, C. Jervis, V. Kotlyarov, A. Letichevsky, and T. Weigert: Leveraging UML to Deliver Correct Telecom Applications. In L. Lavagno, G. Martin, and B.Selic, (eds.). UML for Real: Design of Embedded Real-Time Systems. Kluwer Academic Publishers. Amsterdam (2003).
6. A. Letichevsky, J. Kapitonova, A. Letichevsky Jr., V. Volkov, S. Baranov, V.Kotlyarov, T. Weigert: Basic Protocols, Message Sequence Charts, and the Verification of Requirements Specifications. Computer Networks, vol. 47, 662–675 (2005).
7. J. Kapitonova, A. Letichevsky, V. Volkov, and T. Weigert: Validation of Embedded Systems. In R. Zurawski, (eds.). The Embedded Systems Handbook, CRC Press, Miami (2005).
8. A. Letichevsky, J. Kapitonova, V. Volkov, A. Letichevsky, jr., S. Baranov, V. Kotlyarov, and T. Weigert: System Specification with Basic Protocols. Cybernetics and System Analyses, vol. 4, 479–493 (2005).
9. A. Letichevsky, J. Kapitonova, V. Kotlyarov, A. Letichevsky Jr, N. Nikitchenko, V. Volkov, and T. Weigert: Insertion modeling in distributed system design. Problems of Programming, vol. 4, 13–39 (2008).
10. R. Milner: Communication and Concurrency. Prentice Hall (1989).
11. R. Milner: Communicating and Mobile Systems: the Pi Calculus. Cambridge University Press (1999).
12. C.A.R. Hoare: Communicating Sequential Processes. Prentice Hall (1985).
13. Cardelli, L. and A.D. Gordon: Mobile Ambients. In Foundations of Software Science and Computational Structures, Maurice Nivat (eds.), LNCS vol. 1378, pp. 140–155. Springer (1998).
14. Y. Gurevich: Evolving Algebras 1993: Lipari Guide. In E. Borger (eds.), Specification and Validation Methods, Oxford University Press, pp. 9–36 (1995).
15. C. A. R. Hoare and He Jifeng: Unifying Theories of Programming. Prentice Hall International Series in Computer Science (1998).
16. J. Meseguer: Conditional rewriting logic as a unified model of concurrency. Theoretical Computer Science, 73–155 (1992).
17. A. Letichevsky, J. Kapitonova, V. Volkov, V.Vyshemirsky, A. Letichevsky Jr: Insertion programming. Cybernetics and System Analyses, vol. 1, 19–32 (2003).
18. J.V. Kapitonova, A.A. Letichevsky, and S.V. Konozenko: Computations in APS. Theoretical Computer Science, 145–171 (1993)
19. Insertion Modeling System, http://apsystem.org.ua
20. Dexter Kozen David Harel and Jerzy Tiuryn: Dynamic Logic (2000).
21. C. A. R. Hoare: An axiomatic basis for computer programming. Communications of the ACM, vol. 12(10), 576–580 (1969).
22. R. W. Floyd: Assigning meanings to programs. Proceedings of the American Mathematical Society Symposia on Applied Mathematics, vol. 19, 19–31 (1967).
23. A.A. Letichevsky, J.V. Kapitonova, V.A. Volkov, V.V. Vyshemirsky: Insertion Modelling. Cybernetics ans System Anilises, vol. 1, 19–23 (2003).
24. F. Rossi, P. van Beek, T. Walsh: Handbook of Constraint Programming. Elsevier Science Inc., New York, NY, USA (2006).

25. R. Bartak: Tutorial on Filtering Techniques in Planning and Scheduling. The English Lake District, Cumbria, UK (2006).
26. K. Marriott, P. Stuckey: Programming with Constraints: An Introduction. MIT Press (1998).
27. Farmer, Wolf, Goat and Cabbage Problem, http://wiki.visual-prolog.com/index.php?title=Farmer,_Wolf,_Goat_and_Cabbage.

Is Your Ontology a Burden or a Gem? – Towards Xtreme Ontology Engineering

Olga Tatarintseva[1], Vadim Ermolayev[1], Anna Fensel[2, 3]

[1] Department of IT, Zaporozhye National University,
66 Zhukovskogo st., 69600 Zaporozhye, Ukraine
tatarintseva@znu.edu.ua, vadim@ermolayev.com
[2] FTW Forschungszentrum Telekommunikation Wien GmbH,
Donau-City-Straße 1, A-1220 Vienna, Austria
[3] STI Innsbruck, Technikerstr. 21a, A-6020 Innsbruck, Austria
anna.fensel@sti2.at

Abstract. One of the commonly acknowledged shortcomings of Semantic Technologies that prevents their wide adoption in industry is the lack of the commitment by the intended domain experts and users. This shortcoming becomes even more influential in the domains that change sporadically and require appropriate changes in the respective knowledge representations. This discussion paper argues that a more active involvement of the intended user community, comprising subject experts in the domain, may substantially ease gaining the required commitment of the critical mass of the domain users to the developed domain ontology. As a possible approach for building an instrumental platform for that, the paper suggests the use of the Semantic MediaWiki based collaboration infrastructure for maintaining and discussing ontology descriptions by the community of its intended users and developers. We also report how a prototypical ontology documentation wiki has been used for gaining the commitment of ontology users in the ACTIVE European project.

Keywords: ontology engineering methodology, stakeholder commitment, OntoDocWiki, xtreme ontology engineering.

1 Introduction

Building and refining practically useful knowledge representations in different domains is one of the major challenges in making semantic technologies publicly accepted today. The problem is not only in creating the proper encodings of the tacit knowledge of subject experts, or user behavior observations [1] but also in gaining a commitment to the developed ontologies by the users who are supposed to exploit these modules of formalized and explicit knowledge – either directly in their daily activities or by empowering their software tools. Evidently creating ontologies is not a routine task. It requires substantial intellectual effort.

Moreover, refining ontologies, making them better covering the intended requirements of the user community, is even more challenging and effort consuming. As well known from knowledge elicitation practice, *five* subject experts will most

definitely have *seven*[1] different opinions. So, the commitment of those individuals can only be reached if a knowledge representation is aligned and harmonized alone their subjective and tacit interpretations of the domain knowledge.

Gaining a commitment to the ontology by a wider group or a community of intended users is even more complex. One reason is that the majority of users have to adapt themselves not only to the suggested knowledge representation but also to the knowledge carried by this formal representation – which could both be novel to them. In our opinion the difficulty of gaining users' commitment is the major obstacle for a broader adoption of the semantic technologies in industries and the reason[2] for the criticism expressed to those technologies. The problem becomes even more challenging for the knowledge about the domains that change frequently. Ontologies describing those domains have to be changed accordingly. The changes in the knowledge representations have to be accepted by the subject experts and the users.

Hence, offering support for facilitating a better and less effort consuming comprehension, alignment and harmonization of knowledge representations by a user community may become a substantial step forward in reusing domain knowledge by knowledge workers and their software systems. Contemporary ontology engineering methodologies put insufficient emphasis on offering ways to gaining such a commitment. The analysis of this shortcoming is given in Section 3.

We believe that a more active involvement of the intended ontology users in the processes of ontology development and refinement is required for lowering their comprehension barriers. A software tool facilitating this active involvement will inevitably be a collaboration platform that allows discussing knowledge representations and expressing opinions and arguments by any community member.

Developing tools for collaborative knowledge engineering and knowledge reuse is one of the mainstreams in the semantic technologies community. However, the vast majority of the tools available today are tailored to the use by knowledge engineers, but not by domain experts or users. One interesting exception is the development of ontology games and collaborative (social) semantic mark-up tools for Web 2.0. Yet, these approaches yield too lightweight models – insufficiently expressive for the majority of industrial applications. The analysis of the state of the art in collaborative platforms for ontology engineering, onto-gaming and semantic mark-up on the Social Web is given in Section 4.

One possible solution for the outlined problem is making these divergent courses meet. Tool support for ontology engineering would benefit from adopting "croud-sourcing" features of collaborative knowledge representation development by Social Web users. A meeting point that will allow for the proper comprehension of knowledge structures is a collaborative platform for presenting and discussing the documentation of the ontologies by the subject experts and intended users alone the development process. There are several obstacles on this way. One is the lack of a proper incentive mechanism motivating subjects to take their active part. Another one lies in the nature of the work to be done – it is out of the scope of the core professional competence of intended audience. Yet more obstacles are caused by the lack of the tool support for: (i) the development and versioning of the ontology

[1] The numbers are indicative.

[2] … apart of the incurred computational overhead.

documentation in line with the evolution of the ontology; (ii) the discussion of ontology documentation as a representation of knowledge that is more easily comprehendible by the users than the code of the ontology. Last but not least is the need for a mechanism of reaching and spreading consensus among the participants. The requirements to the envisioned collaborative platform are presented in Section 5.

Our experience in developing and experimenting with a prototype collaboration platform for involving domain specialists in the active discussion of domain knowledge representations is presented in Section 6. The prototype platform implements some of the outlined requirements. It is based on the Semantic MediaWiki [2] with an extension for moderated discussions. The prototype platform has been used in ACTIVE Project (http://active-project.eu/) for representing and discussing the PSI Suite of Ontologies (http://isrg.kit.znu.edu.ua/ontodocwiki/) describing projects and processes in microelectronic engineering design.

2 Why is an Ontology Often a Burden?

Given the effort and intellect invested into the development of ontologies as consensual descriptive formal models of domains of discourse [3] in the last two decades, it could have been expected that ontologies had already become the core enablers for the ICT infrastructures and services in many industrial branches. However this is not entirely the case. "Unfortunately, the number and quality of actual, "non-toy" ontologies available on the Web today is remarkably low" [4]. Several technical limitations and practical challenges preventing easy adoption remain unsolved. These barriers for technology and methodology uptake *as perceived by industries*[3] (c.f. [4]) are as follows:

(i) **Unjustified benefits**. Industrial users and policy makers tend to think that using ontologies in their industrial setting is an artificial requirement to a large extent. They (sometimes wrongly) assume that the required information could be presented and processed in a more easy way using lighter-weight mark-up languages (like XML) and corresponding parsers.

(ii) **Considerable effort expectations**. Industries consider that even if the need for an ontology is justified, the effort required for building it is too large to be acceptable for the incurred benefits of use.

(iii) **Insufficient expressivity and comprehension gap**. Industrial technical specialists fear that the expressivity of ontology specification languages is insufficient for fully and adequately describing their subject domain. Consequently, it can not be granted that the intended users of the ontology easily grasp the meaning of the ontology elements as intended by the knowledge engineers who created the code.

(iv) **Computational overhead and poor scaling**. Software developers in industry estimate that ontology based software solutions are too heavy-weight. The software spends too much computation power for ontology processing and

[3] This information has been acquired from a several year experience of industrial partnerships in knowledge intensive research and development projects.

reasoning because of the complexity of the problems that are solved. As a result ontology-based applications scale quite poorly to be acceptable in industrial settings. More lightweight solutions are demanded.

(v) **Insufficient maturity of ontology engineering process.** If it is not their core competence, industrial engineers will seek for and accept a methodology that is well defined and based on the use of standardized working patterns. However, the leading experts in the field state that the development of ontologies is still much more a craft work or a non-trivial mental exercise than a rigorous and standardized engineering process. One of the particular shortcomings is that ontology engineering cycles are too long. The result is that we can not build ontologies that adequately follow the changes in quickly evolving domains.

Last but not least, and as a consequence of the combination of the outlined barriers, people in industries hardly believe that ontologies will solve their practical problems and help effectively in the development of their applications. Therefore it is often difficult to obtain their commitment to the ontologies offered to or developed for them.

It may be also noticed that the requirements the industries implicitly put up-front in (i–v) are not properly balanced and are sometimes clashing. For example a desire to have easily comprehendible lightweight knowledge representations is contrary to the demand of more expressive power for a more adequate representation of a domain. One good approach to resolve those clashes is to offer a layered representational structure with a more coarse and easy to grasp descriptions on the top down to fully detailed and formally coded knowledge representation modules in the bottom. As suggested in [5] those layers may be offered as different representation facets to different categories of specialists and at different development phases.

We do not intend to resolve all the fears of industrial experts in this paper. Our objective is to evaluate the existing ontology engineering approaches, development praxis, and methodologies by looking at how they facilitate better and broader commitment to the developed knowledge representation artifacts and, by that, relax some of the existing barriers.

3 Shortcomings of Ontology Engineering Methodologies

Knowledge Engineering as a subfield of Artificial Intelligence or broader – of Computer Science is a vibrant research discipline for already more than two decades. It involves integrating knowledge into computer systems. Knowledge has to be therefore represented in a way that a computer system is able to process. These representations are often elaborated as ontologies using the instruments provided by Ontology Engineering. This discipline develops knowledge representation frameworks to ensure adequate rigor for making the outputs tractable and processible by machines. Formal knowledge representation languages (e.g. OWL 2.0 [6]) are developed and standardized for that purpose.

Ontology Engineering is also concerned about the development of the methods and methodologies for building ontologies to fulfill the requirements of the intended user

audience. The results are sought to cover the user interpretations of the common sense or a target subject domain so much and completely as the expressivity of the formal representation allows. Several ontology engineering methodologies have been developed up to date. In particular METHONTOLOGY [7], DILIGENT [8], On-To-Knowledge [9], Uschold&King [10], Delphi [11], Compendium [12], HCOME [13], CommonKADS [14], NeOn [15] are the methodologies that are mentioned in the literature most frequently. A reader may be pointed to [16] for a more comprehensive list and analytical survey. Recently the methodologies taking into account the economical aspects of ontology engineering appear – e.g. OntoCOM [17].

Among those we are particularly interested in the methodologies which explicitly support: (i) collaborative ontology engineering; and (ii) ontology refinement process with evolving requirements. These methodologies are METHONTOLOGY, DILIGENT, On-To-Knowledge, Compendium and NeOn. In terms of ontology engineering lifecycle all the five methodologies suggest a variation of a process schematically pictured in Fig. 1, where the iterative parts are described in terms of either spiral or iterative waterfall process models.

The differences are:

(i) METHONTOLOGY distinguishes support and development activities and focuses on knowledge elicitation and result evaluation routine
(ii) DILIGENT focuses on distributed deployment and local changes and provides an argumentation framework for harmonization
(iii) On-to-Knowledge distinguishes ontology refinement as a separate important stage of the development process
(iv) NeOn offers a flexible scenario-based decision procedure for choosing the most appropriate lifecycle model and puts significant emphasis on the re-use of ontology patterns and available distributed ontologies
(v) Compendium explicitly concentrates on the ways of organizing collaborative work at knowledge elicitation and evolving prototyping phases

With respect to the relaxation of the barriers for gaining ontological commitment the contribution of all the mentioned methodologies is limited. NeOn suggests re-using good ontology engineering practices in the form of ontology patterns. Implicitly it suggests that using these good practices results in making the ontologies more correct and reliable – thus the commitment to these results is expected to be higher. Only OntoCOM elaborates the incentives for individuals and organizations for introducing ontologies. However it does not suggest mechanisms for gaining the commitment when the ontology is being developed. Compendium offers an approach to collaborative development based on moderated discussions and accounting for the evolution of the requirements. Unfortunately it does not mention the incorporation of the experts who carry the tacit knowledge about the subject domain that has to be elicited. None of the reviewed methodologies pays attention to the presentation of the developed knowledge representation in the form that is easily comprehendible by the intended users. The ontology documentation activity is considered only a support activity to the development process. None of the methodologies, except DILIGENT and NeOn to some extent provide the means for reaching consensus in ontology design decisions. DILIGENT does that by offering a harmonization framework. NeOn suggests consensual seeds in the form of reusable design patterns.

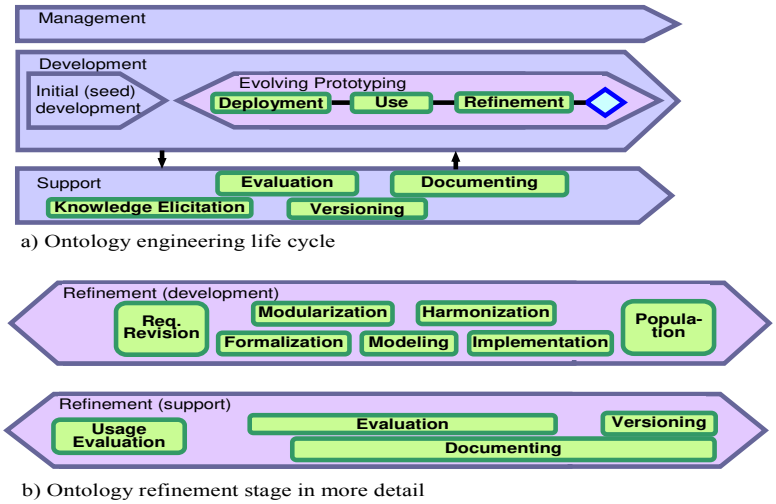


Fig. 1. Ontology engineering lifecycle specified in ISO/IEC 24744 graphical notation for describing methodologies [18].

4 Social Tagging and Games with a Purpose

One of the possible ways to check if the conceptualization of the domain is correct and complete is to evaluate the model against the outcome of users' grasp of the meaning of the content of the representative set of relevant documents. From the other hand, the labels or comments the users put on these documents or resources may be effectively used to infer the conceptualization. Such labels are often denoted as tags or annotations.

If web resources of different modalities are thought of as the representative set of data we find ourselves in the exploding field of collaborative or social tagging and annotation – a substantially characteristic part of the Web 2.0 phenomenon. A good survey of the field of social tagging is offered by Gupta et al in [19]. Tags created by the community of online users are exploited for different purposes. Taxonomy generation is one of the applications particularly relevant to our subject. The shortcomings of social tagging on Web 2.0 as analyzed in [19] as follows: (i) tags are simple bags of words without any more expressive semantics; (ii) tags are often not correct, especially if generated by spammers; (iii) tags are often ambiguous because different users apply terms to documents in different ways; tags are often sparse and do not cover the elements of the resource uniformly. Those shortcomings effect in low quality of tags. The reason is that the taggers do not use the terms of a consensual domain model in their activity.

Semantic annotation and tagging approaches further refine social tagging techniques by offering the collections of terms that are taken from such knowledge representations in the forms of taxonomies, folksonomies, thesauri. Please refer to

[20] for a comprehensive analyses of the requirements and results in the field. However, the backbone knowledge representations have to be obtained before semantic annotation may be undertaken. This remains the work for ontology engineers and is not regarded as a task for non-specialist users.

Hybrid approaches for collaborative tagging and annotation aimed at enrichment of the seed knowledge representations by the activity of the user community doing tags or annotations are also reported. For example [21] reports about the implementation of such a hybrid approach in Digital Libraries by tagging data through a combination of a standardized model, a harvesting protocol and a metadata mapping. It is concluded that both the custodians and users of digital repositories enabled with this collaborative annotation system benefit from the potential value of collaborative tagging without a need for a prior knowledge of the backend annotation systems. The weaknesses of traditional social tagging systems are attempted to be overcome by combining the best features of the Social and Semantic Webs. Unfortunately the problem of motivating users actively taking part in annotating resources remains open even in the reported advanced collaborative semantic tagging systems.

As already mentioned above, good ontologies should match consensual interpretations of domain knowledge by the representative set of domain users as closely and completely as possible. Therefore, ontology engineering in any form is by its very nature a process that has to involve as many domain experts and intended users as affordable. Increasing numbers of people are willing to spend their time using Web 2.0 applications, for example in adding tags and sharing their mark-ups in a group or community. Remarkably, it is totally on the contrary for ontology development – nobody reported about involving big user groups in creating knowledge representations so far. As recognized by many experts in Semantic Technologies one of the possible reasons is that traditional ontology engineering methodologies detach the effort from the benefits (c.f. [22]) hence de-motivating the involvement of those people whose interpretations of the domain are critically required.

One of the promising approaches for motivating more people take part in creating or refining ontologies is offering social software or, alternatively, a game with a purpose to a group of intended users. Several results in this direction are reported in the literature. Ontology creation can be implicitly embedded in social software, namely, social networking portals where users would be creating, evolving and confirming ontology items implicitly in the background, while simply providing information to a the portal for sharing content and communication with other users [23]. Games with a purpose is another approach that has been used mainly for collaborative tagging of resources having different modalities: images [24], music [25] – to mention just few. Gaming approach has also been tried for inferring human intentions from their recorded actions (Common Consensus game, [26]) and for evaluating how well commonsense facts fit to the interpretations of random users (FACTory Game by Cycorp, http://game.cyc.com/). For involving users in creating domain knowledge representations several game scenarios have been developed [22] for ontology building and refinement, ontology matching, annotating content using lightweight ontologies. Even though ontology backed up social networking portals and games with a purpose scenarios differ in relation to the users' motivation to contribute, they are in compliance to the OntoElect approach proposed in the next

Section. Both approaches offer possibilities to identify whenever users start to agree on and share certain ontological items

The positive features that are common to all these gems of related work are: (i) the pattern of user involvement adopted from Web 2.0 is used to motivate people taking part; (ii) all games with a purpose hide their purpose under the gaming scenario – offering fun in reward for providing useful results; (iii) the scenarios are designed in a way that assists in structuring the pool of players by their reputation.

There is also the shortcoming which is inherent to this approach – the knowledge representations or mark-ups that are crafted by non-specialist players can only be lightweight. Otherwise the overhead for ramping-up the players would consume all the offered incentives. Though ontology fragments are aligned with consensual user interpretations, it would be hard to ensure that the quality of those fragments is sufficient for, say, industrial use. Therefore, a joint motivated involvement of ontology engineering professionals, subject experts, and a sufficiently big group of intended users with domain knowledge and expertise is required.

5 OntoElect Approach for Xtreme Ontology Engineering

Charles Petrie in his editorial article [27] argued that the correctness of "… semantics, is evident in its use …". Ontologies are often denoted as descriptive theories that specify domain semantics – so they may only be validated by the users in their daily work in that domain. Martin Hepp in [4] backs up this view by stating that "… commitment can be achieved only by successful joint action – that is, successful usage of the ontology". Emphasizing the role of user commitment he observes that the perceived utility of the ontology grows with the number of users who commit to it. Hence, an effective ontology engineering methodology has to offer a mechanism for gaining the commitment by the intended group of users – the sooner – the better and as broadly as possible. We believe that a correct way to go is to involve the subject experts and the intended users in the development of the ontology at the earliest phase possible.

Our proposal of a possibly effective approach for attracting subject experts to play a more active role in the development and ownership of ontologies is inspired by the observations of social and political life – in *public election campaigns*. Indeed, the desired outcome of an election campaign for every candidate is to gain as much commitment of the electorate as possible. Such a commitment is measured adequately by the number of votes. The candidates compete for the votes by presenting their programmes, making coalitions, taking part in public debates – proving that his or her programme is the best match to the expectations of the majority of the electorate.

In the case of ontology engineering alternative ontology seeds for the same subject domain could be treated as election candidates. Each candidate ontology offered in a, so to say, *ontology election campaign* may be evaluated compared to the other candidates by the ability to answer the competency questions of the electorate. The more competence in answering the requirements of the electorate is demonstrated by the ontology, the more commitment it is potentially able to gain. The members of the electorate are the intended users in the domain. Their commitment could be measured

in simple votes or using a more sophisticated scoring mechanism. A good example of such a mechanism is evaluating submissions in a peer review process. Candidate ontologies may be presented by their election committees composed of the knowledge engineers and subject experts who took part in the initial development of the artifact. The presentations could be compared by the electorate like it happens in politics to the political programs. The candidate ontologies may be invited and take part in the public debates. Their competences will thus be cross-checked by the members of the competing election committees. The results of the debates will provide more evidence to the electorate for making their informed votes. It may happen that none of the candidate ontologies receives the majority in the campaign. Such an outcome in politics may be treated as an event leading to a one more round of elections. In ontology engineering a new election round may also be used constructively for the refinement of the candidates. The development teams may make use of the election results by concentrating on answering the competency questions they failed to answer by the previous revision. Coalitions may also be fruitful if it turns out that merging some candidate ontologies will substantially improve their joint competency. The election committees of the merged ontologies of course have to reach an agreement on how their "societal influence" will be redistributed after their victory in elections. Following the outlined procedure for ontology election, if a particular ontology gets the majority then it can be expected that the electorate's commitment to this ontology is strong enough to ensure its smooth uptake. Hence, a high level research hypothesis in our proposal is:

> *Introducing "democracy" in ontology engineering by incorporating a competitive and transparent procedure of ontology elections makes the process effective, development cycles shorter, and enables better results transfer to industry. The effectiveness is ensured by the fact that the developed artifacts will be appropriately refined following the intended requirements of the users in iterations (election rounds) with active participation of these intended users. The incentive for the active involvement of the domain specialists and knowledge engineers is reciprocal as they share common objectives. Moreover, both parties are naturally motivated by the competitive nature of the process. The iterations become shorter and pursue better defined and more focused objectives that reflect the desires of the domain specialists adequately. Hence, a better transfer is ensured by the fact that the commitment of the intended users to the winner is the strongest among the alternative candidates.*

There are several research questions that have to be answered in more detail and rigor for proving this research hypothesis:

(i) Why will industrial domain specialists be willing to join election committees and vote in election rounds?

This research question is very similar in its nature to the question about the proper motivation for people to join political groups and vote in elections. The answer may be sought by devising appropriate reciprocal incentive schemes motivating industrial domain specialists and knowledge engineers. For that looking at the results in several European projects may be useful. FP7 ACTIVE project derives the recommendation on possible incentive schemes, in particular for ontology development, by looking at the teams of knowledge workers as social structures. The high level objective of the

FP7 INSEMTIVES project (http://www.insemtives.org/) is to bridge the gap between human and computational intelligence and providing incentives for users to contribute to the massive creation of semantic content. A general framework for organizing campaigns may be adapted from the EU infrastructure project SEALS (http://www.seals-project.eu/) which develops a reference infrastructure, the SEALS Platform, to facilitate the formal evaluation of semantic technologies.

(ii) What are the proper ontology representation notation and collaboration platform for transparent election debates?

The first part of this research question is about the proper balance between the expressivity of the notation for representing ontologies and the ease of comprehension of this notation for the users that are not knowledge engineers. The expressivity has to be equal to the tractable subset of the chosen ontology specification language (for example OWL 2.0 – a de-facto standard ontology representation language to date). For ensuring the ease of comprehension we have to take into account that the representatives of the electorate are industrial knowledge workers – the engineers who develop, adapt, or adopt IT solutions in their businesses. A UML-based language is one of the commonly used notations for these professionals. Therefore, a visualized ontology specification in UML or its extension (for example OntoUML [28], or other UML variants with appropriate expressiveness [29]) accompanied by the textual description of ontology elements in a natural language would be appropriate – please refer to Section 6 for more details.

The second part of the question is about the collaboration platform that enables efficient debates. We consider that Semantic MediaWiki (SMW) with extensions is very appropriate as such a platform. An argument in favor of a Wiki-based infrastructure is that it is a Web 2.0 platform that intrinsically supports the exchange of opinions and has been extensively used for collaborative content development, "crowd-sourcing", user community development, etc.

(iii) How to ensure the swift convergence of the series of election rounds to the appearance of a single good scorer ontology?

The answer to this research question has to be sought by looking for the proper set of heuristic rules and policies for ontology elections. It could be rightfully argued that such a statement is a way too succinct to answer the question. However we do not have a more detailed recommendation at the moment and leave this very important issue for future research and experiments.

6 Xtreme Documenting and Ontology Discussions On-Line

One of the hypotheses we pointed out is: the ontology developed with active involvement of the intended users has to be presented in a form that is easily comprehendible by these users. A rich self explanatory notation with a user interface that is native for the target user group has to be exploited for that. We believe that a proper way of presenting the ontology to the subject experts is the documentation of the ontology as it combines the formal definitions of the ontology elements in textual and graphical representation with the informal descriptions of the semantics of those elements. Of course the documentation has to be developed in line with the ontology

design to be available in proper time for discussing the design decisions. Ideally, the documentation describing a concept, a property, an ontological module has to appear at the same time with the appearance of the design of this ontology element. Apart of that, the documentation has to contain the information about the ontological context of the element and the information about its evolution.

As a proof of concept we have developed the electronic documentation site for ontologies – OntoDocWiki (http://isrg.kit.znu.edu.ua/ontodocwiki/). This resource is based on the SMW with LiquidTreads extension (www.mediawiki.org/wiki/Extension:LiquidThreads) as a basic collaboration infrastructure. Currently the resource contains the documentation of the PSI Suite of Ontologies v.2.3. This Suite of Ontologies has been developed in the Performance Simulation Initiative (PSI) project[4] and further refined for the needs of the ACTIVE Project. The wiki articles represent documentation for the Suite of Ontologies (Fig. 2), each individual ontology module (Fig. 3), each individual concept (Fig. 4) have been semi-automatically generated [30] based on the reference specification [31].

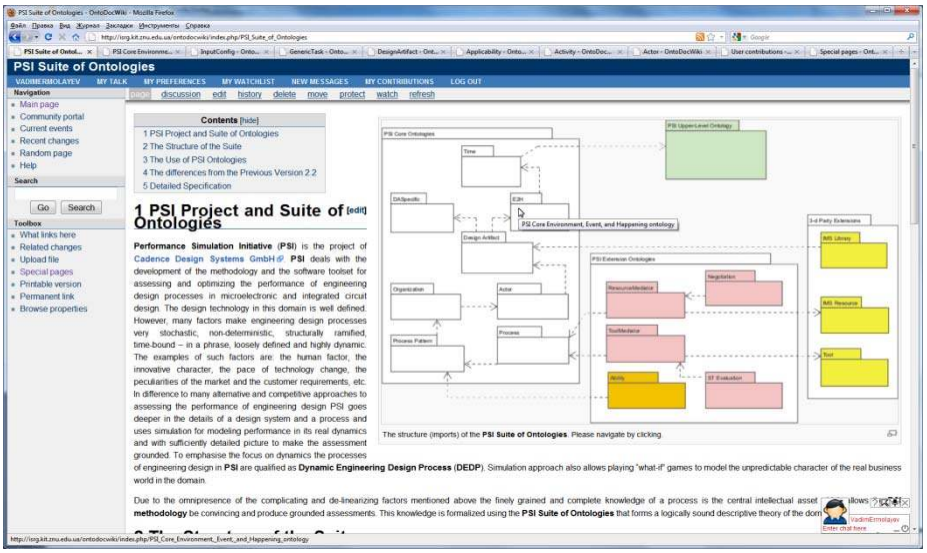


Fig. 2. OntoDocWiki article describing the PSI Suite of Ontologies v.2.3.

As pictured in Fig. 2–4, the articles comply with all the outlined requirements for describing ontology elements in an easily comprehendible and non-professional user friendly, yet informative manner. They combine textual descriptions with the graphical (UML) representation of the ontological contexts. These graphical representations are implemented as image maps and allow easy navigation in the pictured ontological contexts. For example, as pictured in Fig. 2, clicking the package representing the E2H ontology in the package structure diagram opens the article describing the E2H ontology (Fig. 3). Similarly, the class diagram of the particular ontology allows navigation to individual concept articles (e.g. the concept of a

[4] PSI is the accomplished R&D project of Cadence Design Systems GmbH.

Happening, Fig. 4) or related ontological modules represented in the ontological context.

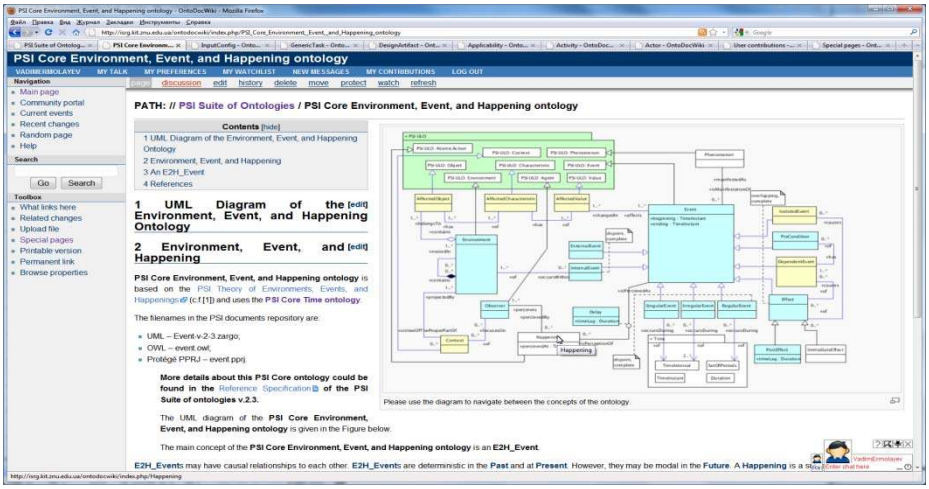


Fig. 3. OntoDocWiki article describing the PSI Environment, Event, and Happening (E2H) ontology v.2.3.

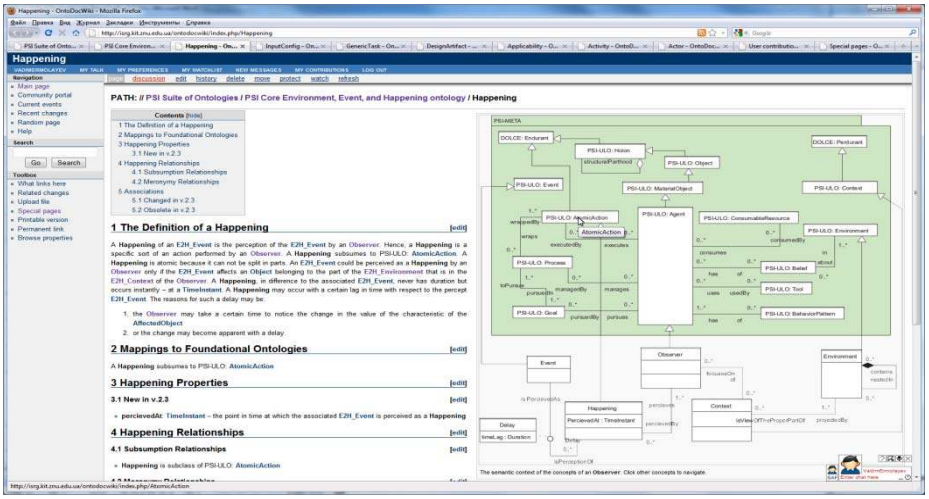


Fig. 4. OntoDocWiki article describing the concept of a Happening of the PSI E2H ontology v.2.3.

At a concept level the documentation is informative enough for allowing a non-professional user to evaluate the semantics of the concept and the surrounding ontological context [29]. As shown in Fig. 4, the article documenting a concept contains the information about:

– The relationship of the concept to the higher-level or foundational ontologies. For example, the concept of a Happening is described as a subclass of an AtomicAction – the concept of the PSI Upper-Level ontology.

– The explanation of the semantics of the datatype properties.
– The object properties grouped by the type of relationship: subsumption (if a concept is a subclass of another concept), part-whole relationships (represented as aggregations or compositions in the UML class diagram), and associations.

The descriptions of individual properties are structured in a way to present the evolution of those properties. For the properties that have been changed in the current version the information about the change is given. For those properties that have been introduced or became obsolete the rationale for this design decision is described. For the properties that may be used differently depending on the deployment of the Suite the variants of use are explained. For example, in some applications, like ProjectNavigator [32] the Suite of PSI Ontologies may be used without its Upper-Level ontology.

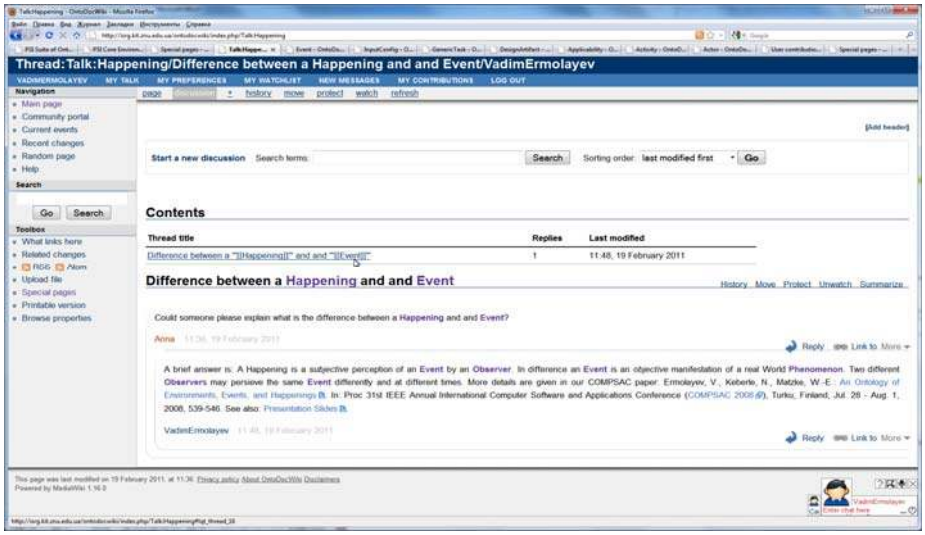


Fig. 5. OntoDocWiki discussion page for the concept of a Happening.

In ACTIVE project the PSI Suite of Ontologies has been used as the background knowledge for describing the subject domain of one of its case studies – knowledge processes in microelectronic engineering design. The fully functional prototype of the ACTIVE Design Project Visualizer [33] has been developed and validated based on this backbone knowledge representation. Prototype development has been done by several project partners. Only one of the partners was the owner of the PSI Suite – the rest were not familiar with these ontologies. Therefore we had to develop an ontology documentation and discussion resource for ramping-up the collegues. It turned out that OntoDocWiki became very helpful in both: explaining the PSI ontologies to the software developers, some with a marginal background in knowledge engineering; and collaboratively refining the ontologies in response to the project requirements. In fact the v.2.3 release of the Suite has been developed in this collaboration based on the extensive use of the OntoDocWiki platform.

Fig. 6. The statistics of use of the OntoDocWiki resource provided by Google analytics.

One of the features that proved to be useful and effective in ramping-up the users was discussions about ontology elements. Fig. 5 pictures an example of such a discussion about the semantics of the concept of a Happening. The utility of a discussion is straightforward – anyone may pose a question or present an argument about the semantics of an element in the ontological context of the related wiki article; anyone else may offer an answer or a counter-argument. The outcome of the discussion is often a better and deeper comprehension of the semantics. It also turned out some times that the discussions led to the changes in the ontology.

Apart of the more active involvement of the users in the ontology development process, the use of the Wiki as a platform allowed making the arguments or the statements in discussions more grounded by linking existing articles to the parts of the discussion statements. For example, the links to the items describing the concepts that are related for the discussion thread were inserted in the headings and the discussion statements using MediaWiki markups for hyperlinks. Those were either internal OntoDocWiki articles (e.g. Event) or the pointers to the external resources (for example [34]) offered as back-up information in support of the discussion statements.

Fig. 6 shows some of the usage statistics for the OntoDocWiki. It is topical to notice that the peaks in access hits (one was on the 12-th of October 2010) depict the activity of users exactly in the periods of ontology discussions that occurred in different phases of the prototype development. It is also interesting that the most frequently visited OntoDocWiki pages were the ones describing the ontology concepts that were the most complex in semantics compared to the others. So, it took people more time and more visits to comprehend these ontology elements. Such information on the intensiveness of the use of different parts of ontology documentation turns out to be extremely important for the knowledge engineering team. Indeed, it objectively measures the complexity of the comprehension of the ontological concepts that could become a problem for gaining the ontological commitment by the community of the intended ontology users. These measures may be valuable in adjusting the tactics and foci in the ontology election campaigns.

7 Concluding Remarks

Ontology engineering as a field has been in intensive research and development for a substantial period of time. As we have hopefully shown in the paper, ontology engineering technology has passed the peak of inflated expectations on the Gartner's hype cycle curve (see e.g. http://en.wikipedia.org/wiki/Hype_cycle). Currently the position is in the proximity of the through of disillusionment, probably a little bit shifted to the slope of enlightenment. Most evidently, semantic technologies in broad and ontology engineering technologies in particular, will be better accepted by industries if and when they reach Gartner's plateau of productivity. Our intension while writing this discussion paper was to analyze the reasons for the disillusionment and, perhaps, to shed the light on the possible way up the slope to the region of industrial maturity. We believe that the right way is marked by the increase in the commitment of the intended industrial users.

We have outlined our views on a possible methodological framework for ontology engineering – OntoElect. We believe that our approach may be capable to relax several barriers on the way of gaining better and broader commitment to the use of ontologies in industrial applications. These beliefs are backed-up by our experience in implementing and using some of the elements of this framework in our research and development work in several European projects. We had positive experience in testing parts of the approach. For example, our work on PRODUKTIV+[5] ontologies happened to be the informal competition of the two separate groups which views further converged to a single (merged) ontology suite – in debates. Another example that highlights the necessity of involving industrial subject experts and users as early as possible was our work on the PSI Suite of Ontologies together with the experts and users from Cadence Design Systems GmbH. The use case for testing the user friendly way for representing ontologies on a collaborative platform was the one in the ACTIVE project – presented in Section 6.

Acknowledgements

PSI Suite of Ontologies has been developed in the PSI project funded by Cadence Design Systems GmbH. OntoDocWiki development has been partially supported by the ACTIVE project funded in part by the European Commission Framework Programme 7. The authors would also like to thank the anonymous reviewers for their thorough comments that helped substantially to improve the paper.

References

1. Gruber, T. R.: Toward Principles for the Design of Ontologies Used for Knowledge Sharing. Int. J Human-Computer Studies 43, 907--928

[5] PRODUKTIV+ is the accomplished R&D project funded by the German Bundesministerium für Bildung und Forschung (BMBF).

2. Krötzsch, M., Vrandecic, D., Völkel, M., Haller, H., Studer, R.: Semantic Wikipedia. J of Web Semantics 5(4), 251--261 (2007)
3. Uschold, M., Grüninger, M.: Ontologies: Principles, Methods, and Applications. Knowledge Eng. Rev. 11(2), 93--155 (1996)
4. Hepp, M.: Possible Ontologies: How Reality Constrains the Development of Relevant Ontologies. IEEE Internet Computing 11(1), 90--96 (2007)
5. Guizzardi, G.: Theoretical foundations and engineering tools for building ontologies as reference conceptual models. Semantic Web 1-2(1), 3--10 (2010)
6. OWL 2 Web Ontology Language Primer, http://www.w3.org/TR/owl2-primer
7. Gómez-Pérez, A., Fernández-López, M., Corcho, O.: Ontological Engineering. Springer, London (2004)
8. Pinto, H.S., Tempich, C., Staab, S., Sure, Y.: DILIGENT: Towards a fine-grained methodology for distributed, loosely-controlled and evolving engineering of ontologies. In: de Mántaras, R.L., Saitta, L. (eds.) 16th European Conf. on Artificial Intelligence. ECAI, pp. 393--397, IOS Press (2004)
9. Sure, Y., Studer, R.: On-To-Knowledge methodology: On-To-Knowledge: Semantic Web enabled Knowledge Management. In: Davies, J. (eds.), J. Wiley and Sons (2002)
10. Uschold, M., King, M.: Towards a Methodology for Building Ontologies. In: Skuce, D. (eds.) IJCAI'95 Workshop on Basic Ontological Issues in Knowledge Sharing, pp.6.1--6.10. Montreal, Canada (1995)
11. Holsapple, C.W., Joshi, K.D.: A collaborative approach to ontology design. Comm. ACM, vol. 45, pp. 42--47. ACM Press, New York (2002)
12. Buckingham-Shum, S., Motta, E., Domingue, J.: Augmenting design deliberation with compendium: The case of collaborative ontology design. In: HypACoM 2002: Facilitating Hypertext-Augmented Collaborative Modeling. ACM Hypertext'02 Workshop, Maryland University (2002)
13. Kotis, K., Vouros, G.A., Alonso, J.P.: HCOME: tool-supported methodology for collaboratively devising living ontologies. In: SWDB'04: Second International Workshop on Semantic Web and Databases, Co-located with VLDB. Springer-Verlag (2004)
14. Schreiber, G. et al.: Knowledge Engineering and Management. The CommonKADS Methodology. MIT Press, Cambridge, USA (1999)
15. Suárez-Figueroa, M. C. et al. D5.4.1: NeOn Methodology for Building Contextualized Ontology Networks. Technical report, The NeOn Project (2008)
16. Pâslaru-Bontaş, E.: A Contextual Approach to Ontology Reuse: Methodology, Methods and Tools for the Semantic Web. PhD Thesis, Freie Universität Berlin, Berlin (2007)
17. Simperl, E., Tempich, C., Sure, Y.: A Cost Estimation Model for Ontology Engineering. In: Cruz, I. F., et al. (eds.) ISWC 2006. LNCS, vol. 4273, pp. 625--639. Springer, Berlin, Heidelberg (2006)
18. Henderson-Sellers, B., Gonzalez-Perez, C.: Standardizing Methodology Metamodelling and Notation: An ISO Exemplar. In: Kaschek, R., Kop, C., Steinberger, C., Fliedl, G. (eds.) UNISCON 2008. LNBIP, vol. 5, pp. 1--12. Springer, Berlin, Heidelberg (2008)
19. Gupta, M., Li, R., Yin, Z., Han J.: Survey on Social Tagging Techniques. SIGKDD Explorations 12(1), 58--72 (2010)
20. Uren, V., Cimiano, P., Iria, J., Handschuh, S., Vargas-Vera, M., Motta, E., Ciravegna, F.: Semantic annotation for knowledge management: Requirements and a survey of the state of the art. Science. Services and Agents on the World Wide Web 4(1), 14--28 (2006)
21. Hunter, J., Khan, I., Gewrber, A.: HarvANA – Harvesting Community Tags to Enrich Collection Metadata. In: Paepcke A, Borbiha J, Naaman M (eds.) 8th ACM/IEEE-CS Joint Conference on Digital Libraries, pp. 147--156. ACM New York, New York (2008)

22. Siorpaes, K., Hepp, M.: Games with a Purpose for the Semantic Web. IEEE Intelligent Systems 23(3), 50--60 (2008)
23. Zhdanova, A.V.: Community-driven Ontology Construction in Social Networking Portals , Web Intelli. and Agent Sys. 6, 1, 93--121 (2008)
24. Von Ahn, L.: Games with a Purpose. Computer 29(6), 92–94 (2006)
25. Law, E. et al.: Tagatune. In: Proc. Int'l Conf. Music Information Retrieval, pp. 361--364, Austrian Computer Soc., Ismir (2007)
26. Lieberman, H., Smith, D., Teeters, A.: Common Consensus: A Web-Based Game or Collecting Commonsense Goals. In: Proc. Workshop Common Sense for Intelligent Interfaces, ACM Conf. Intelligent User Interfaces. ACM Press (2007)
27. Petrie, C.: Pragmatic Semantic Unification. IEEE Internet Computing, vol. 9, no. 5, pp. 95--96 (2005)
28. Guizzardi, G.: Ontological Foundations for Structural Conceptual Models. Telematica Instituut Fundamental Research Series No. 15, Netherlands (2005)
29. Ermolayev, V., Copylov, A., Keberle, N., Jentzsch, E., Matzke, W-E.: Using Contexts in Ontology Structural Change Analysis. In: Ermolayev, V., Gomez-Perez, J.-M., Haase, P., Warren, P. (eds.) CIAO, CEUR-WS/vol. 626 (2010)
30. Ermolayev, V., Tatarintseva, O.: Applied Research and Development in Cooperation with Industry. Information Technologies in Education 5, 16--26 (2010)
31. Ermolayev, V., Jentzsch, E., Keberle, N.: Performance Simulation Initiative. The Suite of Ontologies v.2.3. Reference Specification. Technical Report, PSI-ONTO-TR-2010-1, VCAD EMEA, Cadence Design Systems, GmbH (2010)
32. Sohnius, R., Jentzsch, E., Matzke, W.: Holonic simulation of a design system for performance analysis. In: HoloMAS '07: Proc. of the 3rd international conference on Industrial Applications of Holonic and Multi-Agent Systems, pp. 447--454. Springer-Verlag, Berlin, Heidelberg (2007)
33. Ermolayev, V., Dengler, F., Fortuna, C., Stainer, T., Bosser, T., Melchior, E-M.: Increasing predictability and sharing tacit knowledge in electronic design. In: Warren, P., Simperl, E., Davies, J. (eds.) Context and semantics in knowledge management, Springer Verlag, Berlin, Heidelberg (2011)
34. Ermolayev, V., Keberle, N., Matzke, W.-E.: An Ontology of Environments, Events, and Happenings. In: Proc 31st IEEE Annual International Computer Software and Applications Conference, pp. 539--546. Turku, Finland (2008)

Simulation of the Enhanced Version of Prisoner's Dilemma Game

Tatyana Vlasova, Maryna Vladymyrova and Dmitry Shabanov

V.N. Karazin Kharkiv National University, 4, Svobody Sqr., Kharkiv, 61077, Ukraine
vlasova.tatyana@bk.ru, vladymyrova@gmail.com, d.a.shabanov@gmail.com

Abstract. This paper presents the model and software to explore pair interactions of objects with different behavior and their patterns. The research is based on the enhanced version of a classic prisoner's dilemma game. The non-cooperative finite and infinite pair games with non-zero sums are investigated. Pure and mixed strategies with finite and infinite memory developed by Biology School of V. N. Karazin Kharkiv National University are used to analyze the results.

1 Introduction

This paper presents the model and software to explore pair interactions of biological objects with specified behavior. The research was carried out on request of the herpetology department of V. N. Karazin Kharkiv National University and its results are used in educational process to illustrate some topics of the ecology discipline as well as in biology students' research for studying patterns of pair behavior of some biological species. Our model is based on the classic prisoner's dilemma game [1,2,3] as it is widely accepted as the model to study the pair interactions and behavior of different agents from interactions of animals in nature up to economical transactions in human world [4, 5, 6, 7, 8]. The main goal of the game in its classic version is to get maximal score doing preset number of steps with given values of the fine, the cooperation bonus and the cooperation award. Each participant can remember not more than two its own and the opponent's previous actions.

In distinction to classic case we enhance the rules of the game by allowing more complex behavior of agents depending on their ability to remember their own or opponent's previous steps and on values of fines, bonuses and awards.

The important factor which influences the evolution of living organisms including humans is their communications with environment. It is only typical for such communications to have conflicts of interest between opponents, absence of information about future actions of an opponent and need to foresee its future actions only on the base of the prehistory of similar interactions. The paradox of the game clearly shows the contradiction between group interests and individual ones: what is optimal for the group of two is not good for each member of the group. In fact the same is true for multilevel systems with optimization functions on different levels: their behavior is intuitively unpredictable and even paradoxical.

It is absolutely obvious that the outcome of the game fully depends on the participants' strategies. Here we define strategy in a slightly different way than it is done in the game theory.

Since the objective of this research is to explore the pair interactions in real biological environment it is natural to define strategy as a set of rules used by a participant to make its next step. Because of the infinite variety of strategies for different biological objects we use simulation with strategies constructed by the experimenter. The developed software allows experimenter to set different strategies as input information for simulation. The main goal of experimenter is to find optimal strategy and the value of the game in their classical sense.

2 Model and simulation description

We consider the pair non-cooperative finite and infinite games with a non-zero sum [11; 12; 14] so the general result can be both positive (in the case when participants cooperate during the whole game) and negative. We take the game in its normal form $\Gamma_N =< N; X_1,...,X_i,...,X_n; K_1,...,K_i,...,K_n >$, where Γ_N is the notion of the game, $N = \{1,...,i,...,n\}$ is a set of participants, $X_i = \{x_i\}$ is a set of strategies for the *i*-th participant and $K_i(x_1,...,x_i,...,x_n)$ is the gain function for the *i*-th participant, the value of which is the gain obtained by the *i*-th participant if participants use strategies $(x_1,...,x_i,...,x_n)$. In our model we allow participants to use pure or mixed strategies. Pure strategy is just definite sequence of steps i. e. it can be represented by any element $x_i \in X_i$. Mixed strategy can be a simple set of pure strategies or a set of pure strategies with given probabilities distribution.

Any strategy as a set of rules depends on the participant's memory depth or its type of memory. Here we suppose that each participant remembers its previous actions or the previous actions of the opponent. If the participant has zero memory depth it can make predefined actions during the whole game or make spontaneous actions. If the participant has finite memory depth it uses pure strategy depending on the actions of its opponent. If the participant has infinite memory depth only the absolute value of its current gain influences its actions.

The pure strategy is a response of the participant to the actions of its opponent. We consider the following cases:

No response (leads to spontaneous actions)

The response to the definite set of actions

The response to the small value of the own gain or to the big value of the opponent's gain

The response to the big value of the own losses or to the small value of the opponent's losses

In this case we achieve the goal of maximization of the participant's own gain.

Similarly when the participant remembers only its opponent's history of actions and responds to them we achieve the goal of minimization of the opponent's gain.

We use finite automation with two states which are "accept" and "reject" to model the game [8, 12]. The input information for the automation is as follows: participants'

strategies, values of the gain function, transition rules for a participant's behavior, and initial conditions. Transition rules depend on the type of a participant's strategy which can be pure or mixed. Simulation repeats the preset number of times or till the winning of one of its participants.

There exist four scenarios of simulation each corresponding to four different goals.

First scenario is the experiment between the two chosen strategies ("pure-pure", "mixed-pure", "mixed-mixed"), when the values of gain/bonus/fine (M, L, and K, respectively) are fixed. The simulation results are presented by the graph, where x-axis denotes the sequence of steps and y-axis denotes the quantitative characteristics of the gain/loss (Fig.1).

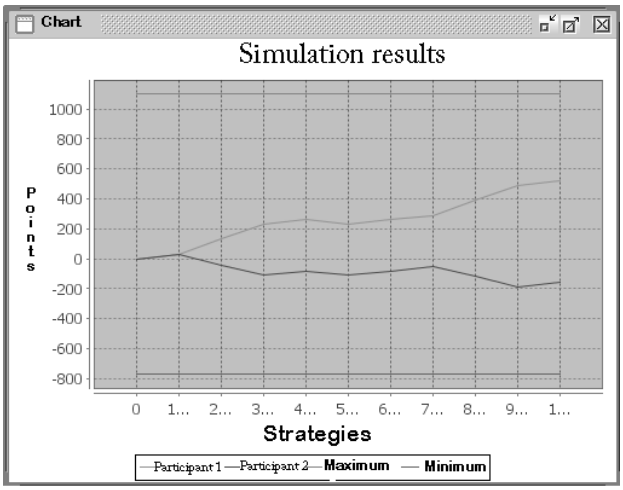


Fig. 1. Simulation results with two strategies (one is random).

The result data of this scenario show that among the strategies with zero and finite memory, the strategy of random steps and zero memory wins in most cases. Among the strategies with finite and infinite memory the finite memory strategy wins in most cases. Analogous conclusions were made previously by R. Dockins [6] when he analyzed the classical version of the game.

Analyzing the obtained results of pair interaction between the pure and mixed strategies we may conclude that the mixed strategy is more advantageous than any of pure strategies. This can be explained by the flexible behavior of the participant with the mixed strategy.

The second scenario is the evaluation of competition between the one fixed strategy and the variety of others. The results of the experiment are presented by the bar chart displaying the number of win points over each of the chosen strategies.

Besides the graphical visualization one can look through the steps history (absolute values of gains or losses) of each of the simulation participants.

The third scenario allows experimenter to approve or disprove the hypothesis that the strategy of a participant depends on the gain function. In order to do it we implemented the feature which allows experimenter to conduct the simulation with fluctuating parameters M, L, K. In this case the results are presented by the graph and

the table showing the data of each participant's results for chosen strategies with respect to varying values of K, L, and M (Fig.2).

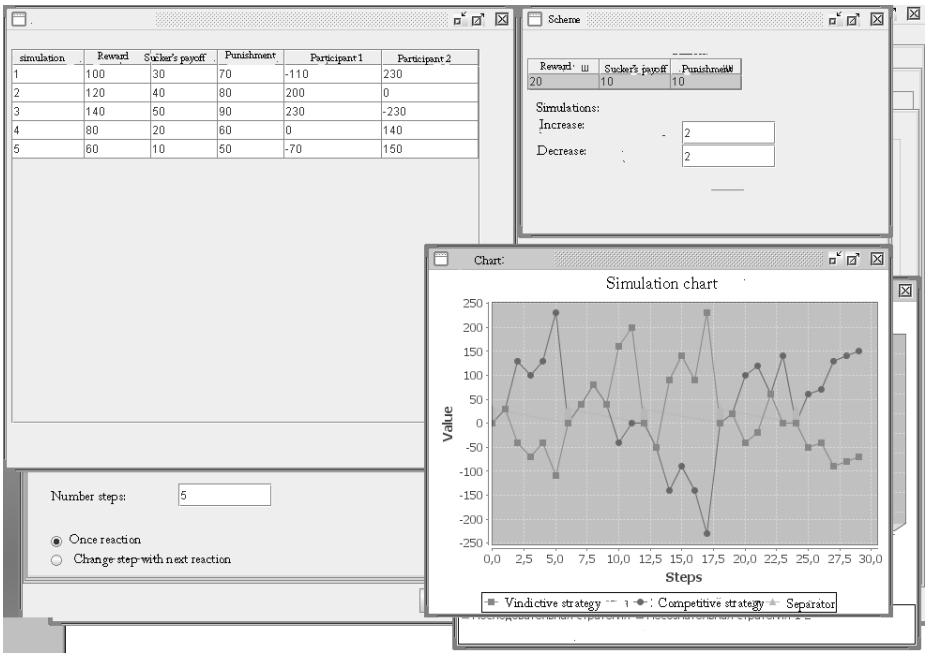


Fig. 2. Simulation results with fluctuating values of K, L, and M.

The results of the experiments show that the outcome of the game depends on the values of the gain function. So one can find experimentally the values of the gain function for any given strategy to be optimal.

Fourth scenario deals with a strategy as an element of the given set. We do not use the formal methods of working with such sets but accepted the way used by experimenters (biologists) to form them. In this case our goal was to find the probability for a strategy from one set to win a strategy from another one. For example our experimenters divided strategies into three sets namely provocative, forgiving and neither provocative nor forgiving exactly in the same way as it is done in R. Axelrod's experiment [4]. In our case a provocative strategy means immediate change of behavior in condition of the participant's own loss (or the opponent's gain) and keeping it till the next loss or till the end of the game. A forgiving strategy means that the participant changes its behavior under the same conditions but keeps it only some limited time (definite number of steps). In some way one can see it as follows: in forgiving strategy the participant only fights back as a response to the smack while in provocative strategy the participant not only fights back as a response to the smack but retaliates. Such division is just conditional as it reflects the view of experimenters.

The results of experiments show that the provocative strategies win in more cases than forgiving ones.

3 Conclusion

The software for simulation the enhanced version of the prisoner's dilemma game to set up experiments and explore pair interactions of objects with different behavior has been developed. The software allows experimenter to set pure and mixed strategies with finite and infinite memory. The simulation depends on its goal: whether it is maximization of the participant's gain or minimization of the opponent's loss. The participant's gain depends on the values of the gain function. If the participants' strategies are fixed then one can find experimentally such values of the gain / bonus / fine that the strategy of one of the participants is optimal. The results of the simulation comply with those given in literature in the case of classic game which can be accepted as an adequacy of the model.

The software was tested and operates at Biology School of V. N. Karazin Kharkiv National University but it can be successfully used at other schools and fields such as economy, sociology, etc.

References

1. Merrill, M. Flood: Some Experimental Games. Research Memorandum RM 789
2. Tucker A. W. On Jargon: The Prisoner's Dilemma. UMAP Journal 1 (1950): 101
3. Rheingold, H.: The Virtual Community – Perennial (2006)
4. Axelrod, R.: The evolution of cooperation. New York: Basic Books (1984)
5. Axelrod, Robert and Hamilton, William D.: The Evolution of Cooperation. Science, 211 (1981) 1390-1396
6. Dawkins, R.: The selfish gene (1993)
7. The Iterated Prisoner's Dilemma Competition [Electronic resource]. – Mode of access: http://www.prisoners-dilemma.com/
8. Adaptive modeler [Electronic resource]. – Mode of access: http://www.softsoft.ru/business/investment-tools/3210.htm
9. Vorobjev, N.N.: Game theory. (1976)
10. Grabovski, B.I.: Cellular automata, as simple models of complex systems. (1995) 412–419.
11. Karlin, S.: Mathematical methods in the game theory, programming and economy (1964).
12. Tseitlin, M.: Research on the theory of state machines and modeling of biological systems. (1969)

ICT infrastructures, Integration and Interoperability

Design and Implementation of a Quality Management System for Electronic Training Information Resources

Hennadiy Kravtsov

Kherson State University, 40 r. Zhovtnya 27, Kherson, Ukraine
kgm@ksu.ks.ua

Abstract: the results on designing and implementation of quality management system of electronic learning resources and its use for the organization and carrying out of monitoring of quality of resources of a higher educational institution are presented. As the illustration of quality monitoring of resources in KSU is used the distance learning system «Kherson Virtual University».

Keywords: quality management system, monitoring and quality management of learning electronic resources, distance learning system «Kherson Virtual University».

1 Introduction

Maintenance of quality of training is one of the primary goals of a university education system. The concept «quality of education» has no standard definition. It is connected by that various groups of consumers put in it their sense and researchers treat it depending on the research problem. Thus it is possible to allocate two basic approaches to concept of quality:

- In the first case for a basis of quality of education the requirement of conformity to the standard is accepted
- In the second case for a basis of quality of education the satisfaction to requirements and expectations of consumers is accepted

Therefore at the analysis of quality of education in university it is necessary to consider two aspects: conformity to standards and satisfaction to requirements of consumers which are students and the university faculty.

One of the major problems of the higher school are the organization of the monitoring system of quality of education and maintenance of constant growth or maintenance at high level of indicators of quality of education. It can be reached by introduction of a control system of quality of educational process and quality of educational services with use of information-communication technologies. In particular quality management of such new forms of training, as distance learning has the important importance. One of objects of the analysis of quality of educational process are electronic information resources (EIR), providing educational process. In particular, distance learning courses are one of the major and most often used EIR [1].

Since EIR are classified as educational electronic editions and are software

products then monitoring of quality of electronic educational resources should be multilevel taking into account their classification signs. Classification principles allow considering the separate characteristics of electronic means of educational purpose for carrying out of monitoring of EIR quality as a whole. Criterion of quality degree of EIR compatibility with standards IMS, SCORM can be chosen.

In work [2] the results of the analysis of criteria of quality and designing of monitoring system of EIR quality in distance learning system (DLS) «Kherson Virtual University», developed in the Kherson State University (KSU) are presented.

2 Structure and architecture of a control system of EIR quality

The EIR quality management system (QMS) is a structural element of architecture of a control system of quality of education in a higher educational institution.

In university taking into account a control system of EIR quality it is possible to present the general scheme of quality management of education as follows (Fig. 1):

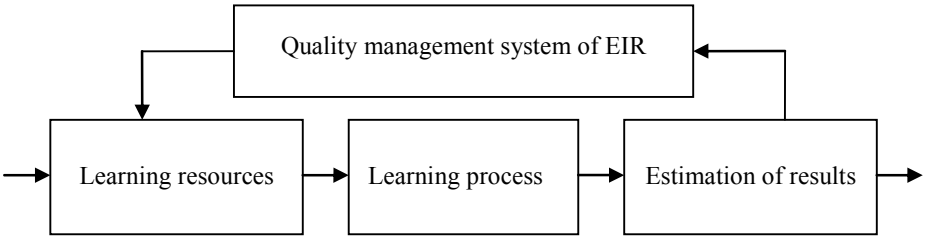


Fig. 1. The quality management system of EIR in architecture of quality management of university education

Thus, EIR QMS plays a role of feedback in the course of training for the purpose of EIR quality assurance and resource maintenance of process of training. Therefore constant functioning EIR QMS taking into account the correct organization of its work should provide high level of EIR quality indicators. On the other hand this system carries out a problem of rejection poor-quality learning EIR, defining their life cycle.

Itself EIR QMS represents the difficult system consisting of objects which carry out the analysis, research and management in the parameters providing EIR quality in the learning course.

The structure of a control system is presented by EIR quality in Fig. 2.

According to the resulted structure EIR QMS managerial process by quality of electronic learning resources consists of a complex of the following interconnected actions. Carrying out of monitoring of EIR quality is a quality assurance major factor, defining, first of all, degree of EIR conformity to educational standards. The important criterion of an estimation of EIR quality is degree of satisfaction of users of these resources of training. The university advisory council supervises over work on carrying out of monitoring of EIR quality and the analysis of results of questioning of students and teachers under program Feedback, defining Estimation criteria of EIR.

Certification of EIR under standard ISO 9000/9001 can serve as an estimation of high quality. At the same time, requirements and recommendations of these standards can serve as criteria of an estimation of EIR quality. The estimation of EIR quality is the tool of improvement of consumer characteristics of these resources, defining directions of researches at support and working out (acquisition) of new electronic resources of training. Acquaintance of the faculty of university with EIR rating promotes increase of motivation of teachers to use of qualitative resources and mastering by new information technologies of training.

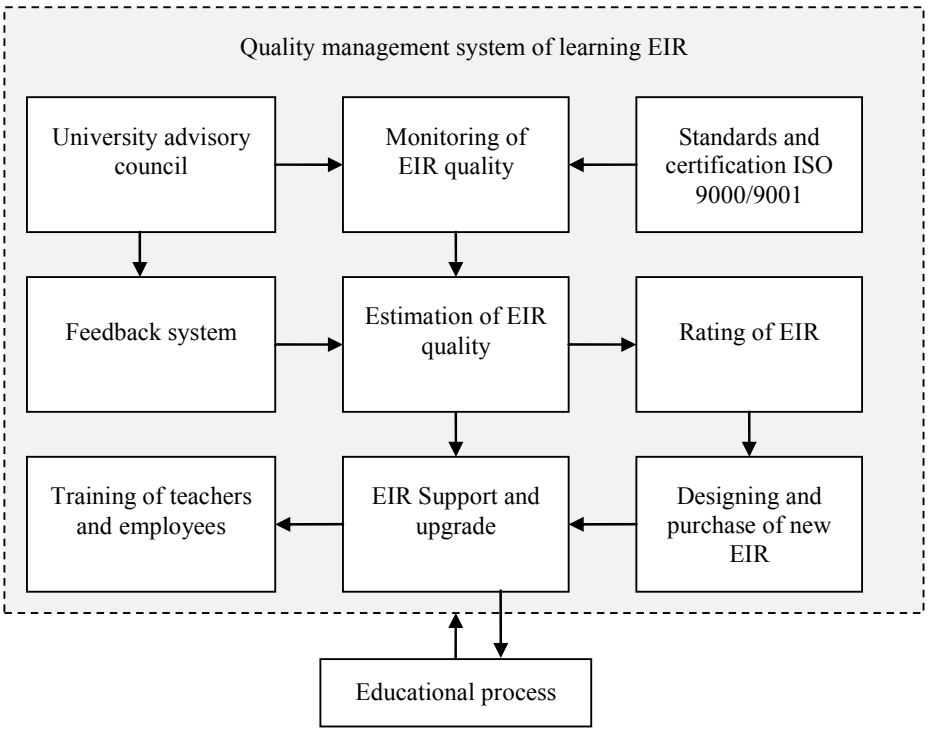


Fig. 2. Structure of a control system of EIR quality.

The control system of EIR quality should contain list of supervising documents, actions and an order of their realization which are reduced to the following in university:

- Administration problems (a policy in the field of quality, the work organization on its embodiment)
- Documentation and planning system
- Quality during working out of plans and programs (competence of developers, level of prepared documents, performance check, a timely estimation of results, entering of necessary changes)
- Quality assurance got EIR training
- Quality at a stage of EIR manufacture (introduction the QMS of working out new EIR)

- Quality check used EIR (entrance checks, the interoperational control, the definitive control, the documentation of tests)
- The control over test means
- Careful research of defective educational resources, detailed finding-out of the reasons of occurrence of defects, carrying out of correcting actions
- Quality of EIR storage, protection against harmful programs
- Documenting of EIR quality, registration of necessary documents
- The analysis of quality and acceptance of corresponding measures
- Personnel training

Let's stop in more details on basic elements of control system of quality of electronic training resources.

2.1 Estimation of EIR quality

The estimation of EIR quality underlies a control system of quality of electronic resources of training. For an estimation of EIR quality it is necessary:

- On a constant basis to carry out monitoring of EIR quality for control of EIR quality
- To have a feedback with users of EIR for the account of wishes in their improvement from positions methodical and program-technology requirements

For carrying out of monitoring of EIR quality it is necessary to develop their criteria of quality. The university advisory council confirms the criteria of EIR quality developed by the methodical commissions. The university Advisory council also confirms recommendations about improvement qualities of EIR received as a result of the analysis of responses of users in Feedback system.

Results of an estimation of EIR quality should be used on the one hand for improvement of their substantial part and satisfaction to technology requirements, on the other hand for publication of a rating of electronic learning resources that also promotes increase of their quality.

2.2 Monitoring of EIR quality

Monitoring of EIR quality possesses a leading role at their estimation of quality. The analysis of electronic resources of training shows, that they have the following classification: to a functional sign they can be carried to training editions, under the form of representation they belong to a category of electronic editions, on technology of creation they represent software product [1]. Therefore monitoring of quality of electronic educational resources should be multilevel taking into account their classification. The satisfaction requirement to the standard international standards what are IMS, SCORM [2] is uniting attribute of multilevel monitoring of EIR quality.

2.2.1 Monitoring of EIR quality to a functional sign

Now for university which includes four groups of the educational information resources differentiated to a functional sign, defining their value and a place the certain typological model of system of educational editions has affirmed as educational process [2]:

- Learning-methodical (methodical instructions, the managements containing materials by a technique of teaching of a subject matter, course studying, to performance of course and degree works)
- Training (textbooks, manuals, texts of lectures, abstracts of lectures)
- Auxiliary (practical works, collections of problems and exercises, reading books)
- Supervising (testing programs, databases)

2.2.2 Monitoring of EIR quality by criterion of compatibility with educational standards

IMS specifications are information model of the description of educational objects. It defines the standardized set of information blocks which are contained by data about an educational resource. The IMS-package which contains educational object consists of two main elements [2]:

- The IMS-manifesto – a special file which describes base resources, the maintenance and the organization of educational object (it is represented in language XML)
- Physical files which make educational object

The similar organization of resources corresponds to modern approaches to work with electronic educational resources, in particular, to the concept of educational object.

The IMS-manifesto is base concept of specification IMS. Conceptually the IMS-manifesto is the multilevel description of data. At the lowermost level there is a description of physical files which form an educational resource. Some descriptive information named metadata which also joins in the manifesto can respond each file.

2.2.3 Types of EIR

Four groups of the educational information resources differentiated to a functional sign which defines their value and a place in educational process are distinguish [2]:

- Program-methodical (curricula and working programs)
- Learning-methodical (the methodical instructions containing materials by a technique of teaching of a subject matter, course studying, to performance of course and degree works)
- Training (textbooks, manuals, texts of lectures, abstracts of lectures)
- Auxiliary (practical works, collections of problems and exercises, encyclopedias, reading books)
- Supervising (testing programs, databases)

Each group of EIR has the distinctive features and the parameters defining quality of this or that educational information resource. So most often educational EIR the electronic textbook (course of lectures) which concerns training resources is used. Among the parameters defining quality of the electronic textbook it is possible to allocate in particular completeness of representation and coherence of training

information materials, presence of the control-help information, conformity to the maintenance of the working program, organization and sequence of a material, ergonomics of the text, presentation of a material: use of multimedia possibilities, interactive systems and modules, modeling possibility, and also testing use, possibility of monitoring of knowledge, self-checking have text formatting, use of tables, schemes, drawings, illustrations, etc. has special importance.

Completeness of the electronic textbook assumes presence of following additional information resources:

- The textbook title page
- The summary it (is desirable)
- The course program
- The list of reductions (if it is available)
- The list of illustrations
- Data on the author
- Actually texts that (heads) № 1, 2, 3, …
- The list of the recommended literature on subjects
- The list of the quoted literature in the end of a course
- Applications (the list of statutory acts, decrees, decisions if they are available)

Along with the electronic textbook the important role is played by a control-help part of resource maintenance of a course which should contain:

- The list of questions and tasks for self-examination studied to each subject-head, section and to all course (or the list of questions and tasks for computer training in the environment of multimedia)
- Subjects of course works and abstracts
- The approximate list of examination questions at all course (or to offset)
- The chronological index (if it is available)
- The index of names (if it is available)
- The index (if it is available)
- The dictionary of terms
- Methodical instructions (or recommendations)

Among all EIR the special role is played by a distance learning courses. It is the basic educational object which is used in distance learning. Its feature consists that it is compound training object which unites various EIR for the purpose of the organization of process of training with use of special program environments – DLS. An example of such program environment which allows to create, keep and use distance courses, is DLS «Kherson Virtual University» [2].

Thus, EIR should be differentiated depending on their type. First of all, it concerns expenditures of working hours on creation of these resources, both time, and intellectual. Therefore at an estimation of concrete EIR quality it is necessary to start with some generalized criterion of working hours input of its creation which can be expressed in weight factor:

- A course of lectures
- The plan-abstract to a course of lectures, laboratory and practical works
- Methodical instructions to carrying out of seminar employment and performance of laboratory works
- The test

- The working program of a course
- Questions to examination/offset, self-checking
- A laboratory practical work
- The collection of problems, exercises, the dictionary
- The methodical grant
- The encyclopedia
- Distance learning course on discipline

It is necessary to notice, that depending on type educational information resources have as the general, and the distinctive criteria of quality which are expressed by quality indicators:

- Completeness of methodical maintenance of discipline
- Authorship of a material
- Completeness of representation of a material
- Conformity to the maintenance of the working program
- Sequence of materials
- Conformity of a material to the world standards
- Degree of use of a resource
- Material organization
- Ergonomics of the text
- Use of links
- Presentation of a material
- Use of multimedia modules
- Use of interactive systems and modules, modeling possibility
- Testing use, possibility of monitoring of knowledge, self-checking
- Use of standard formats of EIR files
- Use of tables, schemes, drawings
- Conformity of a material to level of knowledge of users
- A special-purpose designation of a material for a corresponding audience
- An easy approach to a material
- Stylistic correctness of a statement of a material

Among the basic types of the software for creation e-Learning decisions it is possible to allocate Authoring Packages, Learning Management Systems (LMS), Content Management Systems (CMS), Learning Content Management Systems (LCMS).

Author's products are specially developed for overcoming of those difficulties which teachers face at use of programming languages. These programs usually allow the teacher to develop independently an educational content on the basis of visual programming. A lack of such products is the impossibility to trace and supervise in time process of training and progress of a considerable quantity of trainees. As a rule, they are developed for creation of lessons with an immediate feedback with the trainee, instead of for storage of the information on educational process for long time.

LMS (in Russian-speaking terminology abbreviation DLS is used – "distance learning system") represent a platform for expansion e-Learning, but in some cases can be used and for administration of traditional educational process.

System LMS, in an ideal, should give to each student personal possibilities for the most effective studying of a material, and to the manager of educational process - necessary tools for formation of curriculums, the control of their passage, drawing up of reports on productivity of training, the organization of communications between students and teachers. The student receives from LMS access possibilities to an educational portal which is a starting point for delivery of all educational content, a choice of suitable educational tracks on the basis of preliminary and intermediate testing, use of additional materials by means of special links.

2.2.4 Criteria of EIR quality

The system of monitoring of EIR quality can be based on multicriterion analysis of conformity of these resources to the standard educational standards.

Classification principles allow considering separate characteristics of electronic means of educational appointment for carrying out of monitoring of EIR quality as a whole. Criterion of EIR quality compatibility with standards IMS, SCORM [2, 3] can be chosen.

Let's pass to construction of the general criterion of EIR quality. For a basis of a conclusion of criterion of quality we will accept the standard approach based on consideration of the average factor of quality $K = (a_1k_1 + a_2k_2 + \ldots + a_nk_n)/n$, where a_i – average value of indicators of quality, k_i – value of weight factor of i-type resource [4].

The general criterion of EIR quality can be calculated under the formula

$$K_0 = \sum_{i=1}^{N} a_i t_i \,. \qquad (1)$$

where $a_i = n_i \cdot \gamma_i$ – the quality metrics, n_i – weight factor, $\gamma_i = \sum_{j=1}^{m_i} k_{ij} / k_{iM}$ – average factor of quality, m_i – quantity of metric indicators of quality, k_{ij} – a quality j-indicator, k_{iM} – the greatest possible value of an indicator of quality, t_i – the generalized factor of quality of i-type resource, N – quantity of EIR.

For definition of ratings of faculties and chairs of a higher educational institution it is necessary to enter into consideration relative average criterion of quality $K = K_0 / N$, where K_0 is calculated under the formula (1).

2.3 Feedback system

Demand studying on EIR, as well as on any other intellectual product, is necessary for revealing of their qualities on purpose improvement of their methodical and program-technological properties. The Feedback system with users of EIR serves as the tool for the organization of flexible and all-round interrogations of opinions of students and teachers of university. Usually the system spends questioning in an automatic mode. The built in master of interrogations allows to create easily and simply interrogations, to make to them changes and to spend questioning sessions. The generalized estimation of EIR quality received after statistical processing of results of questioning of users, gives the chance to consider degree of their demand at quality monitoring.

There are specialized systems “Customer Feedback” which help to organize process of questioning of EIR users. Besides data gathering, these systems offer powerful tools of the analysis and the reporting. At the Kherson State University there is an automated feedback system “KSU Feedback” (http://feedback.ksu.ks.ua) which is used for gathering of the information from users of EIR about quality of training, in particular about qualitative characteristics of electronic resources of training.

2.4 Standards and certification ISO 9000/9001

Certification is a documentary acknowledgement of conformity of production to certain requirements, concrete standards or specifications. It is necessary to notice, that conformity to standard ISO 9000/9001 does not guarantee high EIR quality. However conformity to requirements and recommendations of these standards is a necessary condition of high quality of resources of training. The certificate of conformity ISO 9001 is acknowledgement of satisfaction to standard requirements.

Standard ISO 9000/9001 is fundamental, the terms accepted in it and definitions are used in all standards of a series 9000. This standard pawns a basis for understanding of base elements of system of a quality management agrees standards ISO.

Requirements of standard ISO 9000/9001 can be used as criteria at the organization and carrying out of monitoring of EIR quality.

2.5 Advisory council of university

In a control system of EIR quality the university advisory council is the body which is responsible for adequacy estimation of EIR quality taking into account all criteria and indicators of quality. It confirms Position about a control system of EIR quality, defines criteria of their quality, forms rules of carrying out and confirms results of an estimation of quality, and also plans actions for improvement of EIR quality.

The university advisory council defines an order of carrying out of monitoring of EIR quality. It confirms the list of criteria of quality, their weight factors and values of indicators of quality according to (1).

2.6 Support and upgrade of EIR

Support and upgrade of EIR is the important site of work in a control system of quality in respect of elimination of defects, improvement and EIR software at its use in educational process. Support of EIR software is one of phases of life cycle of the software in which course in EIR software changes for the purpose of correction of the lacks found out in the course of use are made, and also for addition of new functionality and efficiency increase. Support software is defined by standard IEEE Standard for Software Maintenance (IEEE 1219), and the life cycle standard is specified ISO 12207.

Support of EIR software is necessary for that maintenance that the software product throughout all period of operation meets requirements of users. Tracing and

the control – key elements of activity on support of EIR software.

The important factor of increase of efficiency usage of EIR is training of users and maintenance with their regular support at work with the current software version.

Support and upgrade of EIR software should consider also updating of capacity of the hardware or a corresponding telecommunication infrastructure.

3 Control system of EIR quality at the Kherson State University

The control system of EIR quality at the Kherson State University works since 2009. It includes all actions and activity of services according to described above the scheme of structure of EIR quality management (fig. 2).

The system of monitoring of EIR quality in DLS «Kherson Virtual University» is based on the multicriterion analysis of conformity of these resources to the conventional educational standards [2]. All resources of electronic library were estimated by criterion K_0 according to (1) with values of weight factors and the indicators of quality confirmed by expert methodical commission of Kherson State University.

Monitoring of EIR quality in distance learning system «Kherson Virtual University» is spent according to the order of rector of university and sets as the purpose, first, to give quality standard rather to great volume of training resources (more than 5000 names), developed by teachers of university, and, secondly, to plan ways of quality improvement of work of faculty KSU to this direction.

4 Conclusions

The system of monitoring of EIR quality can be based on the multicriterion analysis of conformity of these resources to the standard educational standards.

Classification principles allow considering separate characteristics of electronic means of educational appointment for carrying out of monitoring of EIR quality as a whole. Criterion of EIR quality compatibility with standards IMS, SCORM can be chosen.

It is possible to use and other criteria of classification, however, without dependence from appointment, a technique of use or technology of realization, a basis of any didactic means is the teaching material of a studied subject domain. Selection of this material (which it is carried out proceeding from didactic problems and methodical principles) is a prerogative of the teacher. For this reason the computer (distance) course should be the integral multicomponent system reflecting scientific and methodical sights of the author. The corresponding commission of experts of university should give an estimation of quality of distance learning course.

On a basis of multi-criteria analysis taking into account EIR compatibility with the international standards criteria of EIR quality are described.

The basic types of electronic resources of educational appointment for carrying out of monitoring of EIR quality are allocated. For each type of EIR their weight factors and quality indicators are offered. The criterion of quality of an electronic training

resource which is the average characteristic of quality is developed, considers its weight factor and relative indicators of quality.

The offered system of an estimation of quality of electronic training resources is not unique and supposes additions and updating. The estimation of monitoring of EIR quality is given by a corresponding commission of experts of university.

References

1. Bikov V. Yu. Models of the Open Education Organizational Systems: Monograph. - Kyiv: Atika, 2009. - 684 p.: ill.
2. Kravtsov H.M. Quality Monitoring System of University Electronic Information Resources / Information Technologies in Education. 2d Issue. - Kherson. - 2008. - p. 42 - 46.
3. H. Kravtsov, D. Kravtsov. Knowledge Control Model of Distance Learning System on IMS Standard / Innovative Techniques in Instruction Technology, E-learning, E-assessment, and Education. - Springer Science + Business Media V.B. - 2008. - p.195 - 198.
4. H. Kravtsov. Evaluation Metrics of Electronic Learning Resources Quality / Information Technologies in Education. 3d Issue. - Kherson. - 2009. - p. 141 - 147.

On Optimization Criteria for Task Assignment in Cluster and Wide-Area Computing

Dmitriy Litvinov

V.N.Karazin Kharkov National University, Kharkov, Ukraine,
dimalit@yandex.ru

Abstract. Most of techniques proposed for task to processor mapping in distributed computing minimize communication cost that is calculated with model based on the quadratic assignment problem (QAP). I.e. to each pair of tasks and each pair of processors is assigned a cost, and then total cost of particular mapping is computed as the sum of products of the corresponding costs. However, shortcoming of such approach is that it cannot adequately address both bandwidth and contention considerations. Therefore, we propose an alternative communication cost model. It is based on the multicommodity flow (MCF) approach, and can directly compute communication time considering both link bandwidths and contention in the network. We evaluate our model through simulation using NPB Multi-Zone benchmarks on several irregular network topologies. Evaluation results show that MCF-based cost function has higher correlation with actual application run time than popular QAP-based cost functions. Thus, the usage of proposed cost model can potentially improve quality of solutions obtained with the task assignment algorithms.

Keywords: task assignment, wide-area computing, Grid computing, multicommodity flow, mapping problem

1 Introduction

One of the most widely used paradigm of programming multicomputers is "single process multiple data" (SPMD) programming. It assumes that application consists of multiple interacting processes or tasks which are assigned to processors and then executed. If the application is communication-limited rather then computation-limited its runtime heavily depends on network performance. In the case of distributed computing platform with irregular network topology, such as cluster or Grid, this means that application run time also depends on exact way how tasks are assigned (mapped) to the processors.

There exist a number of algorithms that optimize such mapping. Because of NP-hard nature of the problem these algorithms are mainly based on heuristics. In most of researches on this topic application is represented by a task interaction graph (TIG), which we will denote as $G(V, E)$. Nodes of this graph correspond to tasks, and edges represent intertask communication. Both nodes and edges can be weighted with node weight representing amount of computation done by

the task and edge weight representing amount of data transferred between different task pairs. To describe the platform, another graph $G' = (V', E')$ is used. Vertices V' of this graph are processors. The graph is complete. Its weighted edges reflect cost of transferring a unit amount of data between every pair of processors. Sometimes nodes of this graph are also assigned weight which represents cost of performing a unit of computation on that processor. The goal is to find mapping from V to V' which minimizes some objective function π:

$$\pi : V \to V' | c(\pi) \to min \ .$$

Consider cost function $c(\pi)$. In general case it reflects both communication and computation costs. In this paper we focus on communication, and therefore will ignore cost asccociated with computation which generally belongs to a separate problem (load balancing). Usually communication cost is written as

$$c(\pi) = \sum_{(u,v) \in E} w(u,v) d(\pi(u), \pi(v)) \ , \tag{1}$$

where π is a mapping, w is (u, v) edge weight, and d is "distance" (cost of communication) between two processors (vertices of G'). This formulation represents an instance of the *quadratic assignment problem* and therefore such models and similar ones we will call *QAP-based models*.

Now consider distance function d mentioned in (1). It determines the "expence" of communication between each pair of processors. Clearly, it should assign larger costs to communications through slow links. In such a case it should be roughly inversely proportional to the bandwidth that network can deliver between a pair of processors (e.g. maximum flow value). But at the same time it should minimize contention. To account this, d can be made proportional to "hop count" distance between processors. Rationale behind this is that minimizing hop count that a message must travel we also minimize the probability of contention. An example of compromise between these two goals is "RTT" metric, which makes link cost proportional to round trip time (RTT) of a test packet between nodes in question. Clearly, RTT is both inversely proportional to the bandwidth between nodes and directly proportional to the number of links that a message should travel.

However, in either case QAP approach can measure contention only *indirectly* through its *probability*. But with applications where network is a system performance bottleneck it is unwise to model contention in such a rough way. Thus, in this paper we describe communication cost model which can directly account contention. In our model we consider multicommodity flows generated by pairs of processors and route them optimally, such as total communication time is minimized. This time is determined by a "bottleneck" links that have largest traffic/bandwidth ratio. Afterwards, this time is used as a cost of particular task-to-processor mapping. Such formulation is closely related to multicommodity flow feasible problem [1] and similarly can be formulated and solved as a linear program (LP).

2 MCF-Based Cost Function

2.1 Basic Notations

Let $G = (V, E)$ be a (directed) task interaction graph. Vertices of this graph V correspond to tasks and edges E represent intertask communication. Every edge is assigned weight w_{ij} which is amount of data transferred through it.

Let $G' = (V', E')$ be (directed) platform graph. Here vertex set V' represents platform nodes and edge set E' represents links between them. A node can contain zero or more processors. Such a way we uniformely model both compute nodes and network nodes (a.k.a. routers). Every link has bandwidth b_{ij}.

Now suppose we are given some assignment of task to processors (and consequently, to nodes). As the amount of communication between each pair of tasks is known, from these we can derive the amount of communication between every pair of nodes.

Let W be a set of (ordered) pairs of nodes having nonzero communication between them. Amount of data sent between pair $w \in W$ we denote r_w. Let P_w be a set of all possible (simple) paths between two nodes belonging to pair w and let $P = \bigcup P_w$. A routing algorithm used in the network somehow distributes data r_w between these paths. Denote the amount of data sent over path $p \in P_w$ as x_p. Obviously

$$\forall w \in W : \sum_{p \in P_w} x_p = r_w \ . \tag{2}$$

Now consider an arbitrary edge of the platform graph $(i, j) \in E'$. Amount of data that should be sent through it is as follows:

$$v_{ij} = \sum_{p \in P | (i,j) \in p} x_p \ ,$$

i.e. it equals to the total amount of data sent along all paths containing this edge. Minimum amount of time needed to transfer all these data through edge (i, j) is

$$t_{ij}^{min} = \frac{v_{ij}}{b_{ij}} \ . \tag{3}$$

To finish all communication we need to finish the most long lasting one. Therefore, the lower bound on total communication time can be estimated as follows:

$$T^{comm} \geq \max_{(i,j) \in E'} t_{ij}^{min} \ . \tag{4}$$

It can be shown that this value always can be achieved — i.e. for all communicating node pairs $w \in W$ their communication r_w can be performed within amount of time mentioned in (4). So

$$T^{comm} = \max_{(i,j) \in E'} t_{ij}^{min} \ . \tag{5}$$

The value of T^{comm} obtained in such way can be then used as a cost of a mapping. In contrast to QAP-cost it can directly measure contention. We call it

"MCF-cost" because all the flows $x_p \in P_w$ between different source-destination pairs $w \in W$ constitute a multicommodity flow (MCF) and T^{comm} is determined by a "bottleneck" edge of this flow.

2.2 MCF-cost Estimation in General Case

Note that equation (5) for communication time T^{comm} was derived based on the knowledge of the routing algorithm used in the network. Obviously, the value of T^{comm} can be easily estimated if the routing algorithm is static such as in case of tree-structured networks. However, if dynamic routing is utilized we cannot *a priori* know values of x_p which are needed for estimation of T^{comm}. Now we will derive the values x_p for such case too. We will assume that a) network uses dynamic multipath routing and b) that this routing works almost perfectly. In the context of mapping problem we say that routing works perfectly if it can deliver all the data in the shortest possible time.

Let $\boldsymbol{x}$ be a "routing" vector consisting of traffic volumes for all paths in P:

$$\boldsymbol{x} = \{x_p\}, p \in P \ .$$

Given this vector, the values of t_{ij}^{min} (3) and consequently T^{comm} (5) can be determined in a streightforward way. So, our aim is to find such routing vector $\boldsymbol{x}$ that gives

$$\min T^{comm} = T^{min} \ .$$

Let D be a set of all possible routing vectors that satisfy to the equation (2). It can be shown that D is a *convex polyhendron*. As such, any vector $\boldsymbol{x} \in D$ can be represented as a convex combination of vertices of D:

$$\boldsymbol{x} = a_1\boldsymbol{x^1} + a_2\boldsymbol{x^2} + \ldots + a_n\boldsymbol{x^n} \tag{6}$$

$$0 \le a_k \le 1 \ , \sum_{k=1}^{n} a_k = 1 \ ,$$

where $\boldsymbol{x^1} \ldots \boldsymbol{x^n}$ are vertices of the polyhedron D.

Suppose, we are given a routing vector $\boldsymbol{x}$. Assign to every edge (i, j) of graph G' a flow value of

$$F_{ij} = \frac{v_{ij}}{T^{comm}} \ ,$$

where T^{comm} can be found from (5). It can be shown that these flows always are feasible in the network G' and flow on path $p \in P$ $f_p = \frac{x_p}{T^{comm}}$.

As we did it for vector $\boldsymbol{x}$, construct from the path flows f_p vector $\boldsymbol{f} = \{f_p\}$ and similarly to (6) write:

$$\boldsymbol{f} = c_1\boldsymbol{x^1} + c_2\boldsymbol{x^2} + \ldots + c_n\boldsymbol{x^n} \ ,$$

where $c_k = \frac{a_k}{T^{comm}}$ and consequently $\sum_{k=1}^{n} c_k = \frac{1}{T^{comm}}$.

Thus, searching for such $\{c_k\}_{k=1}^{n}$ that $B = \sum_{k=1}^{n} c_k$ is maximized we can minimize T^{comm}.

Finally, recalling that flows $\{f_p\}$ must be feasible in G', we can write problem of minimizing T^{comm} as a linear program with respect to $\{c_k\}$:

$$\begin{array}{l} \text{Maximize} \\ \quad B = \sum_{k=1}^{n} c_k \\ \text{Subject to} \\ \quad \sum_{k=1}^{n} \sum_{p \in P \mid (i,j) \in p} c_k x_p^k \leq b_{ij}, (i,j) \in E' \\ \quad c_k \geq 0, k = \overline{1,n} \ , \end{array} \tag{7}$$

where x_p^k denotes the volume of traffic that goes through path p in a routing vector $\boldsymbol{x}^k$.

3 Preliminary Experiments

To compare proposed MCF-based cost model with the "classical" QAP-based ones we conducted a series of experiments. Using simple heuristic we generated a number of "good" task-to-processor mappings. For each of these mappings we computed QAP-cost, MCF-cost and evaluated by simulation actual application execution time. As our simulation environment didn't support adaptive or multipath routing, we were forced to use static routing only — which is nevertheless common in modern platforms. Experiments showed that MCF-based cost function had higher correlation with actual execution time then QAP-based one.

3.1 Experimental Setup

We conducted our experiments for three different platforms:

1. Compute cluster with hierarchial communication topology and bottleneck at higher levels of the hierarchy ("cluster" topology).
2. Wide-area computing system where nodes are connected via a tree-structured network with homogeneous links ("tree" topology).
3. Wide-area computing system having cycles in the topology graph ("grid" topology).

All three platforms consisted of 16 compute nodes, each having power of 16 GFlops. So, the only difference between them was in interconnect. Topologies of platforms 2) and 3) were derived from the actual topology of wide-area computing system [2].

Using MSG framework of SimGrid [3] we implemented the model of NPB Multi-Zone [4] benchmarks. These benchmarks use overset-grid approach to solve discretized versions of the unsteady, compressible Navier-Stokes equations. With this approach complex domain is covered by a set of partially overlapping meshes or *zones*. Then the equations are solved independently in each zone, and after each iteration the zones exchange boundary values with their immediate neighbors with which they overlap.

With some number of zones assigned to it, each benchmark process repeatedly executes the following steps.

1. Post asynchronous receive request for next step boundary values.
2. Compute current time step using current boundary values.
3. Post asynchronous send reqest with boundary values for neighbouring zones.
4. Finish asynchronous receive requests (1) (synchronize).
5. Go to step 1.

In this model asynchronous communication is used to tolerate network latency and utilize computation/communication overlap.

NPB-MZ benchmarks are particularly suitable to run in a distributed environment because they have relatively low communication-to-computation ratio. At the same time, their communication pattern makes them sensitive to network contention and consequently, particularly appopriate for studying the mapping problem. There are three benchmarks in this suite: LU-MZ, SP-MZ, and BT-MZ. They have very similar communication patterns but while SP-MZ is the most simple and straightforward benchmark, LU-MZ has poor scalability and BT-MZ has additional load-balancing issues.

To make emphasis on communication cost rather than computation cost or other issues we present here results for SP-MZ banchmark only and force the number of tasks equal to the number of compute nodes. So, in all experiments we had one-to-one mapping of tasks to nodes and computation was perfectly balanced. For different mappings only communication produced change in total run time of the application.

For our experiments we used such configuration of the benchmark (class D, 16 tasks) that send relatively large messages which are relatively insensitive to link latencies. Communication matrices for the banchmark were obtained through measurements during test runs on a real cluster and computational complexity of each task was evaluated analitically using formulas from the paper [4].

For generation of "good" random mappings on which we evaluated our model we used simple randomized local search heuristics described in [5]. It begins with a random mapping. Then it choses a random pair of tasks, and if the exchange of their places gives a decrease to target function, it is performed. The algorithm stops when it cannot decrease target function in certain predefined number of tries. Afterwards, the procedure can be repeated with another initial mapping. For our purpuse, we ran the mapping heuristic 1000 times with the stop criterion being 500 tries without improvement of the target function.

3.2 Experimental Results

Figure 1 shows scatter plots with MCF-cost on x-axis and QAP-cost on y-axis for different mappings. To compute QAP-cost for "tree" and "grid" topologies we used RTT-metric, and for "cluster" topology we used "bandwidth"-metric.

It is clear that in all three cases there exists correlation between these two functions. This correlation is particularly high for "cluster" topology: points on the plot are arranged almost in a straight line. However, for two other topologies for each value of QAP-cost there exists some spread of the MCF-cost values. This

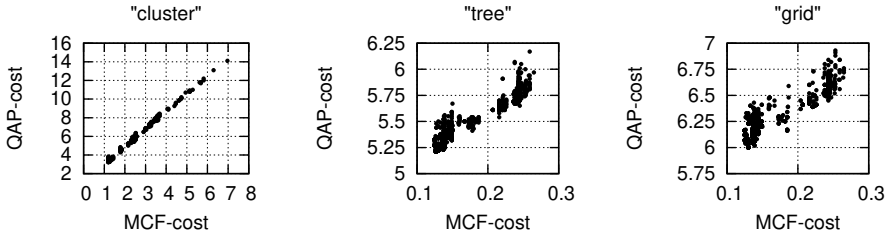


Fig. 1. Dependency of QAP-cost on MCF-cost for different topologies

means that even minimizing QAP-cost can often leave some space for improvement of the MCF-cost and, potentially, the run time of the application. Based on the above arguments, further experiments we conducted only for "tree" and "grid" topologies.

Fig. 2 depicts an example of how QAP-cost, MCF-cost and actual run time may change on successive iterations of the mapping heuristic (figure shows 10 runs (214 iterations) of the heuristic for "grid" topology).

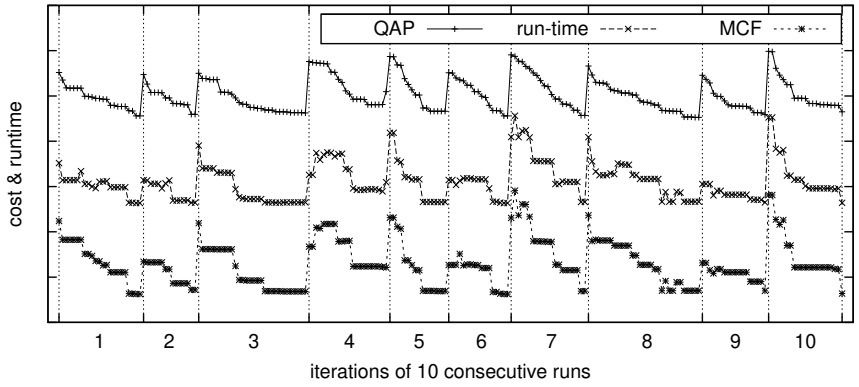


Fig. 2. Heuristic search iterations

While QAP-cost, being a target function of the optimization, constantly decreases, other two curves behave a bit differently. For example, at runs 4, 7, and 8 there are points where actual run time increases, and so does MCF-cost. Generally there are 44 points where QAP-cost goes "wrong" (in opposite direction from the run time). On the contrary, there exists only 15 points where MCF-cost makes such mistake.

To estimate the difference between QAP and MCF costs quantitively, we generated 1000 "final" mappings (i.e. where local search stops) and computed correlation coefficients between each cost function and actual run time of the

application. Additionaly, we separately computed correlation for those mappings that had actual run time falling into best 10% of its full range. Scatter plots are shown on Fig. 3 and numerical resulsts are summarized in Table 1. Clearly, in terms of correlation, for both "tree" and "grid" topologies MCF-cost outperforms QAP-cost. The gain is especially notable for "top 10%" of the mappings which are, obviously, the most important.

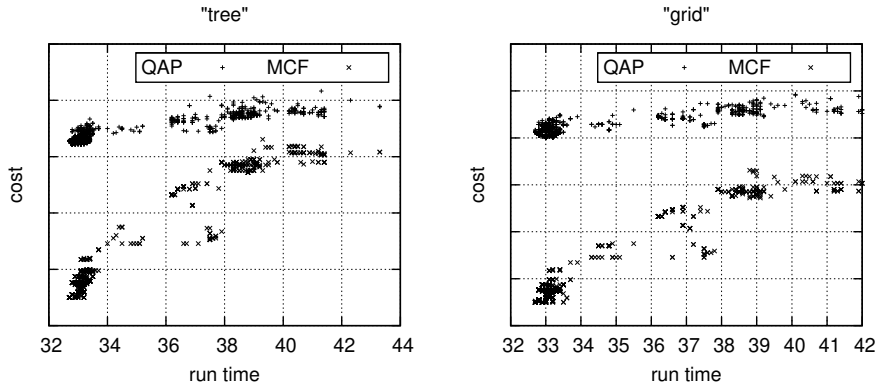


Fig. 3. Dependency between QAP-cost, MCF-cost, and actual run-time

Table 1. Summmary of correlation coefficients

	"tree" topology		"grid" topology	
	100%	top 10%	100%	top 10%
QAP-cost	0.91	0.38	0.86	0.36
MCF-cost	0.97	0.66	0.96	0.55

4 Conclusions and Future Work

Although QAP approach to communication cost estimation is the most widespread in the literature on the mapping problem, other approaches also can be found there. Particularly, in the paper [6] communication cost is estimated as the maximum *occupacy* over links of the platform graph. As the occupacy is defined as the ratio of (traffic) *load* to communication *capacity* of the link, this approach is exactly equivalent to formula (5). However, the paper [6]:

1. Doesn't prove the choise of used flow-based model of communication cost.

2. Considers only static routing and ignores possibility of adaptive and multipath routing.

In this paper we have shown experimentally the advantage of flow-based (MCF) models over traditional distance-based (QAP) models. Certainly, this advantage exists just as long as the messages that tasks send to each other are relatively long, so network latency can be neglected. This condition holds well for NPB-MZ benchmarks running on a medium-sized platform, which we used in our experiments. Also we have shown that for tree-structuted clusters having bottleneck at higher levels of the communication hierarchy MCF-based communication cost model is likely to be equivalent to traditional QAP-based ones and probably has no advantages over them.

Second, theoretical contribution of this paper is that it extends flow-based approach to evaluation of communication cost onto networks with adaptive and multipath routing — i.e. most of modern networks. Furthermore, expression for T^{comm} (7) derived here constitutes theoretical lower bound on the communication time and therefore can be used for evaluation of the efficiency of routing algorithms used in distributed computing platforms. This also suggests that the model presented here can be used for *improvement* of routing algorithms for use in this field.

Based on the conclusions presented just before, we anticipate the following directions of further research in the scope of this paper:

1. Experimental evaluation of proposed MCF-based model of communication cost on platforms that utilize modern adaptive multipath routing algorithms.
2. Development of a mapping technique that uses MCF-based cost model.
3. Extending routing algorithms to utilize the knowledge of structure of the application running on a computational platform with the goal of improvement of routing efficiency.

References

1. Hu, T.C.: Integer Programming and Network Flows. Addison-Wesley, Reading, MA (1970)
2. Saito, H.: Design and Implementation of Scalable High-performance Communication Libraries for Wide-area Computing Environments. Ph.D. thesis, University of Tokyo (2008)
3. Casanova, H., Legrand, A., Quinson, M.: SimGrid: A Generic Framework for Large-Scale Distributed Experiments. In: 10th International Conference on Computer Modeling and Simulation, pp. 126-131 (2008)
4. Van Der Wijngaart, R.F., Jin, H.: NAS Parallel Benchmarks, Multi-Zone Versions. NAS Technical Report NAS-03-010, NASA Ames Research Center, Moffett Field, CA (2003)
5. Orduna, J.M., Silla, F., Duato, J.: On the development of a communication-aware task mapping technique. Journal of Systems Architecture, Volume 50, Issue 4, pp. 207–220 (2004)
6. Taura, K., Chien, A.: A Heuristic Algorithm for Mapping Communicating Tasks on Heterogeneous Resources. In: 9th Heter. Computing Workshop, p. 102 (2000)

A Lightweight Approach to Contact Data Synchronization in Mobile Social Networks

Nikolay Tkachuk[1], Alexey Vekshin[1], Konstantyn Nagorny[1] and Rustam Gamzaev[1]

[1] National Technical University "Kharkov Polytechnic Institute", Frunze str. 21, 61002 Kharkov, Ukraine
tka@kpi.kharkov.ua, {alexeyvekshin, k.nagornyi, rustam.gamzayev}@gmail.com

Abstract. Data synchronization is one of the most critical issues in "always-available" software mobile applications development. In this paper the new approach to resolve this problem in social networks is proposed, which is based on iPhone platform, and utilizes legacy data storage based on MS .Net WCF (RESTFull services) and MS SQL Server. This application provides import of client's contacts data (e.g. from iPhone's address book) created previously in another application (MS Outlook, etc.) into social networks, and supports their updating by further synchronization process. The advantage of proposed approach is it's universality and lightweight, because it does not need to implement any special software adapters and interfaces.

Keywords: data synchronization, mobile application, XML-mapping.

1 Introduction: Research Actuality and Aims

Nowadays many customers need to access information any time anywhere. For this purpose different mobile software- and hardware platforms can be used, which allow to store data in various formats and to provide diverse access interfaces. There is an obvious necessity for centralized data storage and access them in such distributed systems. Thus, data synchronization (DS) issue is one of the most significant problems in development of "always-available" software mobile applications. In this paper we propose an approach to solve a DS problem during development of a mobile client application in social network which is based on iPhone platform and uses legacy data storage based on MS .Net WCF (RESTFull-services) and MS SQL Server. Our system implements the possibility to import into social network the client's contacts data (e.g. from address book), which were created previously in another application (e.g. MS Outlook), and provides their modification with further synchronization. The paper is structured in following way: Section 2 depicts briefly some modern trends in this re-search domain, in Section 3 our approach is represented, the appropriate software solution and its complexity estimation are discussed in Section 4. Finally, Section 5 concludes the paper and gives a short outlook on some future works in this research.

2 Contacts Data Synchronization Issues in Social Networks: Some Modern Trends

Nowadays there are few leading mobile platforms, e.g. iOS, Windows Mobile, Symbian, Android, and most of business customers require solutions which cover all these technologies. In any case the requirement to have contacts data synchronized with remote servers or with other devices is strictly required even for desktop applications, but definitely this is a critically important issue for modern mobile software systems, especially for social networks. This requirement has to be met together with some additional special constraints of mobile applications such as e.g. performance sufficiency and battery life [1].

On Fig. 1 a typical scheme of several interactions in social network is presented. As it is shown on this diagram, each client application: iPhone, MS Outlook, etc., and system's server as well have own storage with contacts data. The centralized database on the server-side contains contacts data from all interacting clients. One of all tasks to be realized in this system is the DS procedure between different clients and centralized database on the server.

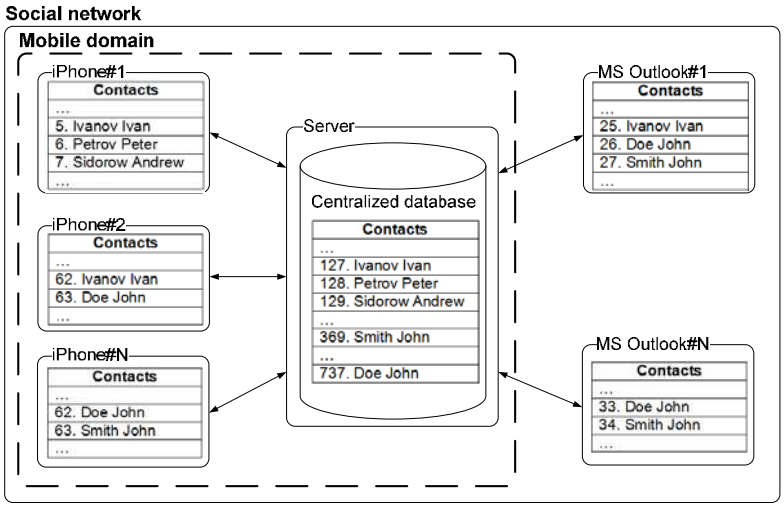


Fig. 1. Typical scheme for DS processes in social networks.

There are some special tools to solve DS problems. For example, SyncML [2] is a specification standard for common DS framework, but the main domain of SyncML is DS for mobile devices which are intermittently connected to network ser-vices. SyncML was specially designed for case, if data to be synchronized have different formats or are used in different software systems. Therefore SyncML's specification is too expensive from the development's efforts point of view, because it contains a lot of redundant options for different network systems development.

ActiveSync [3] is the software tool which allows to synchronize contacts and PIM-data [4] between mobile devices (e.g. mobile phone, communicator, pocket PC) and

server which is running on Microsoft Exchange Server platform. This software requires connection between PC and devices via USB-cable, Bluetooth or IR-port.

MobiLink [5] is the Sybase DS technology from Sybase iAnywhere product line. It is session-based synchronization technology for data exchange between relational data-bases and non-relational data storages. Mobile client using MobiLink technology was implemented for Windows Mobile, but it is not possible to use this solution on iOS platform.

One of open source DS tools is Funambol service [6] with client applications for mobile devices and personal computers, it allows to synchronize contacts, PIM- data, emails and social networks data.

Summarizing features of mentioned above technologies we can conclude that it is too difficult to adopt them by real-life mobile application development taking into account problems with different data exchange formats, communication protocols, etc. Also usage of such tools is actually related with redundant data storage on client's application, and also needs to cleanup non-actual data on the server side. Moreover usage of existing technologies for the DS is expensive from the development costs point of view in rather small mobile software projects.

3 Formal Definitions and Algorithm for Contact Data Synchronization

Proposed DS approach can be represented in a formal way using the following definitions.

Definition 1. *Data synchronization (DS)* is a process, which is given by the tuple

$$DS = \langle XMF, \quad Ph, \quad I \rangle, \tag{2}$$

where XMF is contact identifiers mapping file (see for details Def. 2);

Ph is a set of synchronization phases (see for details Def. 5);

I is a set of queries to a synchronization server (see Def. 6).

Definition 2. *Mapping file (XMF)* is represented by the tuple

$$XMF = \langle CM, \quad LST \rangle, \tag{3}$$

where CM is a set of contact identifiers (see for details Def. 3);

LST is a set of synchronization timestamps (see for details Def. 4).

Definition 3. *Contacts mapping (CM)* is contacts identifiers mapping, represented by subset of Cartesian product

$$CM \subset SrvAb \times ClnAb, \tag{4}$$

where $SrvAb = \{s_k\}, k = \overline{1,m}$ is a set of contacts from server's address book;

$ClnAb = \{a_j\}, j = \overline{1,n}$ is a set of contacts from client's address book.

Definition 4. *Last synchronization time stamp (LST)* is a time of last contacts synchronization,

$$LST = \{t_l\}, \begin{cases} LST = \varnothing, if\ no\ DS\ provided\ yet \\ LST = \{t_1\}, if\ DS\ already\ provided \end{cases}. \quad (5)$$

Definition 5. Synchronization phases (Ph) are procedures to be performed in synchronization process, represented by a following set

$$Ph = \{p_i\}, i = \overline{1,4}, \quad (6)$$

where each phase p_i is a subset of create(C)-, read(R)-, update(U)- and delete(D)-operations:

$$p_i \subseteq \{C, \quad R, \quad U, \quad D\}. \quad (7)$$

Definition 6. Request to synchronization server (I) – is a set of data types, which are transferred from client to server:

$$I \subseteq Q \times W, \quad (8)$$

where $Q = \{q_v\}, v = \overline{1,N}$, N – set of natural numbers;

$W = \{w_z\}, z = \overline{1,3}$, e.g. $W = \{timestamp, contact\ identifier, contact\}$.

Algorithm which is proposed includes following four main phases.

Phase I: fetch modified contacts in SrvAB; modify contacts in ClnAB, update XMF;

Phase II: upload locally modified contacts to a server; retrieve new contact from SrvAB_ID; update XMF with obtained IDs.

Phase III: obtain from XMF the removed ClnAB_IDs; fetch contacts from server with these IDs; *user action request:* restore/delete contacts; *if restore:* insert contacts in ClnAB and update XMF; *if delete:* remove contacts from SrvAB and XMF.

Phase IV: send IDs from XMF to a server; find non-existent contacts in SrvAB by retrieved IDs, return them to a client; *user action request:* restore/delete contacts; *if restore:* upload contacts to a server, update XMF with IDs; *if delete:* delete contacts from ClnAB, update XMF; finally show synchronization report.

In Fig. 2 a sequence diagram of proposed approach in UML 2.0 notation is shown. This view describes a sequence of interactions between User, Client application on mobile device, and Synchronization server. Main synchronization phases are represented as a sequence of method calls or messages. Synchronization process starts after user interaction with mobile device, while synchronization in progress a mobile application displays dialog's confirmation messages, they allow to get control information for user. Interactions between Client application and Synchronization server are method invocations in RESTFull service.

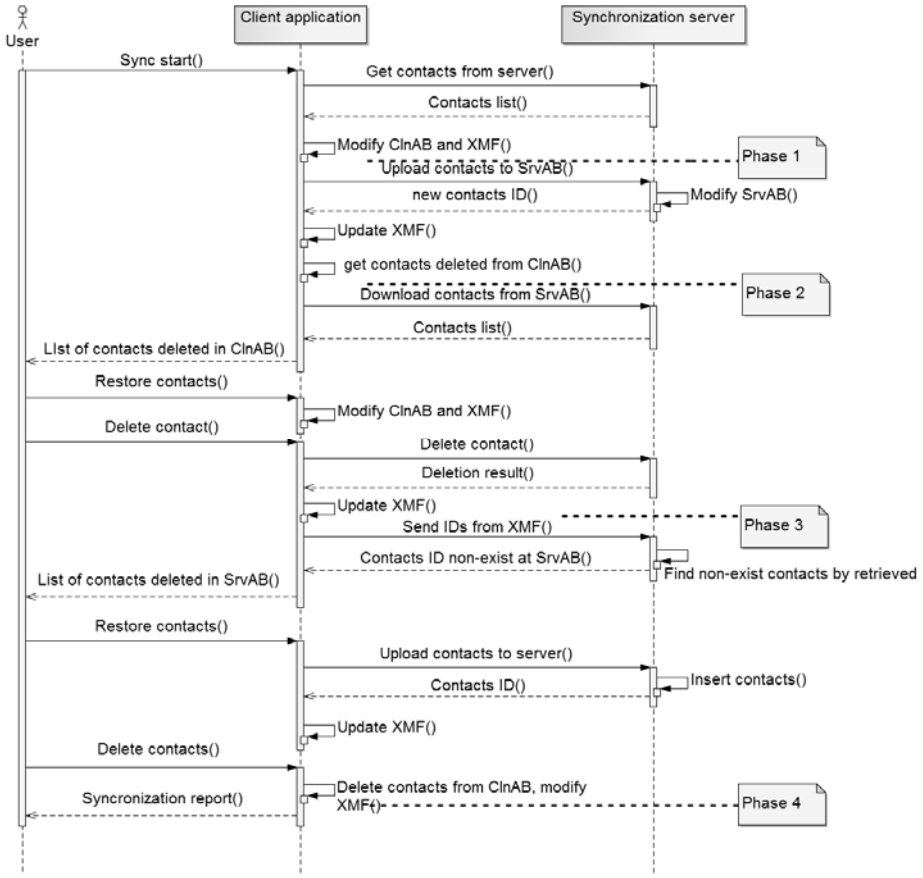


Fig. 2. Synchronization interaction sequence.

In order to provide DS process the appropriate data models, methods and software tools have to be elaborated.

4 Prototype Implementation and Complexity Estimation of Proposed Approach

On Fig. 3 the deployment diagram of typical mobile application including new software components is shown. Main nodes at this diagram are: the Synchronization server and the iPhone as a mobile device. At iPhone side a mobile application is deployed, which consists of following components: 1) *ViewController* is an application controller, which handles events, invokes *Model* and *ABProvider*; 2) *Model* is a component, which implements business logic of mobile application; 3) *ABProvider* is a component to provide access to iPhone's local address book and to

implement CRUD-operations; 4) *SyncServiceProvider* is a component to access remote REST-service and to utilize preparing request and parsing response; 5) *MappingFile* is a component accesses XMF. At the Server side the *SyncServer* node is presented with following components: 1) *SyncService* is a RESTFull-service implemented using C# and WCF technology to access server data with CRUD-operations; 2) *Centralized* database is a central storage of contacts data. As communication protocol the HTTP is used.

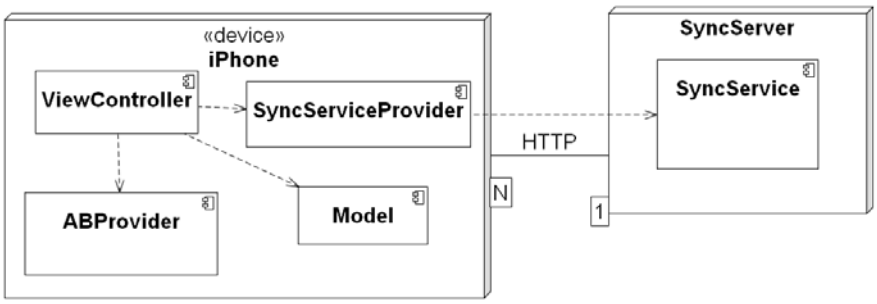


Fig. 3. The main nodes and components of proposed approach.

In a case of usage proposed approach next several components have to be implemented: 1) *XMF* mapping file, to store identifiers; 2) *SyncServiceProvider* synchronization service client. That is why from our point of view the proposed approach has less complexity as compared with another DS tools, e.g. like SyncML and Funambol (see Section 2).

In order to compare these approaches correctly, we need to describe typical software components for each tool in the same notation, and to estimate their complexity in some way.

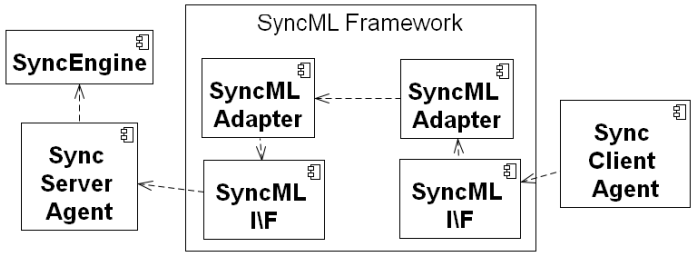


Fig. 4. SyncML-based DS approach components.

E.g., in case of SyncML framework there are some components are needed to implement for DS process: 1) 2 SyncML-adapters; 2) SyncML-engine; 3) SyncML client agent; 4) 2 SyncML I\F (API to SyncML-adapter); 5) SyncML server agent.

The SyncML based DS-architecture is shown in Fig. 4 as the UML component diagram.

In case of Funambol tool it is also needed some components to be implemented: 1) a SyncML-adapter, 2) Input and Output synclets (Java adapter classes); 3) Synchronization Sources (BTW: additionally some back-end classes have to be used in this approach, but they should not be taken into account for our comparison). In Fig. 5 the components of Funambol-based solution are shown [7].

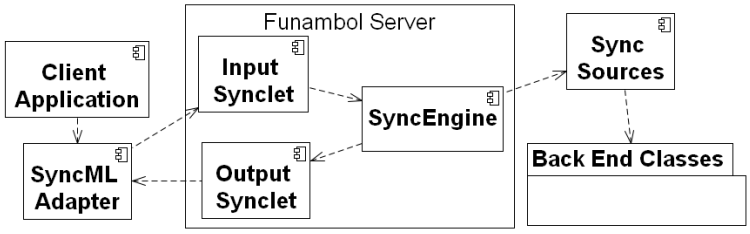


Fig. 5. Funambol-based DS approach components.

As usual frameworks have to be compared by their functionality and structures (see e.g. in [8, 9]), so to calculate complexity it is possible to calculate number of components, divided in some groups. In our case there are two groups of components: the components which have to be implemented, and the components which are already available. To calculate complexity the following formal expression can be used:

$$C = \sum_{i=1}^{z} \rho_i \cdot c_i \,, \tag{9}$$

where: z is the number of component groups;

ρ_i is the weighting coefficients for components of i-th group;

c_1 is the number of components in i-th group.

The following test case uses weight coefficients calculated with Analytic Hierarchy Process (AHP) method [10], the appropriate coefficients are: $\rho_1=0,17$, $\rho_2=0,83$. With respect to expression (8) there are the following final values of complexity estimation:

1. for proposed approach: $C_{proposed} = 0,17 \cdot 2 + 0,83 \cdot 0 = 0,34$;
2. for SyncML framework: $C_{SyncML} = 0,17 \cdot 7 + 0,83 \cdot 0 = 1,19$;
3. for Funambol tool: $C_{Funambol} = 0,17 \cdot 4 + 0,83 \cdot 1 = 1,51$.

Results of complexity estimation expose that the proposed approach has less complexity than SyncML and Funambol both.

5 Conclusions and Future Work

We have presented the lightweight approach for contact data synchronization in mobile social networks, which allows to reduce development costs and to elaborate reusable software solutions. Of course, there are some problems in our approach which were not discussed in this paper. For example, we did not take into account the fact that contact data in any social network are surely private information, therefore a correct synchronization procedure has to provide an appropriate data security options, etc. Another critical issue in the proposed approach is an intensive data exchange process between a lot of client applications and centralized data storage, and it can lead to "bottleneck" effect in the synchronization framework. That is why in future work we are going to solve these problems, and additionally to improve our approach in the way of advanced analysis of contacts data to be synchronized in order to prevent possible semantic errors and data missing.

References

1. Mobile Platform Benchmarks. A Methodology for Evaluating Mobile Computing Devices. Daniel McKenna. Transmeta Corporation (2000)
2. SyncML Data Synchronization Protocol, http://www.openmobilealliance.org/Technical/release_program/ds_v1_2_2.aspx
3. Microsoft Active Sync, http://msdn.microsoft.com/en-us/library/aa913903.aspx
4. The Return of the PDA, http://memex.org/thereturnofthepda.html
5. Designing Mobile Applications: Why Sync Is Central, Sybase iAnywhere, (2007)
6. Funambol Project, http://www.funambol.com
7. Funambol Documentation, https://www.forge.funambol.org/download/documentation.html
8. Compare JavaScript framework, http://www.ibm.com/developerworks/web/library/wa-jsframeworks/
9. Best Web Frameworks, http://www.bestwebframeworks.com/
10. Saaty, T.L.: The Analytic Hierarchy Process: Planning, Priority Setting, Resource Allocation. McGraw-Hill (1980)

Virtual Laboratory for Distance Learning: Conceptual Design and Technology Choices

Evgen Kozlovsky[1] and Hennadiy Kravtsov[1]

[1]Kherson State University, , 40 r. Zhovtnya 27, Kherson, Ukraine
{evgen, kgm}@ksu.ks.ua

Abstract: Questions of designing and a choice of technologies of creation of virtual laboratory for the distance learning system are considered. Distance learning system «Kherson Virtual University» is used as illustration.

Keywords: distance learning system, virtual laboratory, Rich Internet Application.

1 Introduction

Objects of research are systems distance learning systems and virtual laboratories (VL). Research objective is designing and description of technology choices of virtual laboratory development in distance learning system for expansion of educational possibilities of distance education.

Virtualization from the point of view of education is a process and result of communicative interaction of subjects and objects of education in the virtual educational environment which specificity of the maintenance is defined by concrete subjects and objects during the interaction.

The wide concept of virtuality should be narrowed to area of the concepts describing virtuality from positions of processes modeling in a concrete subject domain. Virtuality is a set of prospective interactions of real objects representations among themselves, and also possible consequences, i.e. conditional interactions of concepts images and objects formally represented as real.

Concept "laboratory" is a place specially organized for execution of laboratory works, experiments and a place for search of decisions in the field of fundamental sciences, or for the tasks solutions in a certain application area of knowledge.

The concept of the virtual laboratory has not the accurate settled definition, therefore often companies interpret it at own discretion, and they adjust it under concrete needs of their products. We will introduce concept of virtual laboratory, being based on the resulted definitions. So, the virtual laboratory is a virtual environment in which the possibility of research of conducts of models of objects, their sets and the derivatives, set with the certain stake of detailing in relation to the real objects, in a certain field of knowledge is organized.

Analysis of the Internet resources shows that the overwhelming majority of projects of virtual laboratories are not conforming to the concept of laboratory. More

frequent it is Internet sites where the texts of laboratory works are present, supported by media maintenance. Frequently only one process without management possibility is displayed in accompaniment. If in the context of such "laboratories" are any objects they are very limited in possibilities, and they are procedural programs, but not objects, without possibility to modify or to complement the motion of going processes.

That is, there is a certain discrepancy of purpose and the name of the majority of the Internet resources – «virtual laboratories». From this follows the research problematic – creation of the project and development of the software of virtual laboratory.

Besides, it is necessary to notice, that modern IT possibilities allow to expand concept of virtual laboratory – to apply it not only for fundamental sciences but also for more wide range of fields of knowledge, for example rendering support at studying of arts, linguistics, the rights, philology, etc.

In department of multimedia and distance learning technologies KSU there are already preconditions to creation of such virtual laboratory. In the asset of department there is an experience of creation of declarative virtual systems of processes modeling, in particular, the demo-project «Electrotechnics» in which the algorithms of creation and an estimation of quality of interaction processes in electric circuit are laid. The distance course "Cytology" is the software product the purpose of which is support of process of execution of virtual educational laboratory works on the course of the discipline with the same name. Besides, there is an experience of creation of virtual educational training simulators, such as, a microscope and the electric motor, allowing studying the details of construction and principles of its work. Also, there is a long-term experience of teaching of discipline «Modeling of physical processes», students create virtual models for calculation and visualization of processes in various sections of course of physics under the guidance of employees of department. Thus, we have concepts of creation of virtual laboratory which can unite in itself support training process of students on the whole spectrum of fundamental and applied disciplines.

As we understand, the virtual laboratory should function inside the system of distance learning. Using the virtual laboratory works is capable to expand qualitatively possibilities of distance learning, and also effectively support execution of practical works.

Interviewing of teachers and students – users of distance learning system «Kherson virtual university» (DLS KVU) has shown, that answers for the question «What tutorials do you prefer?» show the positive growth of requirement and interest to non-standard approaches in education (Fig. 1). There is already a category of people who consider that the possibilities, which are integrated in the standards of distance learning (DL), are not enough today. The wider toolkit for execution of virtual laboratory works is needed for active users of DL.

Progress of DLS has caused the transferring to the virtual environment the basic forms and ways of education - the groups, courses, tests, libraries, virtual rooms of communication, virtual boards with wide exposition facilities of information. The following stage of development DLS is obvious - it is necessity for creation of the universal virtual laboratory, which will have the digital analogues of laboratory offices of university, with all necessary tools. In such laboratory the support of

scientific researches of students and the control at all stages of cognitive process are provided.

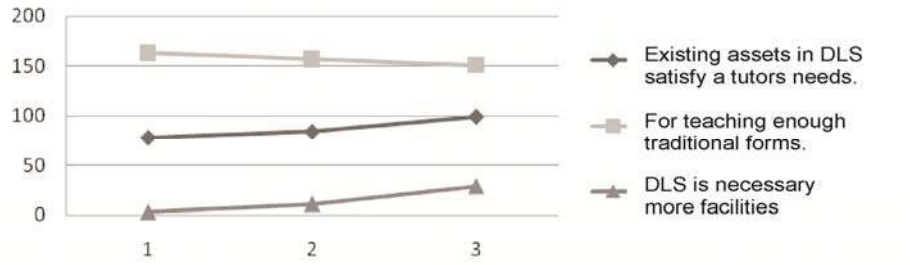


Fig. 1. Results of interviewing of users DLS KVU.

2 Analysis of existence of interaction between DLS and virtual laboratories

There is a big variety of DLS. The systems are free-extended, brand and author's systems. All systems are developed according to the international standards of distance learning, which allow providing the compatibility of components and the possibility of their repeated use in other systems, guaranteeing their invariance and unequivocal treatment. In opinion of the authors of DLS KVU any system has viability if it completely conforms to standards, and also it goes on a step ahead, introducing new educational means, it expands possibilities for the users, it introduces innovations which will not contradict the basis – standards, such as IMS and SCORM.

In the standards of DL virtual laboratories are not regulated, there is only a generalized description of objects of education. Requirements to such objects of training are in two key concepts: first of all the property of isolation, as a black box with the entrance and target strict-designated parameters, they are parameters of initialization of object and the parameters describing results of work. The educational resource should correspond to the standard at level of statement of an educational task, provide execution of educational task and should give an estimation of results, for the account in the rating estimation system. Packing of this type of a resource for carrying over to other educational system should be provided. The system resource will not contradict the standard at performance of these conditions [2].

The analysis of experience of introduction of virtual laboratories in DLS shows, that the problem of development the virtual laboratory as a part of DLS was yet not set by developers.

More often, laboratories reflect only one field of knowledge, usually it is sections of classical sciences, such as an electricity, analytical chemistry, optics etc. Also it is necessary to notice, overwhelming number of laboratories are in a format of distributive local software products which are delivered on separate computers, or on a small local area network, school or high school.

On-line Laboratories are presented in smaller volume and variety. Thus this class of software products is actively developed, superseding off-line versions from the market of educational.

On-line Laboratories are presented in smaller volume and variety. Thus this type of software products is actively developed, superseding from the educational market off-line resources.

Both these of type of virtual laboratories are frequently used limited, not in distance learning system, but only as additional facultative resources. In them there are not or partially realized modules of support of educational process. Usually the control of knowledge is realized in the form of self-checking questions, or the elementary test, the results of modeling are simply displayed to the user, but are not gotten to group journal of progress automatically.

The virtual educational and engineering laboratories of the industrial enterprises intended for automation of engineering work are the separate class. They are used in highly specialized areas of engineering science. These laboratory cabinets model real industrial problems of the concrete enterprise with high level of detailed elaboration. The main attention is usually given to realization of procedures of mathematical modeling, calculation and optimization of studied objects or processes during their creation [2].

During projecting and development of virtual laboratory the most effective approach will be focused on this class of software products, they have powerful analytical and mathematical appliance, the detailed level of imitation of processes.

The urgency of creation of virtual laboratory is caused not only development of DL. There are a number of disciplines in which laboratory researches mean considerable charges of educational institutions on machine tools, tools, preparations, reactants, etc. Besides as practice shows, students can't execute laboratory work correctly, after theoretical preparation in first time. Often for a successful execute the laboratory work it is necessary to make experiments several times, and then the expenses can essentially increase. Besides, in educational institutions, far not always there are means for purchase and completion of laboratory offices by all necessary. Virtual models are used in cases when the experiment is dangerous, expensive, occurs in inconvenient scale of space and time (too long-term, too short-term, lingered), is impossible, unique, not visual etc. [3]. In these cases the virtual laboratories become the necessity, it gives the chance to spend educational process with smaller expenses, or it will allow saving means, preparing students at first on special training apparatus, for the subsequent transition for real expensive laboratory stands.

In our opinion, the virtual laboratory is capable to expand a spectrum of given educational services considerably. Its use will allow expanding possibilities of interaction of all participants of educational process. The authors put the objective – to create virtual laboratory as an educational resource in DLS «Kherson virtual university».

3 System architecture

3.1 General approaches of realization and place of virtual laboratory in DLS

Modern possibilities of network technologies, universal prevalence of high bandwidth telecommunication channels, and also the width of possibilities of means of programming, in which it is comfortably to develop the web-oriented software products, allow saying that virtual educational means are expedient to create in the form of web-services.

Considering that fact that DLS KVU is the Internet application, the designing and development of VL should be executed as a web-oriented application. Though, it is not necessary to reject the possibility of creation of the software adapted under a local network with possibility at work to contact a server for data exchange.

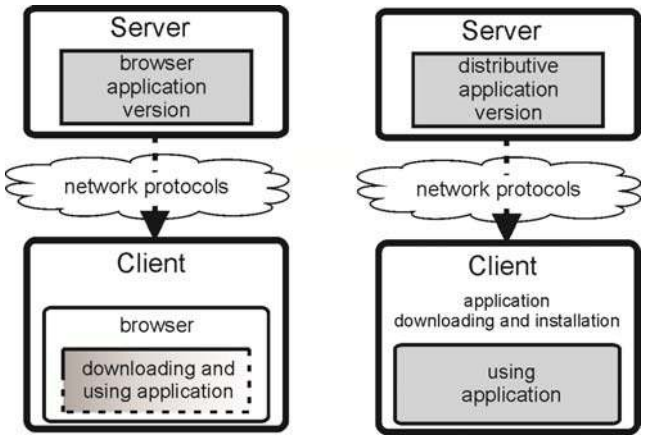


Fig. 2. Architecture web-oriented and locally network applications.

In fig. 2 the basic approaches of work of client-server applications are presented. In fig. 2 (*a*) the model of the web-oriented applications in which the client receives and uses the freshest versions of the software is presented. In fig. 2 (*b*) the model of locally network applications in which the software is installed on the computer of the user is presented. The user should contact the developer and receive corresponding modules of updating for reception of updating of the software.

The virtual laboratory has considerable volume of the information which at the first approach can demand much disk space and operative memory that will occupy definitely superfluous resources at each application launch. Frequently all laboratory modules are not necessary for user at once, in this case an optimum variant is to make possibility for loading only the basic environment of the certain working module and

necessary section of library of objects. Then the user will save the traffic of resources and can increase the effectiveness of work.

It is necessary to notice, that in both cases there are advantages and lacks, therefore the realization of both ways of installation of applications will be the best variant then the concrete user can choose the way of installation more suitable for him, considering its personal needs.

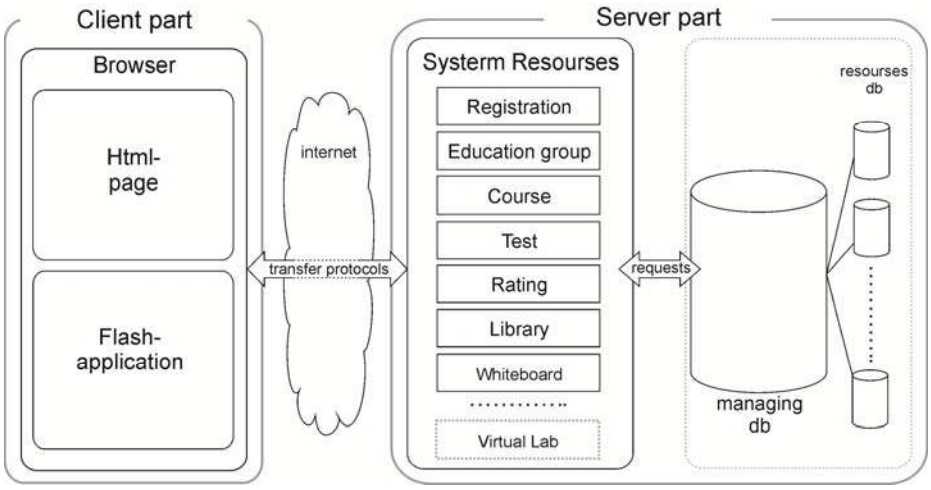


Fig. 3. Virtual laboratory as a resource in DLS «Kherson virtual university».

In fig. 3 the place of virtual laboratory in DLS «Kherson virtual university» is shown. The laboratory is considered by developers as one of the important resources of system. Educational and the control facilities are an integral part of educational process in the distance form.

Developers put the task to create such mechanisms:

1. Possibility of creation of the working model or construction
2. Possibility of carrying out of various transformations and change of statuses (editing) of model or construction
3. Possibility to carry out necessary calculations and measurements of parameters of model or construction virtual measuring devices

Let's consider the method of creation of the module of virtual laboratory in the form of the client-server application in format Rich Internet Application (RIA) [4]. The choice of this model is connected with advantages which are given by this format to developers and users. First of all it is necessary to note the possibility of the user to work on any computer with Internet connection in any browser with installed module Flash Player. The interface and level of interaction of the user with RIA application can be similar with functionality of traditional desktop applications. Developers and designers receive freedom in realization of ideas, considering specificity of concrete area of knowledge.

Today level of use of module Flash Player of users is 96 % [5]. Universal prevalence of Flash-technologies allows speaking about availability to users of such approach in realization of the module of virtual laboratory.

There is a wide variety of developing tools for creating RIA applications. These are such technologies as Adobe Flash\Flex, Backbase, Google'sGWT framework, java applications, Microsoft ActiveX controls, Microsoft Silverlight, etc. [4] Our choose of technology for developing client application is Adobe Flex, firstly of all because it installed on the wide prevalence on computers of users, and also it has full multimedia possibilities of Flash that allows to open the essence of modeling of processes VL widely.

3.2 Scheme of components of the module «Virtual laboratory»

In fig.4 the scheme represents the architecture of the program module «Virtual laboratory» is presented. In the server part of the application there are three basic components – the module kernel, the connected databases and the mathematical processor.

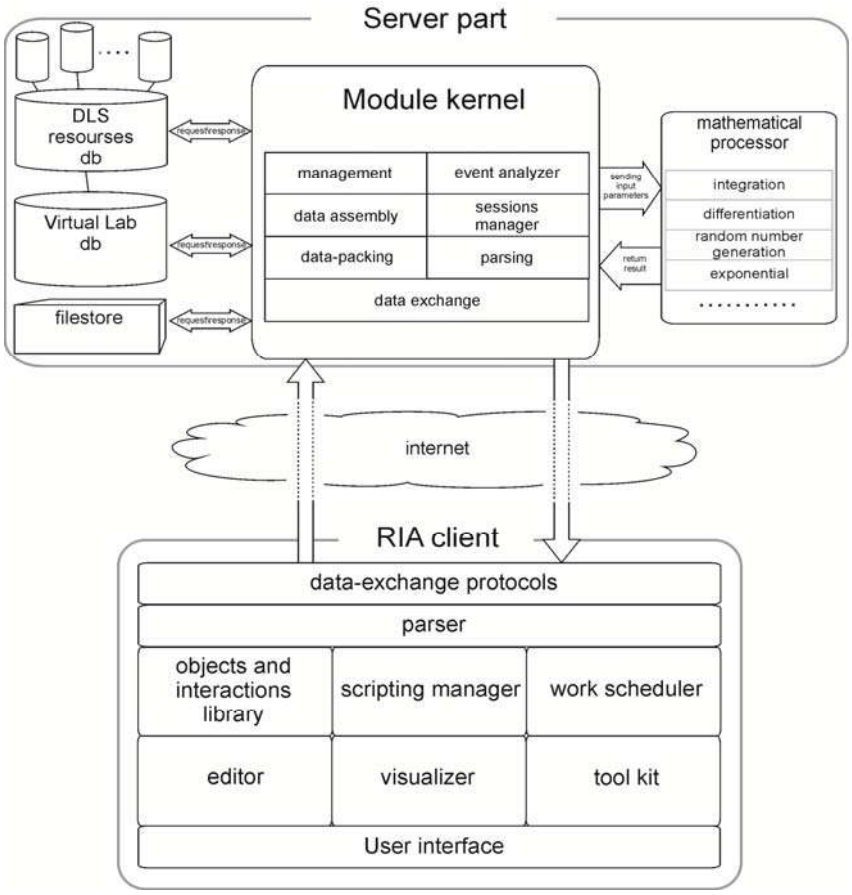


Fig. 4. The scheme of model of virtual laboratory.

The set of the basic classes of management are In the module kernel:

1. "Management" is responsible for the rights of accounts of users, provides synchronization of resources of users, and connects options of the components used by the concrete user.
2. «Event analyzer» writes down, orders and stores all actions of users for the subsequent reproduction.
3. «Data assembly» provides creation of packages of data and files of resources for library of the user with the subsequent sending in repository on a server at the start of the work.
4. «Sessions manager» creates virtual rooms, connects to them users, and provides authorization and administration of rooms. It is responsible for reception and data transmission between users and a server.
5. The module «Data-packing» creates packages from resources which will be sent to the client, and also for maintenance of integrity of data and creates metadata of these packages of resources.
6. Module "Parsing" provides transformation of arriving inquiries and messages from the client application and transfer of given data to operating modules.
7. "Data exchange" is the interface of web services of interaction with client applications.

In module VL databases of DLS «Kherson virtual university» are used. First of all, it is base of authorization, statistics, also sending of data about progress of users in base estimation is realized, etc. Except used DLS bases, the module has own database, where records about job of all components of the module, the information about objects, records of events, and also links with storehouses of resources of libraries are stored.

The mathematical processor serves for maintenance of difficult mathematical calculations which cannot be carried out by client computers. These are calculations of the solution of the equations of the higher orders, the calculations demanding use of mathematical methods of correlation, specification, iterations, and calculations of errors, an analysis, and others.

The client part of the software of VL has components for creation and work with virtual models:

1. «Objects and interactions library» contains sets of objects of a certain subject domain, and also the ways of the interaction defined by the current field of knowledge.
2. «Scripting manager» allows adjusting interactions between objects, setting their parameters, imposing conditions. Also it is intended for the description of properties and interactions of the new objects created in the editor.
3. «Work scheduler» has two operating modes. In elapsing mode of laboratory research the user should according to offered course of work make all necessary operations for result reception. In the mode of creation of new work this component writes down history of events, allows to correct them, and to form the scenario of performance of work.
4. In "editor" VL the designing of the new objects is made. Users can on the basis of existing VL of library, preparing them, create other objects. Using in parallel the component of scripting, users can specify and finish new parameters and methods of interaction with other objects, finish new ways of behavior.

5. "Visualizer" is the component in which there is laboratory research, provides display of all designs of objects. In this module the current statistical data, such as values of parameters of system are also displayed, resultants of a direction of movements are specified. Directly here there is a process of modeling and management of a work scenario.
6. "Tool kit" is control facilities of user and changes of the objects which are in components the Visualizer and the Editor.
7. «User interface» – is set of methods of display and interaction of the user and system.

3.3 Description of job of users with the interface of the program module «Virtual laboratory» in DLS KVU

In DLS KVU there are two basic roles of users which work in virtual laboratory - tutor and student. Working with module VL is carried out in two modes: work of the tutor in editor VL in a mode of creation of virtual laboratory work and work of the student in the training mode in VL. The tutor has the rights to create new models in the editor and to look through them, and the student can use ready model during work in VL.

Process of work of the tutor with VL begins with creation of new model of virtual lab work (VLW) in a window of the editor. First of all, the tutor always bases the educational activity on the working program of discipline. In the working program training basic elements, such as lectures, practical and laboratory works are reflected. The first step of creation of virtual laboratory work is creation of scenario. At this stage the tutor creates scenario VLW in the form of algorithm which realizes during laboratory work.

Then, the tutor consistently fills the model of VLW with objects from library VL, realizing the scenario of its performance. The tutor sets to objects parameters, methods of interactions, designs from them blocks, and thus forms definitive scenario VLW.

When the new laboratory work is completely designed, it is passed to the server which processes it and compiles ready model of laboratory work.

The student at performance of VLW has the necessary set of resources of training: tools, objects for work according to the scenario. In conformity with the requirement of isolation of module VLW, it should provide an estimation of quality of performance VLW. Therefore, after performance of laboratory work by the student, the module of performance VLW passes results of work in the form of estimation to a server.

Thus, the process of work in module VL allows to model educational process in real research laboratory, to create, to train and make control of the work.

4 Conclusions

In the article the project of model is developed and technologies of designing the program module «Virtual laboratory» for distance learning system «Kherson virtual

university» are described. The technology of designing of program module VL is based on the client-server application in format Rich Internet Application.

References

1. Kravtsov, H., Kravtsov, D.: Knowledge Control Model of Distance Learning System on IMS Standard / Innovative Techniques in Instruction Technology, E-learning, E-assessment, and Education. – Springer Science + Business Media V.B. pp. 195—198 (2008)
2. Solovov, A.V.: Virtual educational laboratories in an engineering education. The education industry. Release 2. – M: MGIU, pp. 386—392 (2002)
3. Mukhin O.I. Modelling of systems, http://stratum.ac.ru/textbooks/modelir
4. WorldLingo Multilingual Archive. RIA, http://www.worldlingo.com/ma/enwiki/en/Rich_Internet_application
5. Comprehensive Aggregate Internet Usage Statistics, http://www.statowl.com/custom_ria_market_penetration.php
6. Kolesov J.B., Senichenkov J.B.: Imitating modelling of difficult dynamic systems, http://www.exponenta.ru/

Methodological and Didactical Aspects of Teaching ICT

Practice in Software Engineering course: "what and how to study"

Yuriy Solyanik, Maryna Vladymyrova,
Iryna Zarets'ka, and Grygoriy Zholtkevych,

V.N. Karazin Kharkiv National University,
4, Svobody Sqr., Kharkiv, 61077, Ukraine
u.solyanik@gmail.com, vladymyrova@gmail.com,
zar@univer.kharkov.ua, zholtkevych@univer.kharkov.ua

Abstract. The goal of this paper is to share the experience of V. N. Karazin Kharkiv National University in Software Engineering training, especially in organizing students' practical work so that they gain competences required by modern software industry based on world standards.

As software development became now comprehensive industry there is strong demand for highly qualified specialists all over the world. As any industry it is based on standards for products as well as for processes. So any university graduate planning to work in this industry should know these standards and be able to work with them including tailoring to the concrete situation. It is especially true for the graduates majoring in Computer Science (CS). Usually these knowledge and skills are taught in the course of Software Engineering (SE) which is adjourned to the senior years when all the fundamental concepts of CS and technological skills have been already gained. Unfortunately all manuals on SE studies including Computing Curricula present only core topics to discuss and learn but no hints on how to organize practice with visible results of students' growing as specialists just ready for industrial work. It is up to the University to decide the ways to develop such competences in graduates. Our university developed its own approach to provide necessary competencies so the goal of this paper is to share our views and experience with other universities as well as to have feedback as to advantages and disadvantages of our approach.

There are several definitions of SE. Let us take this one: "Software engineering is an engineering discipline that is concerned with all aspects of software production from the early stages of system specification to maintaining the system after it has gone into use" [1]. Another one tells that SE encompasses knowledge, methods and tools for defining software requirements and performing software design, software construction, software testing and maintenance tasks. Anyway as any engineering discipline SE is regulated by number of normative documents and standards. Among world leading standardization organizations in this field are ISO, IEC, IEEE, ESA. Their standards concerning SE cover all the parts of this discipline. It is just natural to build the subject of SE on the foundation of the normative documents and standards. Studying the majority of standards

in the course of SE we base our practice mostly on the ESA standards [2] as they usually include all the information from similar standards of other organizations but are more verified as they deal with critical software. According to these standards we divide Software Life Cycle (SLC) onto six phases which are User Requirements phase (UR), Software Requirements phase (SR), Architectural Design phase (AD), Detailed Design phase (DD), Transfer phase (TR) and Operation and Maintenance phase (OM). It is essential to teach SE in such a way that students could go through each phase of SLC performing its tasks and preparing corresponding artifacts and documents strictly in compliance with the standards. Knowing as many standards as time allows and being able to apply them practically will make it possible for students to perform agile tailoring of them depending on company policy and concrete type of project when working in industry.

The whole process of studying looks like this. According to SWEBOK [3] SE discipline consists of two big areas which are Software product engineering and Software management. In fact we have these parts in two different courses but they go in parallel with common ideas and projects to work on. Both courses are taught to the graduate students (fifth year) so the main goal of both of them is to systemize and generalize all the knowledge and skills gained by students before via series of conceptual lectures made by students themselves in form of presentation and practical work of real industrial level. So we organize semester long business game on SE (both parts) with students working in teams on real projects and playing different roles during the process of development. In fact they all go through being business analysts, system analysts, system architects, quality assurance personnel, team leaders, project managers and technical writers. Moreover they usually work on projects that are needed by University subdivisions or have real customers and timescales with several students of younger ages (usually fourth and third years of study) to their subordination making exploratory and experimental prototyping, coding, unit and integration testing, etc., which can be estimated for them as course work or even bachelor project. This heightens the responsibility of graduate students not only for the projects to be done in time but for undergraduates to have good marks under their leadership not to mention their own marks on the SE subjects. All graduate students are divided into teams consisting usually of 3 to 4 graduate students (depending on the project scope) plus 2 to 3 undergraduates. Each team works on a separate project but reviews the project of its peer team. The process of peer reviewing is not less important than working on their own project as it allows students to see mistakes and blunders as well as successful features more clearly. All the steps, activities and solutions are thoroughly documented and reviewed which at the end gives the full picture of students' progress and results.

The topics of conceptual lectures in SE are presented in the table. In fact all the key areas and units are well presented in SWEBOK [3], PMBOK [4] and Computing Curricula [5, 6] so we only packed them into topics and added some modern technological aspects [7]. In fact we use a lot of standards and literature in this course, it will take several pages to present all of them, so we put only

those we directly refer to in this paper into the list of literature below.

Software Product Engineering		**Software Management**	
Topic	**Hours**	**Topic**	**Hours**
Taxonomy of standards in SE	1	State of the art in SE and Software Management. Software Project Management, Software Quality Management, Software Configuration Management. Four Ps: Project, Product, Process, Personnel	2
SLC concepts and models (including classical and modern ones like Agile, SCRUM, KANBAN, etc.)	2	Project goals. Project processes scheme. Six management processes. Business-planning of the project	1
User requirements elicitation methods, tools and artifacts	2	Organizational management.	2
Software requirements specification methods, tools and artifacts	2	Personnel and project environment management. General information about supporting processes. Communication management	2
System models, types and classification	2	Planning. Standards of planning. Different representations of the plan (Gant diagrams, network planning, etc.). Creating the plan. Diagrams analysis. Critical path, critical chain. Survey of tools SPMP	4
UML 2.x, history, development, usage	3	Infrastructure of planning: database of processes, base line of process stability, processes' assets. Methods of possibilities evaluation (PSP, TSP, CMM)	1
Formal methods of requirements specification	2	Software Management. Configuration. Standards, basic concepts, methods, tools SCMP	4
Quality of software, metrics of quality	2	Software size metrics. Methods of data collection. Software size evaluation	4

Software Product Engineering		**Software Management**	
Topic	**Hours**	**Topic**	**Hours**
Software architecture, architectural styles and patterns, POSA, MDA, SOA	4	Software cost evaluation. Standards, models (COCOMO,COCOMO2, SLIM, etc), tools	4
Architectural design processes, methods, tools and artifacts	2	Risks. Uncertainties. Standards. Risk management and control. Quantitative and qualitative risk analysis. Models, methods, tools. Risks planning	4
Verification and validation of software, methods, tools and artifacts	4	Quality Management. General principles. Product quality. Quality Assurance (QA) organization: organizational structure of the project. SQMP	4
Detailed design processes, methods, tools and artifacts	2	Software reliability evaluation	3
Transfer, operation and maintenance processes, methods, tools and artifacts	2	Project closing, analysis and summaries	1

Now about the practical work. We prepare a number of real projects with real customers and timescales (usually semester long to develop and further to continue coding, testing and maintaining by undergraduates from the corresponding team). After teams were formed and projects selected by teams they prepare vision documents in standard form while several first lectures acquaint them with the main concepts and activities to follow. Then they proceed with the projects going iteratively through each phase of SLC.

The first phase - UR one - takes usually 3 to 4 weeks which is longer than needed because at this time students are not yet well immersed into subject. They work directly with customers and domain experts, prepare questionnaires and make surveys to define project scope and boundaries, assess operational environment, determine users' roles, and elicit user requirements at most thoroughness and completeness. Sometimes exploratory prototyping is needed (usually made by undergraduate members of a team). Then it takes time to learn standards and prepare the draft version of the first full document in standard form which is URD (User Requirements Document) with all requirements identified and attributed and status sheet attached. This version is reviewed by peer team and all the discrepancies found are documented and discussed during the review meeting. Then several iterations with corrected URDs and updated status sheets follow until the final version is approved and signed by both parts. At the same time the acceptance test plan is being worked out and documented.

As to the management activities students define their projects goals from the management point of view and elicit main processes, find the appropriate structure of their team organization, distribute roles and write down the responsibilities of each role. They also get acquainted with the MS Project tool.

The second phase overlaps first one and usually begins when first URD draft is submitted for the review. Second phase usually takes the same time as first or a little longer as it supposes analysis of the requirements and logical model construction as well as system requirements specification. At this time students have already studied different system models and software quality metrics, which facilitates the process of a logical model construction and working with both functional and non-functional requirements in quantitative form to be properly verified. They use various CASE tools for model construction and specification. At the end of this phase the Software Requirements Document (SRD) is prepared with obligatory traceability matrix attached. Again iterative process of documented peer reviewing and making changes takes place, which ends with the approval and signing of SRD.

As to the management activities this phase is dedicated to planning. Students plan their work on the project using MS Project tool, identify tasks, define their sequence and duration of each of them, and determine types of relations between tasks. Then they assign real terms and resources for each task, e. g. form the project's schedule and construct the critical path using MS Project tool and developing their own program to build a critical path just to compare the results. It allows them to evaluate the real duration of the project. Students also consider different solutions to the problem of resource overloading. At this phase the SPMP in standard form is being prepared and submitted for the review.

The third phase is the most hard for students because it requires all their previous knowledge and skills, sometimes they are compelled to learn new technologies and make exploratory prototyping to solve architectural problems usually rising from non-functional requirements and to apply appropriate architectural styles and patterns. They have to construct a physical model with detailed description of each system component, process, software module and physical node of software deployment which is not only difficult to perform but time consuming as well even using advanced CASE tools (not to mention regular returns to the previous documents). At the end of this phase the standard ADD (Architectural Design Document) is submitted for the review which also takes several iterations before final approval.

As to the management activities this phase is dedicated to the Configuration Management. Within the frame of their project students create repository to store all the artifacts of the project, make checkouts, updates and commitments, add files and folders, assign tags and create alternative branches of development process using TortoiseCVS and TortoiseSVN tools. They also prepare Software Configuration Management Plan (SCMP) in standard form and submit it for the review. Another students' activity at this phase concerns with software cost estimation. They use Costar and USC-COCOMOII 1999.0 with CostXpert tools to

estimate the cost of software for the early design model and for post-architecture. For their projects students assign the values of Scale Drivers, Cost Drivers, Function Points or SLOC, Resource Costs and make necessary calculations and reports. The SPMP draft made at the first phase is now being improved with reference to the risk plan developed further.

The fourth phase heavily overlaps the third one as it is usually done by undergraduate members of the team and starts just as the critical architectural solutions have been made. It consists of coding, unit and integration testing with all documentation required by standards. At this time graduates begin writing PHD (Project History Document) as time goes to the end of semester. The main output document of this phase - DDD (Detailed Design Document) - takes too much time to prepare so graduates usually only supervise its preparation and as our experience shows never have time to verify it thoroughly. This is mainly the responsibility of undergraduates to finish coding and testing as well as transfer releases to customers and maintain them during the operation before provisional acceptance and after it. At this moment we try to have new younger students to learn this software to be able to maintain it.

As to the management activities this phase is dedicated to risk control and management. Students make the risk plan for their projects which includes all steps of working with risks e. g. risk identification, risk estimation, methods of responding to risk events and control of these responses. They use PertMaster Project Risk v7.6.0006 tool for this work. At this phase they also prepare Software Quality Management Plan (SQMP), specify and improve SCMP made at the previous phase and consider problems of management according to the plans. At the end of this phase the process of project closing is being considered, analyzed and the whole project is summarized.

Certainly the time limits and kinds of projects students develop do not allow them to go practically through all the management activities and artifacts considered in theoretical study, but even what is done gives students some management skills to be developed in their further work as no industry newcomer begins his or her career as a software project manager.

As a result our graduate students learn to work with requirements and specify them, make sound and grounded architectural solutions, carry out management activities and present all information in compliance with world standards. All that is being done in a team work with peer verification and audits made by teaching staff. As to the benefits for undergraduates they learn a lot from their senior mates which is good in itself, plus get ready for analytical work next year. Sure a lot of work is done by each member of a team with all those paper artifacts being meticulously prepared according to standard forms. A lot of them might seem excessive for real middle size industrial software company using agile methods and techniques. But what we think is: it is better to master the whole process and use only parts of it than to know only some parts when it is necessary to use the whole process.

References

1. Sommervile, Ian: Software Engineering. Pearson Education, New York and London (2001)
2. ESA Software Engineering Standards. ESA Board for Software Standardization and Control (1991)
3. Software Engineering Base of Knowledge (SWEBOK). Revision of IEEE (2010)
4. A Guide to the Project Management Body of Knowledge, 4th ed. PMBOK Guide (2008)
5. Computing Curricula 2001. Computer Science. Final Report. The Joint Task Force on Computing Curricula. IEEE & ACM (2001)
6. Computing Curricula 2005. The Overview Report. The Joint Task Force on Computing Curricula IEEE & ACM (2005)
7. Futrell, R. T., Shafer, D. F., Shafer, L. I.: Quality software Project Management. Addison - Wesley Publishing Company, New York (2002)

Influence of the Labor Market upon the Forming of Competence of Future IT Specialists

Dmitriy E. Shchedrolosev

Kherson State University, 40 r. Zhovtnya 27, Kherson, Ukraine
dim@ksu.ks.ua

Abstract. The article deals with the professional competence in the structure of the personality of the future software engineer from the point of view of providing a successful career growth.

Keywords: software engineer, career, professional competence.

1 Introduction

Under conditions of rapidly changing technologies, the system of education is required to prepare a large number of IT professionals who possess a certain level of professional competence and personal qualities. Honorary Professor of Griffts University Ian Laue said that university education can no longer provide students with knowledge and skills they need in their professional career because most of these skills are simply not there [7: 67].

Since the state of national information resources and means to activate them identify potential successful development of the state, ensuring its national interests, special significance is attached to training highly qualified specialists in IT technologies that can provide the necessary IT-level of the modern society.

In modern society that is tied with IT, the system of education is required to prepare a large number of corresponding competent professionals. Training in information and communication technology must be flexible enough since professional skills that may be demanded by employers, quickly change over the years that a young person spends in training [7].

Important is the fact that modern programming is collective and individual programmer persons are closely related to their usefulness for the whole team, and therefore require individual skills of teamwork, leadership skills, some knowledge in the field of psychology and management. In our opinion, the feature of a successful IT professional is not a fixed set of knowledge and skills in a particular industry, but the formation of the competences that provide career growth. Building a learning process based on the competence approach is the most effective way for providing specialist training to meet the modern demands of society.

The aim of the article is to define factors that influence upon the formation of the IT specialist's competences and determine possibilities of their forming in the educational system of a university.

2 Analysis of the Latest Researches and Publications

Analysis of specific tasks in a matter of programming and professional skills of programmers at different times were made by psychologists and educators F. Brooks, D. Weinberg, N. Wirth, L. Grishko, A. Dijkstra, S. McConnell, M. Smulson, B. Shneyderman and others.

The question of competence as such was studied by N. Bibik, B. Elkonin, J. Kolomynsky, I. Lerner, A. Markov, P. Myasoid, N. Nechaev, A. Nikiforov, L. Petukhova, J. Raven, I. Rodyhina, M. Skatkin, G. Selevko, L. Khoruzhaya, A. Hutorskiy and others.

The author's experience as a teacher and head of IT department along with the expressed problems of training of future software engineers determined the need for the research, the purpose of which was to define the structure of the professional competence of IT professionals, and write this article.

3 Outline of the Main Research

The problem of competence approach to the preparation of future software engineers makes a clear understanding of not only substance but also the structure and features of professional competence in the field of ICT.

According to the definition in the explanatory dictionary of Russian language, competence is knowledge, credibility, competence is a range of issues, events in which the person has a high authority, knowledge, experience, and range of permissions [12]. Dictionaries give the following definitions of the term "competence": 1) knowledge, credibility; 2) a range of issues, events in which the person has a high authority, knowledge, experience, and permissions [12].

Under the authority we mean the ability and perceived willingness of individuals to implement a system of acquired knowledge, skills and desire to solve actual problems in specific circumstances with certain possibly foreseen consequences and responsibility for their actions. In our view, this definition describes the most successful IT professional competence [6]. L. Petukhova's structure offers personal competence in which the main principle is to separate fundamental entities and entities based on the abilities and personal deposit and the need for appropriate educational environment as a complex of factors, which requires adequate and timely response in the relevant areas and considers major components of competency - its expertise (knowledge, skills, abilities), goals (needs, values, motives, trains, ideals, etc.), quality (the ability to manifest synergy, adaptation, scaling and interpretation, self-development, integration and transfer of knowledge from one industry to another, etc.) [6:60-64].

Considering the professional programmer, scientists (E. Dijkstra [3] M. Smulson [9] B. Shneyderman [13]) distinguish the qualities that are inherent to programmers who are directly associated with the creation of software and human and psychological traits that should be inherent to the programmer.

Analysis of the history of technology and programming languages shows changes in requirements for knowledge, skills, thinking style and professional qualities of the

programmer. The feature of new knowledge in programming is that they are built on new technology, high level of abstraction, sometimes on a synthesis of old knowledge, unlike in medicine, physics, and law, where new knowledge is deepening and refining of the old one.

Under the conditions of rapidly changing technologies, the education is required to prepare a large number of IT professionals who possess a certain level of professional competence and personal qualities. However, training in information and communication technology must be flexible because professional skills that may be demanded by employers quickly change over the years that a young person spends in training [8].

In our opinion, the main factor that determines the success of any project lies in people, programmers in our case. The impact on people in organizations is determined by two main factors: the hierarchical structure of the organization and corporate culture of the organization produced by common values, social norms, and behavior settings that regulate social norms. For a programmer understanding himself as part of the team is vital, which is particularly important in the process of working at a huge project when the development team includes specialists from various fields in order to fully consider all the system capabilities when making any decisions. In order to receive recognition by the partners, achieve high self-esteem, harmonious teamwork, every programmer has to go all the way from simply making money.

Professional activities of a programmer in the company can be classified according to the following:

- Skill level - junior, middle, senior
- Technological direction - Java,. Net, C + + ...
- Role in the project – project manager, analyst, architect, technical leader, developer, configuration manager, quality manager, quality engineer, expert in customer relations

Researching of articles, forums, blogs and personal experience of working with a team of IT professionals indicates that in order to support professional level, a programmer must constantly be aware of many new technologies, new methods to know the solution of certain problems, fully replenish their knowledge and skills. Usually in the first five to seven years it is rather fluent, which is associated with features of age (the human brain easily learns new information at the age of 18-25 years). Further professional career of a programmer has certain characteristics, so graduates have to be aware of their own future prospects straight from the start either to become managers, heads the team of web development, academics, or change specialty.

Training in information and communication technology must be flexible enough since professional skills that may be demanded by employers, quickly change over the years that a young person spends in training.

To determine what competencies and personal qualities are required by the future software engineer for a successful career, we'll consider requirements of today's employer for each step of the career of IT professionals by giving an example of snippet of grades of one international IT company (Fig. 1). The figure offers one of possible variants of roles distribution structure, which can vary from one project to another.

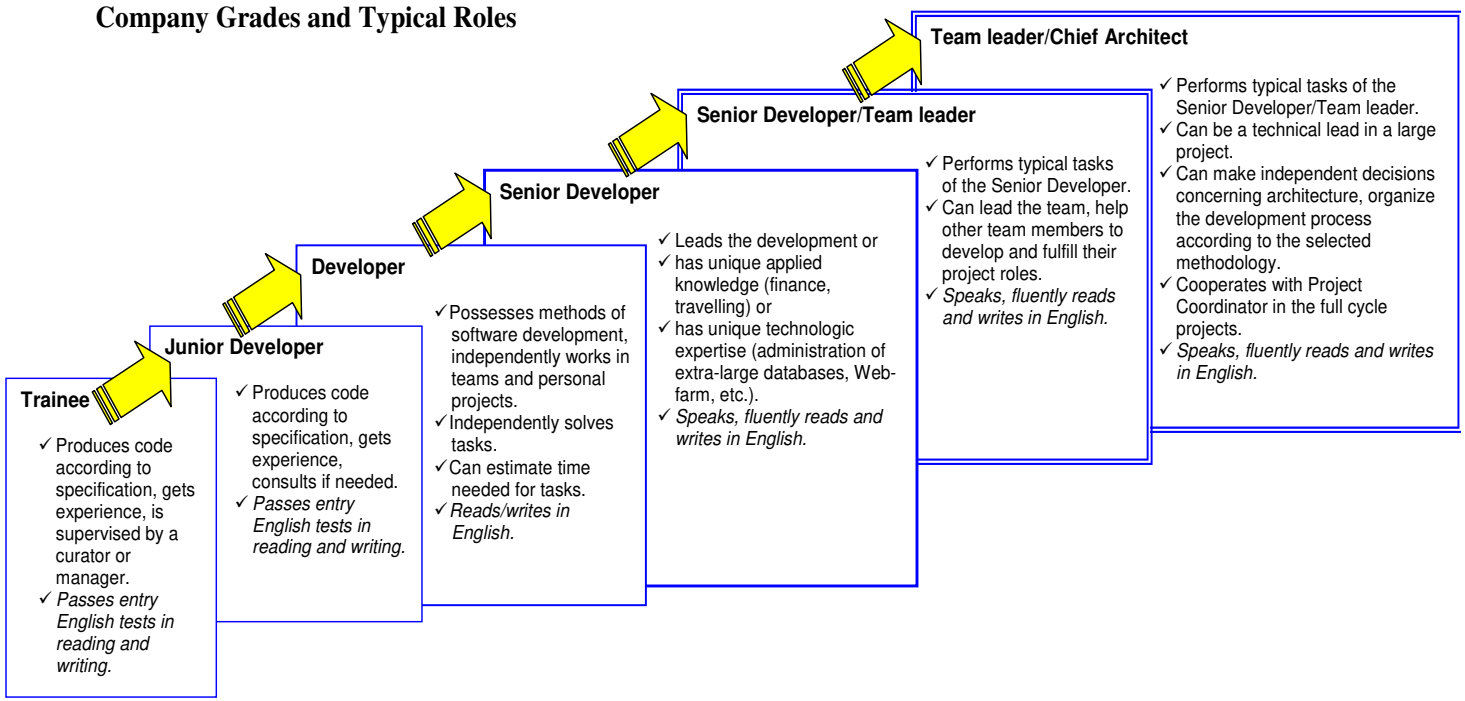


Fig. 6. The steps of career growth of a software engineer in a company.

This scheme does not cover the whole range of specialties in IT industry. Among the most important specialties are: system analysts, requirements engineers, software testers, quality assurance engineers, and some others. The purpose of the scheme is to demonstrate the peculiarities of requirements for various career stages of developer.

As we can see from the figure, the employer at each step separately writes the requirements for English proficiency, personal qualities, and only a few grades deal with certain unique knowledge, for example, applied areas of technological expertise.

Accordingly, under the competence approach, the emphasis should be shifted from learning according to certain state standards of professional knowledge and skills in programming ability to forming an ability to act, make decisions, be energetic in all spheres of public life, educate such skills as teamwork, leadership qualities, responsibility, ability to reflect, capacity for independent learning and mastering new technologies in life, self-education skills, planning, logical and algorithmic thinking, dedication, perseverance, ability to independently make decisions, quickly adapt to new challenges, and a broad outlook on the subject region. In addition, there is a demand for specific knowledge of psychology and management, including project management. An effective mechanism for ensuring high quality learning process is feedback.

Our analysis allowed to consider the components of professional competence in the structure of programming engineer's personality that are required at different steps of professional growth.

Level of graduates must at least meet a minimum level of trainee, novice. Requirements to knowledge that disclose social, personal, scientific, instrumental, general prerequisites for development and professional competence and skills system that reflects their graduate details are spelled out in the industry standard for the specialty "Informatics" [4, 5].

Obviously, the formation of professional competence of programmers should take place throughout the whole period of study at the university while learning the disciplines of each specialty curriculum. However, analysis of the current situation shows that training in computer science is technologically oriented. The professional technical issues should be central in the knowledge base of the programmer, but all aspects of human-computer interaction and the application of programming results are almost completely ignored. Often first-class programmers disregard knowledge in other areas or believe that the level of competence in their field can replace the knowledge of other subjects [1].

A specialist should has a more specific professional thinking, the main characteristics of which are critical attitude to progress, the ability to offer new and a skill of taking into account the impact of all significant internal and external factors that ensure reliable operation of the proposed material.

For the formation of appropriate competencies it is necessary to adjust the content-part curriculum areas in various disciplines related to the preparation of IT professionals and create conditions so that instead of getting a ready knowledge students could gradually develop special skills to self-disclose university disciplines, synthesize information flows necessary for the formulation of knowledge with accordance to the given program.

For example, according to the specialty curriculum of the Informatics specialty, a course in Psychology is studied at the third year in the fifth semester in the volume of

54 hours per semester plus end credits. In our opinion, for the formation of competencies necessary for a successful career the attention of the engineer-programmer should be focused on the following two areas: psychology and management. In the "Recommendations on Teaching Computer Science at the Universities (Computing Curricula 2001: Computer Science)" there is the theme of "Social and professional issues".

Future leaders and managers of teams for the sake of organizational efficiency do not simply have to get knowledge on the course of psychology, but know, for example, that any person belongs to one of three types [11]:

- leader - a man who seeks to manage other people and projects for whom personal success comes first. In the team consisting of leaders only, there will be a constant struggle for power, even the best ideas will not be brought to realization, and exchange of ideas will probably be missing.
- techie - a person who derives pleasure from the process of finding a solution, such as programming, to whom it is not important that there are leaders and subordinates, etc. The main thing is to solve problems. The team consisting only of the specialists of this type will not keep the budget and time, everyone will sit in his corner and solve the task that he likes most and not the one which is necessary at the point.
- friendly - these people pass information and promote the results of other people. According to psychologists, 60% of women belong to this category. The team consisting mainly of friendly people will have fun at holidays, the most pleasant atmosphere, but productivity will be low.

Another feature of the programmer is moving from one project to another. It requires the ability to switch attention. Important modern features include copyright compliance, working with legal software, and quick mastery of a particular subject area. An ability to independently make a decision, quickly adapt to new challenges, a broad outlook in the subject area are becoming the main qualities of a professional engineer and programmer.

Conclusions

Experts in the field of IT are very popular in society, but active development and broad application of technology, a variety of technologies and improvement of software process lead to a very wide range of areas, which in turn demands experts with very different professional knowledge and competence. Therefore, the applied aspects of training, even within one profession, are a very complicated matter. The relevant matter is the mutual enrichment of basic and applied parts of training IT professionals through the active use of ICT in the learning process, distance education and modern techniques that help develop a wide range of competencies.

The feature of programming is the need to solve different types of tasks in a certain subject area and build mathematical models. There are still unsolved problems of harmonious combination of fundamental and applied aspects in the construction of methodical training of future software engineers.

Since it is important for universities to form a competitive graduate, IT specialist, they must pay attention not only to the formation of knowledge of certain fundamental and professional disciplines, but also organize the training process so as to best promote the development of certain personal qualities in students.

Especially important is recognition of the fact that to be a specialist is a process, not a phenomenon, and the aim is not to become at some point a highly skilled programmer in order to learn nothing new anymore.

References

1. Белая О.А. Психология программирования: человеко-машинный аспект информационных технологий /Белая О.А., Новиков Б.А., Одинцов И.О. – Материалы второй открытой всероссийской конференции «Преподавание информационных технологий в России». - Государственный научно-исследовательский институт информационных технологий и телекоммуникаций http://www.ict.edu.ru/vconf/index.php?a=vconf&c=getForm&d=light&id_sec=168&id_thesis=6748&r=thesisDesc
2. Гришко Л.В. Вимоги до професійних якостей програміста. Вісник Черкаського університету, Випуск 173. Серія Прикладна математика. Інформатика. - Черкаси, 2009. - С. 116-120.
3. Дейкстра Э. Дисциплина программирования: Пер. с англ. – М.: Издательство “Мир”, 1978. – 274 с.
4. Галузевий стандарт вищої освіти України. Освітньо-кваліфікаційна характеристика бакалавр. Галузь знань 0403 Системні науки та кібернетика. Напрям підготовки 040302 Інформатика. Міністерство освіти і науки України. Київ, 2010. – 32 с.
5. Галузевий стандарт вищої освіти України. Освітньо-професійна програма підготовки бакалавр. Галузь знань 0403 Системні науки та кібернетика. Напрям підготовки 040302 Інформатика. Міністерство освіти і науки України. Київ, 2010. – 94 с.
6. Петухова Л.Є. Теоретико-методичні засади формування інформатичних компетентностей майбутніх учителів початкових класів. Дисертація на здобуття наукового ступеня доктора педагогічних наук зі спеціальності 13.00.04 – Теорія та методика професійної освіти. - Південноукраїнський державний педагогічний університет ім. К.Д. Ушинського, Одеса — 2009, - 552 с.
9. Сейдаметова З. С. Навчальна дисципліна «Введення в спеціальність» і адаптація студентів першого курсу комп’ютерних спеціальностей. Проблеми освіти: Наук.-метод. Зб.. Кол. Авт. – К.: Інститут інноваційних технологій і змісту освіти МОН України, 2007. Вип. 50. – С. 66 – 70.
10. Семеріков С.О., Теплицький І.О. Фундаменталізація як основа розвитку інноваційної вищої освіти./ Семеріков С.О., Теплицький І.О. // Збірник наукових праць Кам’янець-Подільського Національного університету імені Івана Огієнка. Випуск 15. Серія: Педагогічна. Частина IV. Лісабонська стратегія європейської інтеграції в галузі освіти як визначальний чинник інновацій в підготовці фахівця. 2009. - С. 249-251.
11. Смульсон М.Л. Психологія розвитку інтелекту: Монографія. – К., 2001. – 276 с.
12. Співаковський О.В. Теорія і практика використання інформаційних технологій у процесі підготовки студентів математичних спеціальностей /Співаковський О.В. – Херсон: Айлант, 2003. – 229 с.

13. Терехов А.Н. Психология программирования. Групповая разработка и организация коллектива .Лекция из курса «Введение в технологию программирования». http://citforum.univ.kiev.ua/SE/project/terehov/2.shtml
14. Толковый словарь русского языка: В 4 т. /Под ред. Д.И. Ушакова. – М., 1935. – Т.1.
15. Шнейдерман Б. Психология программирования: Человеческие факторы в вычислительных и информационных системах. Пер. с англ.– М.: Радио и связь, 1984. – 304 с.

Author Index