Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 103 (2017) 162 – 167

Procedia Computer Science

XIIth International Symposium «Intelligent Systems», INTELS'16, 5-7 October 2016, Moscow, Russia

Conceptual model of problem-oriented heterogeneous distributed computing environment with multi-agent management

I. Bychkov[a], G. Oparin[a], A. Tchernykh[b], A. Feoktistov[a]*, V. Bogdanova[a], S. Gorsky[a]

[a]*Matrosov Institute for System Dynamics and Control Theory, SB RAS, Lermontov str., 134, Irkutsk, Post Box 292, 664033, Russia*
[b]*CICESE Research Center, Computer Science Department, Carretera Ensenada-Tijuana No. 3918, Zona Playitas, Codigo Postal 22860, Apdo. Postal 360, Ensenada, B.C., Mexico*

Abstract

One of the promising approaches to the efficient management in a heterogeneous distributed computing environment is the use of multi-agent systems. When researchers want to solve tasks in such an environment, they need utilities for the description of the processing in the specific domain, and knowledge about a software-hardware infrastructure. This multi-component knowledge is used for effective decision-making. Practice and experience of multi-agent systems show that knowledge representation is non-trivial problems. We proposed an approach to the design of the conceptual model of a heterogeneous distributed computing environment based on multi-agent management. An important component of the model is algorithmic knowledge, which provides the support of planning, resources allocation and dynamic decomposition of a problem to subproblems. This knowledge is represented as a semantic network, which describe semantic relations between concepts. Unlike to Semantic Web Services and Semantic Grids, our approach provides advanced planning abilities to support parallelism and scalability. We develop specialized high-level tools for designing and using the conceptual models.

Peer-review under responsibility of the scientific committee of the XIIth International Symposium "Intelligent Systems"

Keywords: multi-agent systems; knowledge representation; distributed computing environment.

* Corresponding author.
E-mail address: agf65@yandex.ru

Peer-review under responsibility of the scientific committee of the XIIth International Symposium "Intelligent Systems"
doi:10.1016/j.procs.2017.01.043

1. Introduction

High-performance computing systems for solving complex scientific problems in various subject domains are an essential part of the modern research. Heterogeneous environments such as Grid and Cloud-computing systems provide scalable solutions. In large distributed environments, to support scalability, the decentralized management is more preferable in comparison with the centralized management[1]. A well-established approach to the decentralization is the multi-agent technology[2]. It can significantly improve a management quality, parallelism and efficiency[3]. This type of management assumes that users pass a part of the rights and responsibilities to program agents, which united into a multi-agent system (MAS)[4]. Each agent has knowledge about itself, and the environment. Tools for updating knowledge allow agents to adapt to the changing reality.

We consider three groups of users: administrators of the computing environment, application developers, and end users (scientists), who solve their problems. Each user operates with its own set of concepts related to his activities. An effective decision-making solution in the MAS is based on the knowledge exchange between agents within the single system of concepts.

The multi-agent management in problem-oriented environments requires to take into account the subject domain features of solving problems[5]. A large range of tools for MASs exists and successfully applied in practice. However, representation the heterogeneous knowledge of subject domains in a MAS is still a challenging problem.

Our contribution is multifold. We propose an approach to design of conceptual models for problem-oriented heterogeneous distributed computing environments that allows users to aggregate and unify the heterogeneous and uncoordinated knowledge about various aspects of the problem into the single system of concepts.

A conceptual model is a particular case of the semantic network represented by oriented graph, where vertices correspond to problem domain objects (concepts) and edges define binary relations between vertices. Thus, the designed model reflects the semantics of the domain in the form of concepts and their relationships used for the agent knowledge conceptualization.

Our approach is based on models in the applied software packages. The knowledge representation opportunities of such models are extended to support distributed computing using the MAS, which is described in[6,7]. As a result, the MAS works with different problem domains, unlike many MASs used in practice that are developed for a single problem domain.

The conceptual model includes description algorithmic knowledge about various problem domains, software-hardware infrastructure of the environment and administrators' knowledge. Unlike to approaches that based on using the semantic meaning of information and computational resources in distributed computing, our approach provides advanced planning abilities to support parallelism and scalability (see Section 5).

2. Model core

The most basic concept of the model is an object described by name, class, and a set of attributes. An attribute is a number, object link, or uncertainty symbol. Links provide access to an object. There are two types of links: simple that directs to one object, and multiple that points list of objects.

The model supports the following relations between objects: "one-to-one", "many-to-many", "one-to-many", "many-to-one". Multiple object relations are created by lists. Lists are objects of special class. Lists unite the objects of same class and provide access to these objects as to the single object.

There are types and classes that are build-in into the model. New classes can inherit the properties of existed classes.

The model core includes classes of the problem domain objects, object attributes and relations between objects. These components are needed to describe algorithmic knowledge, knowledge about software-hardware infrastructure of the computer environment, and administrator knowledge.

Knowledge of a problem domain is expressed by objects with conditions represents as productions set. Productions are objects of determined class.