Turkish Journal of Computer and Mathematics Education Vol.12 No.13 (2021), 5385-5390

Research Article

Proposing an SLR exploring the assumptions and requirements of the negotiation protocol in MAS

Sumeet Mathur

Email ID: sumeet.mathur@hotmail.com

Article History: Received: 11 January 2021; Accepted: 27 March 2021; Published online: 4 Jun 2021

Abstract:- The multi-agent system (MAS) community has put noteworthy endeavors in recording design patterns. In spite of the substantial body of work, design patterns for MAS have gotten the consideration they merit, neither in the agent-situated programming community, nor among programming expert. Albeit a large portion of these approaches have just begun investigating the point, there is a looming requirement for building up an exploration guide, just as recognizing logical and technological challenges in this degree. The examination demonstrates that there is an absence of a standard format for reporting design patterns for MAS, which hampers the utilization of patterns by practitioners,) relationship between patterns are ineffectively portrayed, which results in an absence of diagram of the pattern space, patterns for MAS have been utilized for an assortment of use areas, which supports their high potential for practitioners, and arrangements of design patterns for MAS are limited to explicit pattern lists, a progressively all encompassing perspective on the pattern space is absent. From our examination, we plot various guidelines that are vital for future work on design patterns for MAS and their appropriation by and by.

Keywords: Multi- Agent system, design, pattern, technology, software, etc.

I. INTRODUCTION

Capturing design information as patterns is a typical practice in standard software engineering. Design patterns permit reuse of best practices and maintaining a strategic distance from most exceedingly terrible. The value of patterns has been demonstrated observationally. Design patterns improve software's quality properties, similar to viability and re-convenience, and accelerate the advancement time. These components are vital practically speaking, particularly for undertaking chiefs, since improving them diminishes costs. Amid the most recent decade, the multi-agent system (MAS) community has put huge endeavors in recording design patterns. Regardless of the substantial body of work, design patterns for MAS have not gotten the consideration they merit, neither in the agent-arranged software community, nor among software practitioners, the creators express that "One of the principle reasons why standard software engineers don't profit by MAS patterns is that they just don't have any acquaintance with them".

Agent properties According to Odell (2007) among key agent characteristics are:

- autonomous - equipped for to autonomous activities without outer mediation so as to accomplish objectives,
- interactive - speaking with different agents and additionally condition,
- Adaptive - changing, improving conduct in light of changes. Different attributes:
- intelligent - fit for utilizing autonomous and flexible activities for accomplishing objectives and its state is directed by learning,
- flexible - combaining reactive, proactive and social attributes,
- reactive - checking the environment and responding, preplanning and reorganizing so as to adjust to environment changes,
- proactive - setting an arrangement to achieve an objective and moving in the direction of the arrangement,
- social - creating social relations,
- mobile - equipped for transporting itself,
- trustworthy - honest and don't purposefully act in surprising ways

As MAS have highlights that are generally considered as key to engineering complex appropriated applications, it is critical to give a reasonable outline of existing patterns to make this information available to practitioners. Keeping that in mind, we played out a systematic literature review covering the primary distribution settings of the field since 1998. From the 500 studies considered, 40 were incorporated into the investigation, bringing about 100 patterns. In this paper we report the aftereffects of this literature review, and from our discoveries, we plot various guidelines that, as we would like to think, are critical for future work on design patterns for MAS and their selection by and by.

Design patterns have been embraced in numerous controls, for instance brain research and social sciences. With regards to software engineering, patterns bolster better design decisions, improve communication among stakeholders, and spare time by reusing demonstrated solutions. The idea of design pattern was presented in building architecture by Christopher Alexander during the 1970s. As per Alexander, building structures and town arranging ought to be upheld by design patterns. These patterns comprise of three layers as appeared in Figure 1.

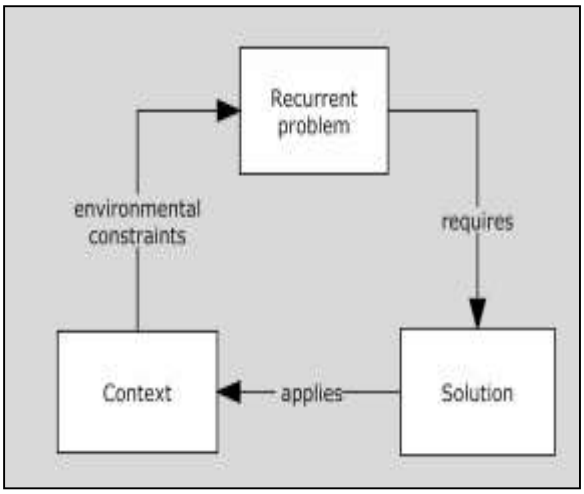

FIGURE 1 PATTERN ANATOMY

The first layer embodies a repetitive issue. An issue emerges in a circumstance known as a unique situation - i.e., the second layer. The third layer is the arrangement, that is, an outstanding and demonstrated answer for an issue in a unique situation. Design patterns are intended to be conventional, so their application and execution may change. Moreover, design patterns are down to business, yet tried arrangements, as they are gotten as a matter of fact with building genuine, solid systems. Design patterns are frequently arranged and gathered in a type of an inventory. A list fills in as a library of skill of fruitful arrangements; henceforth it is a successful instrument for learning and educating. The Gang of Four (GoF) proposed the first arrangement of software design patterns. This pattern inventory utilizes a two dimensional grouping dependent on degree and reason. The inventory contains 23 design patterns that were beforehand undocumented. Buschman's order is another well-known association of software design patterns. This index advances usefulness and basic standards, and uses likewise a grouping along two measurements: granularity and reason. Other prevalent inventories are Fowler's pattern index, J2EE outlines for vast scale endeavor applications, and Schmidt's inventory that reports design patterns for simultaneousness.

Wooldridge (2009) states that multi-agent systems emerged as a new field in computer science from following trends:

- ubiquity: spreading of computers preparing power all over the place,
- interconnection: communication between procedures, building circulated systems,
- intelligence: building devices equipped for human-like intelligence,
- delegation: giving over the control, mechanizing and designating human errands to computers and robots,
- human-orientation: the manner in which we collaborate with individuals turns into the manner in which we associate with computers

A) Principles of MAS

An agent can be rationalized as an autonomous element, with expendable information, driven without anyone else's input created or prompted goals. Also, agents can watch the encompassing condition through a recognition layer, and perhaps interact with it, as well as with different agents. MAS are generally made out of approximately coupled agents interconnected and organized in a system. The degrees of cooperation among agents, the sort of application, and agent interaction show generate a broad range of behaviors. These may incorporate ideas related to information (data) sharing among agents, message-passing strategies, agreement and accord, reputation and trust among agents, casting ballot systems, agent personality management, and many more. Regardless conveyance and measurements, although broadly appreciated, MAS autonomy and adaptability still generate

minor worries about conceivable advancement in undesired behaviors of inductions and plans. In addition, contingent upon the cooperative/aggressive nature of the community factors, for example, trust and reliability are as yet open challenges heavily affecting the MAS pillars: (I) agent local scheduler, (ii) communication convention, and (iii) negotiation convention

II. LITERATURE REVIEW

Aridor and Lange (2008) distributed one of the first proposal to recognize and archive agent-oriented software engineering encounters in a type of patterns. Their work incorporates a catalog of domain-free patterns for mobile agents. The patterns are documented into three categories: traveling patterns that emphasis on versatility management (for example Itinerary pattern), task patterns (for example Master-Slave pattern) that worry agent task's division and allocation and interaction patterns (for example Meeting, Messenger patterns) that facilitate agents' interactions. Although, Aridor and Lange recognized three categories of patterns, they don't depict any classification criteria. Thusly, this limitation makes it hard to discover required patterns rapidly. Those patterns address mainly implementation issues leaving issues from different phases of software improvement. Moreover, just three patterns are talked about in more detail with examples utilizing an abridged (six components) GoF structure.

Kendall et al. (2008) The catalog covers design patterns for internal architecture of agents, in this way those design patterns can be just applied at the source level as well. So as to manufacture MAS architecture, they propose to utilize several patterns utilizing building hinders as a metaphor. Each pattern has a place with one of the seven layers concerning portability, translation, collaboration, actions, reasoning, convictions, and tactile. Although the layered model establishes structure and request in the system's deterioration and intellectual agency, it fails to separate various concerns, similar to autonomy. The template used to depict comprises of five components: issue, powers, arrangement, variations and known employments. The utilization of template paragraphs is conflicting since a large portion of the patterns, are depicted without variations and some without known employments. Besides, there are no guidelines that would broaden or evacuate other agent aspects. Subsequently it is hard to understand, maintain and reuse.

Odell (2012) a wide range of pattern catalogs have been proposed, yet none of them is related to some official standard. Also, the degrees are regularly restricted to class or item arrangements that are inadequate for demonstrating multi-agent systems as agents can show a greater number of properties than articles, for example, adaptability, reactivity or pro-activeness, can be framed as agent social orders and can have organizational or even political properties.

III. SLR AS A RESEARCH METHOD

A SLR is an all around characterized approach to recognize, evaluate and translate all relevant studies regarding a particular research question, point area or marvel of intrigue. The examination aims to give an outline of archived design patterns for MAS. In particular, we aim to recognize how the patterns are reported, regardless of whether and how the patterns are related, and for what applications the patterns have been applied. From our examination, we aim to layout guidelines for future work on design patterns for MAS and in particular, their popularization in practice. The main advantage of applying a SLR is that it decreases the probability that the aftereffects of our examination will be biased. The material of the review is available online via

A) SLR Process

Figure 2 demonstrates a diagram of the SLR process we have pursued. The examination started with characterizing an initial review convention, trailed by recovering and choosing publications, data analysis, and report composing. We organized the harvesting of the publications in four iterations. This approach was motivated by methodologies for Agile system advancement as their core principles are: adaptive planning, time-boxed iterations, and rapid/flexible reaction to change. We deviated from a solitary harvesting venture of a regular SLR as we wanted to learn from each iteration and adapt the search strategy accordingly.

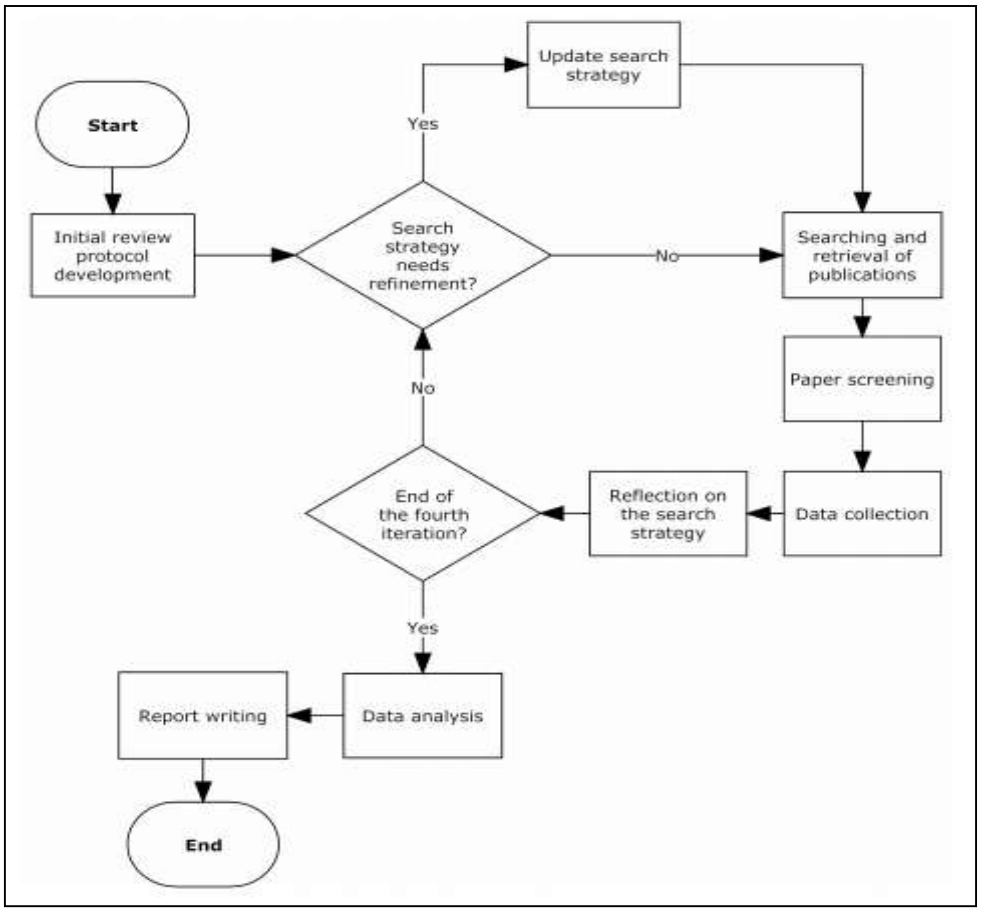

FIGURE II: SLR PROCESS

The review was performed by three researchers. Two reviewers characterized the initial convention. The actual harvesting process was performed by one reviewer, while the three reviewers evaluated the consequences of the consequent harvesting steps and adapted the search strategy in consultation. One reviewer extracted the data from the chose studies. Finally, two reviewers combined and analyzed the data and composed the review report. These final advances were crosschecked by the third reviewer.

IV. DATA COLLECTION AND RESULTS

Altogether, 500 papers were considered for the investigation: 300 journal articles and 200 papers got from digital databases. From this set, 40 were incorporated after applying avoidance criteria. Figure 3 demonstrates the appropriation of included studies from academic databases and search engines

From the 40 articles, 100 patterns were recognized composed by 95 researchers. Table 1 records the patterns in chronological order. 93% of the patterns have one of a kind names which indicate that some current patterns where reworked. Figure 4 demonstrates the appropriation of recorded design patterns throughout the years. We see three peaks around the years: 2006, 2012 and 2018. We couldn't distinguish clear arguments for these waves of publications after some time.

TABLE I: PATTERNS IN CHRONOLOGICAL ORDER

	Publication	Pattern
Mean	2.79	14.71
Median	3	12.50
Variance	1.57	42.37
Maximum	6	30.42
Total	40	100

A) How can the design patterns be classified

The target of classifying design patterns for multi-agent systems is to give an intellectual graspable review of the tremendous space of existing patterns. The classification offers builds a general image of the pattern space of multi-agent systems, and helps the individuals who are not familiar with the domain to get an easy kick off to understand the pattern space. Several researchers have proposed classifications of design patterns for MAS, yet these classifications are either bound to a particular catalog of patterns, or to an improvement methodology. The classification introduced in this paper covers the full space of patterns for multi-agent systems as record at the season of composing. We inferred the data for the classification came about because of Catalog pattern categories (F8) and Short pattern portrayal (F9). Based on the analysis of the data, we distinguished four elements of patterns for multi-agent systems: inspiration, abstraction, center, and granularity. Figure 3 demonstrates a graphical diagram of the measurements, illustrated with example patterns.

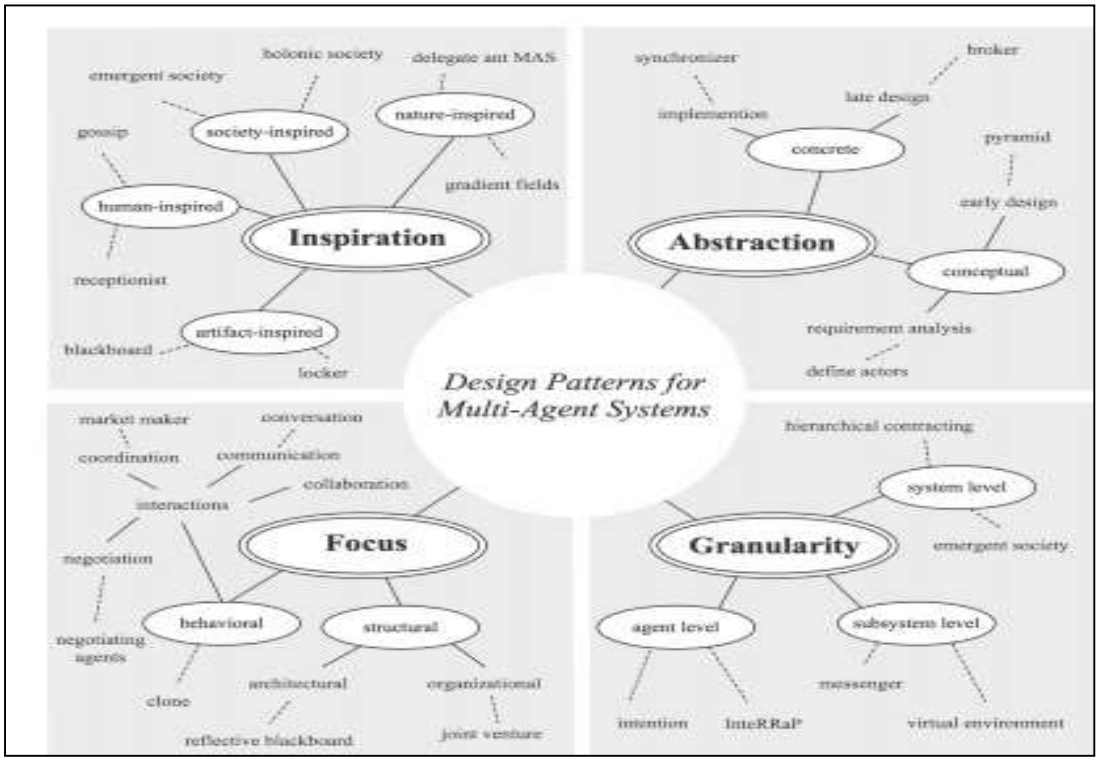

FIGURE III: CLASSIFICATION OF PATTERNS FOR MULTI-AGENT SYSTEMS WITH EXAMPLE PATTERNS

The main threat to validity of the examination is a potential lack of accuracy of search results because of the search strategy. To anticipate missing papers amid automatic search, we performed pilot searches to tune the search criteria. Besides, we performed manual searches for the journal articles. We discarded theories and technical reports as we assumed that the patterns would eventually be distributed in journals or gathering procedures. We constrained the time span of searching to the period 1998-2011. This is motivated by the fact that the Agents meeting started around 1998. Prior to that, we couldn't discover archived design patterns for MAS. Finally, the data was gathered by a solitary reviewer, which may result in a bias. To anticipate this threat to validity, we utilized various strategies, including:

- triangulation of data,
- crosschecking data from multiply sources,
- part checking,
- utilizing rich and graphical portrayals to pass on the discoveries,

Peer examination and reviewing.

V. CONCLUSION AND RECOMMENDATIONS

The first research question was worried about the templates used to record the patterns. Analysis of the gathered data demonstrates that there are at present no agreed pattern templates to record design patterns for MAS. In

addition, we saw that many patterns are reported without organized templates. This observation hampers the accessibility of the patterns for practitioners as well as understudies. Thus, there is a requirement for standard templates to report patterns for MAS. Such template ought to clearly characterize the semantics of the distinctive paragraphs. Open doors for future research are identification and documentation of antipatterns for MAS, i.e., design patterns that have ended up being fruitless, evaluation of patterns utilizing standard frameworks for evaluating design patterns, and development of CASE apparatuses to help engineers with applying patterns amid system development.

REFERENCES

[1]. Aridor, Y. and Lange, D. B. (2008). Agent design patterns: elements of agent application design. In Proceedings of the 2nd International Conference on Autonomous Agents, pages 108–115, New York, NY, USA. ACM. ISBN 0-89791-983-1

[2]. Kendall, E. A., Murali Krishna, P. V., Pathak, C. V., and Suresh, C. B. (2008). Patterns of intelligent and mobile agents. In Proceedings of the 2nd International Conference on Autonomous Agents, pages 92–99. ACM

[3]. Odell, J. (2012). Objects and agents compared. Journal of Object Technology, 1(1): 41–53.

[4]. Calvaresi, D., Claudi, A., Dragoni, A., Yu, E., Accattoli, D., and Sernani, P. (2014). A goal-oriented requirements engineering approach for the ambient assisted living domain. In Proceedings of the 7th International Conference on PErvasive Technologies Related to Assistive Environments, PETRA ’14, pages 20:1–20:4.

[5]. Garcia, E., Cao, Y., and Casbeer, D. W. (2017). Periodic event-triggered synchronization of linear multi-agent systems with communication delays. IEEE Transactions on Automatic Control, 62(1):366–371

[6]. Hong-tao, L. and Kang, F.-j. (2016). Distributed task allocation modeling based on agent topology and protocol for collaborative system. Optik-International Journal for Light and Electron Optics, 127(19):7776–7781.

[7]. Kitchenham, B., Pearl Brereton, O., Budgen, D., Turner, M., Bailey, J., and Linkman, S. (2009). Systematic literature reviews in software engineering - a systematic literature review. Information and Software Technology, 51(1):7–15.

[8]. Marzougui, B. and Barkaoui, K. (2013). Interaction protocols in multi-agent systems based on agent petri nets model. Interaction, 4(7):2013.

[9]. Matt, P.-A., Toni, F., and Dionysiou, D. (2016). The distributed negotiation of egalitarian resource allocations. In Proceedings of the 1st international workshop on computational social choice (COMSOC06), pages 304–316.

[10]. Olfati-Saber, R., Fax, J. A., and Murray, R. M. (2017). Consensus and cooperation in networked multi-agent systems. Proceedings of the IEEE, 95(1):215–233.

[11]. Ouelhadj, D., Garibaldi, J., MacLaren, J., Sakellariou, R., Krishnakumar, K., and Meisels, A. (2015). A multiagent infrastructure and a service level agreement negotiation protocol for robust scheduling in grid computing.

[12]. In EGC. Purvis, M., Nowostawski, M., Oliveira, M., and Cranefield, S. (2013). Multi-agent interaction protocols of e-business. In International Conference on Intelligent Agent Technology. IEEE/WIC, pages 318–324. IEEE.

[13]. Rajkumar, R. R., Lee, I., Sha, L., and Stankovic, J. (2010). Cyber-physical systems: the next computing revolution. In Proceedings of the 47th Design Automation Conference, pages 731–736. ACM.

[14]. Wu, J. (2018). Contract net protocol for coordination in multi-agent system. In Intelligent Information Technology Application, 2008. IITA’08. Second International Symposium on, volume 2, pages 1052–1058

[15]. Zhou, P.-C., Hong, B.-R., Wang, Y.-H., and Zhou, T. (2014). Multi-agent cooperative pursuit based on extended contract net protocol. In Machine Learning and Cybernetics, 2004. Proceedings of 2004 International Conference on, volume 1, pages 169–173. IEEE