19th ICCRTS
"C2 Agility: Lessons Learned from Research and Operations"

Commanding Heterogeneous Multi-Robot Teams

Topics
Experimentation, Metrics, and Analysis
Modeling and Simulation
Architectures, Technologies, and Tools

Names of Authors
Thomas Remmersmann, Ulrich Schade, Alexander Tiderko
Fraunhofer Institute for Communication, Information Processing and Ergonomics FKIE

Point of Contact
Thomas Remmersmann
Fraunhofer Str. 20
53343 Wachtberg
GERMANY
{thomas.remmersmann,ulrich.schade,alexander.tiderko}@fkie.fraunhofer.de

ABSTRACT

Bolstering up a military unit, e.g., an infantry platoon on a recce mission, by robots often is a double-edged sword. On the one hand, the robots are able to support the soldiers in multiple ways, they can transport bulky equipment, they can enter risky spots, and they may have sensor suits that help to detect dangers of all kinds. On the other hand, robots need to be equipped with energy sources that are cumbersome by themselves. In addition, the robots must be commanded. In order to optimize the support for their unit, it is often necessary to have a team of heterogeneous robots, e.g., UAVs as well as UGVs all with different sensors and specific abilities. Such a team, however, is even harder to command than a team of homogeneous robots.

Our research aims at simplifying commanding teams of heterogeneous robots. In order to achieve this aim, we use the standards BML (Battle Management Language) and ROS (Robot Operating System) to communicate with the robot team. BML is used since our approach to commanding robots is from the language point of view very similar to commanding simulated units. Thus, we use language constructions modeled on those we developed as part of the NATO research groups on BML, NATO MSG-048 and NATO MSG-085. Currently, we are testing to use one single mobile GUI, also modeled on our NATO research groups' results. That GUI is implemented on a tablet and enables the controller to command a team of two UAVs and four UGVs. All the robots can be equipped with different sensor suits.

This article presents our solutions about the following topics essential for the described challenge. First, the robots have to introduce themselves by communicating their current abilities and their status to the commander. Second, the commander gives the commands to the robot team in a mission kind fashion. Thus, third, there has to be a kind of intelligence in the robot team that calculates sub tasks out of a given command and distributes these sub tasks to the team members taking the members' specific abilities into account. Fourth, the robot team has to fuse sensor data in order to send a unified picture back to the commander in order to contribute to the operational picture.

1. Introduction

In 2013 at the 18th ICCRTS, we presented the basic ideas how to command and control a multi robot system (MRS) using Battle Management Language (BML) [9]. In this follow-up paper, we present our progress with respect to reporting. In particular, the reports in question include the robot's reports about their equipment, e.g., the attached sensors, and as a consequence the robot's reports about their capabilities. As all communications between the MRS and its controller is expressed in BML, these reports also are formulated in BML. Based on the reports, we in addition present some use cases for which the planning is adjusted considering the reported equipment. Taking these uses cases as examples, we also show how plans can be reviewed by the user. Finally, we present some more reporting capabilities, again using BML, for which sensor data is compressed to information maps and interpolations in order to provide the controller a better operational picture good without overwhelming him with tons of detail information.

The use of BML as C2 standard is essential for our approach. BML allows expressing orders and reports on an abstract level. BML orders are sent to the MRS. The MRS has its own intelligence that determines the details about how to execute the orders taking the different capabilities of the robots into account. This corresponds to the interpretation of BML orders by a simulation system. As an effect the controller is freed from the burden to handle all the details of the robotic actions, like collision avoidance. BML reports are sent form the MRS to the controller's GUI. Like the orders, the reports also include a degree of abstraction in order to provide the controller with exactly the information he should be aware of suppressing irrelevant details. This again, is in analogy to the communication with simulation systems for which, for example, the rate of positional reports from the simulated units has to be dropped significantly [4].

The paper at hand is structured as follows. First, we will shortly recapitulate the basics of our approach (section 2). Then we will how robots report about their capabilities (section 3). The knowledge about the readiness of the robots and about their capabilities is the major input of the planning component that helps to take burden off the controller. This planning component is presented in section 4. Finally, in section 5, we will present measurements are reported back to the BML-GUI and how the respective results are presented to the controller (section 5). This leads to a general conclusion and outlook (section 6).

2. Assumptions and Preconditions

In order to command and control the MRS, the controller uses a specific BML-GUI which is presented in more details in section 2.3. The GUI supports the formulation of BML orders and displays the content of the robots' reports. It has its origins in the C2LG-GUI we originally developed for the NATO MSGs [3, 4, 6] to command and control simulated units.

2.1 Robot Operating System (ROS)

As we want to integrate different kinds of robots into the MRS, different operating systems, middleware, and communication protocols had to be dealt with. We decided to use the Robot Operating System (ROS) [7], developed and maintained by Willow Garage, as communication standard. ROS offers well-defined data types for most kinds of data. It is open source and hence free available, and it has a constantly growing community contributing to the software repository. The integration of ROS into the corresponding middleware is not in the focus of this paper, but see [5] for details on the integration topic. In order to connect the robots' ROS systems with the BML-GUI, a ROS component called BMLConnector had been developed in Python. Whenever the BMLConnector receives a BML order it transforms it into a defined ROS message, called BmlTask. The BmlTask message is then published as a ROS topic. For reports, the respective ROS messages are translated back to corresponding BML reports which then are sent to the BML-GUI so that for example the robots' positions can be displayed on the map of the BML-GUI. The robots also report about their operational state, the task status and detection of suspected enemy units. Further kinds of reports and their handling within our approach are presented in more detail below.

2.2 Battle Management Language (BML)

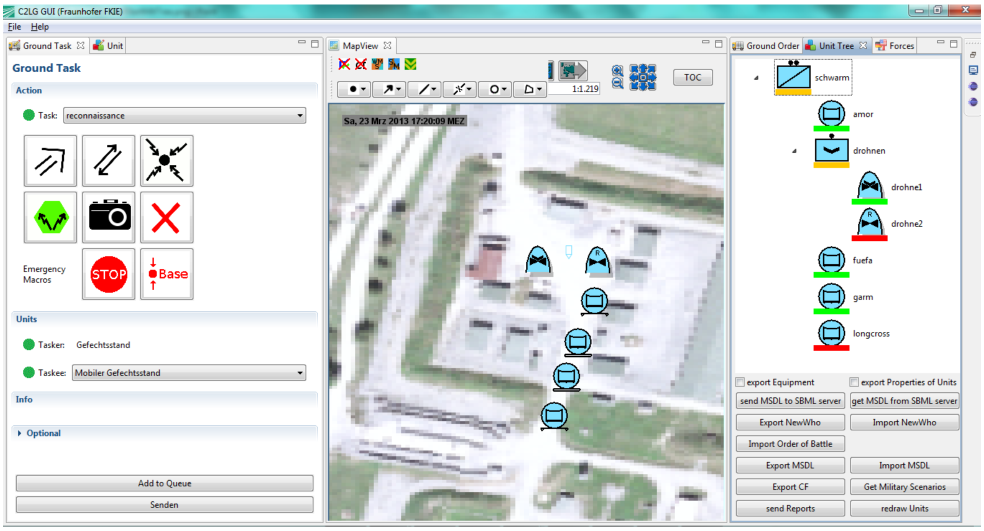
Battle Management Language (BML) is an artificial, unambiguous, human-readable language and open standard that has originally been defined to express and to exchange orders, reports and requests between Command and Control systems (C2 systems) on the one side and simulation systems on the other side. The idea to develop a language for message exchange from C2 systems to simulation systems and back has been discussed in the SISO since 2000, cf. [1] for details about the early SISO discussions. But BML became operational not until the work of NATO MSG-048 “Coalition Battle Management Language” [4] and its successor, NATO MSG-085 “Standardization for C2-Simulation Interoperation”.

Since BML must be unambiguous to allow automatic processing, the first step initiated by NATO MSG-048 was to define it as a formal language [10, 11]. On that basis XML schemata had been derived that serve as base for the BML message exchange. In meantime, the concept of BML and its application for the interaction between C2 systems and simulation systems had been validated in numerous experiments and demonstrations, among them a demonstration at Ft. Leavenworth in December 2013 that included five C2 systems and four simulation systems from six different nations.

It is self-evident that a language that can be used to command and control simulated units in a simulation system might be a valuable tool to command and control multi robot systems.

2.3 BML-GUI

In order to command and control the MRS, Fraunhofer FKIE had developed an interface for that purpose. Since the interface is used to command and control a MRS by BML orders and because that interface receives BML reports to be displayed for the controller, we will denote the interface BML-GUI in the following. The BML-GUI is based on the C2LG-GUI, FKIE uses in the demonstrations and experiments of NATO MSG-048 and NATO MSG-085. Here, the C2LG-GUI took and takes the role of a C2 system in those experiments on coupling C2 systems to simulation systems. In analogy, the BML-GUI took and takes the role of a C2 system in the demonstrations and experiments that include the MRS. The BML-GUI has been implemented on a standard workstation so that it can be used stationary in a control center, cf. figure 1 for a screenshot. It also has been implemented on a tablet (cf. figure 2) so that the MRS can be commanded by a mobile controller.



Task: reconnaissance

Tasker: Gefechtsstand

Taskee: Mobiler Gefechtsstand

Sa, 23 Mrz 2013 17:20:09 MEZ

Figure 1: BML-GUI, stationary version for control centers; left panel is for expressing orders (spatial aspects can be expressed by clicking the map); middle panel displays the map; right panel displays additional information, e.g., the icons of the robots and their status.

We focus on commanding the MRS. In order to express a valid order the user has to express a) the type of task that has to be executed, b) the part of the MRS that he wants to task (taskee), c) the spatial constraints of the task, and d) the temporal constraints of the task. The following tasks can be ordered: move, patrol, observe, recce, take a high solution picture, cancel, emergency stop, and emergency return to base. In order to simplify the expression of these tasks, the GUI has one button for each task marked with a respective icon, cf. figure 1 which shows the GUI. The tasks buttons are displayed in the left panel. In order to express the taskee, that is the part of the MRS that is supposed to execute the task, the GUI provides the user with the MRS's order of battle. This order of battle is built automatically at the start of an operation. The robots introduce themselves by communicating their current capabilities and their status. The capabilities and their status are used as input for the planning process. The general status of each robot from the MRS is presented on the GUI as colored bar below the symbol of respective robot (green for ready, red for not ready, and yellow if some robots are ready and some others are not in the case that the symbol represents more than one robot, e.g., the whole MRS). In the planning process, the status is also used to incorporate only those robots that are ready into the execution of a task. This makes the MRS and its planning tool tolerant to failure of single robots and also allows adding more robots to the MRS if necessary. The controller can chose among single ready robots or the whole MRS to determine the taskee. Of course, a robot's status may change during an operation, e.g. by running low on energy. In that case, the robot in question will return to the base. It might even become necessary to adjust a robot's sensor suit to the operation. In that case, after its sensors are exchanged (and the energy is refilled), the robot will introduce itself anew and its set of capabilities and its actual general readiness is actualized.

Figure 2: BML-GUI, mobile version

3. Reports on Capabilities

In [8] we described how the robots report back their general status, their position, the tasks progress and detected units. Since that time, we extended the set of report types in order to allow the robots to report about their capabilities and to report about measurements. We first will take a look at the reports on capabilities (section 3) and how that is used in planning the process of executing a given order (section 4). Finally, we will take a look at reports about measurements (section 5).

Reports about general status use the standard BML report format as given in (1a). An example is provided in (1b). Reporting about capabilities is modelled on reports about equipment as have been incorporated into standard BML, i.e., BML for message exchange among C2 systems and simulation systems, due to research by NATO MSG-085. Equipment reports use the format given in (2a). An example is provided in (2b). These equipment reports have been tested during the December 2013, BML experiments in Ft. Leavenworth, KS. The format for capability reports as used in our experiments on commanding MRS by BML is given in (3a) with a respective example in (3b).

(1a) [report] *own status-gen* ReporterIdentification Status-Value AtWhere When Certainty Label
(1b) [report] *own status-gen Longcross OPR at [50.123,7.123] at now RPTFCT report-169;*

In example (1b), the robot called *Longcross* reports to be ready (operational = *OPR*) while being at the given location. The report refers to the current point in time (*now*) and is reported as fact (*RPTFCT*).

(2a) [report] *own* whoRef *has* EquipmentIdentifier operational count When Certainty Label
(2b) [report] *own Coy_391_2 has Dingo operational 3 at now RPTFCT report-196;*

In example (2b), the second company of PzGrenBtl 391 (*Coy_391_2*) reports that it has three Dingos operational. The report again refers to the current point in time (*now*) and is reported as fact (*RPTFCT*).

(3a) [report] *own* whoRef *has* EquipmentIdentifier operational count When Certainty Label
(3b) [report] *own Robot_A has 3DScanner operational 1 at now RPTFCT report-196;*

In example (3b), the robot called *Robot_A* reports that it has one 3D Scanner operational. The report once more refers to the current point in time (*now*) and is reported as fact (*RPTFCT*).

4. Planning

The knowledge about which of the MRS's robots are ready and the knowledge about their capabilities allows for dynamic planning. Only those robots that are ready and that contribute to the execution of an ordered task are taken into account for that planning.

In order to clarify the point, we would like to discuss an example. Let us assume that the MRS consists of four ground robots and two UAVs. Let us further assume that two of the ground robots are wheel-driven and the other two use tracks. If this MRS is ordered to patrol a muddy path, the planning tool might send the two wheel-driven UGVs to the path's end points and divide the path into two sub paths so that one track-driven UGV and one UAV can be send to each subpath. Then, at each subpath, one GUV and one UAV can run or fly up and down. However, if for example, one of the UAVs is not ready because of low energy, the planning has to consider the other UAV to fly up and down the whole path. In order to make the example even more complex, let us assume that one of the track-driven UGVs has a chemical sensor and the other one has a nuclear sensor instead and the same is true for the UAVs. In this case it might be a good idea to assign the UGV with the chemical sensor and the UAV with the nuclear sensor to the same subpath, leaving the UGV with the nuclear sensor and the UAV with the chemical sensor for the other subpath, so that as result the whole path is check for chemical and nuclear abnormalities.

The planning tool should run on the MRS in order to free the controller of cumbersome detailed planning processes. However, it also should provide feedback about calculated plans to the controller so that he can interfere if needed. In our case, that feedback is implemented as a kind of BML report. In general, the planning tool receives the high-level BML order from the controller and starts the disaggregation program for the ordered task. The program generates low-level tasks with constrains. Those low-level tasks are sent back as suggested course of action to the GUI as a sequence of BML orders, cf. (4) for the sequence that refers to the patrol example discussed above. The user then can review the tasks and can manipulate them if needed.

(4)
move Planner HANNA to PatrolEndPoint1 start at now order-1;

move Planner AMOR to PatrolEndPoint2 start at now order-2;

move Planner Longcross to PatrolEndPoint1 start at now order-3;

move Planner Garm to PatrolMiddlePoint start at now order-4;

move Planner Psyche2 to PatrolEndPoint2 start at now order-5;

patrol Planner Longcross from PatrolEndPoint1 to PatrolMiddlePoint strend order-3 order-6;

patrol Planner Garm from PatrolMiddlePoint to PatrolEndPoint2 strend order-4 order-7;

patrol Planner Psyche2 from PatrolEndPoint2 to PatrolEndPoint1 strend order-5 order-8;

In example (4), the planner has divided the route to be patrolled into two parts (from PatrolEndPoin1 to PatrolMiddlePoint and form PatrolMiddlePoint to PatrolEndPoint2). The planner suggests sending the UGV HANNA and the UGV Longcross to PatrolEndPoint1, the UGV AMOR and the UAV Psyche2 to PatrolEndPoint2, and the UGV Garm to PatrolMiddlePoint. The UAV Psyche1 is not considered since it is not yet ready because of low energy. After arriving at their respective point, Longcross, Armor and Psyche2 are ordered to patrol along the respectively denoted part of the route. “strend order-3” means “start to execute this order after order-3 is finished”.

5. Reports on Measurements

In this section, we will take a look at reporting on measurements. This, in particular, includes reports about temperature and about the concentration of chemical gases or concentration of radioactivity. Measurements reports follow the form as given in (5a). In (5b) an example is provided. Reporting on measurements are not yet included in the BML standard, as used in NATO MSG-085 and will be published by SISO. It obviously, however, constitutes a useful addition to the standard that also can be used in C2 system to simulation system couplings if simulated units need to send respective reports.

(5a) [report] Phenomenon ReporterIdentification SensorIdentification MeasuredValue AtWhere When Certainty Label
(5b) [report] *Temperature Longcross Weather-Sensor0815 16.5degree at Hades ongoing at 20140131120000 RPTFCT report-256;*

In the example (5b), robot Longcross reports that the temperature at Hades is currently 16.5 degrees. This is reported as fact for the point in time “year = 2014, month = 01, day = 31, hour = 12, minute = 00, second = 00”.

Instead of showing the user all the measurement reports in detail we generated interpolations and display those as overlays over the BML-GUI map (which is in the central panel of the GUI, cf. figure 1). It’s possible to generate two kinds of overlays. One simplifies the interpolation into 4 kinds of areas (harmless, alarming, hazardous to health and deadly). This gives even an untrained controller a quick overview about the situation. The other interpolation has soft transition and gives for every point the measured value (e.g., the degree of gas concentration).

Figure 3 : simplified visualization

Figure 4: showing soft transition between measured points

6. Conclusion and Outlook

BML can be used to command and control a MRS. BML helps to reduce the burden of controlling since orders can be expressed on an abstract level (mission command [12]; for more details on mission command as formalized and established to the Prussian army by Helmuth Graf von Moltke cf. [2, 8]). In sum, a single controller can operate the MRS. Even more, the use of the BML-GUI as described above allows for a mobile controller since there is an implementation of the GUI on a tablet.

In future, we would like to elaborate on the planning tool that takes the BML order as expressed on the BML-GUI by the controller and transform it into a sequence of more explicit BML orders to be sent to the members of the MRS. Since these orders are also expressed in BML, the whole set can be sent back to the GUI so that the controller can check them and modify if needed. An elaborated planning tool will further increase the applicability of our approach.

References

[1] C. Blais, M.R. Hieb & K. Galvin, *Coalition Battle Management Language (C-BML) Study Group Report*. 2005 Fall Simulation Interoperability Workshop (05F-SIW-041), Orlando, FL, September 2005.

[2] T.N. Dupuy, *A Genius for War: The German Army and General Staff, 1807-1945*. Englewood Cliffs, NJ: Prentice-Hall, 1977.

[3] B. Gautreau, T. Remmersmann, D. Muniz, T. Serrano, H. Henderson, N. de Reus, L. Khimeche, J. Martinet, E. Pedersen, J. Lillesoe & D. Liberg, *Lessons Learned from NMSG-085 CIG Land Operation Demonstration*. 2013 Spring Simulation Interoperability Workshop (13S-SIW-031), San Diego, CA. April 2013.

[4] K. Heffner, N. de Reus, L. Khimeche, O.M. Mevassvik, M. Pullen, U. Schade, J. Simonsen & R. Gomez-Veiga, *NATO MSG-048 C-BML Final Report Summary*. 2010 Fall Simulation Interoperability Workshop (10F-SIW-039), Orlando, FL, September 2010.

[5] M. Langerwisch, M. Ax, S. Thamke, T. Remmersmann, A. Tiderko, & B. Wagner, *Realization of an Autonomous Team of Unmanned Ground and Aerial Vehicles*. International Conference on Intelligent Robotics and Applications (ICIRA 2012), Montreal, Canada, October 2012.

[6] M. Pullen, S. Levine, K. Heffner, L. Khimeche, U. Schade, M. Frey, N. de Reus, N. le Grand. P. de Krom, O.M. Mevassvik, A. Alstad, R. Gomez-Veiga, S. Galan Cubero & A. Brook, *Integrating National C2 and Simulation Systems for BML Experimentation*. 2010 Euro Simulation Interoperability Workshop (10E-SIW-008), Toronto, Canada, June 2010.

[7] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. B. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, *ROS: an open-source robot operating system*. International Conference on Robotics and Automation, 2009.

[8] V. Regling, *Grundzüge der Landkriegsführung zur Zeit des Absolutismus und im 19. Jahrhundert*. In Militärgeschichtliches Forschungsamt (Ed.), Deutsche Militärgeschichte 1648-1939. Hersching, Germany: Pawlak, 1983.

[9] T. Remmersmann, A. Tiderko & U. Schade, *Interacting with Multi-Robot Systems using BML*. 18th International Command and Control Research and Technology

Symposium (ICCRTS), Fairfax, VA, June 2013.

[10] U. Schade & M.R. Hieb, *Formalizing Battle Management Language: A Grammar for Specifying Orders*. 2006 Spring Simulation Interoperability Workshop (06S-SIW-068), Huntsville, AL, April 2006.

[11] U. Schade & M.R. Hieb, *Battle Management Language: A Grammar for Specifying Reports*. 2007 Spring Simulation Interoperability Workshop (07S-SIW-036), Norfolk, VA, April 2007.

[12] J. Storr, *A Command Philosophy for the Information Age – the continuing relevance for mission command*. In: D. Potts (Ed.), The Big Issue: Command and Control in the Information Age. Washington, DC: CCRP, 2003.