

Invited Poster:

makeSense: Easy Programming of Integrated Wireless Sensor Networks

Fabio Casati^{*}, Florian Daniel^{*}, Adam Dunkels[†], Stamatis Karnouskos⁺, Patricio Moreno Montero^{*}, Luca Mottola[‡], Felix Jonathan Oppermann[‡], Gian Pietro Picco^{*}, Kay Römer[‡], Patrik Spieß⁺, Stefano Tranquillini^{*}, Paolo Valleri^{*}, Thiemo Voigt[†]
[†]Swedish Institute of Computer Science, [‡]University of Lübeck (Germany),
^{*}ACCIONA Infraestructuras (Spain), ⁺SAP Research (Germany), ^{*}University of Trento (Italy)
Contact e-mail: thiemo@sics.se

Abstract— WSNs are expected to play a critical role in the next computing revolution, as depicted in the visions of Cooperating Objects and the Internet of Things. However, designing and developing WSN software is currently very difficult. This may prevent WSNs from reaching large-scale adoption, especially in industry. The *makeSense* project aims at enabling an easier integration of WSNs in business processes, by allowing business process experts and WSN developers to express the high-level functionality required, while leaving low-level details to the compiler and run-time system. We envision the results of *makeSense* to be not only a landmark for WSN software development, but also a new way to look at WSN programming that increases productivity and business value, enabling a far-reaching adoption in key industrial domains.

I. INTRODUCTION

Wireless Sensor Networks (WSNs) are a key component towards the integration of the physical and virtual worlds, as depicted in the visions of Cooperating Objects and Internet of Things. However, their widespread adoption in industry is still limited. Factors that hinder a more widespread adoption are the difficulty of WSN programming, as acknowledged in the CONET research roadmap [2], and the limited support for integration with existing IT infrastructures. In particular, although several programming abstractions are available in the literature [3], almost none of them explicitly supports the integration of WSNs with business processes.

The EU-funded *makeSense* project enables such integration by devising programming abstractions to express the high-level WSN functionality within existing business process modeling concepts. This allows for seamless specification of the behavior of the WSN and the surrounding business process. Low-level details are then left to a dedicated compiler and run-time system. The name, *makeSense*, reflects both purpose and ambitions of the project. The first part of the name, *make*, refers to the *make* tool, the software development utility that relieves developers of software development details.

Section II of this paper illustrates the *makeSense* approach and the overall architecture. Section III elaborates on the expected results of the project and concludes.

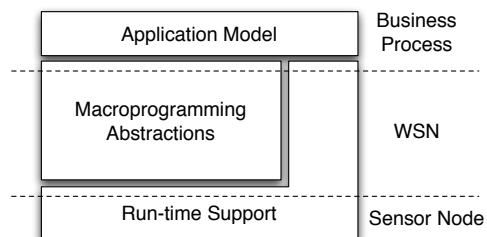


Fig. 1: *makeSense* architecture.

II. APPROACH

Consider an automatic building ventilation system that integrates with an on-line meeting room reservation application. The current approach is to manually ventilate the room either at fixed intervals irrespective of any meetings, or to trigger the ventilation manually, e.g., by the meeting participants. Smart control strategies for ventilation systems may allow to adjust the ventilation levels to the actual demand, based on a room's scheduled and monitored occupancy, while assuring adequate environmental quality. This can save up to 30% of the energy used for air conditioning in a building. The latter account for more than 40% of the energy consumption in Europe.

The smart control system uses WSNs to check the presence of people and the CO₂ levels in rooms. The CO₂ monitoring starts 15 minutes prior to a meeting. If CO₂ is above a specific threshold, the system automatically triggers the ventilation. This check continues periodically. Additionally, 15 minutes after the scheduled start of the meeting, the system starts monitoring the presence of persons. If presence is detected, the system updates the status of the room in the reservation application to “occupied”; otherwise, the room status is set to “available” and the periodic monitoring of CO₂ stops.

To ease the design and implementation of applications such as the one described above, *makeSense* follows an approach consisting of three layers, as depicted in Figure 1:

- The *application model* layer integrates sensor networks with business application systems by allowing WSN behavior to be expressed within a business process model.
- The *macro-programming* layer provides a network-centric

