Information and Software Technology 55 (2013) 673–689

Contents lists available at SciVerse ScienceDirect

Information and Software Technology

journal homepage: www.elsevier.com/locate/infsof

Suitability assessment framework of agent-based software architectures

Ghassan Beydoun [a,*], Graham Low [b], Paul Bogg [b]

[a] The University of Wollongong, Wollongong, Australia
[b] The University of New South Wales, Sydney, Australia

ARTICLE INFO

Article history:
Received 9 December 2011
Received in revised form 11 October 2012
Accepted 29 October 2012
Available online 17 November 2012

Keywords:
Modelling
Distributed intelligent systems
Agent
Multi Agent Systems
Software development

ABSTRACT

Context: A common distributed intelligent system architecture is Multi Agent Systems (MASs). Creating systems with this architecture has been recently supported by Agent Oriented Software Engineering (AOSE) methodologies. But two questions remain: how do we determine the suitability of a MAS implementation for a particular problem? And can this be determined without AOSE expertise?
Objective: Given the relatively small number of software engineers that are AOSE experts, many problems that could be better solved with a MAS system are solved using more commonly known but not necessarily as suitable development approaches (e.g. object-oriented). The paper aims to empower software engineers, who are not necessarily AOSE experts, in deciding whether or not they should advocate the use of an MAS technology for a given project.
Method: The paper will construct a systematic framework to identify key criteria in a problem requirement definition to assess the suitability of a MAS solution. The criteria are first identified using an iterative process. The features are initially identified from MAS implementations, and then validated against related work. This is followed by a statistical analysis of 25 problems that characterise agent-oriented solutions previously developed to group features into key criteria.
Results: Key criteria were sufficiently prominent using factor analysis to construct a framework which provides a process that identifies within the requirements the criteria discovered. This framework is then evaluated for assessing suitability of a MAS architecture, by non-AOSE experts, on two real world problems: an electricity market simulation and a financial accounting system.
Conclusion: Substituting a software engineer's personal inclination to (or not to) use a MAS, our framework provides an objective mechanism. It can supplant current practices where the decision to use a MAS architecture for a given problem remains an informal process. It was successfully illustrated on two real world problems to assess the suitability of a MAS implementation. This paper will potentially facilitate the take up of MAS technology.

1. Introduction

A distributed intelligent system is a collection of interacting intelligent individual components which cooperate to solve global goals as well as solving their local goals [41,43]. The agency metaphor, as applied to such individual components, has proved fruitful in modelling their behaviour and host systems. Indeed this has led to the acceptance of 'Agents' as highly autonomous, situated and interactive software components. They autonomously sense their environment and respond accordingly. A distributed system formed from coordination and cooperation between agents is known as a Multi Agent System (MAS). The diverse knowledge and capabilities of individual agents within a MAS facilitate the achievement of global goals that cannot be otherwise achieved by a single agent working in isolation [73]. MASs have been shown to be highly appropriate for the engineering of open, distributed or heterogeneous systems [35,38,57].

Distributed MAS have been developed by Distributed Artificial Intelligence researchers since the 1980s. However it is more recent that many Agent Oriented Software Engineering (AOSE) methodologies have been proposed to guide the development of MAS (e.g. MaSE [27], GAIA [74], PROMETHEUS [52], MOBMAS [69] and TROPOS [13]). Such methodologies define various modelling languages, steps, techniques and models to produce MAS [3]. Whilst software architecture researchers aim at formalising the description of the system to facilitate the transition into design [48], software engineering researchers working on MAS architectures (aka AOSE researchers) aim to create requirement analysis concepts and tools to convert the problem description into agent based requirement models (to a varying degree of formalism). They often use different agent based constructs and target different development settings or phases. Gaia [76] for instance supports the development cycle

* Corresponding author.
E-mail addresses: beydoun@uow.edu.au (G. Beydoun), G.Low@unsw.edu.au (G. Low), paulb@unsw.edu.au (P. Bogg).

0950-5849/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.infsof.2012.10.007

of MAS from analysis to low level design. Prometheus [52,66] defines an agent-based development process of three phases – system specification, architectural design and detailed design – to develop MAS based on a specific agent architecture (BDI architecture (Belief – the agent's knowledge of the world, Desire – the agent's goals, Intentions – the goals that the agent is committed to achieve at certain moment)). Adelfe [7,8] is oriented to the development of adaptive MAS, i.e. systems that can adapt themselves to unpredictable, evolutionary and open environments. PASSI [21] and its evolution ASPECS [22] focus on agent societies to describe a complete development process from requirements specification to implementation. TROPOS [2,15] covers the analysis and design phases of MAS development and is based on the i^* requirements elicitation approach [75].

The notion of architecture in software engineering aims at reducing the cost of development [48]. Whilst MAS architectures can also contribute to cost management of a project [6], they are often pursued as a complexity management/problem solving tool which may in some cases allow tackling new problems [41,43]. The decision to apply a MAS architecture and possibly an AOSE methodology to a given problem remains an informal ad hoc process, based on the software engineer's inclination to use such architectures and/or past experience of using such architectures applied to similar problems. Given the small number of software engineers that are familiar with MAS and AOSE, many problems that could benefit from a MAS approach are solved with other approaches which may not be the best approach for a particular problem. This can indeed be in some cases an overlooked opportunity for a very productive and cost effective approach. Research has recently shown that, when suitable, MAS architectures can lead to large increases in the productivity of developers and programmers [6].

The lack of familiarity with MAS and AOSE has no doubt contributed to the delay in the much anticipated adoption of AOSE in many medium to large-scale projects. As pointed out in [1], as a particular technology matures, its accessibility to non-experts increases. We believe that AOSE has sufficiently developed and it is timely to facilitate access to this technology for non agent-experts. In this spirit and to overcome the above barrier to the successful adoption of AOSE, this paper provides a framework to guide a software engineer, who may not be an AOSE expert, to decide whether a MAS architecture may be an appropriate solution for a particular problem. The selected development approach (e.g. *object-oriented*, *service-oriented*, *agent-oriented*) would depends on both the suitability of the resultant implementation as well as factors such as project cost and availability of experienced staff. We assume that the requirement gathering has been accomplished before the framework is used. In other words, we assume that any doubts about the requirements and the domain have been resolved. As such, to apply the framework, it is not necessary to have deep expertise about the application domain.

The rest of this paper is organised as follows. Section 2 describes related work and introduces previous attempts to establish types of problems suited for MASs. Section 3 proposes features of problems to which MASs are suited, based on an analysis of problems for which MASs have been developed. Further analysis on these features establishes relationships that indicate different problems have different degrees of suitability for MASs. Section 4 uses this analysis to formulate a framework for determining the suitability of a MAS to a problem. This framework is placed within the context of a software development lifecycle. Section 5 describes an application of the framework to describe the suitability of a MAS to the problem of modelling an electricity market place for assessment of an electricity market simulation and a financial accounting system. Section 6 concludes the paper.

2. Related work

An increasing acceptance of AOSE as an alternative approach to software development has led a number of researchers to ask how and when AOSE would be preferred over other approaches [51,72,78]. This question has appeared prominently amongst researchers in agent-oriented development and is yet to be answered [78]. Towards finding an answer, the primary focus in this paper is in defining a measure of *suitability* defined as the extent to which a software solution adequately addresses the features of a particular problem domain. In reality, the answer to the question *"Could one suitably use a MAS approach to solve this particular problem?"* could be based on an acceptable degree of suitability combined with an acceptable level of project and personnel-based estimates such as cost, time, and expertise.

Apart from a few application-based research works (e.g. cases presented in Table 5 later in this paper), none have demonstrated an analysis of suitability at the same level of depth of [37,54,55]. One of the earliest reviews of MAS suitability and general benefits can be found in [37]. In particular, this work analyzed the suitability of MAS for various telecommunication applications. The main conclusion was that MASs were suitable for telecommunication applications when key system requirements involve *distribution, robustness, responsiveness* and *flexibility*. This conclusion was later confirmed for other domains (such as logistics and space exploration) [54]. Similar conclusions were found in [55] for manufacturing and defense applications, but with additional system requirements of *openness, complexity* and *cooperative problem solving*. Whilst the analysis was convincingly thorough in those problem domains considered by [37,54,55], no domain-independent validations or generalisations were made. We believe that it has been briefly attempted only in AOSE in the context of creating new methodologies.

AOSE researchers directly involved with methodologies have attempted to formulate the suitability of MASs in trying to scope their methodologies. Notably [51] attempted this from the perspective of *"management, usage, and technical"* requirements. In their approach, they identified a small set of requirements, performed a survey-based validation, and compared two methodologies. In relation to MAS suitability, the technical requirements identified were: *legacy, distribution, environment, dynamic structure, interaction, scalability* and *agility*. This work is perhaps closest to the work presented here in that they identified suitability criteria and attempted a validation. However, their validation was limited to expert reviews and they did not perform a comprehensive domain-independent analysis. Another notable related effort that appeared at the same time as [51], is the EURESCOM project [29]. This resulted in the following domain-independent characterisation of when MAS is suitable:

- Complex/diverse types of communication are required.
- The system must perform well in situations where it is not practical/ possible to specify its behaviour on a case-by-case basis.
- Negotiation, cooperation, and competition between entities is involved.
- The system must act autonomously; and/or
- High modularity is required (for when the system is expected to change).

Similar suitability criteria to [27] were suggested by [25] with one addition:

- (when) There is decentralised or distributed information availability (e.g. in competitive situations, or communication failure somewhere).