Integration of Autonomous UAVs into Multi-agent Simulation

Martin SELECKÝ [1], Tomáš MEISER[2]

[1]Dept. of Cybernetics, Czech Technical University, Technická 2, 166 27 Prague, Czech Republic
[2]Dept. of Computer Science, Czech Technical University, Technická 2, 166 27 Prague, Czech Republic

martin.selecky@agents.fel.cvut.cz, tomas.meiser@agents.fel.cvut.cz

Abstract. *In the past several years Unmanned Aerial Vehicles (UAVs) have gained a lot of attention in both the research field and in the field of commercial deployment. Recently researchers started to study problems and possibilities connected with the usage, deployment and operation of teams of multiple autonomous UAVs. These multi-UAV scenarios are from their nature well suited to be modelled and simulated as multi-agent systems.*

In this paper we present solutions to the problems that we had to deal with in the process of integration of two hardware UAVs into an existing multi-agent simulation system with additional virtual UAVs resulting in a mixed reality system where the hardware and virtual UAVs can co-exist, coordinate their flight and cooperate on common tasks. The hardware UAVs are capable of on-board planning and reasoning and can cooperate and coordinate their movement with one another as well as with the virtual ones.

Keywords

Unmanned Autonomous Vehicles, multi-agent systems, deployment.

1. Introduction

Nowadays the popularity of UAVs is growing thanks to the low cost of their deployment and maintenance and the possibility to operate them in areas inaccessible or dangerous for human pilots. To manage more sophisticated tasks such as area surveillance and monitoring or multiple target tracking, teams of multiple UAVs should be deployed which requires more complex control, coordination and cooperation mechanisms. These mechanisms have been already studied and developed for virtual UAVs as a part of multi-agent systems for example in AgentFly system [4] or MAS proposed by Baxter and Horn in [2] that would be very well suited to be applied to real hardware UAVs.

Further significant challenge in working with hardware air vehicles is that the design/test cycle is considerably longer than for ground robots or virtual entities. Flight experiments mean a greater level of risk, since even minor errors can lead to serious crashes. That is why mixed reality simulations, that allow integration of real hardware UAVs and simulated virtual ones to co-exist in one environment, are helpful and necessary in the development process.

In [3], [6] or [7] multi-agent system's principles have been used to issue commands to hardware UAVs. In these works, however, a central planning mechanisms are used and the UAVs only follow the precomputed plans.

Bürkle et al. [8] presented deployment of control mechanisms for teams of hardware VTOL micro UAVs. These UAV teams are capable of on-board planning and some cooperation on mutual tasks, however they do not coordinate their flight in terms of collision avoidance. The system also does not allow co-existence of hardware UAVs together with simulated ones.

We integrated two hardware fixed wing UAVs to the AgentFly multi-agent system [4] that is used for simulation of UAVs and air traffic, allows complex coordination and cooperation of agents and provides collision avoidance mechanisms. We modified the system to allow both hardware and virtual UAVs to act and interfere in a common mixed reality environment and we equipped the hardware UAVs with a Gumstix computer to enable on-board planning, reasoning and communication with other hardware or software UAVs.

This paper is organized as follows. First, we will briefly describe the AgentFly system in the section 2 and the UAVs and their computational and communication equipment in section 3. Section 4 shows the modifications done to the AgentFly system and hardware equipment for the UAVs deployment and in section 5 we describe the results of performed experiments. Section 6 concludes this paper.

2. AgentFly Multi-Agent System

AgentFly is a Java based multi-agent system designed for simulation of air traffic and UAV missions (see Figure 1). It is build on top of A-globe multi-agent platform [5] and it provides the simulated entities with communication frame-

work, trajectory planning and collision avoidance and cooperation mechanisms.

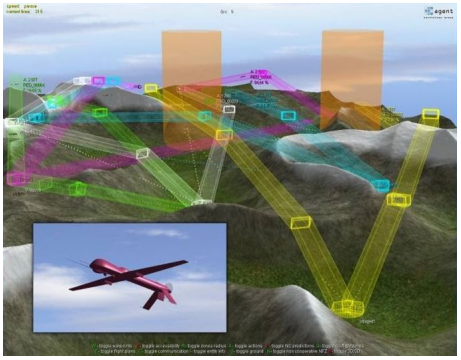


Fig. 1. AgentFly multi-agent system.

Each simulated aircraft in AgentFly system is composed of two software agents - *Pilot agent* and *Plane agent*. Pilot agent is responsible for the UAV's reasoning - it calls the trajectory planner and handles the communication with other UAVs for the purposes of cooperation and collision prevention. Plane agent provides the Pilot agent an interface for high-level plane control and executes the flight plans. It also simulates the aircraft's on-board instruments like radar, GPS or clock.

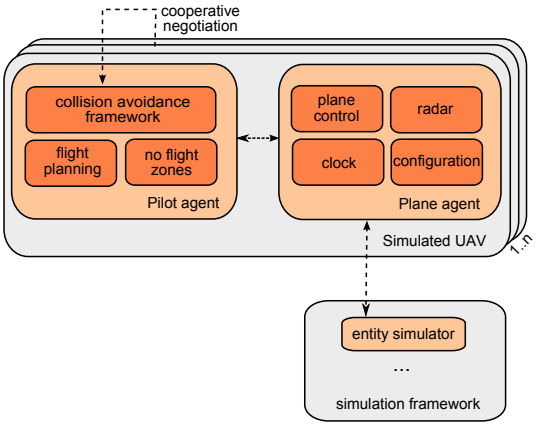


Fig. 2. Design of simulated virtual UAV.

System provides simulated aircrafts with mechanisms for peer-to-peer collision avoidance - either *cooperative* where the UAVs exchange their flight plans and negotiate about the avoidance manoeuvres, or *non-cooperative* where the UAVs cannot (for example because of incompatible protocols) or do not want (for example hostile airplanes) to exchange their plans and the collision is solved by flight trajectory prediction.

The flight plans consist of a sequence of flight manoeuvres - straight flight, left/right turn, up/down turn. They represent so called Dubins curves [1] that are based on a fact that any two positions (points with directions) in space can be connected by a sequence of a turn of a given radius, straight segment and another turn. Each manoeuvre is represented by a starting point, turn angle or acceleration, speed and duration.

During the transition process from software simulation to mixed-reality we replaced two of these virtual aircrafts with hardware commercial off-the-shelf fixed wing UAVs by Procerus Technologies.

3. Hardware UAV Platform

We selected the Procerus UAV (Figure 3) because of its relatively small cost, built-in cameras and sensors, ease of hardware integration and eventual repairing and included Kestrel autopilot. The Kestrel autopilot is capable of waypoint navigation thus removing the necessity of low level UAV control and it provides telemetry and GPS info about the aircraft via RF modem.

Fig. 3. Unicorn UAV by Procerus Technologies.

To provide a computational capacity for on-board planning and communication we equipped the UAV with Gumstix Overo Fire computer on module (COM) (Figure 4). This computer is based on ARM Cortex-A8 architecture with 512MB RAM running at 720MHz with Linux OS and Java Embedded to enable running of the modified AgentFly multi-agent system.

Fig. 4. On-board computer Gumstix Overo Fire.

Figure 5 shows the block design of the deployed hardware UAV. The Kestrel autopilot is connected to the sensors and actuators to provide low level UAV control. By connected 869 MHz Microhard modem the autopilot distributes telemetry and GPS info to the ground station and receives control commands in case when manual control is required or the autonomous on-board planning and control algorithms fail. The autopilot is also connected to the Gumstix Overo COM by RS-232 line. By this line it distributes the teleme-

try and GPS to the on-board planner and receives standard control commands - mainly sequence of waypoints in return. The communication with other hardware or virtual aircrafts is carried out by additional 2.4 GHz XBee modem.

Both modems have their counterparts on the ground station PC to pass the telemetry and communication to the simulation and vice versa.

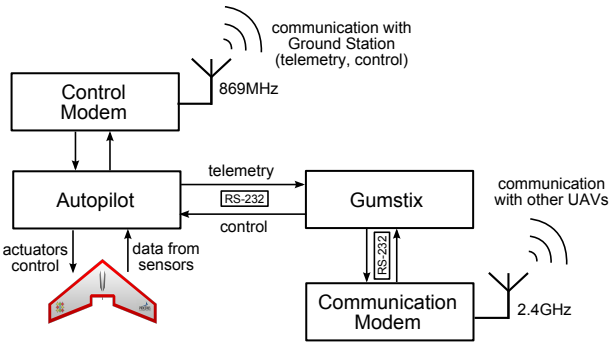


Fig. 5. Scheme of hardware UAV design.

4. Modifications made to the multi-agent system

The software integration of the hardware UAVs to the multi-agent system is depicted in Figure 6. It can be seen that the hardware UAV entity consists of an on-board part responsible for the planning, flight execution and other reasoning and a part located on the ground station PC responsible for visualization and exchange of position and telemetry (block in the figure denoted as "radar") information between the real and virtual entities.

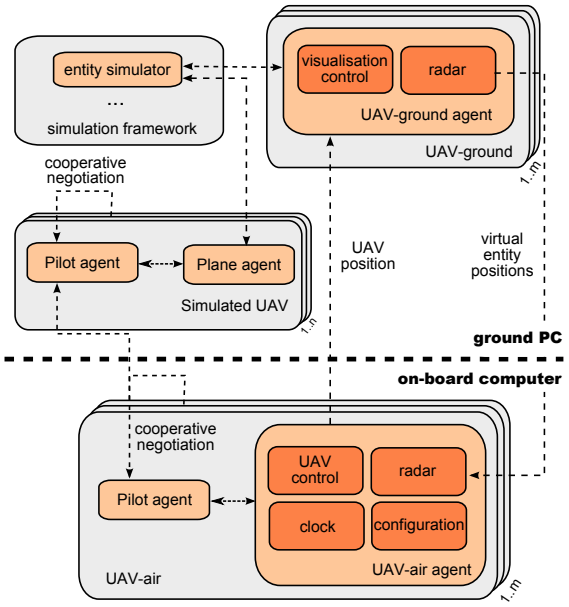


Fig. 6. Design of modificated MAS for mixed hardware and virtual UAV simulations.

For the integration of the hardware UAV, some parts of the simulation and planning software had to be modified for several reasons:

- **Waypoint navigation** of the Kestrel autopilot was not compatible with the AgentFly's manoeuvre-like flight plan structure.
- **Wind** in the real environment significantly influenced the plan execution precision.
- **Position uncertainty** caused by errors of sensors and actuators or environmental effects.
- **Unreliable communication** and low bandwidths of the RF modems caused problems in the collision avoidance and cooperative mechanisms.
- **Simulation time** needed to be synchronized otherwise it caused problems in cooperative collision avoidance negotiations.

In the following we will describe the necessary changes that had to applied to deal with these problems.

4.1. Waypoint Navigation

As said before, the Kestrel autopilot provides the operator with waypoint navigation capability. The waypoints are uploaded as GPS coordinates, thus we had to sample the UAV's plan which is represented as a list of flight manoeuvres. The manoeuvres are sampled according to the acceleration or turn angle - the more rapid change in speed or angle, the more dense is the sampling (see Figure 7).

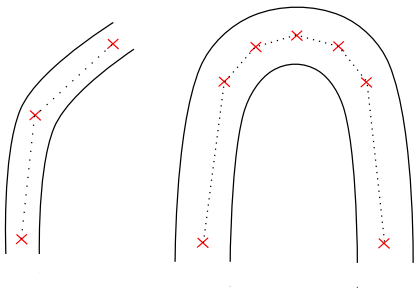


Fig. 7. Flight manoeuvres sampling. The larger is the turn angle the more dense is the sampling.

Another problem is that the planner works with Cartesian coordinates. The sample waypoints thus have to be transformed to GPS coordinates using linearisation of the world at the point specified by the position of the ground station. This position represents the origin of a Cartesian coordinate system with x-axis pointing to the east, y-axis to the north and z-axis up (see Figure 8). This linearisation causes errors that are less then 10cm in measurements of distances and less then 8m in the measurements of altitudes at distance 10km away from the origin and 500m higher.

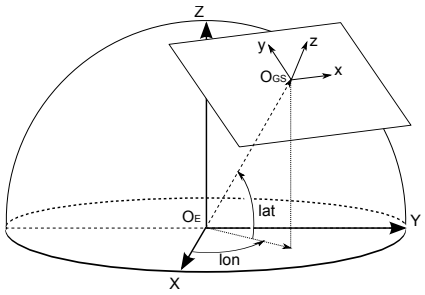
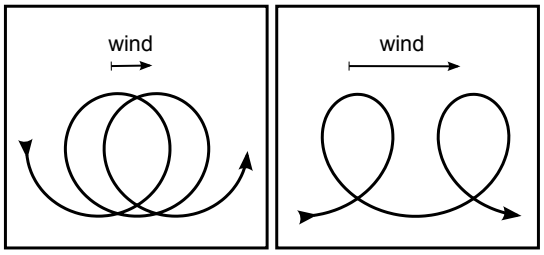


Fig. 8. Linearisation of the coordinate system.

4.2. Planning in Wind

The wind has a significant effect on UAV's plan execution precision. Apart from gusts of wind that drift the aircraft and cause position uncertainty discussed in the next section, the stable wind especially influences the minimal turn radius of the UAV - one of the main parameters of the trajectory planner. The aircraft flying against wind is capable of turning with much smaller turn radius than in case of flying in the wind direction. In these conditions the original planner that was based on planning with Dubins curves (i.e. straight segments and turns with constant radius) created trajectories that were impossible to be followed in presence of wind.

We modified the planner to use trochoidal curves. The trajectories then can be constructed as Dubins curves in a coordinate system that moves in the same direction and speed as the wind (see Figure 9) and can be followed much more precisely even in strong wind.



(a) $\frac{|\mathbf{wind}|}{|\mathbf{velocity}|} = 0.1$ (b) $\frac{|\mathbf{wind}|}{|\mathbf{velocity}|} = 0.3$

Fig. 9. Effect of wind on the flight trajectory.

4.3. Position Uncertainty

Because of the errors of the sensors and actuators and effects of the environment, such as gusts of wind, the aircraft's position estimation is not and cannot be precise. For effective UAVs coordination and control, this uncertainty must be modelled and the planning algorithms must take it into account.

We distinguish two types of errors - Δ_T for *time related errors* and Δ_D for *distance related errors* (see Figure 10). The distance related error is a deviation of the UAV from the planned spatial trajectory. It can be caused by sudden wind gusts, wind changes, high airspeeds or imprecise autopilot control when the aircraft is unable to follow the trajectory correctly. The time related error is then deviation of the UAV from the time plan. It is caused by imprecise autopilot velocity control, by accumulated small delays that emerge when the autopilot repairs small spatial deviations from plan, or by high wind speeds when it is difficult to keep desired aircraft's velocity.

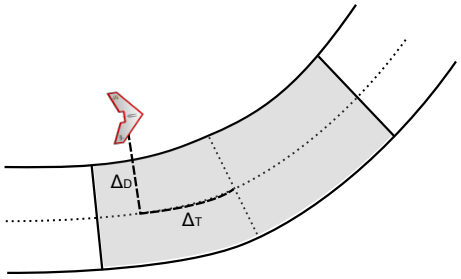


Fig. 10. Two possible error types in plan following- Δ_T for time related error and Δ_D for distance related error.

To handle these uncertainties we use so called *safety zone* around the UAV. Generally, the worse is the flight plan execution performance, the bigger the safety zone needs to be to keep the plane far apart from obstacles or other planes.

When specifying the dimensions of the safety zone we distinguish two different safety ranges - *safety time range* s_T and *safety distance range* s_D (see Figure 11). The safety time range is given by the maximal allowed time related error Δ_T and the safety distance range is given by the maximal allowed distance related error Δ_D. When the UAV gets out of any of these ranges, trajectory replanning is scheduled. The safety time range is bigger than the safety distance range because it is generally more difficult for the aircraft to keep up with the plan in the time domain.

The safety zone is used during the planning procedure so that it is wrapped along the planned trajectory and it cannot cross any obstacle, no-flight zone or other planned trajectory in time and space.

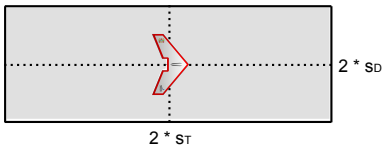


Fig. 11. Two different safety ranges - *safety time range* s_T and *safety distance range* s_D.

4.4. Unreliable communication

Communication is one of the most crucial features of any multi-agent system. In AgentFly system all the collision avoidance and cooperative negotiations such as plan and positions exchange and coordination commands are conducted by message exchange. In the simulation the messages are

transferred by reliable TCP/IP protocol but in the real environment the UAVs use RF modems with limited bandwidth, with possible interference of their signal with other RF devices and with significant signal attenuation with the distance.

The largest portion of the communication bandwidth is used by plans exchange during cooperative collision avoidance solving. Figure 13 shows required bandwidth for worst case collision avoidance negotiation in superconflict scenarios where all the UAVs are all flying against each other with one collision point in the middle of them (see Figure 12). We presume the initial distance of UAVs to be 1.5km with requirement to solve the conflict 10s before the collision point.

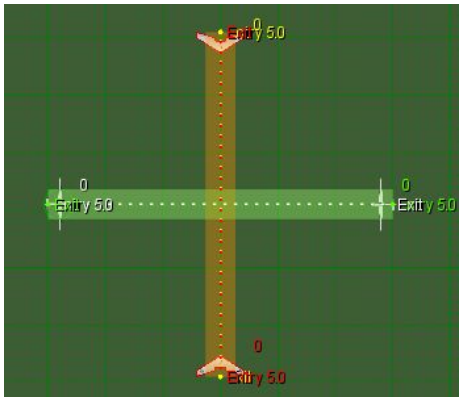


Fig. 12. Superconflict collision avoidance scenario.

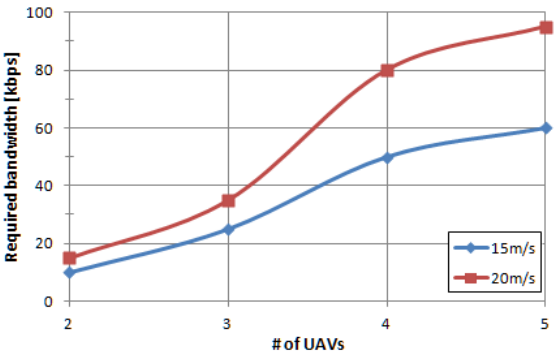


Fig. 13. Required bandwidth for collision avoidance negotiation in superconflict.

Apart from the plan exchange there is additional bandwidth required for manual safety control, telemetry broadcasts and communication management control. This needs 10-30kbps of additional bandwidth. Commercial modems that are capable of communication at a distance of at least 1.5km have maximal RF bandwidth around 115kbps which with the additional and safety bandwidth is not enough to handle even the 4 UAVs supeconflict scenarios. Moreover operating the modems at full speed requires significant portion of CPU time. That is why we decided to use two independent RF modems operated by on-board CPU units - Kesterl autopilot and Gumstix COM as was shown in Figure 5.

4.5. Simulation time

Collision avoidance algorithms as well as other cooperation and coordination mechanisms need to have synchronized time to work properly. There are two ways to synchronize the simulation times.

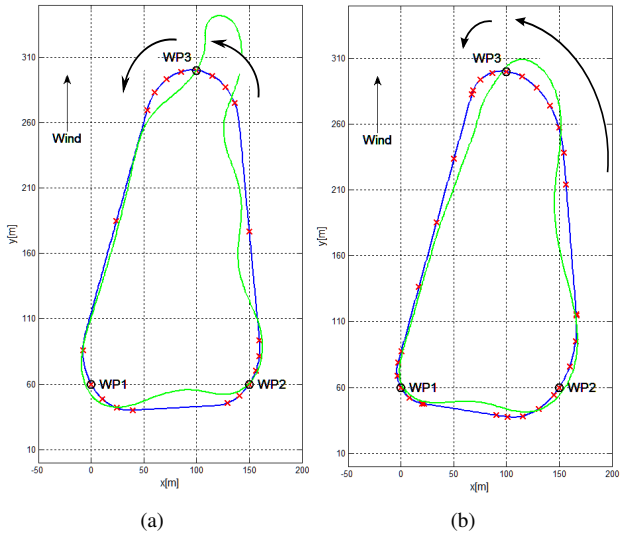
Because all autopilot modules as well as the ground station PC are connected to GPS modules, we can synchronize the clocks from the time information contained in GPS updates. This method can give very precise time information, however as we have found out, Kestrel autopilot sometimes provides wrong time information in GPS updates that need to be recognized and taken out from the measurements.

The other way is to pass the ground station's simulation time during a registration process of the hardware UAVs to the system and then start to measure time from that moment. In this method there is a problem in the information transfer time - it can take up to one second to transfer the registration data along with the simulation time which can have significant effect on the time synchronization. However we decided to use this approach because the synchronization error is smaller than the time errors from the GPS updates.

5. Experiments

To test and verify the above described modifications, we performed several field tests.

First, we compared the plan execution precision in cases with plans found by the original planning algorithm and with plans found by our modified algorithm for planning in wind. The tests were conducted in a presence of approximately 5 m/s wind, the UAV's airspeed was 15 m/s and the scenario consisted of three way-points placed as shown in Figure 14.



(a) (b)

Fig. 14. The effects of planning with (a) Dubins curves and (b) trochoidal curves.

The blue line in Figure 14 is the ideal planned trajectory that starts in the first way-point (WP1) then proceeds via WP2 and WP3 and ends again in WP1. The green line is the real recorded trajectory of the UAV and the black arrows emphasize the different turn radii in individual cases. It can be seen that the trajectories created with Dubins curves with constant minimal turn radius were impossible to be followed at some points. On the other hand the trajectories created with trochoidal curves that were adapted to the wind strength and direction were followed much more precisely.

Another experiment we performed was a mixed reality collision avoidance test. We prepared scenario with one hardware and one virtual UAV flying against each other. The purpose of this experiment was to test the functionality of collision avoidance mechanism between the real and virtual aircrafts as well as to test the sufficiency of the RF bandwidth required for the communication. The experiment was successful, the UAVs needed less then 10s to solve the collision situation which with airspeeds of 15 m/s correspond to 300m distance flown from the beginning of the negotiation.

In the future we would like to perform experiments with two hardware UAVs and later adding one and more virtual ones to study the scalability of the problem.

6. Conclusion

In this paper we presented the problems and solutions connected with the integration of hardware fixed wing UAVs into an existing multi-agent system for UAV simulation. Thanks to this we are now able to verify the functionality of the collision avoidance and cooperative mechanisms in a real environment as well as find their possible bottlenecks and limitation such as the maximum available RF bandwidths for communication or limited computational capacity of onboard computers.

Our work is a first step to a full deployment of AgentFly system on real hardware UAVs that could operate independently in coordinated teams during tactical missions.

In the future work we would like to extend the tactical cooperation possibilities of the UAVs and also deploy autonomous VTOL quadcopters next to the fixed wings and provide them with mechanisms of mutual flight coordination and cooperation.

Acknowledgements

The project described in the paper was supervised by Ing. M. Rollo, PhD., FEE CTU in Prague and supported by the Czech Ministry of Defence grant OVCVUT2010001.

References

[1] DUBINS, L.E. On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents. In *American Journal of Mathematics*, JSTOR, 1957, vol. 79, no. 3, p. 497 - 516.

[2] BAXTER, J.W., HORN, G.S. Controlling teams of uninhabited air vehicles. In *Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, ACM, 2005, p. 27 - 33.

[3] BAXTER, J.W., HORN, G.S., LEIVERS, D.P. Fly-by-agent: Controlling a pool of UAVs via a multi-agent system. In *Knowledge-Based Systems*, Elsevier, 2008, vol. 21, no. 3, p. 232 - 237.

[4] ŠIŠLÁK, D., VOLF, P., KOPŘIVA, Š., PĚCHOUČEK, M. AGENTFLY: A Multi-Agent Airspace Test-bed. In *Proceedings of 7th International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2008)*, May 2008.

[5] ŠIŠLÁK, D., REHÁK, M., PĚCHOUČEK A-globe: Multi-Agent Platform with Advanced Simulation and Visualization Support. In *Web Intelligence*, IEEE Computer Society, 2005, p. 805 - 806.

[6] BEARD, R.W., McLAIN, T.W., NELSON, D.B., KINGSTON, D., JOHANSON, D. Decentralized cooperative aerial surveillance using fixed-wing miniature UAVs. In *Proceedings of the IEEE*, 2006, vol. 94, no. 7, p. 1306 - 1324.

[7] SCERRI, P., VON GONTEN, T., FUDGE, G., OWENS, S., SYCARA, K. Transitioning multiagent technology to UAV applications. In *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems: industrial track*, International Foundation for Autonomous Agents and Multiagent Systems, 2008, p. 89 - 96.

[8] BÜRKLE, A., SEGOR, F., KOLLMANN, M. Towards autonomous micro uav swarms. In *Journal of intelligent & robotic systems*, Springer, 2011, p. 1 - 15.

About Authors...

Martin SELECKÝ

graduated from Faculty of Electrical Engineering Czech Technical University in Prague in January 2010. Currently he works as a researcher in the Agent Technology Center (ATG) and he is pursuing his PhD degree in Artificial Intelligence and Biocybernetics at Department of Cybernetics, FEE CTU. He studies methods of automated planning and path planning and works on a deployment of AgentFly project technologies on autonomous UAVs and light sport aircrafts.

Tomáš MEISER

holds the bachelor degree from Faculty of Electrical Engineering Czech Technical University in Prague from June 2009. Currently he studies for his master degree in Artificial Intelligence on FEE CTU in Prague. At same time he works as a researcher in the Agent Technology Center (ATG), where he studies capabilities of modern wireless communication platforms employed in dynamic scenarios.