A Design of the ViaBots Model for Industrial Assembly Line Application

Mara Pudane[a], Arturs Ardavs[b], Egons Lavendelis[c], Leonards Elksnis[d], Aiga Andrijanova[e], Peteris Ecis[f], Arturs Oniscenko[g] and Agris Nikitenko[h]
Department of Artificial Intelligence and Systems Engineering, Riga Technical University, Kalku 1, Riga, Latvia
{mara.pudane, arturs.ardavs, egons.lavendelis, leonards.elksnis, aiga.andrijanova, peteris.ecis,

Keywords: Adaptive Systems, Multi-agent Systems, Viable Systems Model, Heterogeneous Multi-robot Systems.

Abstract: Due to growing requirement of Industry 4.0 and general robotics infiltration into everyday life and industry applications, the adaptive heterogeneous multi-robot systems have become highly significant topic. While the adaptivity as a phenomena has not been researched for a long time in robotic systems, the organisational theory has analysed the adaptivity in long term, or viability, for several decades. ViaBots is organisational theory based framework for technical systems, that defines the functions that the system must fulfil to be viable. The goal of this paper is to present a design of the ViaBots model in case of heterogeneous multi-robot system, in particular, for an industrial assembly use-case.

1 INTRODUCTION

The number and functionality of robotic devices, including autonomous robots, robot manipulators etc., in the past years have grown rapidly. Such a rise in the possibilities has also boosted new requirements for robotic systems in general; they call for fully adaptive multi-robot systems that can adapt to the changes in the system itself, as well as to the fluctuations of the external environment (Dario, 2017). Such adaptive systems would offer multiple benefits, including fault-tolerance, enhanced behaviours of the system, and minimal human intervention. Only then it would be possible to talk about true autonomy of robotic systems.

However, it is hard to achieve such general long-term adaptivity that allows the system to adapt to different tasks that are not foreseen in the original design. There are some robotic systems, that manage to achieve at short-to-middle-term autonomy in specific cases, such as adaptivity to workload or changes in system configuration (Ardavs et al., 2019). However, there are little systems that are designed to adapt to potential rapid changed.

Adaptivity and systems' ability to persist over long periods of time has been researched extensively in the Organisational theory. The Viable Systems Model (VSM) defines the functions and functional dependencies that the system needs to be adaptive in long term, i.e., viable (Beer, 1985). VSM has been applied to technical systems as well and has showed promising results.

The remainder of the paper is organized as follows. In Section 2 we explain the existing problems in the development of adaptive heterogeneous multi-robot systems, in particular, the need for theoretical model for multi-robot systems' functional organisation, and the integration of heterogeneous robots. In Section 3, the VSM is described as well as its mappings for a technical system. Section 4 details application of the model for

[a] https://orcid.org/0000-0002-9188-5478
[b] https://orcid.org/0000-0002-4430-8037
[c] https://orcid.org/0000-0001-9912-035X
[d] https://orcid.org/0000-0001-8801-6071
[e] https://orcid.org/0000-0002-2509-6235
[f] https://orcid.org/0000-0001-6088-5873
[g] https://orcid.org/0000-0002-2189-4053
[h] https://orcid.org/0000-0002-5701-3094

Pudane, M., Ardavs, A., Lavendelis, E., Elksnis, L., Andrijanova, A., Ecis, P., Oniscenko, A. and Nikitenko, A.
A Design of the ViaBots Model for Industrial Assembly Line Application.
DOI: 10.5220/0009173709330940
In *Proceedings of the 12th International Conference on Agents and Artificial Intelligence (ICAART 2020) - Volume 2*, pages 933-940
ISBN: 978-989-758-395-7; ISSN: 2184-433X

the use-case design and finally, in Section 5, conclusions and future work is given, as this is research in progress.

2 RELATED WORKS

The related work section is divided into two parts: first of all, the concept of adaptivity in the multi-robot systems so far is reviewed, looking at existing examples of the multi-robot systems frameworks, as well as adaptivity as a concept in other areas. Secondly, the methods for creating heterogenous multi-robot systems are reviewed, with the focus on collaboration among robots.

2.1 The Logical Structure of the Heterogeneous Multi-robot Systems

In order to achieve a long-term operation, several architectures have been proposed for adaptive multi-robot systems. Some examples of such approaches include NASA's more universal Autonomous Nano Technology Swarm (ANTS) concept (Vassev et al., 2012), or narrower biology-inspired intelligent control (Jafari & Xu, 2019). Unfortunately, while most of the studies concentrate on specific missions or applications of robot teams, only very few propose formal frameworks or methods of system design. One of such approaches uses Event-B and PRISM design methods to derive technical design of the system through iterative steps and assesses probability of goal achievement thereby providing guidance for further developments (Tarasyuk et al., 2013). Similarly, in (Gerostathopoulos et al., 2016) it is proposed to design self-adaptive system by using the invariant method. The developed framework, IRM-SA, tackles the design complexity of invariant methods thus enabling the design of such systems.

Current requirement for the robotic systems has created an emerging need for general formal framework for design and development of resilient and adaptive systems (Ardavs et al., 2019). This framework would have to fulfil two main tasks – ensure systems' adaptivity to the changes in the environment and within itself; and reduce the complexity of the design process (Ardavs et al., 2019).

In the organizational theory, a concept of the long term adaptivity, or viability, has been researched for several decades, resulting in the VSM (Beer, 1985). Essentially, the VSM describes the functions and functional dependencies that the systems need to be viable. While the VSM initially was only used to analyse and improve organisation related aspects, such as information flows (Kirikova & Pudane, 2014), in the recent years due to the new requirements of multi-robot systems, VSM has been adapted to some technical systems. Examples of these include the design of the smart distributed automation systems (Bonci et al., 2019) as well as cyber security management (Spyridopoulos et al., 2014). To authors' knowledge, the most extensive research on VSM applicability in the technical systems was done in (Ardavs et al., 2019) where the technical-systems-appropriate VSM design was developed and adapted to conveyor belt simulation in a multi-agent environment. The results showed that using the VSM as a logical-level framework enables agent self-organisation and improves the output of the system. This leads to conclusion that VSM-based model would improve the performance of a real system as well and is applicable to industrial assembly task done by highly heterogeneous multi-robot system at the conveyor belt. Another reason to believe this conclusion, is that the fractal nature aside, the functions of VSM are similar to MAPE-K loop introduced by IBM (IBM Coorporation, 2005).

2.2 MAS as a Logical Abstraction Layer for Multi-robot System

Another challenge regarding heterogeneous multi-robot systems is enabling the control of the whole system. As the complexity of robots is growing, the control of each robot separately becomes increasingly detailed. The control of multiple robots is an even more challenging task. Additionally, different robots of various manufacturers have different interfaces and operating systems which leads to difficulties in the integration.

To cope with the complexity in various domains, a common approach is to develop a layered design and decision-making architectures. In multi-agent systems, for example, design is usually divided into micro and macro level where the micro level concerns the design of a single agent as opposed to macro level which includes designing multi-agent system as a whole (Wooldridge, 2009). Another related approach to heterogeneous systems, is aggregate computing which is alternative way for implementing such systems (Beal et al., 2015). While (Bures et al., 2016) criticizes MAS as an approach to implement complex heterogeneous systems, we see VSM as a tool to mitigate the critic, in particular, of lack of the architecture in multi-agent systems.

Similarly, the decision making process must be divided in the multi-robot systems. In this case, three abstraction layers must be used (Lavendelis & Nikitenko, 2015). On the lowest level, the robots control their mechanisms and perform discrete actions. On the middle level, the decisions are made that are needed to fulfil tasks (such as going to particular location). At the top level, the multi-robot management decisions are made. These decisions include cooperation, task distribution and other high-level tasks that are arguably the most challenging.

In several cases, the developers have used multi-agent layer for the management of heterogeneous systems, such as autonomous transport vehicles' systems (Martin et al., 2019), or multi-robot systems (Lavendelis & Nikitenko, 2015). In the latter case, each agent represents a physical device (robot, sensor, manipulator, etc.) within this abstraction layer. The high-level decisions are negotiated among agents; each agent manages the corresponding device. This allows to abstract from low level details while performing such tasks as allocation and reallocation, collaborative mapping, coordination, etc.

Use of multi-agent paradigm also allows building open systems since a new agent representing a new device can be added to the system. Still, mechanisms to adapt the situation to the changes in the composition of agents are necessary. Since the VSM deals with the management of the whole system, the multi-agent system is the level where the VSM is implemented.

3 ViaBots MODEL

The ViaBots model consists of the VSM functions transformed into agent-related terms and concepts.

3.1 The Viable Systems Model

The viable systems model consists of five functional blocks, called subsystems (Beer, 1985). These functional blocks: S1, S2, S3, S4 and S5 respectively represent five main functional groups: Operation, Coordination, Control, Intelligence and Policy. The instances of the S1 subsystem (Operation) are the units that perform the actual work in the system, i.e., everyday tasks, while the rest of the subsystems represent the Management functions. S2, or Coordination performs the tasks of local management. One S2 instance is attached to every S1 instance. S2 instances negotiate resources, and if no consensus can be acquired, turn to S3. S3 (Control) is global level management. The main function of S3 is to oversee the execution of the global plan by observing the work of S1 and S2 and, if needed, redistributing resources. The plan is changed in close coordination of S3 which is informed of the current situation in the organisation, and S4 (Intelligence) which observes external environment and holds the model of changes in the external environment. Finally, S5 (Policy) is the subsystem that generates general policy for the system as well as fulfils the role of representation.

There are different types of dependencies among functions (Pudane, 2013). The type *a* link is between S1 and S2, as well as between S1 units and external environment. The goal of these links is to manage complexity, i.e., reduce the number of states with which system needs to deal. The *b* type link is coordination and negotiation link that exists among S2 instances. Type *c* link is directed link that represents monitoring and is located between S3 and S1, as well as between S4 and external environment. Type *d* link is a cooperation or integration link between S3 and S4. Finally, type *k* link is a directed control link that exists between S5 and S3, S4; S3 and S2, and S2 and S1.

3.2 The Adaptation of VSM to Technical Systems

To adapt the VSM to multi-agent system, several mappings were done (overview in Tables 1 and 2). First, all the functions of VSM subsystems were identified and mapped either to agent tasks, or communications (e.g., negotiations), and the subsystems themselves where mapped to agent roles. The roles can then be comprised into agents dynamically. Such an approach gives more flexibility since there are no limitations on what physical units must perform the functions. Then, the roles were correspondingly mapped into behaviours and tasks were implemented as procedures. Each agent that carries out one or several behaviours, can now represent either a software agent, or a robot. The Table 1 contains identified functions, corresponding tasks, task mappings into roles and then – agents. Table 2 contains the interaction design with functions that correspond to channels, and multi-agent system interaction types. The mapping in detail is discussed in (Ardavs et al., 2019).

4 USE-CASE DESIGN

The benefits of the ViaBots model is best seen in a larger scale and complexity (Ardavs et. al, 2019), for this reason use case is a physical assembly line. Similarly to the simulation in (Ardavs et. al, 2019), it

Table 1: Task allocation to roles and role allocation to agents in ViaBots model.

VSM	Function	Task	Role	Agent
S1	Depends on the system	1..n tasks depending on the variety of the agents	1.. n roles depending on the domain	Either physical units that carries out roles (i.e., robots), or the ones that will run on one computer or in one domain. Assigned to agents having variety reduction and amplification capabilities
Type A	Attenuators and Amplifiers	Attenuate variety task and amplify variety tasks between external environment and system		
S2	S1 element coordination	Negotiation Task	S2	
	Solving of conflicts of S1			
	S1 resource relocation			
Type A	Attenuators	Attenuate variety task between Operation and Management		
S3	Resource distribution	The resource distribution task	S3	Assigned to one agent irrelevant from the already assigned roles
S3	Interpretation of policy decisions	Interpret high level instructions to lower level task		
S3	Development planning according to environment and system states	Opinion task-current situation		
S4		Opinion task-future states	S4	Assigned to one agent irrelevant from the already assigned roles Assigned to agent having monitoring environment capabilities
S4	Suggest. for safety policy	Safety and resilience task		
S4	Learning	Learning task		
S4	Management of external contacts	Search task		
S4	Environment monitoring	Env. monitoring task		
S5	Representation	HCI task	S5	Assigned to one agent irrelevant from the already assigned roles
S5	Investment in structure and policy formation	Policy task		

Table 2: The interactions in ViaBots model.

In VSM		**Function**	**MAS interaction type**
System	Channel		
S2	Type B	S1 element coordination	Coordination protocol
S2		Solving of conflicts of S1	Negotiation protocol
S2		S1 resource relocation	Resource negotiation protocol
S2	Type K	Control channel function	Sent once, message-received response
S3	Type K	Control and monitoring over S1 and S2	Sent once, message-received response
	Type C		*Inform* message
S3 and S4	Type D	Development planning according to environment and system states	Negotiation protocol
S4	Type C	Monitoring of system itself	*Inform* message
S5	Type C	Monitoring of cooperation of S3 and S4	*Inform* message
	Type K		Sent once, message-received response
Channel	Type A	Amplify variety between Operation and Management	Broadcast

was decided to build a real-life conveyor belt with multiple robots as workers.

4.1 The Conveyor Belt Use-case

The following robots are used with the conveyor belt: in the robotics community well-known packaging robot Baxter with fixed base and two synchronized 5 DOF manipulators [1], an industrial 5 DOF ABB manipulator [2] and several custom-made 3 DOF manipulators. Additionally, a humanoid robot with human-like hands – Pepper[3] is used in case no other robot is capable to do the tasks. Such a set of robots is highly heterogeneous in terms of the manipulators

[1] Rethink Robotics, 2015. *Baxter™ SDK API Documentation*, retrieved from http://api.rethinkrobotics.com/.

[2] ABB Robotics, 2019. Robotics product range. Creating the flexible, collaborative and connected Factory of the Future. ABB Robotics.

[3] Softbank Robotics, 2019. *Pepper*, retrieved from: https://www.softbankrobotics.com/us/pepper.

used, time necessary to do the task and parts available to each robot. At the same time, multiple robots can do any or most of the tasks. This enables modelling an assembly line with different robots and as a consequence necessity to find appropriate task allocation among the robots to optimize the performance of the whole assembly line.

As tasks for the conveyor belt, boxes with round holes were designed. Robots must insert cones in these holes in a specific pattern (Figure 1). The different types of parts are achieved by adding a marker, to what type the box belongs (i.e., what is the cone pattern for this particular box type). There are two sizes of the cones, and not all the robots can insert both types of the cones.

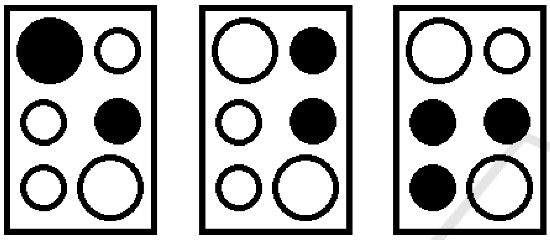

Figure 1: The tasks for the assembly line - different layouts.

4.2 The Implementation of the Tasks of the ViaBots Model

The tasks of the ViaBots model were adapted to the conveyor belt use-case (overview in the Table 3). The S1 functionality in this case is part assembly that is modelled as moving the cones from the dispenser to the boxes. The complexity of the environment is reduced by robot sensors; we added infrared sensors to the conveyor belt to simplify the sensor data processing. Negotiation is performed by S2 agents based on the time needed for task completion (some manipulators are slower than others, it takes more time to pick up bigger cones) and availability of the details. If manipulator runs out of cones, it is defected and cannot continue working until additional cones are added to dispenser.

The interaction among Operation and Management is performed only when one of the subsystems S1 cannot continue its work. Such a situation can occur in the following cases: (a) there are no more cones of a particular size; (b) the robot that has the needed cones is working for another S1 unit. In such a case, S3 redistributes resources, i.e., makes the S1 units share the same robots.

S4 holds the environment model, i.e., it foresees the upcoming assembly task sequence; together with S3 they choose appropriate resource redistribution, and S3 can ask for external resources (i.e., ask appropriate resources to the human user). The assembly task sequence can be changed as a change in the external environment.

Due to complexity in real-life implementation, S5 was left to user. This concerns overall configuration of the conveyor belt.

Table 3: The adaptation of ViaBots model tasks to conveyor belt use-case.

In VSM	Task	Implemented as
S1	1..n tasks depending on the variety of the agents	Robot tasks of picking up a part and placing it in the box in the corresponding place.
Type A	Attenuate variety task and amplify variety tasks between external environment and system	The environment was simplified by adding infrared sensors to announce that the box had arrived. The robot must pick and place the detail, based on fixed position. S1 is considered to be the robot manipulator together with infrared sensor.
S2	Negotiation Task	Negotiating among S2 units based on time for part insertion and available robots
Type A	Attenuate variety task between Operation and Management	Intervention required when errors or resource lacking occurs
S3	The resource distribution task	Calculating the times for task execution and enforcing the new tasks
S3	Interpret high level instructions (policies) to lower level task	Implemented through human interaction with system
S3	Opinion task based on current situation	Weights of the calculated part sequence
S4	Opinion task based on future states	Weights of the foreseen part sequence
S4	Safety and resilience task	Calculating the number of left-over cones.
S4	Learning task	Part sequence which S4 learns; in this case the part sequences will be fixed.
S4	Monitoring of environment task	Due to simplicity of environment acquired through S1 and S2
S5	HCI task	Performed by user
S5	Policy task	Performed by user.

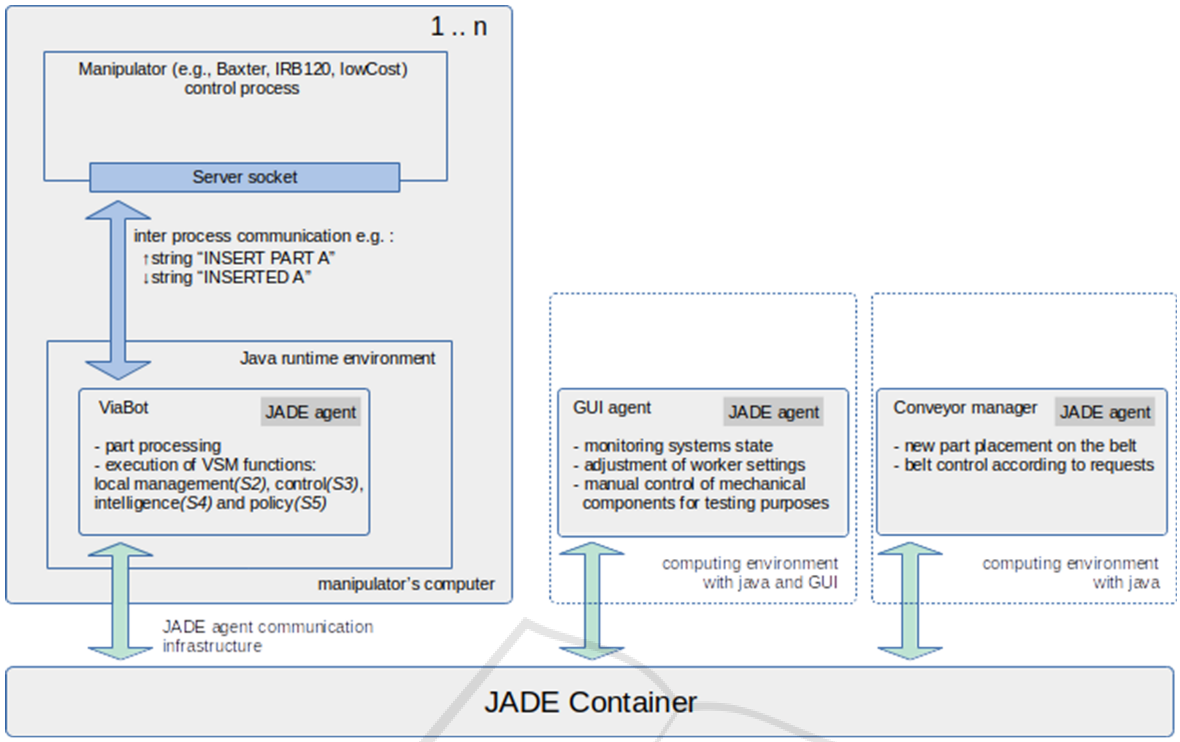

Figure 2: Deployment diagram of the assembly line use-case.

4.3 Overall Scheme of the Use-case

The deployment diagram of the use-case is depicted in Figure 2. The multi-agent system of the use-case was implemented in JADE (Bellifemine et al., 2007). JADE container is populated with an agent for each of the manipulators by the conveyor belt. Each of these agents via simple ACL messages (FIPA, 2002) manages the corresponding manipulators. The subsystem roles, according to ViaBots model, are implemented as JADE agent behaviours and are assigned to agents dynamically. Depending on the starting sequence, one of the agents runs not only S1 and S2, but also S3 and S4 functions. Physically, the agents will run on the manipulators' computers. Additionally, on a separate computer, a GUI agent captures the states of the conveyor belt and provides the user interface to interact with conveyor belt system. Finally, Conveyor manager is a specific agent that would control the conveyor belt and move it, when needed.

In Figure 3, a general sequence diagram of interaction among the behaviours is depicted. Conveyor agent serves as a sensor to all the S1 systems and message to S1 agents when the new box has arrived. It also places incoming boxes on the belt. All the S2 Behaviours will contain a task that determines, if the incoming box requires insertion of the given unit's type. If the box refers to particular S1 unit, the message is sent to S2 of new task, otherwise, the box is ignored. Then, the S2 requests to corresponding S1 agents if they are able to perform the task and in what time. If the answers are positive, S2 picks one of the agents to perform the work. Otherwise, the message is sent to S3 that there are no resources available. S3 gathers the information from all the S2 on current situation, as well as information from the S4 on the future states. Finally, S3 generates a new resource distribution. If resources cannot be redistributed, the working cycle ends.

5 CONCLUSIONS AND FUTURE WORK

The work presented in the paper is on-going research and the use-case is about to be tested to acquire first results. The theoretical results achieved earlier in (Ardavs et al., 2019) based on a simulated environment has proven the usefulness of the VSM in the technical systems It was proven in the simulator that overhead calculations are not significant, especially with large number of robots. The purpose of the implementation on a realistic assembly line done within this paper is to collect data from close to

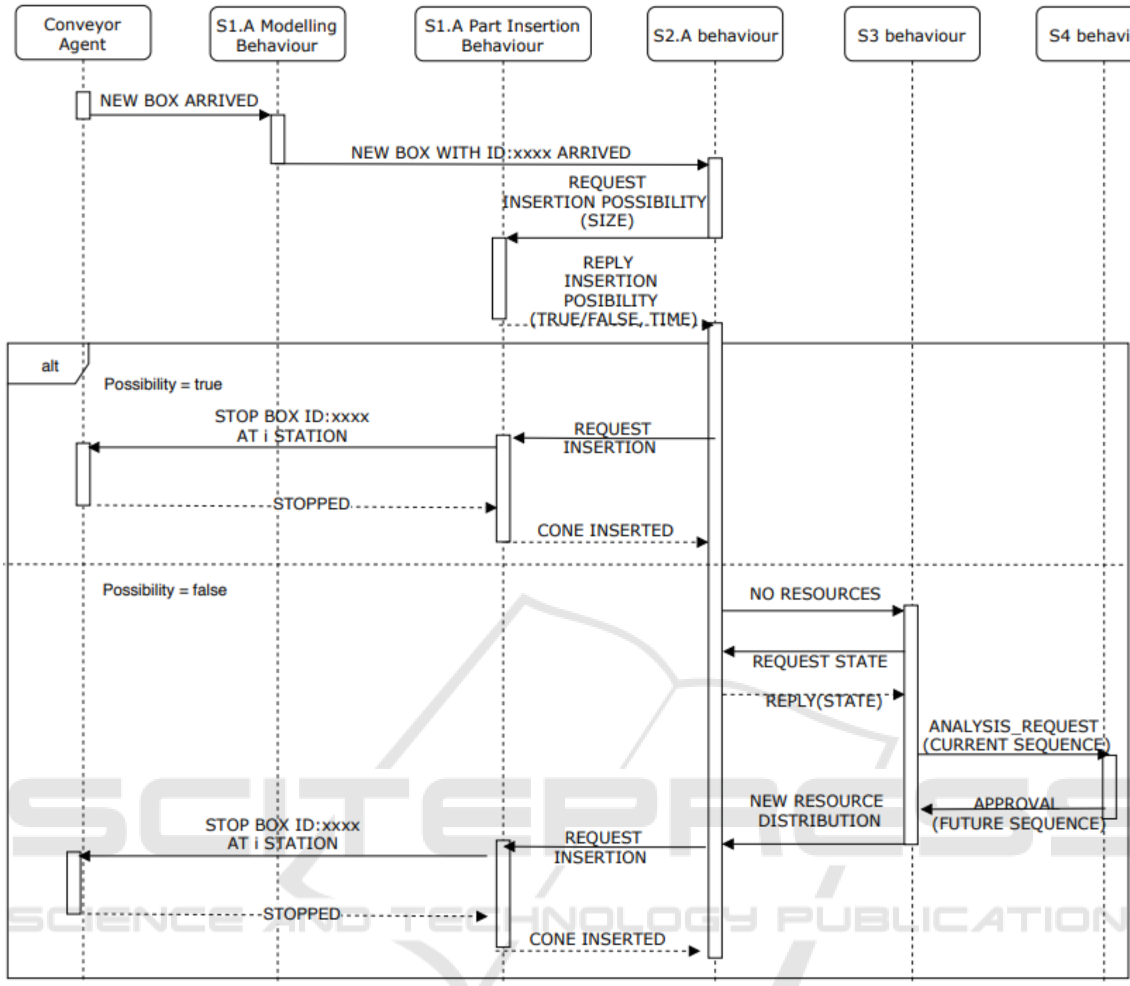

Figure 3: Scenario sequence diagram of the assembly line use-case.

real environment and further analyse the efficiency of the proposed model. The ultimate goal of the research is to create a framework for long term adaptivity of heterogeneous multi-robot systems. This would enable such systems as robotized assemblers or other production lines to adapt to unexpected events or slight changes in the tasks done. At the moment, such adaptation can be done only by human operator.

The future work is to do experiments in real environment to compare the efficiency of the ViaBots-model-based assembly line against the one with fixed configuration to prove that the results achieved in simulated environment are valid.

It is planned to introduce unpredictable events to enable test long-term autonomy in these scenarios. It must be considered that any system (including natural systems) can adapt only to tasks that can be physically done with available tools. For this reason. the unpredictable events in conveyor case will be different kinds of assembly tasks, such as a box with different hole configuration (i.e., 9 holes).

Later on the model will be applied to different mobile robot based scenarios to validate its applicability to various tasks since the aim of the research is to develop a general model that can be applied to various multi-robot systems.

ACKNOWLEDGEMENTS

This work has been supported by the European Council Seventh Framework Program FLAG-ERA project “Rethinking Robotics for the Robot Companion of the Future” (RoboCom++).

REFERENCES

Ardavs, A., Pudane, M., Lavendelis, E., Nikitenko, A., 2019. Long-Term Adaptivity in Distributed Intelligent Systems: Study of ViaBots in a Simulated Environment. In *Robotics,* 8, 25.

Beal, J., Pianini, D., Viroli M., 2015. Aggregate Programming for the Internet of Things. In *Computer*, 48, 9, pp. 22-30.

Beer, S., 1985. *Diagnosing the Systems for Organizations*, John Wiley & Sons: Pennsylvania, USA.

Bellifemine, F.L., Caire, G., Greenwood D., 2007. *Developing Multi-Agent Systems with JADE* (Wiley Series in Agent Technology). John Wiley & Sons, Inc.: Hoboken, NJ, USA.

Bonci, A., Pirani, M., Cucchiarelli, A., Carbonari, A., Naticchia, B., Longhi, S., 2018. A Review of Recursive Holarchies for Viable Systems in CPSs. In *Proceedings of IEEE 16th International Conference on Industrial Informatics (INDIN)*, Porto, Portugal, July 18-20, 2018, pp. 37-42.

Bures, T. Plasil, P., Kit, M., Tuma, P., Hoch, N., 2016. Software Abstractions for Component Interaction in the Internet of Things. In *Computer*, 49, 12, pp. 50-59.

Dario, P., 2017. Fet-Flagship proof-of-concept Project: Rethinking Robotics for the Robot Companion of the future. Presentation at *Ro-Man 2017: 26th IEEE International Symposium on Robot and Human Interactive Communication, August 28, 2017*.

Foundation for intelligent physical agents (FIPA), 2002, *FIPA ACL Message Structure Specification.*

Gerostathopoulos, I., Bures, T., Hnetynka, P., Keznikl, J., Kit, M., Plasil, F., Plouzeau, N., 2016. Self-adaptation in software-intensive cyber–physical systems: From system goals to architecture configurations. In *Journal of Systems and Software*, 122, pp. 378-397.

IBM Coorporation, 2005, An architectural blueprint for autonomic computing, Autonomic Computing, White Paper, Third edition IBM: USA.

Jafari, M., Xu, H., 2019. A biologically-inspired distributed fault tolerant flocking control for multi-agent system in presence of uncertain dynamics and unknown disturbance. In *Engineering Applications of Artificial Intelligence*, 79, pp. 1-12.

Kirikova, M., Pudane, M., 2014. Viable Systems Model Based Information Flows. In Proceedings of 17th East European Conference on Advances in Databases and Information Systems, New Trends in Databases and Information Systems, *Advances in Intelligent Systems and Computing,* 241(1), pp. 97-104.

Lavendelis, E., Ņikitenko, A., 2015. Software Abstraction Layer Based Multi-Robot System Technology. In *STO-MP-AVT-241 - Technological and Operational Problems Connected with UGV Application for Future Military Operations*, Poland, Rzeszow, 20-22 April, 2015. Rzeszow: NATO STO, pp.18.1-18.8.

Martin, J., Casquero, O., Fortes, B., Marcos, M., 2019. A Generic Multi-Layer Architecture Based on ROS-JADE Integration for Autonomous Transport Vehicles. In *Sensors*, 19, 69.

Pudane, M., 2013. *Knowledge Flow Analysis using Viable Systems Model,* Master Thesis, Riga Technical University, Riga, Latvia (in Latvian).

Spyridopoulos, T., Maraslis, K., Tryfonas, T., Oikonomou, G., Li, S., 2014. Managing cyber security risks in industrial control systems with game theory and viable system modelling. *In Proceedings of 2014 9th International Conference on System of Systems Engineering (SOSE)*, Australia, June 9-13, pp. 266-271.

Tarasyuk, A., Pereverzeva, I., Troubitsyna, E., Laibinis, L., 2013. Formal Development and Quantitative Assessment of a Resilient Multi-robotic System. In *Proceedings of: SERENE 2013: International Workshop on Software Engineering for Resilient Systems*, Kiev, Ukraine, October 3-4.

Vassev, E., Sterritt, R., Rouff, C., Hinchey, M., 2012. *Swarm Technology at NASA: Building Resilient* Systems. In IEEE IT Professional, 14, pp. 36-42.

Wooldridge, M., 2009. An Introduction to MultiAgent Systems. Wiley Publishing, 2nd Edition.