Oleksii E. Kovalenko[1]

MODELS AND MEANS FOR SERVICE AGENTS ORCHESTRATION IN SITUATION MANAGEMENT SYSTEMS

The issues of interaction in agent-oriented systems are discussed in the article. The review of theoretical and technological tools to support the coordination of agents is performed. Grouping of agents sets into families according to the stages of situational management tasks is proposed. The model of integrating behavioral and coordination aspects of agents is developed.

Keywords: *situation centre; situation calculus; agent-oriented systems; agents coordination; action theory.*

Олексій Є. Коваленко

МОДЕЛІ ТА ЗАСОБИ ОРКЕСТРУВАННЯ СЕРВІСНИХ АГЕНТІВ У СИСТЕМАХ СИТУАЦІЙНОГО УПРАВЛІННЯ*

У статті розглянуто питання організації взаємодії в агентно-орієнтованих системах. Проведено огляд теоретичних і технологічних засобів підтримки координації агентів. Запропоновано принципи розподілу множин агентів на сімейства у відповідності до задач етапів ситуаційного управління. Розроблено модель інтеграції поведінкових і координаційних аспектів у агентів.

Ключові слова: *ситуаційні центри; ситуаційне числення; агентно-орієнтовані системи; координація агентів; теорія дії.*

Форм. 2. Літ. 33.

Алексей Е. Коваленко

МОДЕЛИ И СРЕДСТВА ОРКЕСТРОВКИ СЕРВИСНЫХ АГЕНТОВ В СИСТЕМАХ СИТУАЦИОННОГО УПРАВЛЕНИЯ

В статье рассмотрены вопросы организации взаимодействия в агентно-ориентированных системах. Проведен обзор теоретических и технологических средств поддержки координации агентов. Предложены принципы разделения множеств агентов на семейства в соответствии с задачами этапов ситуационного управления. Разработана модель интеграции поведенческих и координационных аспектов в агентах.

Ключевые слова: *ситуационные центры; ситуационное исчисление; агентно-ориентированные системы; координация агентов; теория действия.*

Problem statement. Situation management systems based on information technologies are used in various spheres of human activity (Jakobson et al., 2005; Jakobson et al., 2007). Situation management (SM) is considered "as a framework of concepts, models and enabling technologies for recognizing, reasoning about, affecting on, and predicting situations that are happening or might happen in dynamic systems during pre-defined operational time" (Jakobson et al., 2007). The multi-agent system (MAS) (Wooldridge, 2002) due to its characteristics directly fit for solving SM problems (Buford et al., 2006). MAS can be used as supporting tools for organizing and serving integrated environments of SM. Such type of SM environment we call as agent-based environments of situation management (ABESM).

[1] Institute of Mathematical Machines and Systems Problems of the National Academy of Sciences of Ukraine, Kyiv, Ukraine.

* статтю підготовлено на основі доповіді на XII-му міжнародному науковому семінарі «Сучасні проблеми інформатики в управлінні, економіці, освіті та екології» (1–5 липня 2013 р., оз. Світязь – Київ).

SM is a complex multilayered technology with variety of interrelated tasks on each layer. Appropriate agents on various layers perform the functions of dispatching, coordination, services seeking and providing security etc. Agents during functioning must use the appropriate knowledge. Knowledge of an agent is a fragment of knowledge domain of a SM problem. Knowledge domain of a SM problem is an element of a SM model.

Latest research and publications analysis. One of the main features of the ABESM functioning is the collective work of a group of experts, which includes a person who makes decisions, a group of analysts and ABESM. Methodological principles of multiprofile situation management systems (SMS) activity as a purposeful organizational structure involves identifying such aspects of activity as (Hills and Jones, 2012):

1) mission statement – generalized goal, formulated in the context of SMS strategy;

2) vision statement – environmental assessment and a clear definition of perspectives and performance criteria;

3) strategic management – continuous process associated with the formulation of organizational solutions for efficient and productive activity in the context of mission, vision and goals of global activity objectives;

4) goals, objectives, targets tree – hierarchical set of final states or outcomes to achieve the aims of the activity of SMS;

5) policies – organizational mechanisms to ensure the achievement of defined objectives;

6) activity models – set of methods, tools and algorithms that define and describe the operational procedures at all levels of an organizational structure.

One of the conditions for the successful activity of any organizational structure, SMS including, is its effective management that takes into account the specifics of its operation. Significant results in this direction were obtained at the end of the last century in the work of Henry Mintzberg (Mintzberg, 1992) and Alvin Toffler (Toffler, 1970). Successful computerization of process management requires appropriate models for collective activity in information systems.

The following technological stages can be identified in the operation of SMS (Morozov et al., 2009):

1) monitoring of the control object;

2) determination of participants (analysts, experts, making decisions persons) for situational management processes;

3) systemic analysis of the situation (analysis of current and forecasting of anticipated states of management object, simulation of events evolution in management object etc.);

4) decision-making through meetings and discussions;

5) decisions working off (bringing to performers and implementation monitoring);

6) retrospective analysis (assessment) of decisions' quality and their implementation performance.

The agent-based approach for situation management supporting in large-scale disaster relief operations was proposed by J. Buford et al. (2006).

Coordination of agents in ABESM realized through orchestration and choreography was studied in (Blake, 2003; Busi et al., 2005; Misra, 2005; Havey, 2008). Agents activity can be maintained by linguistic means, such as situation calculus (Levesque et al., 1998; Giunchiglia and Lifschitz, 1999), pi-calculus (Milner, 1993; Milner, 1999; Ryan and Smyth, 2011), orchestration languages (Andrews et al., 2003; Viroli, 2004; Lapadula et al., 2007; Montesi et al., 2007; Kitchin et al., 2009; Montesi, 2013;), agent communication language (Labrou and Finin, 1998; FIPA-ACL, 2002; Luck et al., 2004), action description languages (Gelfond and Lifschitz, 1998; Letichevsky and Gilbert, 1998; Giunchiglia and Lifschitz, 1999; Reiter, 2001; Lee et al., 2013) etc.

Unresolved issues. The main feature of SMS functioning is the need to address specific custom tasks for the purposes of situational management using the typical SMS infrastructure. Decision management of each task is performed by an authorized person – a person who makes decisions (PMD) – with the involvement of specialists from different organizations to perform the roles of experts, analysts, consultants, using tools and techniques of modern information technologies. Thus, SMS establishes an organizational structure with permanent staff to ensure the functioning of SMS and temporary groups of users who use the services of SC to solve their problems of situational management. These organizational structures require the creation of adequate software tools for providing information services to participants of the SM process and adaptation of these services to the conditions of solving each problem. A variety of information services provided by the SMS on the one hand, and the short list of information services to solve specific problems, on the other one, cause the need for coordination of SMS staff and expert groups with the PMD at the stages of preparation, implementation, monitoring and implementation analysis of decisions during SM. Such coordination should be supported by appropriate information technologies and software tools.

Different stages of SM are performed by different agents groups in ABESM. The peculiarities of each stage must be taken into account. Therefore, various types of agents are used during the implementation of each stage situation management process at ABESM. Software agents act as active elements that ensure the integration and adaptation of services to address specific problems of situational management. Interaction between agents is determined by management model adopted in SMS. The agent models for each stage of SM must take into account communication and coordination aspects – situation formalization, concurrent processes control, orchestration, messaging.

The objective. Situation formalization, concurrent processes control, orchestration and messaging mechanisms are determined by the peculiarities of ABESM organizational structure. Integration and harmonization aspects of orchestration based on the coordination of ABESM agents are the objectives of this article.

Research findings. The most used model of agent in MAS is BDI (Beliefs-Desires-Intentions) model. This model is supported by FIPA. Agents behavior in MAS described by Basic Action Theory (Reiter, 2001) that formalizes the action theory. Social aspects of agents behavior may be described by the theory of structural functionalism (Parsons, 1975). Structural functionalism, or simply functionalism, is the framework for building a theory that views society as a complex system, the parts of which work together to promote solidarity and stability (Macionis, 2010).

The basic action theory typically describes action as behaviour caused by an agent in a particular situation. Agent's desires and beliefs determine its intentions and behaviour. The basic concepts in situation calculus are situations, actions and fluents. Briefly, actions are what make the dynamic world change from one situation to another when performed by agents. Fluents are situation-dependent functions used to describe the effects of actions. Fluent domain of definition is the set of all possible situations. There are two kinds of them, relational fluents and functional fluents.

To describe a dynamic domain in situation calculus, one has to decide on the set of actions available for agents to perform, and the set of fluents needed to describe the changes these actions will have on the world. So, basic action theory describes situational aspect of agent's activity.

The a basic action theory (BAT) T in its general form consists of 5 components (Reiter, 2001: 58):

$$T = \Sigma \cup T_{SSA} \cup T_{PA} \cup T_{UNA} \cup T_{s_0}, \quad (1)$$

where Σ is the set of foundational axioms for situations; T_{SSA} is the set of successor state axioms (SSA) for functional and relational fluent; T_{PA} is the set of precondition axioms (PA) for actions; TUNA is the set of unique names axioms (UNA) for actions; T_{s0} is the set of first-order sentences that are uniform in s_0, so that s_0 is the only term of the sort situation mentioned by the sentences of T_{s0}. There are two binary predicates: P_{oss} and $\subset$. $P_{oss}(a,s)$ states under which conditions action a is applicable in situation s; $s \subset s`$ defines an ordering relation on situations (which are interpreted as sequences of actions) and states that s is a proper subsequence of $s`$. Thus, no sentence of T_{s0} quantifies over situations, or mentions P_{oss}, $\subset$ or the function symbol *do*. Finally, T_{s0} describes the initial situation.

According to T. Parsons (1968) social behavior is associated with normative, voluntary, semiotic (symbolic) aspects. Normative aspect depends on commonly accepted values and norms. Voluntary aspect is associated with the will of a subject, providing some independence from the environment. And semiotic (symbolic) aspect defines the symbolic mechanisms of social behavior regulation.

The pi-calculus (π-calculus), continuation of calculus of communicating systems (CCS), allows participants of communication process (for example, agents) to be communicated along channels themselves, and in this way it is able to describe concurrent activity (computations) whose connections configuration may change during the activity (computation). So, pi-calculus describes the communicative aspect of agent's activity.

General coordination mechanism in hierarchical agent-based environments is orchestration. Orchestration describes the automated arrangement, coordination, and management of complex computer systems, middleware, and services. In the context of agent's paradigm orchestration defines the policies and agent's hierarchy through coordinated activity, provisioning, and change management. This creates an agent-aligned infrastructure that can be scaled up or down based on the needs of each application. Orchestration also provides centralized management of the resource pool, including billing and metering. Orchestration languages describe controlled coordination aspect of agent's activity.

Agent communication languages (including FIPA-ACL) define the structure and the semantics of agent's communicative acts which provides the general mechanism of messaging between agents. Agent communication languages describe messaging aspects of agents' interactions.

Action calculus and action description languages are the means of events and context driven behavior of agents. Action description languages are formal languages for describing the effects and executability of actions. The best known action language is PDDL (Planning Domain Definition Language) and its agent-oriented version MA-PDDL. PDDL based on SRIPS (Stanford Research Institute Problem Solver) ideas and mathematically presets as quadruple $\langle P,O,I,G\rangle$, were P is the set of conditions, O is the set of operators (i.e., actions) (each operator is itself a quadruple $\langle \alpha,\beta,\gamma,\delta\rangle$, each element being a set of conditions), I is the initial state, given as the set of conditions that are initially true, G is the specification of the goal state (this is given as a pair $\langle N,M\rangle$, which specify which conditions are true and false, respectively, in order for a state to be considered a goal state). Action languages describe event-dependent aspect of agents' behavior.

In result the integrated behavioral model of service agent is presented as tuple:

$$A_b = \langle T,P,C,M,D\rangle, \qquad (2)$$

where T is a means of situation description; P is a means of communicative control in changeable communication environments; C is a means of coordination mechanism; M is a means of messaging between agents; D is a means of action description.

Conclusions and further studies perspective. Functioning of the situation management system is based on the coordination of its components. The model of coordinated activities is created to ensure proper functioning of a situational management system according to its organizational structure, that is supported by relevant software agents. Software agents are organized in a hierarchy according to the levels of abstraction and specialization of their functions, that ensure access to services during the process of situational management. Coordination of functioning is carried out by software agents on the basis of orchestration of lower-levels services by the agents of higher-level and choreography of services at the same level of situational management process during the specific problems solving. Models of integration of behavioral and coordination aspects for agents were developed during the study. These models can be adapted for particular information platforms and technologies. So, further studies may be dedicated to analyzing peculiar properties of various information platforms and technologies and developing working agents-based environments for situation management.

References:

Ситуаційні центри. Теорія і практика / За ред. А.О. Морозова, Г.Є. Кузьменко, В.А. Литвинова. – К.: Інтертехнодрук, 2009. – 348 с.

Andrews, T., Curbera, F., Dholakia, H. et al. (2003). Business Process Execution Language for Web Services, version 1.1, 2003.5 // ifr.sap.com.

Blake, M.B. (2003). Coordinating multiple agents for workflow-oriented process orchestration. Information Systems and E-Business Management, 1(4): 387–404.

Buford, J.F., Jakobson, G., Lewis, L. (2006) Multi-Agent Situation Management for Supporting Large-Scale Disaster Relief Operations. International Journal of Intelligent Control and Systems, 11(4): 284–295.

Busi, N., Gorrieri, R., Guidi, C., Lucchi, R., Zavattaro, G. (2005). Choreography and orchestration: A synergic approach for system design. In: Service-Oriented Computing-ICSOC 2005 (pp. 228–240). Springer Berlin Heidelberg.

Foundation for Intelligent Physical Agents (2002). FIPA Communicative Act Library Specification // www.fipa.org.

Gelfond, M., Lifschitz, V. (1998). Action languages. Linkoping Electronic Articles in Computer and Information Science, 3(16): 1–19.

Giunchiglia, E., Lifschitz, V. (1999). Action languages, temporal action logics and the situation calculus. Linkoping Electronic Articles in Computer and Information Science, 4(40): 1–19.

Havey, M. (2008). Modeling Orchestration and Choreography in Service Oriented Architecture // www.packtpub.com.

Hill, C.W.L., Jones, G.R. (2012). Strategic Management: An Integrated Approach. 10th ed. Mason: South-Western Cengage Learning. 960 p.

Jakobson, G., Buford, J., Lewis, L. (2007). Situation Management: Basic Concepts and Approaches. In: Popovich, V.V., Schrenk, M., Korolenko, K.V. (eds.) Information Fusion and Geographic Information Systems 2007. LNG&C, vol. XIV, pp. 18–33. Springer, Heidelberg.

Jakobson, G., Lewis, L., Matheus, C.J., Kokar, M.M., Buford, J. (2005). Overview of situation management at SIMA 2005. In: Military Communications Conference. MILCOM 2005. IEEE, pp. 1630–1636. IEEE Press.

Kitchin, D., Quark, A., Cook, W., Misra, J. (2009). The Orc programming language. In: Formal Techniques for Distributed Systems (pp. 1–25). Springer Berlin Heidelberg // orc.csres.utexas.edu.

Labrou, Y., Finin, T. (1998). Semantics for an agent communication language (pp. 209–214). Springer Berlin Heidelberg.

Lapadula, A., Pugliese, R., Tiezzi, F. (2007). A calculus for orchestration of web services. In: Programming Languages and Systems (pp. 33–47). Springer Berlin Heidelberg.

Lee, J., Lifschitz, V., Yang, F. (2013). Action Language BC: Preliminary Report. In: Proceedings of International Joint Conference on Artificial Intelligence (IJCAI).

Letichevsky, A., Gilbert, D. (1998). A general theory of action languages. Cybernetics and Systems Analysis, 34(1): 12–30.

Levesque, H., Pirri, F., Reiter, R. (1998). Foundations for the situation calculus. Linkoping Electronic Articles in Computer and Information Science, 3(18).

Luck, M., McBurney, P., Preist, C. (2004). A manifesto for agent technology: Towards next generation computing. Autonomous Agents and Multi-Agent Systems, 9(3): 203–252.

Macionis, G. (2010). Sociology 7th Canadian Ed. Pearson Canada Inc.

Milner, R. (1993). The polyadic π-calculus: a tutorial (pp. 203–246). Springer Berlin Heidelberg.

Milner, R. (1999). Communicating and mobile systems: the pi calculus. Cambridge University Press.

Mintzberg, H. (1992). Structure in Fives: Designing Effective Organizations. Englewood Cliffs, N.J.: Prentice-Hall. 312 p.

Misra, J. (2005). Computation Orchestration. In: Engineering Theories of Software Intensive Systems (pp. 285–330). Springer Netherlands.

Montesi, F. (2013). Process-aware web programming with Jolie. In: Proceedings of the 28th Annual ACM Symposium on Applied Computing (pp. 761–763). ACM.

Montesi, F., Guidi, C., Lucchi, R., Zavattaro, G. (2007). JOLIE: a Java Orchestration Language Interpreter Engine. Electronic Notes in Theoretical Computer Science, 181: 19–33.

Parsons, T. (1968). The structure of social action: a study in social theory with special reference to a group of recent European writers. New York: Free Press. 775 p.

Parsons, T. (1975). The Present Status of "Structural-Functional" Theory in Sociology. In: T. Parsons. Social Systems and The Evolution of Action Theory. New York: The Free Press.

Reiter, R. (2001). Knowledge in action: logical foundations for specifying and implementing dynamical systems (Vol. 16). Cambridge: MIT Press.

Ryan, M.D., Smyth, B. (2011). Applied pi calculus // www.bensmyth.com.

Toffler, A. (1970). Future Shock. New York: Random House. 505 p.

Viroli, M. (2004). Towards a formal foundation to orchestration languages. Electronic notes in theoretical computer science, 105: 51–71.

Wooldridge, M. (2002). An Introduction to Multi-Agent Systems. John Wiley and Sons, NY.

Стаття надійшла до редакції 8.08.2013.