CONET Newsletter

INFSO-ICT-224053 www.cooperating-objects.eu **Issue nr.16, January 2012**

Contents

Editorial

Welcome to the sixteenth issue of the CONET newsletter. CONET is the EU FP7 network of excellence on Cooperating Objects, merging the fields of embedded systems for robotics and control, pervasive computing and wireless sensor networks. CONET focuses on establishing the field of Cooperating Objects within the research and industrial community, thus strengthening the position of Europe in the research landscape.

We'll take the opportunity to present some of the work being done within the CONET consortium, preceded by the member profile section presenting Schneider Electric Automation GmbH.

SPINE2 is a software framework specifically designed for visual and rapid development of distributed signal processing applications on different C-like WSN platforms. In the context of the REWSN cluster, SPINE2 is applied to support BSN applications in the e-Health area. However, on-going efforts are being devoted to enhance SPINE2 for more general WSN application domains.

The article "*Towards the envisioned cooperative ATM paradigm"* analyses the required cooperative integration among all future air traffic management actors and the expected improvements planned to achieve the foreseen goals in terms of efficiency, capacity balance and safety.

The last article describes the EU FP7 makeSense project that aims at providing a unified programming framework and a compilation chain that, from high-level business process specifications, generates code ready for deployment on WSN (wireless sensor networks) nodes.

If you are interested in obtaining up-to-date information about the CONET project please visit our website at: **http://www.cooperating-objects.eu**

We hope you will enjoy this issue ■

Member Profile: Schneider Electric Automation GmbH

Schneider Electric

http://www.schneider-electric.com

As a global specialist in energy management with operations in more than 100 countries, Schneider Electric offers integrated solutions to make energy safe, reliable, efficient, productive and green across multiple market segments. The Group has leadership positions in energy and infrastructure, industrial processes, building automation, and data centres/networks, as well as a broad presence in residential applications.

The Schneider Electric Automation GmbH with its main offices Marktheidenfeld and Seligenstadt is part of the Schneider Electric Group. The company develops and manufactures in its activities Machine Solutions and System Consistency within the Industry Business organization particular hardware and software products for automation solutions in machine and plant construction.

In the Industry market, the products and services solutions cover:

- Processes automation
- Machine control and monitoring
- Power supply & distribution
- Energy monitoring and control
- Utility management (lighting, ventilation, elevators, intruder alert, etc.)
- Single site or multi-site production data management
- Critical power

Main customers are engineering firms, systems integrators, OEMs, large industrial companies, panel builders and electrical equipment distributors, end users.

Schneider Electric has a strong participation and leadership in European R&D projects, including EU FP6 NoE IPROMS, EU FP6 STREP Inlife, EU FP6 STREP RI-MACS, EU FP6 SOCRADES, EU FP6 NoE CONET and EU FP7 IMC-AESOP.

In the context of CONET, Schneider Electric works on the application of service-oriented architectures and associated Web Service technology to the communications between cooperative objects. Schneider Electric has also substantial experience in the application of collaborative agent-based technology in industrial automation for implementing real-time decision-making mechanisms. This expertise will be instrumental in the development of test and simulation platforms, as well as in the blending of agent technology with Web Services ■

The SPINE2 framework

By G. Fortino, S. Galzarano, UNICAL and R. Giannantonio, Telecom Italia

Wireless Sensor Networks (WSNs) currently represent one of the most promising technologies for supporting the next pervasive and ubiquitous computing systems. They have been successfully exploited in many different application areas and in future they will play an increasingly important role. However, the development of applications for WSNs is an extremely challenging, error-prone and time consuming task, because of the lack of easiness in programming. It is therefore important to provide design methodologies and programming frameworks which enables rapid prototyping of WSN applications.

SPINE2 is a software framework specifically designed for the development of distributed signal processing applications on WSNs.

SPINE2 is an evolution of SPINE [1] and is implemented for reaching a very high platform independency by supporting C-like programmable sensor architectures. Moreover, while SPINE is centred on a programming model based on functions, SPINE2 offers a new programming abstraction based on a *task-oriented paradigm* in order to best fit the requirements of collaborative distributed data-processing applications in resource-constrained environments.

The task-oriented programming abstraction

The development methodology of SPINE2 aims at providing an abstract representation for the behaviour of distributed applications running on the sensor nodes, by omitting low-level details related to a specific platform and thus reducing the programming complexity.

With respect to the task-oriented approach, an application can be simply specified as a set of interconnected *tasks*. Each task represents a particular activity, such as a sensing operation, a processing function or a radio data transmission. A task connection represents a relationship between tasks that generally consists in a temporal and data dependency. In this way, the set of interconnected tasks forms a direct graph that explicitly defines the flow of computing operations representing the high-level description of the data-processing application.

Typically, a data-processing application (see Figure 1 for a simple distributed application involving only two nodes and not specified Processing tasks) supported by the framework consists in:

(i) accomplishing the needed sensor readings;

(ii) passing the sensed data to processing functions which carry out some signal processing operation;

(iii) sending results to other nodes of the network (i.e. the basestation), eventually for further data elaboration.

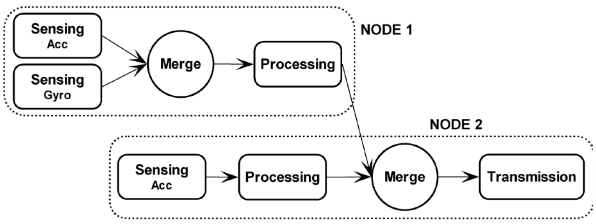

Figure 1: example of task-oriented application

Distributed and collaborative applications can then be programmed as a dynamically schedulable and reconfigurable set of tasks. Different tasks can be assigned to each node of the network and tasks can be controlled at execution time via proper message exchange.

Such a task-based programming abstraction, which capture both data- and control-flow mechanisms, allows for a better application definition that, in turn, will lead to a more clear and effective activities scheduling. Moreover, designing an application as a composition of elementary blocks enables rapid application reconfiguration and thus simpler application maintenance. Furthermore, by adopting such a task-oriented approach, the framework itself can be easily enhanced in functionalities, by simply adding new task definitions representing further computing capabilities.

The framework architecture

SPINE2 is composed of two components: one is implemented on the coordinator of the WSN, the other is implemented on the sensor nodes.

The coordinator-side part of the framework is a Java application running on a laptop (or a handheld device) through which the developer manages the sensor network and configures the task-based application by means of proper APIs. They allow to specify the set of tasks constituting the application, their configuration parameters, how they are connected together and on which node every single task has to be instantiate. Moreover, it gathers the processed data coming from the WSN application and eventually passes them to a higher-level user application for further data processing. To enable developers to easily configure and manage the task-based application, rather than directly using the SPINE2 APIs, an user-friendly tool is available (see Figure 2).

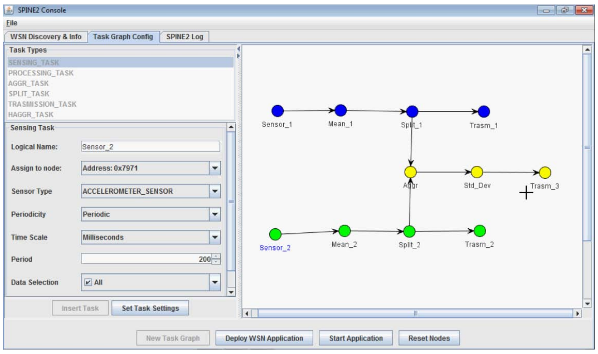

Figure 2: the SPINE2 GUI Console

The node-side part of the framework represents the middleware engine running on top of the sensor node operating system. It is responsible of handling the messages coming from the coordinator for configuring the portion of task-graph assigned to the node. Moreover, it implements the appropriate runtime system for interpreting and executing the task-based application. Its architecture is designed for supporting a high portability of the framework and, in particular, according to the *software-layering approach* (see Figure 3) for hiding the heterogeneity of different platforms, a basic software layer (or *core framework*), which provides basic functionalities, is defined for a set of heterogeneous platforms based on a similar programming language and adapted to each different platforms through platform-specific modules.

Specifically, the core framework provides the basic node execution logic (such as task management, application-level message handling and onboard sensor abstraction management) whereas the platform-specific modules are the components needed for accessing the services and the resources provided by a specific sensor node platform on which the software architecture is running. The core framework is implemented in C language, so that it can be easily ported to every C-like compatible sensor platforms, without the need for changing its implementing code. The platform-dependent components have to be specifically provided for each further new sensor platform and cannot be therefore reused for different hw/sw technologies. In practice, they represent adaptation components (or platform-specific drivers) that have the function of bridging the core with a particular operating system, through well-defined interfaces. Currently, SPINE2 supports TinyOS- and Z-Stack-based platforms.

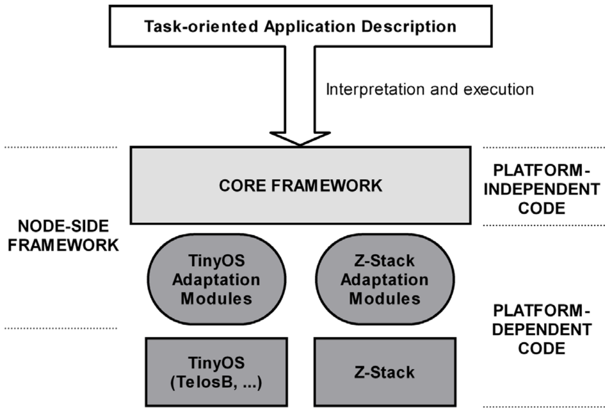

Figure 3: The software-layering approach for developing the node-side part of the framework

Conclusions and future research work

Thanks to the task-oriented programming abstraction and the sensor node software architecture, the SPINE2 framework has the following characteristics:

- *Platform independency and quick portability*: the framework can be rapidly ported for supporting different C-like sensor platforms, by simply redefining only the platform-dependent components (such as sensors and radio drivers) of the framework architecture.
- *Extensibility*: the task-oriented paradigm allows a straightforward addition of new functionalities, by defining new tasks representing further computing capabilities, without any change to the underlying run-time logic or to the other task definitions.
- *Modularity*: the framework architecture composed of several and independent functional modules allows for a more rapid and effective framework maintenance and improvement.

Any modification made by the framework developers affects only the corresponding modules without involving the rest of the architecture.

In conclusion, SPINE2 allows developers to specify their applications only once, independently from the actual platforms adopted, thanks to the fact that they are built through the defined task-based abstraction whose constructs are interpreted and "executed" by the same basic software component, the core framework. This approach enhances code reusability (also for heterogeneous WSN), and avoids the need for physically redeploying the node side code.

The SPINE2 framework has been used and is being using for developing several applications, currently mainly related to the WBSN (Wireless Body Sensor Network) domain, such as a gait analysis application based on a left-right Hidden Markov Model (HMM) [2], and a distributed action recognition system [3], which is based on the template matching classification approach. All these applications require a distributed processing of signals acquired by a set of body-worn sensor nodes.

Moreover, for better supporting more general distributed WSN applications involving many sensor nodes, the SPINE2 framework will be extended with the concept of "task group" defining an abstract set of tasks which are dynamically instantiated (and interconnected) at runtime on the basis of the dynamic changings of the WSN composition and topology.

The SPINE2 framework will be soon released as an open source project, similarly as it has been done with SPINE version 1.x (stay tuned on **spine.deis.unical.it**).

References

[1] Bellifemine F., Fortino G., Giannantonio R., Gravina R., Guerrieri A., Sgroi M., "SPINE: A domain-specific framework for rapid prototyping of WBSN applications" Software Practice and Experience, Wiley, 41(3), 2011, pp. 237-265, DOI:10.1002/spe.

[2] Raveendranathan N., Galzarano S., Loseu V., Gravina R., Giannantonio R., Sgroi M., Jafari R., Fortino G., "From Modeling to Implementation of Virtual Sensors in Body Sensor Networks". IEEE Sensors Journal, 2011, to appear. DOI:10.1109/JSEN.2011.2121059

[3] Panuccio P., Ghasemzadeh H., Fortino G., Jafari R., "Low-Power Action Recognition with Optimal Sensor Selection: An AdaBoost Driven Distributed Template Matching Approach". In *Proc. of the 1st Int'l Workshop on Mobile Systems, Applications, and Services for Healthcare (mHealthSys)* jointly held with ACM SenSys 2011, Seattle, USA, Nov, 2011 ■

Towards the envisioned co-operative ATM paradigm

By Enrique Casado and David Scarlatti, Boeing R&T Europe

Introduction

Nowadays the current Air Traffic Management (ATM) system is involved in a big transformation that will suppose a major change in the ATM paradigm. This modification is motivated by the high increase of traffic that will occur in the forthcoming years. The future system will be expected to handle at least 25 million flights a year in Europe of all types of vehicles, (fixed-wing aircraft or rotorcraft) and systems (manned or unmanned), which will be integrated into and interoperable with the overall air transport system with 24-hour efficient operation of airports [1].

One of the main features to be developed is the enhancement of the collaborative decision making process that will support the paradigm of Free Flight, ensuring safe operations within a more dense traffic flow. The Free Flight paradigm relies on the capability of flying the user's preferred trajectory as much as possible while safety is not jeopardized. In current operations, safety issues are managed directly from ground. However, in the future ATM system, the responsibility of maintaining safety will relay not only on the Air Traffic Controllers (ATCO) but also on the flight crew. To execute properly this shared responsibility, the system will provide a common infrastructure where all information will be accessible by any aircraft or ground-based systems in order to elaborate their own decisions in a co-operative and distributed manner.

This new ATM environment will enclose different systems that will cooperate among them to increase the efficiency of the global system without impacting safety requirements. They can be classified into ground-based systems (such as Arrival Managers, Departure Managers or Traffic Flow Managers) and on-board systems (such as Flight Management System). The main difference between both is that the former are fixed systems deployed in a specific location and the latter are mobile systems that fly within the airspace or taxi through the airport platform. The challenge to be addressed is the integration of heterogeneous mobile objects (aircraft) with different performances and equipments in a global system capable of supporting collaborative decision making. Thus, it is required to design and develop effective procedures to mange properly all the information coming from all these mobile objects. These procedures will allow accessing the relevant information according to the responsibility of each specific system, and the generation of a sequence of actions derived from the interpretation and post-processing of such data.

The main advantages of the application of cooperative procedures in the ATM field are:

- Common situational awareness. Both air traffic managers and airspace users share the same information which is widely distributed by the global information system.
- Better information accuracy, reliability and accessibility. All systems will have access in real time to the information shared by any other systems in the network.
- Optimization of airspace use. The whole system will be able to balance automatically capacity and demand, reducing the misuses of the airspace. This situation will be translated into operational improvements due to a better allocation of the resources.
- Reduction of ATC directives. With less ATC tactical interventions, it will be possible to executed trajectories very close to those preferred by users. This will imply optimum and more efficient flights, even when adjustments are required.
- Seamless integration of unmanned air vehicles (UAVs). Cooperatives systems sharing a common network for exchanging information will provide the infrastructure required for the integrations of the UAVs into the managed airspace.

Collaborative Decision Making

The concept of Free Flight is at the core of the future ATM system, where the responsibility of maintaining separation minima will rely on both air traffic controllers and flight crew. Closely related with this, Collaborative Decision Making (CDM) processes will support this future operational concept facilitating the access to the information to all stakeholders and distributing the responsibility of the actions among them considering different levels of hierarchy according to the actual conditions.

- The main objectives of CDM [2] could be summarized in:
- Provision to all cooperative actors, including airlines and air navigation service providers (ANSPs), with real-time access to all the information shared through the network infrastructure (weather updates, aircraft position, delays…).
- Improvement of the available information fusing data from different sources (aircraft, ground stations, operational centres …).
- Enabling accessibility to the right information at the right time to all involved actors (traffic managers, air traffic controllers, airspace users…).
- Establishing the procedures and mechanisms for allowing collaborative actions that improve the performances of the system, reducing at the same time any adverse impact to the airspace users.

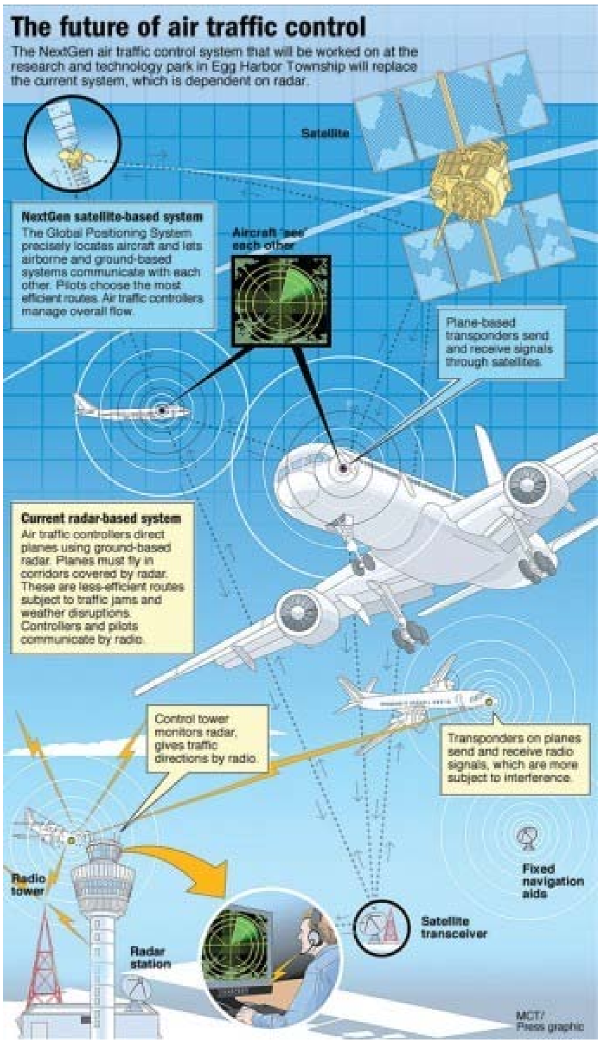

Figure 1: The Future of air traffic control (www.flightschoollist.com)

The final aim of CDM is to foster collaboration between airspace users and traffic managers to achieve a more efficient utilization of the airspace.

The new ATM paradigm [3] will deploy new process and procedures to support a full collaborative decision making (CDM) environment, where all interventions will be negotiated among all active players while safety conditions are preserved.

CDM will be enabled by different on-board and ground-based systems whose responsibilities are focused on sharing the critical information required for seamless operations in a robust and reliable manner. The technologies behind these concepts can be grouped in three main areas:

- Automatic Dependant Surveillance – Broadcast (ADS-B). The capabilities added by this technology will be a better determination of air-craft positioning and its dissemination through surroundings traffic for optimum operations.
- Airborne Separation Assurance System (ASAS). The shift of responsibilities form ATCO to the flight crew will be based on the capability of autonomous self separation. Each aircraft will ensure that minimum distance with other traffic is respected autonomously in all cases.
- System Wide Information Management (SWIM). The new features of the ATM system will be based on sharing the information among all stakeholders using a common infrastructure which will manage the flow of information according to the privileges of each of them and its criticism regarding the future procedures.

System Wide Information Management

The future ATM paradigm will represent a huge scenario of mobile (aircraft) and static (ground-based) cooperating objects. This new operational concept makes extensive use of a network infrastructure that establishes proper connectivity among all sensors and objects. This network infrastructure will support not only large scale (in fact worldwide) deployment but also highly heterogeneous systems. This heterogeneity can be characterized by four different properties:

- Mobility. Not all the systems connected to the information management system have the same location all the time. Some sensors are considered static (fix locations at ATC centres or airports), while others are dynamical (on-board systems).
- Security. The distributed information ranges from public (i.e. airport conditions) to reserved and secret data (aircraft position and intended trajectory).
- Data volume. The amount of shared information will depend on the characteristic of the generator and receiver systems, but usually it will vary from huge packages (e.g. complete weather forecast reports) to small packages (e.g. air traffic control queries).
- Safety. The system will support safety critical data such as aircraft separation distances or sense and avoid data, as well as, ordinary data such as flight plans or airline preferences.

The network infrastructure includes not only the physical layers to transmit the data (wires, satellites, or radio modems) and the related architecture, but also the upper layers including protocols and middleware technologies for ensuring a proper and robust connectivity of all participants.

One of the most relevant requirements to this network infrastructure is the interoperability. The global infrastructure has to be capable of providing accurate information with higher quality of service to any system included into the network. Not only it has to support all new on-board or on-ground systems designed and developed according to the purpose of being connected to the network, but also it must support all other systems, including legacy ones, which had been adapted to be part of the community. Although the final aim is to have all systems integrated into the same network to provide the same service to all users, there will be different deployment stages during which many diverse users equipped with a variety of capabilities have to interoperate seamlessly without affecting safety.

The System Wide Information Management aims at integrating the ATM network in the information sense, not just in the systems sense. The fundamental change of paradigm forms the basis for the migration from the former one-to-one message exchange concept to the many-to-many information distribution model. This architecture can be defined as geographically dispersed sources which collaboratively update the same piece of information, while geographically dispersed destination nodes require to gain situational awareness based on the changes applied to this piece of information [4].

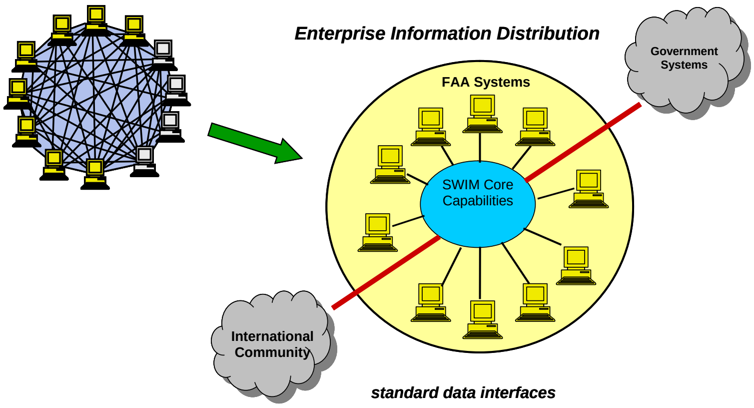

Figure 2: FAA vision of the SWIM concept

The main operational enabler for the implementation of this new operational concept is the capability of managing the quality, integrity and accessibility of shared ATM information. These data are named virtual information pool and represent a high complex and growing web distributed and fast changing system of data generation, edition and distribution [5].

Based on the assumption that SWIM will interconnect many cooperating objects, static or dynamic, in a distributed architecture, the expected benefits regarding the current point-to-point system are:

- Network is organized for achieving a common goal. The main target for deploying this infrastructure is to increase the efficiency of the global ATM system for an increasing traffic while safety is at least maintained or even improved when possible.
- Cooperative and modular approaches. The responsibility over the generation and edition of the data is partitioned according to the different needs and requirements of the systems which publish or subscribes the information.
- Better communication and interaction. The system will support the continuous update of the virtual information pool, and facilitate the ac-cess to these data according to the privileges assigned to each participating system.

Cooperative requirements in the future ATM system

The current ATM system is operating making use of procedures and infrastructures developed since decades. This system is no capable of assuming the increasing requirements coming from the users (airlines) which at the end respond to a social demand of faster and more reliable communications.

At the end of the transformation process of the ATM environment, the final goal is to establish a sound connectivity among all participating systems. Due this, many research areas in the domain of cooperating objects are suitable of being incorporated to the new developments that will support the enhanced system capabilities. In particular the following topics have special interest.

- Routing algorithms for networks with high speed mobile nodes, that defines the optimum path for the information packages from the publisher to the subscriber.
- Sense and Avoid cooperation algorithms. Due to the shift of responsibility form ground to on-board systems, the aircraft will require the capability of identifying surrounding traffic and recognizing unsafe circumstances. Once this kind of situation comes up, the aircraft will need to have the ability for solving it autonomously.
- Robust secure encryption with limited computational resources. The performance of the network is directly linked to the computational requirements and the encryption of messages is a must for a proper information management. Techniques for optimizing the use of the resources and even more, the distribution of the effort in critical situation will add value to the general aim of the network.
- Distributed Negotiation. The new ATM paradigm changes the negotiation process distributing the responsibility of each decision, instead of maintaining it in a unique role (usually ATC). For implementing this new functionality is required to define the process and procedures for a distributed negotiation which will be adapted according to the timeframe established for a final resolution.
- Sensor fusion. The network will interconnect disparate and heterogeneous sensors working together. The challenge is not to overload the ATCO, the flight crew or any system with all the data managed by the network but to provide with the right information at the right time according to the needs of the user.

- Scalability. The system has to be capable of integrate an increasing number of system maintaining the levels of performance and without interfering the safety of operations. Thus, scalable mechanisms for adapting the network according to the actual number of connected players have to be considered as part of the development.

Cooperative procedures and collaborative decision making are the cornerstones of the envisioned ATM system. All actors and stakeholder will be aware about any change in the network and any relevant decision will be assumed considering the impact on every single element (or at least on the surrounding cooperative agents). Thus, cooperation as sharing information to facilitate surrounding operations, and collaboration to take the right decision at the right time represent the mayor advances in the ATM field to cope with the envisaged growth of the air traffic.

References

[1] European Commission, Directorate-General for Research and Innovation and Directorate General for Mobility and Transport, Flightpath 2050 Europe's Vision for Aviation, March 2011

[2] Michael O. Ball, Robert L. Hoffman, Dave Knorr, James Wetherly; Assesing the benefits of Collaborative Decision Making in Air Traffic Management, 3rd USA/EUROPE ATM R&D Seminar; Napoli, April 2000

[3] L.Guichard, S.Guibert, H.Hering, J.Nobel, D.Dohy, JY.Grau, K.Belahcène, (2004) Paradigm SHIFT Operational Concept Document V1.0 EEC Note No 01/2005

[4] ICAO DOC 9854 Global Air Traffic Management Operational Concept.

[5] Strategic Research Agenda, Advisory Council for Aeronautics Research in Europe (ACARE), 2004 ■

makeSense: Easy Programming of Integrated Wireless Sensor Networks

By the makeSense Team (Swedish Institute of Computer Science, University of Lübeck, University of Trento, SAP AG, ACCIONA Infraestructures S.A.)

The industrial adoption of wireless sensor networks (WSNs) is hampered by two main factors. First, there is a lack of integration of WSNs with business process back-ends. Current approaches typically consider the WSN as a stand-alone system, leaving the integration with the back-end infrastructures to application developers. This requires considerable effort and expertise.

Second, programming WSNs is still challenging as it is mainly performed at the operating system level. The many existing WSN programming abstractions are hard to use since they typically focus on one specific problem.

To address these challenges, the EU FP7 makeSense project aims at providing a unified programming framework and a compilation chain that, from high-level business process specifications, generates code ready for deployment on WSN nodes.

Scenarios and Goals

A paradigmatic example of our target scenarios is ventilation in buildings, an application case where project partner ACCIONA provides significant domain expertise. Fans are commonly operated at a fixed rate, independent of room occupation, resulting in unnecessary ventilation of unoccupied rooms and over-ventilation of sparsely occupied ones, ultimately wasting energy. A smarter strategy may consider room occupation, resulting in sustainable building management, as shown in Figure 1.

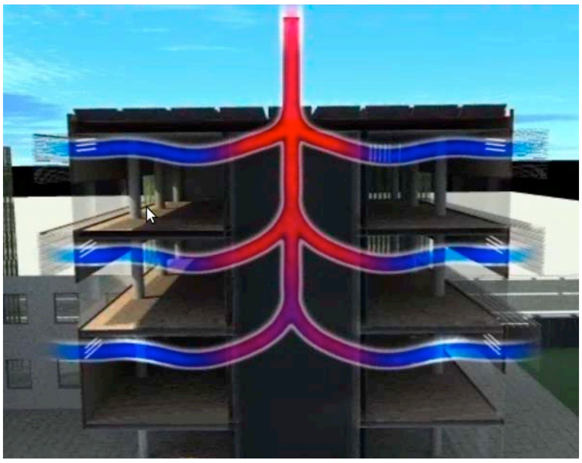

Figure 1: Building ventilation.

Consider an office environment, where employees book meeting rooms on the Web through a back-end process notifying the expected participants. Room ventilation is minimal when no meeting is scheduled. Sensors and actuators driven by the business process increase ventilation before the meeting and until human presence is detected or CO_2 levels are above threshold.

Realizing this system requires a tight integration between the business process and the network of sensors and actuators immersed in the environment, as the application logic needs to extend to the latter. Moreover, implementing the processing for adaptive ventilation complicates application development, as it departs from the traditional data collection — most common in WSN applications — to encompass possibly distributed control loops.

Numerous application domains share similar requirements. To address them, the design of the makeSense framework revolves around three fundamental goals:

- we must *seamlessly integrate* with existing business process technology, providing an adoption path that complements, instead of disrupting, existing methodologies and technologies with WSN ones;
- we must be *modular* and *extensible*. As we aim for our system to be useful across several real-world applications, extensibility is key to ensure that the programming abstractions and their implementation can be easily adapted to the specificity of the target domain as well as to unforeseen needs;
- we must *self-optimize* w.r.t. high-level performance goals. This ability to self-adapt is necessary to support long-lasting, operational business processes immersed in the physical environment and subject to the vagaries of wireless communication.

Approach

Our architecture is based on the separation of concerns provided by a distinction in layers of functionality: i) an *application* layer concerned with business processes and their modelling; ii) a *macroprogramming* layer concerned with the distributed execution of activities within the WSN; and a *run-time* layer concerned with the low-level aspects supporting the above and enabling self-optimization.

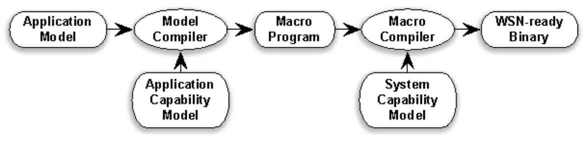

Figure 2: Compiling business process models to WSN-executable code.

A model-driven approach connects the three layers, as shown in Figure 2. The application model represents a holistic, network-agnostic view of the entire business process, i.e., including the WSN and the process back-end. At this level, we use and extend the Business Process Modelling Notation (BPMN), as shown in Figure 3 for the aforementioned ventilation scenario. By introducing new attributes, the modeller can specify a new *intra-WSN participant*, containing the logic executed by the WSN. As the latter is resource-constrained, we allow only a subset of BPMN elements to be used there. Furthermore, we introduced a new *WSN activity* type.

As WSNs are inherently distributed systems, we also introduced a *Target* attribute for lanes and activities within the intra-WSN participant, that allows specifying where the respective logic should be executed, based on labels that are relevant at the modelling layer. Further notations are used to specify performance requirements (e.g., a certain level of reliability, or a minimum lifetime).

Two compilation steps achieve the semantic link among layers. The modelcompiler takes as input the application model and an application capabilitymodel. The latter is a coarse-grained description of the WSN, providing information such as the type of sensors/actuators available and their operations. The model compiler translates these descriptions into a program written in a custom macro-programming language, serving as an intermediate language closer to the reality of WSN systems, yet high-level enough to be potentially used directly by a developer.

Here, our intent is to provide a framework where the abstractions contributing to the language are decoupled, can leverage on existing implementations, and can be changed or extended easily to suit specific application needs. This design goal biased the design of the entire language. To properly identify the units of functionality, reuse, and extensions we defined a notion of *meta-abstraction*, implemented through different "concrete" abstractions. Abstractions provide the key concepts enabling interaction with the WSN according to common patterns, e.g., many-to-one and one-to-many communication. Their composition can be achieved by using common control flow statements, provided by a core language that serves as the ``glue'' among macroprogramming abstractions. The core language in our case a stripped-down version of Java tailored for WSNs

The macro compiler takes as input the macro-program generated by the model compiler and a system capability model. The latter provides finer-grained information on the deployment environment (e.g., how many sensors of a given type are deployed at a location). The macro-compiler generates executable code that relies only on the basic functionality provided by the run-time support available on the target nodes. By leveraging

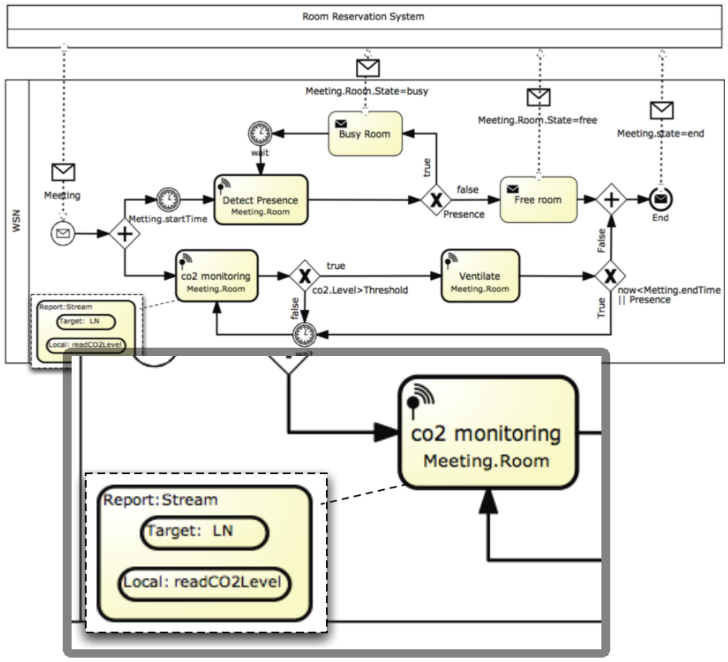

Figure 3: BPMN model for a fragment of the ventilation scenario.

the system capability model, the macro compiler can generate different code for different nodes, based on their application role. The executable code contains the mechanisms enabling self-optimization.

This functionality adapts the system configuration to changing requirements based on the developer-provided performance goals. In the ventilation scenario of, high data reliability required to accurately monitor the people's presence will correspond to different protocol settings compared to situations with no on-going meetings, when energy preservation is the major performance concern.

To enable self-optimization, we gather run-time information from the WSN (e.g., network topology and protocol performance) and feed it to a reinforcement-learning algorithm that explores the space of possible protocol configurations in simulation. Based on the corresponding performance, we derive self-optimization policies that determine how to set protocol parameters depending on the current application performance state, and distribute them back to the deployed network.

Status and Outlook

We implemented the extended BPMN meta model in Signavio Core Components, an open source, browser-based BPMN editor. Our prototype implementation is focused on the model-to-macrocode transformation. Future work includes extending a BPMN runtime with the lightweight messaging protocol in JSON. The macro-compiler prototype is implemented as a multi-pass compiler employing the ANTLR parser generator. Currently, the compiler is primarily optimized for maintainability and extensibility. We also implemented a basic set of concrete abstractions for the macroprogramming language. The self-optimization functionality is currently a separate stand-alone prototype written in Java, soon to be integrated into the makeSense framework. The prototypes will be released as open source.

In this article, we presented the ideas and early results underlying the makeSense project, tackling the problems of unification of previously existing stand-alone programming abstractions for WSN, as well as integration of WSN with business back-ends. As these two problems are currently hampering WSN adoption in industry, we believe that makeSense will have a profound impact on the use of WSN in this domain.

ⓘ **http://www.project-makesense.eu** ■

Announcements

CONET 2012: The Third International Workshop on Networks of Cooperating Objects

April 16th, 2012, Beijing, China

ⓘ **http://conet2012.cister-isep.info/**

Paper submission deadline: January 30, 2012

DPMSS 2012: 1st International Workshop on Data-intensive Process Management in Large-Scale Sensor Systems (DPMSS): From Sensor Networks to Sensor Clouds

May 13-16, 2012, Ottawa, Canada

ⓘ **http://dpmss2012.deis.unical.it**

Paper submission deadline: January 31, 2012

ESARS 2012: 2nd International Conference on Electrical Systems for Aircraft, Railway and Ship Propulsion

16-18 October 2012, Bologna, Italy

ⓘ **http://www.esars.org**

Provisional full papers deadline: March 15, 2012

EWSN '12: The 9th European Conference on Wireless Sensor Networks

February 15-17, 2012, University of Trento, Italy

ⓘ **http://ewsn12.disi.unitn.it**

MobiPST 2012: The Second International Workshop on Privacy, Security and Trust in Mobile and Wireless Systems

July 30 to August 2, 2012, München, Germany

ⓘ **http://ocu-stars.okcu.edu/ksha/mobipst2012.html**

Paper submission deadline: March 9, 2012

NGCUV 2012: IFAC Workshop on Navigation, Guidance and Control of Underwater Vehicles

April 10-12, 2012, Porto, Portugal

ⓘ **http://www.fe.up.pt/~ngcuv2012**

Paper submission deadline: January 15, 2012

RoboSense 2012: The International Workshop on Cooperative Robots and Sensor Networks

August 27-30, 2012, Niagara Falls, Ontario, Canada

ⓘ **http://www.coins-lab.org/events/RoboSense12/**

Paper submission deadline: February 20, 2012

A special edition of a Springer Book “Cooperative Robots and Sensor Networks” will include selected best papers of this Workshop

SESENA 2012: 3rd International Workshop on Software Engineering for Sensor Network Applications

June 2, 2012, Zurich, Switzerland

ⓘ **http://www.sesena.info**

Paper submission deadline: 17 February 2012

WFCS 2012: 9th IEEE International Workshop on Factory Communication Systems COMMUNICATION in AUTOMATION

May 21 - 24, 2012, Lemgo/Detmold, Germany

ⓘ **http://www.init-owl.de/wfcs2012/**

Work-in-progress paper submission deadline: February 10, 2012

Latest News

BonnMotion 2.0 released!

Some of the new features:

- 3D support
- New models
- SLAW
- SteadyStateRandomWaypoint
- RandomWaypoint3D
- and others
- Export of DisasterArea movements to standard NS-2/NS-3
- Import of GPX traces
- Code validation tool

More information (including changelog and source code) can be found at:

ⓘ **http://bonnmotion.net.cs.uni-bonn.de/**

EWSN '12: The 9th European Conference on Wireless Sensor Networks

February 15-17, 2012, University of Trento, Italy

ⓘ **http://ewsn12.disi.unitn.it**

Register @ http://www.cooperating-objects.eu to receive future issues of the CONET Newsletter