Challenges in the Planning, Deployment, Maintenance and Operation of Large-Scale Networked Heterogeneous Cooperating Objects

Pedro José Marrón, Chia-Yen Shih, Richard Figura, Songwei Fu, Ramin Soleymani
Networked Embedded Systems Group
University of Duisburg-Essen
Bismarckstr. 90, 47057 Duisburg, Germany
{*pjmarron,chia-yen.shih,richard.figura,songwei.fu,ramin.soleymani*}@*uni-due.de*

Abstract—**Efficient deployment and management have been identified to be key challenges for the acceptance of solutions based on Cooperating Objects (COs). The operations for CO deployment and management can be classified into five phases in each of which several challenges and issues are emphasized. This paper presents the PLANET project whose goal is to tackle these challenges and to provide support for issues regarding large-scale CO deployment and operations. Moreover, two application scenarios, Wildlife Monitoring and Automated Airfield, are considered to demonstrate the capability of the PLANET solution.**

I. INTRODUCTION

Wireless sensor networks have been part of the research agenda for several years and have become since then part of the core enabling technologies that make smart cities and ubiquitous computing possible. The first applications of such systems were relatively simple and included the monitoring of the environment or of animals in their own habitat [1] [2]. Even in these first types of simple applications, the challenges faced by researchers were daunting at first and required the development of new techniques to deal with uncertainty and the real world [3].

With time, applications have become even more complex and are starting to include heterogeneous systems such as combinations of unmanned vehicles (aerial or terrestrial) and sensor networks. Furthermore, data is at least as complex as time series and no longer limited to simple scalars (like temperature and humidity). This makes writing generic and flexible solutions for these types of applications a big challenge. In general, we can say that current sensor network applications have different characteristics that require:

- The simulation of the environment in such a way that solutions can be designed and developed without having to go physically to the field, install the sensor network and collect feedback from it.
- Automatic interactions between the real deployment and the simulations performed in the lab, thus requiring a system to allow for the feedback from the real world to the simulation tools used to model the system in the first place.
- The capability to deal with complex data (not only scalar values like temperature or humidity). Complex data are in the most generic case time series that contains timestamped information about a complex signal, let it be audio, video or raw data (vibration, etc.) collected from the environment.
- The ability to deal with heterogeneous devices that interact with each other in such a way that they all cooperate to achieve a common goal.

Systems for wireless sensor networks that do not provide support for all these characteristics and requirements are bound to fail in the real world. In the best case, they will not be able to operate at the level required by the user. In the worst case, they will not be able to work in practice at all or provide faulty information that does not take into account all the aspects required by the application.

For these reasons, it is imperative that new applications and systems that support the development of these systems solve all of these issues satisfactorily because they can be applied to real-world environments.

The issues and requirements presented above are a subset of the challenges identified as part of the research roadmap written by the CONET consortium [4]. In it, and after the input not only from the consortium but also from a number of experts surveyed, it was possible to identify the most important research issues for cooperating objects. In the roadmap, Cooperating Objects are defined as follows:

> “Cooperating Objects (COs) consist of embedded computing devices equipped with communication as well as sensing or actuation capabilities that are able to cooperate and organize themselves autonomously into networks to achieve a common task. The vision of COs is to tackle the emerging complexity by cooperation and modularity. Towards this vision, the ability to communicate and interact with other objects and/or the environment is a major prerequisite. While in many cases cooperation is application specific, cooperation among heterogeneous devices can be supported by shared abstractions.” [4]

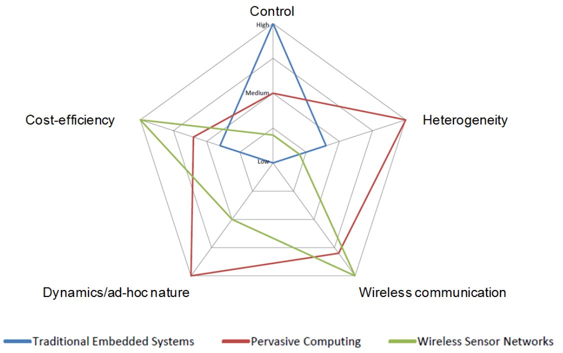

Figure 1. Key functional aspects in different system concepts

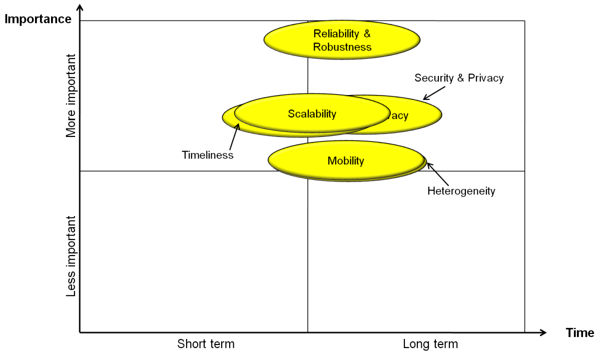

Figure 2. Matrix of Non-functional Properties area

Figure 1 shows a graphical representation of the field and the traditional areas incorporated into it. From the picture it seems clear that no solution that has been traditionally used in each area is able to cope with the complexities of the others. Therefore interdisciplinary solutions are the only ones that will be able to work in practice.

Regarding the research issues mentioned above, Figure 2 shows the estimation of the surveyed experts regarding importance and complexity of non-functional properties in the Cooperating Objects area. As can be seen, non-functional properties such as mobility, security, timeliness, etc. are considered, together with heterogeneity and deployment issues the most important ones to work on, but also some of the most challenging.

Among these non-functional properties, robustness sticks out being one of the properties that researchers and experts consider crucial for the acceptance of Cooperating Objects technologies. Robustness in a system like the one explained above can only be achieved if we have a system that is able to achieve the following goals:

- Is able to monitor itself and determine whether or not the network is behaving as expected.
- Is able to heal itself in case there are problems with the current status of the network or its performance is below the limit defined by the user.

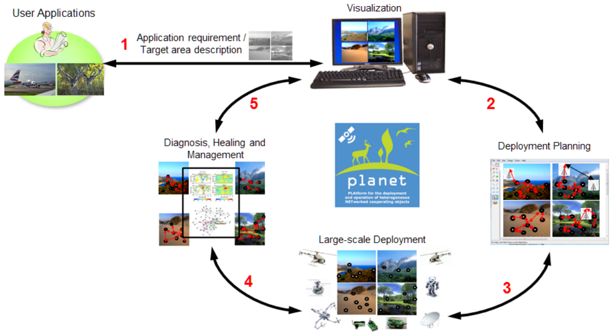

Figure 3. The PLANET approach. *Airfield photo: Phillip Capper*

- Is able to provide feedback to the developer or to the user in order to mitigate or even completely avoid future problems.

Therefore, the goal of the PLANET project [5] is to create an integrated platform that supports the efficient deployment, maintenance and operation of large-scale deployments of heterogeneous Cooperating Objects.

The remainder of the paper is organized as follows. Section II gives an insight on the solution proposed by PLANET as well as a description of the challenges that need to be solved in the complete implementation. Section III provides details on the experiments and use cases planned and, finally, section IV concludes the paper.

II. CHALLENGES OF THE PLANET SOLUTION

Figure 3 shows the overall philosophy of the PLANET project as well as the phases that an overarching solution should have. As can be seen in the picture, there are five phases needed to tackle the aforementioned challenges. Each one of these phases has its own set of challenges that we detailed in the following sections.

The complexity of the solution can be appreciated if we take into account that the solution to each one of the challenges needs to interoperate with the rest in order to provide a common platform and solution that works in different environments, as we will detail in section III.

Additionally, the solutions provided by PLANET have to take the following issues into account:

- The cooperating objects used to implement the application are highly heterogeneous with devices that range from a simple networked sensor to a complex Unmanned Aerial or Ground Vehicle.
- Security and safety are two of the most critical aspects of the solutions presented since the network has to operate with humans in the loop. This extends to the notion of intrusion detection for the cases where critical infrastructures such as airports are monitored by the PLANET system.
- Connectivity[6] and scalability present two difficult non-functional properties that the networks have to

integrate. This has to be achieved even in the presence of communication failures and/or in the case where the area that needs to be monitored is not physically reachable. For this particular case, data mules and the combination of the use of mobile and static nodes should be used.

A. User Input Processing Phase

To provide support for a user application, the PLANET framework assumes that the user should provide information about the deployment environment as well as application-specific systems used for the CO simulations in the form of models. In addition, the user should specify his requirements to PLANET with respect to all aspects including deployment plan generation, deployment operations, CO Control, application performance[7], network monitoring and maintenance. The PLANET framework processes these user inputs and uses them to configure different components of the framework for system operations in different phases.

In this first phase, the most important challenges to tackle are the following:

- The definition of a specification language that can be used by a user to describe the deployment environment. This implies not only the specification of the environment itself but also of the actors such as sensors, robotic platforms, etc. that operate in this environment.
- The development of tools that allow for the refinement of these models taking into account feedback from the system after its real deployment has been performed. This implies the development of incremental simulation environments that allow partial changes without the recomputation of the whole environment.

Therefore, PLANET aims at defining a specification language that allows the user to describe complex and a variety of configuration parameters including application-specific models, deployment and application performance metrics, pre-deployment simulation, network deployment, monitoring and recovery. Such specification language can help in efficiently configuring the PLANET framework components in the initial phase.

B. Pre-deployment Phase

The goal of the pre-deployment phase is to create a deployment plan that takes into account the current state of the environment. The plan contains information such as deployment positions and trajectories of the static and mobile Cooperating Objects to be installed in the target environment. Especially, PLANET addresses the capability of deploying COs using unmanned aerial and ground vehicles (UAVs and UGVs)[8]. Thus, the effect of dynamic factors (e.g., the wind, the speed of the deployment vehicles) needs to be considered when creating the deployment plan.

In this phase, the most important challenges to tackle are the following:

- The design and implementation of planning tools that are able to generate deployment plans for the deployment of static and mobile nodes using autonomous vehicles. These planning tools have the constraint that the created plans need to be feasible in practice. Such feasibility is difficult to achieve due to unavoidable imprecision of simulation models and the complexity of deployment optimization with heterogeneous COs. Therefore, the definition of deployment metrics that are used to evaluate the deployment plans is crucial for the plan generation. However, it is not trivial to define a deployment metric to evaluate the deployment plans created for various application requirements.
- The accurate estimation of the position of static nodes and the moving schedule of mobile nodes so that the entire area of interest is covered. Coverage can be defined as either static or dynamic, as determined by the case where a mobile CO patrols an area providing temporal sensory data during its movement. Additionally, different parts of the network might have complementary or even conflicting coverage requirements, which further increases the complexity of the coverage solution.
- The evaluation of the deployment plan using the simulation tools with user-defined performance metrics. It is challenging to design a simulation tool that precisely simulates the interactions between heterogeneous COs. Especially, the simulator needs to cope with different simulation models provided by the user. Other issues like scalability[9] and extensibility also need to be considered. An alternative is to integrate multiple simulation tools each of which is used to simulate a limited set of COs. However, in order to integrate different simulators, common input/output format of the simulators[10] needs to be defined. Moreover, time synchronization[11] between different simulator presents an extremely problem to deal with.

Therefore, PLANET aims at creating optimized deployment plans by developing the planning tool that provides optimized coverage solution. The usability of the deployment plan is further verified by running the application simulation with the deployment plan. Thus, one important feature of the planning and simulation tools is that they are capable to generate the deployment plan in the presence of potential errors and inaccuracies derived by the simulation models. Furthermore, given the fact that not all applications have the same requirements, the planning and simulation tools require that the user should be able to determine the deployment and performance metrics in order to generate optimized deployment plans.

C. Network Deployment Phase

In this phase, network deployment operations are launched to perform CO deployment in real-life following the deployment plan generated in the previous phase. Most importantly, the deployment operation can be performed by autonomous vehicles in PLANET. The challenges found in this phase are the following:

- The execution of the deployment procedure in such a way that a collection of autonomous UAVs and UGVs are able to carry COs and place them at the positions specified by the deployment plan. To accomplish a successful deployment, it is important to precisely coordinate and synchronize the operations between deployment vehicles. Such coordination[12] and synchronization are difficult to achieve due to the strict requirements on the efficient and reliable communication between on-duty deployment vehicles.
- The need to identify the actual deployment positions of COs[13][14]. Due to the dynamics of the environment and deployment vehicles, the COs may not be dropped at the specified positions. Therefore, acquiring the actual deployment position becomes necessary to ensure the coverage of the deployment. However, such localization information could be difficult to obtain depending on whether there is pre-existing infrastructure to support localizing COs. Without localization support, additional assistant objects, e.g., anchor nodes with GPS, need to be deployed first in order to provide position information about the deployed COs. Moreover, the precision of the localization techniques also presents an important issue to be tackled.
- In addition to deployment position, the deployment status such as connectivity and coverage[15] needs to be gathered to be able to automatically determine whether re-positioning of nodes is required. The position and deployment status reporting requires a data delivery path to the deployment control center. In the situation that there is no pre-installed network infrastructure, either the deployed nodes form an ad-hoc network, or additional gateway nodes need to be deployed in order to report the status information. In the case of a large-scale deployment, a hierarchical network is required to perform efficient data delivery. Moreover, issues including data aggregation and reliable data collection need to be considered to ensure the integrity of status report data.

Therefore, PLANET aims at providing CO deployment support by providing techniques for coordinating operations of deployment vehicles, localizing the deployed nodes, reporting actual node positions and deployment status. Moreover, the deployment operation in PLANET needs to be *adaptive*. That is, given the status information collected by the deployment vehicles, PLANET is able to determine whether node repositioning is required in order to achieve full coverage.

D. Deployment Debugging Phase

In this phase, the network of cooperating objects has been deployed and has been put into debugging mode[16]. In this mode, the application logic is performed, and application-specific data is collected to analyze the level deployment completion. The network is monitored and the health of the system is determined while the application is running. The following challenges play a crucial role:

- The design and implementation of the deployment analysis tool to ensure the CO deployment has met the user requirement. The performance metrics defined by the user are main elements used by the analysis tool to determine the success of the deployment. Additionally, complex deployment diagnosis algorithms are required in case that the deployment fails to reach the application performance requirement.
- The design and implementation of non-intrusive monitoring algorithms that enable the gathering of information regarding the health and performance of the network in order to validate the expected results as estimated by the simulation tool in the first phase.
- The capability to tune the network parameters in such a way that they continue within the performance expectation of the user, as defined in the first phase.

Therefore, PLANET aims for providing a deployment analysis tool that determines the completion of the CO deployment, and identifies the need for re-deployment based on the gathered monitoring information and the performance evaluation using the user-defined performance metrics.

E. Network Operation Phase

In this phase, the network has been taken out of debug mode and is able to operate in normal mode. After several iterations in the previous phases, the network reaches this phase because given the information, modeling and monitoring capabilities of the network, it is not possible to find a better solution than the one proposed. This does not mean that the network configuration is optimal, just that the used methodology is not able to determine a better solution given the aforementioned constraints. However, there are still a number of challenges that need to be dealt with in this phase:

- The non-intrusive monitoring [17] of a network in operational mode. The challenge is, in this phase, even more difficult than in the previous one since the monitoring overhead affects a running application. The goal is obviously to make measurements that affect as little as possible the normal operation of the network.
- The non-intrusive reporting of information and alarms, if needed, that will trigger another iteration from the network or the intervention from the user. This should happen even if the network only has limited information about its state and in a fast and accurate way, trying to avoid false positives as much as possible [18].

Figure 4. Doñana Biology Reserve. *Source: CSIC*

Figure 5. Highly Automated Airfield. *Sources: (left) Mario Roberto Duran Ortiz; (right) Phillip Capper*

- The automatic detection and suggestion of changes that need to take place in the network in order to repair it or improve its performance. In general, this could imply the following changes to the current deployment:
  - Changes to the location and position of static cooperating objects.
  - Changes to the routes of mobile objects because of the unexpected effects with certain obstacles in the network not foreseen in the pre-deployment phase.
  - Removing nodes that are misbehaving as seen by the performance evaluation and the metrics of the network.
  - Adding new nodes in certain areas in order to improve on a specific metric.

Therefore, PLANET aims for providing light-weighted and low-overhead monitoring solutions that efficiently and accurately detect network failures[19]. Moreover, failure recovery and network healing techniques are also expected to developed in order to maintain the continuous operations of the deployed CO network.

III. EXPERIMENTAL EVALUATION

As stated in the previous sections, the purpose of the PLANET project is not only to come up with solutions that work well in each one of the phases described. The ultimate goal is to show that our approach works in practice under different conditions and scenarios that differ fundamentally from each other.

For this reason, there are two main settings considered for the scenarios: The Doñana Biological Reserve (DBR) and an Automated Airfield Scenario (AIR). The former is a world heritage site located in the south of Spain that contains a variety of animals as well as four different types of terrains that make it very challenging for Cooperating Object applications. The latter is a fully automated airfield built also in the south of Spain for the purpose of testing UAVs and their interactions in a safe setting.

It seems obvious that both settings are distinct enough that providing solutions for one of them cannot be transferred without changes to the other. Naturally, the kind of applications (or use cases) that can be tested in each scenario is very wide and, in cooperation with biologists and aerospace engineers, we have identified the following use cases for Doñana (DBR) and the airfield scenario (AIR):

- **DBR1: Pollution monitoring**, where unmanned aerial and ground vehicles together with a pre-deployed sensor network will ensure the health of the water by detecting the presence of pollutants using different cross-validating techniques.
- **DBR2: Animal Monitoring and Tracking**, where a mobile network of sensors installed on different types of animals will determine their behavior as well as their relative positions to well-known beacons. Unmanned aerial and ground vehicles will be used as data mules if the spread of animals in a large area is so sparce that it is impossible to guarantee connectivity.
- **DBR3: Documentation of Animal Behavior**, where several unmanned aerial vehicles will be used to document the behavior of the Greylab goose using high definition cameras during the day and night time.
- **DBR4: Aerial Stratification of Insects**, where unmanned aerial vehicles will be used to sample insects at different altitudes and to correlate this information with that of sensors installed in bats. This will allow the improvement of the understanding of the key ecological interaction between bats and insects.
- **AIR1: Automated Mission Service Provision**, where a network of unmanned aerial vehicles are able to coordinate their missions by combining information about their own data and sensor information from the airfield using a pre-deployed sensor network that is assumed not to fail.
- **AIR2: Perimeter Security Service Provision**, where the infrastructure of the airfield, composed of sensors and unmanned ground vehicles that patrol the area, are able to detect an intruder in the perimeter and to act upon it.
- **AIR3: Sensor Healing Service**, where unmanned ground vehicles are used to heal the sensor network by carrying the appropriate sensors to the locations where they failed in order to re-establish connectivity, coverage, etc.
- **AIR4: Emergency Communication Service**, where unmanned aerial vehicles need to establish an ad-hoc network among themselves since their satellite connection to the control tower is lost. The unmanned

vehicles are supposed to relay data from the sensors on the ground to the other vehicles on the air using multi-hop communication.
- **AIR5: Emergency Airfield Tower Control Failure Service and Landing Aid Service**, where an unmanned ground vehicle will re-establish the connection to unmanned aerial vehicles using a mobile, low-overhead control tower that can be carried and moved to the appropriate position as needed in order to assist in the landing of vehicles that have lost some of their sensing and tracking capabilities.

IV. CONCLUSION AND FUTURE WORK

The challenges presented in this paper clearly show the necessity to integrate existing solutions for individual problems in such a way that the combined system exhibits an emergent behavior that cannot be achieved with the invidivuals solutions alone. Moreover, the additional constraints on the uncertainty of the environment as well as the capability of the PLANET platform to use knowledge from the real world to refine its internal model, will make it possible to apply it to the most heterogenous environments and in cases where more theoretical solutions will fail.

We, therefore, believe that only solutions that combine the capabilities of different disciplines into one integrated solution will be able to cope with the complexity of real-world applications and, in the long run, be successful in real deployments.

REFERENCES

[1] K. Martinez and R. Ong, “Glacsweb: a sensor network for hostile environments,” in *IEEE SECON*, 2004, pp. 81–87.

[2] J. Polastre, R. Szewczyk, A. Mainwaring, D. Culler, and J. Anderson, “Analysis of wireless sensor networks for habitat monitoring,” 2004.

[3] K. Langendoen, a. Baggio, and O. Visser, “Murphy loves potatoes: experiences from a pilot sensor network deployment in precision agriculture,” *Proceedings 20th IEEE International Parallel & Distributed Processing Symposium*, p. 8 pp., 2006.

[4] P. J. Marrón, S. Karnouskos, D. Minder, and A. Ollero, Eds., *The Emerging Domain of Cooperating Objects*. Berlin, Heidelberg: Springer, 2011, iSBN 978-3-642-16945-8.

[5] Planet. [Online]. Available: http://www.planet-ict.eu/

[6] A. Ghosh and S. Das, “Coverage and connectivity issues in wireless sensor networks: A survey,” *Pervasive and Mobile Computing*, vol. 4, no. 3, pp. 303–334, 2008.

[7] C.-F. Chiasserini and M. Garetto, “Modeling the performance of wireless sensor networks,” in *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, vol. 1, march 2004, pp. 4 vol. (xxxv+2866).

[8] L. Parker, B. Kannan, X. Fu, and Y. Tang, “Heterogeneous mobile sensor net deployment using robot herding and line-of-sight formations,” in *Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, vol. 3, oct. 2003, pp. 2488 – 2493 vol.3.

[9] E. Egea-Lopez, J. Vales-Alonso, A. Martinez-Sala, P. Pavon-Mario, and J. Garcia-Haro, “Simulation scalability issues in wireless sensor networks,” *Communications Magazine, IEEE*, vol. 44, no. 7, pp. 64 – 73, july 2006.

[10] T. Voigt, J. Eriksson, F. Österlind, R. Sauter, N. Aschenbruck, P. J. Marrón, V. Reynolds, L. Shu, O. Visser, A. Koubaa, and A. Köpke, “Towards comparable simulations of cooperating objects and wireless sensor networks,” in *Proceedings of the Fourth International ICST Conference on Performance Evaluation Methodologies and Tools*, ser. VALUETOOLS ’09, 2009, pp. 77:1–77:10.

[11] Z.-Y. Jin and R. Gupta, “Lazysync: A new synchronization scheme for distributed simulation of sensor networks,” in *Proceedings of the 5th IEEE International Conference on Distributed Computing in Sensor Systems*, ser. DCOSS ’09. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 103–116.

[12] M. Alighanbari, Y. Kuwata, and J. P. How, “Coordination and control of multiple uavs,” in *in The American Control Conference*, 2003, pp. 4–6.

[13] G. Mao, B. Fidan, and B. D. O. Anderson, “Wireless sensor network localization techniques,” *Comput. Netw.*, vol. 51, pp. 2529–2553, July 2007.

[14] L. Hu and D. Evans, “Localization for mobile sensor networks,” in *Proceedings of the 10th annual international conference on Mobile computing and networking*, ser. MobiCom ’04, 2004, pp. 45–57.

[15] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, and C. Gill, “Integrated coverage and connectivity configuration in wireless sensor networks,” in *Proceedings of the 1st international conference on Embedded networked sensor systems*, ser. SenSys ’03. New York, NY, USA: ACM, 2003, pp. 28–39. [Online]. Available: http://doi.acm.org/10.1145/958491.958496

[16] M. Ringwald and K. Roemer, “Monitoring and debugging of deployed sensor networks,” in *GI/ITG Workshop on Systemsoftware for Pervasive Computing*, October 2005.

[17] D. Minder, M. Handte, and P. Marron, “Tinyadapt: An adaptation framework for sensor networks,” in *Networked Sensing Systems (INSS), 2010 Seventh International Conference on*, june 2010, pp. 253 –256.

[18] S. Rost and H. Balakrishnan, “Memento: A health monitoring system for wireless sensor networks,” in *Sensor and Ad Hoc Communications and Networks, 2006. SECON ’06. 2006 3rd Annual IEEE Communications Society on*, vol. 2, sept. 2006, pp. 575 –584.

[19] J. Chen, S. Kher, and A. Somani, “Distributed fault detection of wireless sensor networks,” in *Proceedings of the 2006 workshop on Dependability issues in wireless ad hoc networks and sensor networks*, ser. DIWANS ’06. New York, NY, USA: ACM, 2006, pp. 65–72. [Online]. Available: http://doi.acm.org/10.1145/1160972.1160985