Decentralized coordination of heterogeneous ensembles using Jadex

Oliver Kosak, Constantin Wanninger, Andreas Angerer, Alwin Hoffmann, Andreas Schierl, Hella Seebach

Decentralized Coordination of Heterogeneous Ensembles Using Jadex

Oliver Kosak, Constantin Wanninger, Andreas Angerer, Alwin Hoffmann,
Andreas Schierl, and Hella Seebach
Institute for Software & Systems Engineering, Augsburg University, Germany
E-Mail: {kosak, wanninger, angerer, hoffmann, schierl, seebach}@informatik.uni-augsburg.de

Abstract—**Multi-agent systems can be a viable choice for realizing self-organizing systems consisting of reconfigurable software components. We present a real-world system consisting of heterogeneous air and ground robots whose behavior and coordination is orchestrated by a MAS in a decentralized manner. The system is able to cooperatively transport large-scale measuring equipment and is used for environmental observation, such as in-situ measuring of temperature.**

Index Terms—**Swarm Robotics; Cyber-Physical Systems; MAS; Heterogenuous Ensembles; Sensor Systems; UAV**

1. Introduction

The ability to coordinate is an important aspect in systems consisting of multiple robots that should achieve a certain task cooperatively, deal with unexpected events, and adapt to changing tasks. Currently, a robots autonomy is usually limited to fulfilling its basic functionality. Unmanned Aerial Vehicles, for instance, are only able to fly to prescribed waypoints and hold their position. Coordination, however, requires a degree of flexibility that has to be provided at a higher level of abstraction. Our approach employs a distributed multi-agent system (MAS) based on the Jadex Active Components Framework [1] for coordinated planning and task allocation in an ensemble of heterogeneous robots. MAS are specially suited for this purpose, as they 1) encapsulate lower level functionality in agents that can make autonomous, situation-aware decisions 2) they abstract from the distributed execution of agents (e.g., communication on hardware level). In our demonstration, we show that these properties enable a heterogeneous ensemble of robots to cooperatively execute different a-priori unknown tasks. Additionally, this system is used to transport a large-scale temperature sensing device to perform environmental measurements. Details on the architecture we developed for the coordination of multiple robots are provided in Sect. 2. Sect. 3 further describes different application scenarios.

2. Approach

Our MAS-based architecture is shown in Fig. 1. It is geared to the application problem class *Multi Robot-Multi Task-Problem* [2]. The contributions thus are 1) autonomous

solving of tasks by coordination of multiple robots; 2) contributing to multiple tasks with a single robot in parallel; 3) providing users with a high-level task abstraction to work with; and 4) allow for issueing tasks dynamically at runtime of the system. Currently, ths system is used to coordinate varying numbers of flying robots to achieve user-defined goals (e.g., flight formations) in a decentralized manner. There are two types of devices in our architecture: *Self-organizing autonomous Devices* (SoD), each controlling the robot they are attached to, and a semi-autonomous *User Device* for instructing SoDs.

The *User Access Agent* (located on the *User Device*) offers an interface, which allows an operator to create new tasks (e.g., T_1 = "*Generate line formation!*" or T_2 = "*Fly a path along coordinates X, Y and Z!*") and subsequently distribute these tasks to an ensemble of SoDs. For distribution, the *User Access Agent* identifies *Blackboard Agents* (BA, located on SoDs) within its communication range, and transmits the task to them. When finally a task is solved by an ensemble of SoDs, the *User Access Agent* is informed by the BAs and provides the operator with this information.

A status of a task can be $\{unsolved, active, solved\}$. BAs that receive a new task classify its status as *unsolved* and gossip [3] the task and its status to other BAs on reachable SoDs. This is necessary, as the communication graph in a network might not be fully connected at all times and knowledge about tasks and their status can be lost with a certain probability. By gossiping information about tasks and their status, the chance of task fulfillment can be maximized while the risk of executing a task multiple times is minimized – always keeping in mind that the loss of information can never be canceled out completely [4]. For simplification, we assume a reliable WiFi communication.

The System-of-Systems characteristic of our architecture originates from the idea that each SoD runs several agents, so that a SoD itself represents a MAS. Each agent on an SoD encapsulates a specific functionality that contributes to the self-organizing characteristics of the SoD as a whole. In our architecture, BAs are not only able and responsible for distributing information about tasks and their status in the system, but also to decide whether tasks can be solved by a single SoD (e.g., T_2) or require cooperation of multiple SoDs (e.g., T_1). According to this decision, the *Blackboard Agent* forwards a task either to the local *Individual Coordination Agent* (ICA) in the first case or to
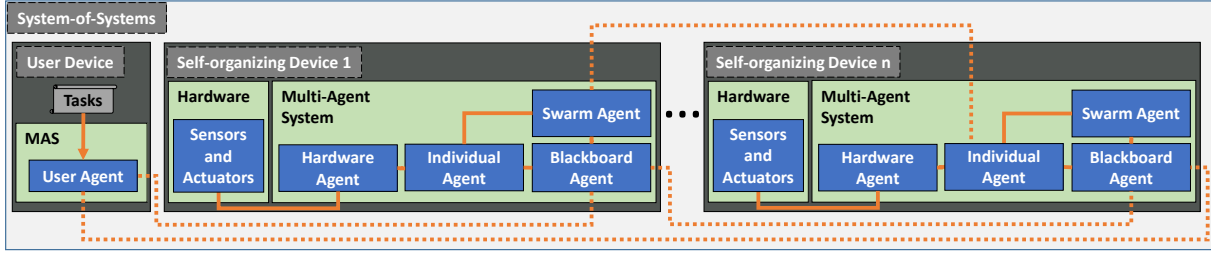
Figure 1. System-of-Systems architecture: Each of the n *Self-organizing Devices* (SoD) provides the platform for an equal set of agents that form a Multi-Agent System running on the device. In addition to this, a *User Device* represents the instructional part, offering an interface to a human operator for introducing new tasks. Solid lines indicate intra-device communication, dotted lines inter-device (wireless) communication for a scenario where the *Swarm Coordination Agent* on SoD$_1$ coordinates n *Individual Coordination Agents* on its local device as well as on remote devices.

the *Swarm Coordination Agent* (SCA) in the latter case. Both then further analyze the tasks.

The SCA has the responsibility to solve tasks that need cooperation. Therefore, it is able to transform an original task T_i into a set of n subtasks, $T_i^{sub} = \{T_i^1, ..., T_i^n\}$, where each of them can be solved by a single SoD. The task decomposition is currently done by analyzing spatial information given in the task (e.g., desired flight formations). These information are also used to identify situations where coordination of the subtasks execution is necessary (e.g., cooperative transport needs coordinated movement). Additionally, the SCA tries to find and recruit a team of n ICAs (on the local or a remote SoD) that is able to solve T_i together. Having recruited a suitable team, the SCA starts a task activation routine where the status is set to *active*, its local BA is informed to further gossip the update, and one appropriate task $\in T_i^{sub}$ is assigned to each cooperating ICA. As soon as every cooperating ICA has solved its assigned task completely or partially, and informed the coordinating SCA, the *Swarm Coordination Agent* triggers a finalizing routine which sets the status of T_i to *solved*, informs its local BA to gossip the update, and dissolves the previously formed team of ICAs.

If, in contrast to that, the *Blackboard Agent* classifies a task as solvable by a single ICA, this task is transmitted to the local ICA. It runs a routine similar to the task activation process of the SCA including information exchange with its local BA about status updates on the task. In both cases, the *Hardware Management Agent* provides the hardware interface to currently attached hardware as it has knowledge about necessary protocols and other specific hardware properties. The ICA uses this interface to determine the device's qualification for tasks as well as to correctly address actuators for executing actions necessary to solve a task (e.g., "*Fly to coordinate X!*").

3. Application

The architecture presented above is employed to coordinate an ensemble of multiple UAVs and a ground robot. This system is used to precisely capture the spatial distribution of temperature in the environment by coordinated movement. Our activity is part of the geographic measurement campaign ScaleX 2016 in the pre-alpine observatory project TERENO[1]. In the domain of environmental geography, UAVs are, in general, an interesting and relatively cheap alternative to stationary remote sensing devices as UAVs are able to perform measurements *in-situ*, i.e., at the exact location of interest. The availability of small, light-weight sensing devices is, however, a crucial factor. The Distributed Temperature Sensing (DTS) approach allows for measuring temperature along a fibre-optic cable [5]. Due to the light weight of the cables, they can be mounted on UAVs. By coordination of multiple UAVs, great flexibility for in-situ measurements in different heights and along various fibre patterns is obtained. The DTS logging unit (Oryx DTS[2]) has to be installed on a ground robot (also integrated as SoD), due to its weight. The same software architecture is used to set up an indoor demonstration at SASO 2016. Instead of GPS, an indoor tracking system supplies each device with data for self-localization. For a safe demonstration, we will place our robots inside a secured cage of 5 x 9 meters. By issuing tasks via the UAA, the ensemble will be able to fly certain patterns, and to change and precisely maintain its flight formation despite the ensemble's size. More details including pictures and videos are available online[3].

References

[1] A. Pokahr, L. Braubach, and K. Jander, "The Jadex project: Programming model," in *Multiagent Systems and Applications*, ser. Intelligent Systems Reference Library, M. Ganzha and C. L. Jain, Eds. Springer, 2013, vol. 45, pp. 21–53.

[2] B. P. Gerkey and M. J. Matarić, "A formal analysis and taxonomy of task allocation in multi-robot systems," *The International Journal of Robotics Research*, vol. 23, no. 9, pp. 939–954, 2004.

[3] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–114, Aug 2002.

[4] K. P. Birman, "Network perspective," in *Guide to Reliable Distributed Systems: Building High-Assurance Applications and Cloud-Hosted Services*. Springer, 2012, pp. 101–143.

[5] M. J. Zeeman, J. S. Selker, and C. K. Thomas, "Near-surface motion in the nocturnal, stable boundary layer observed with fibre-optic distributed temperature sensing," *Boundary-Layer Meteorology*, vol. 154, no. 2, pp. 189–205, 2015.

1. http://www.imk-ifu.kit.edu/scalex.php
2. http://www.sensornet.co.uk/oryx-dts-sensors/
3. http://www.isse.de/software/saso2016