Experimenting with Computer Swarms: a Mobile Platform based on Blimps

Carsten Buschmann, Dennis Pfisterer and Stefan Fischer
Institute of Operating Systems and Networks
Technical University of Braunschweig
Mühlenpfordtstr. 23, 38106 Braunschweig, Germany
{buschmann|pfisterer|fischer}@ibr.cs.tu-bs.de

In the SWARMS research project [1], we explore the development of applications for huge collections of mobile devices communicating via radio technologies. Our approach is inspired by the analogy of biological swarms and flocks with the following characteristics:

- Relatively resource constrained individuals achieve complex joint operation by local, non-centralized interaction.
- Individuals move independently from each other.
- Individuals might fail at any time due to malfunction or death.

We are especially interested in a problem-centric way of application programming: we prefer to describe the problem and let the swarm decide how to solve it. Our approach is to provide services such as communication, coordination or context awareness that help to hide the distribution from the application and support self-organization. They are based on a new communication paradigm we call the *distributed virtual shared information space (dvSIS)*. The dvSIS describes the state of both the swarm as well as the environment and is based on information that the different devices acquire and publish. No device has complete knowledge (that is, no device sees the whole dvSIS), and must therefore rely on partial information. Due to environmental conditions, node failure, and so forth, the views of different devices might be inconsistent.

Since we do not only want to prove the feasibility of our ideas by simulations, we have developed a first version of a middleware (which we call "SWARMSware") that realizes the dvSIS and hides the environment's complexity from the application. Communication is similar to the well-known publish/subscribe paradigm except that we neither use central instances nor a fixed division into information sources and consumers. The content-based mechanism XCast [2] controls the information dissemination in the network. The dvSIS consists of semi-structured self-describing information components that are defined by an XML schema. These components comprise meta-information describing the context of data acquisition (such as position and reliability) or the data itself (such as the level of aggregation or scope) that let the application evaluate information. Based on this meta-information, the application can decide whether or not to integrate received information into its own view of the dvSIS.

A swarm coordinates the behavior of its members based on their properties (capabilities and states), the swarm's context and the required individual roles to complete a task. Devices carry information on both their capabilities (e.g. available sensors or network interfaces) and state (e.g. location or remaining battery capacity) as part of their local view on the dvSIS. Tasks that groups of swarm members need to fulfill can be derived from the context. A global task could for example be "discover topology", a regional task would be "find group leader". Each task defines different roles with certain requirements. An example for a role is "fixed network gateway" requiring the appropriate network connections. After local checking which roles a device might be able to fulfill, all devices negotiate the distribution of roles.

The SWARMSware especially aims at so called sensor networks [3]: collections of devices which comprise sensing, computing, and radio frequency (RF) communication capabilities. Because devices must be cheap, the employed sensors may be inexact and unreliable. To derive trustworthy readings from rather undependable sensors [4], aggregation techniques such as multilateration and sensor fusion need to be employed. To aggregate corresponding readings from multiple swarm members within certain proximity, devices demand for synchronized clocks. Apart from a common understanding of time, swarm members also require knowledge about their position (absolute or relative to others). Thus, the middleware provides support to time synchronization, numeric location awareness and symbolic representations of locations.

The SWARMSware has been implemented for, among others, the wireless sensor mote ESB 430/2 developed at the Freie Universität Berlin [5], yielding a simple but immobile experimentation environment. In order to also enable experimentation with complex mobile 3D-scenarios, we integrated these motes with mobile vehicles, in this case small blimps. Each blimp is equipped with a sensor mote running the middleware and controlling the blimp's engines. Additional indoor positioning systems feed information on the devices' current position into their dvSIS. We are currently experimenting with two different systems: the first one exploits the differential time of flight property of ultrasound and infrared light in combination with a magnetic compass sensor to gain information on the orientation about the vertical axis. The second, more advanced system relies on the measurement of magnetic field vectors.

Based on this platform, we have designed two sample applications. The first application is concerned with spatial coordination. Here, an external control instance issues commands such as "form circle" or "form rectangle". The swarm members must then coordinate their behavior in order to achieve this task. Our interest is rather on the negotiation of e.g. which of the blimps represents an edge and which form vertices than on path planning etc. Later on, we plan to extend the functionality to blimps forming more complex shapes like letters etc. The second application builds on top of the first one. It is concerned with constantly creating a temperature map of a specified area, i.e. a building. During their flight, the blimps take temperature samples that are then merged into the map (as part of the dvSIS). This information is constantly transferred to a gateway PC from where it can be accessed. In this application, data fusion is the major challenge.

Acknowledgement This work is part of the project "SoftWare ARchitecture for Mobile self-organising Systems". The SWARMS project described in [1] is funded by the German Research Foundation (DFG), where it is part of the focal program "Basic Software for Self-Organizing Infrastructures in Networked Mobile Systems" (SPP 1140). For further information, see www.swarms.de.

[1] Carsten Buschmann, Stefan Fischer, Norbert Luttenberger and Florian Reuter: Middleware for Swarm-Like Collections of Devices, in IEEE Pervasive Computing Magazine, Vol. 2, No. 4, 2003

[2] Jochen Koberstein, Florian Reuter, Norbert Luttenberger: The XCast Approach for Content-based Flooding Control in Distributed Virtual Shared Information Spaces—Design and Evaluation. In: Karl, H., Willig, A., Wolisz, A. (Eds.): First European Workshop on Wireless Sensor Networks, Berlin, Germany, Jan. 2004, Springer Lecture Notes in Computer Science 2920, pp. 188–203.

[3] I.F. Akyildiz, S. Su, Y. Sankarasubramanian and E. Cayirci: Wireless Sensor Networks; A Survey, Computer Networks, Vol. 38, No. 4, March 2002, pp. 393 – 422

[4] Robert Szewczyk, Joseph Polastre, Alan Mainwaring, David Culler: Lessons from A Sensor Network Expedition, In: Karl, H., Willig, A., Wolisz, A. (Eds.): First European Workshop on Wireless Sensor Networks, Berlin, Germany, Jan. 2004, Springer Lecture Notes in Computer Science 2920, pp. 307-322

[5] Website of the Embedded Sensor Board ESB 430/2: http://www.scatterweb.com