

Enabling Wireless Sensor Networks: Integration of WSNs into Development Environments

Tomasz Naumowicz, Freie Universität Berlin

Wireless Sensor Networks

Wireless Sensor Networks (WSNs) are wireless ad-hoc networks consisting of many embedded sensor nodes deployed in the field. They are used for high-precision data gathering in use cases for which setting up a wired infrastructure is too difficult or too expensive.



WSN in agriculture

Typical applications include environmental or habitat monitoring, object tracking, and perimeter security.

The Challenge

Wireless Sensor Networks are not easily accessible to the industry and researchers from other fields than computer science. Today, the required embedded development is a very time consuming and error prone task. Debugging support is limited.

In order to make the WSNs more attractive as a tool, seamless integration into commonly used development environments and a high level of abstraction are needed.

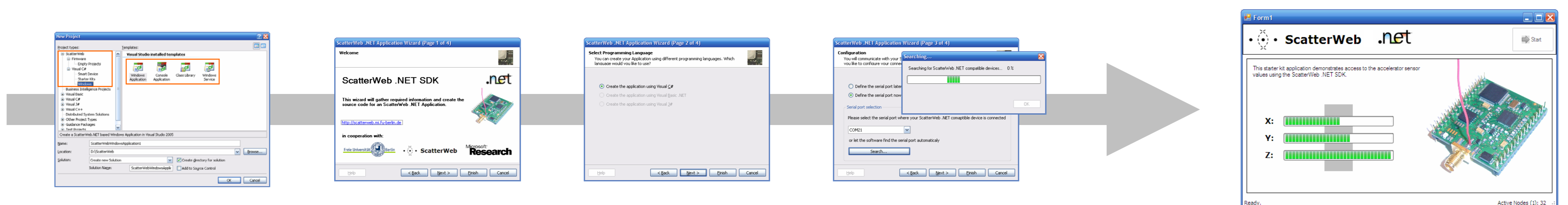
Up until now, research in the area of WSNs has mostly focused on hardware design, self-organisation, a plethora of routing algorithms, or energy saving patterns.

The upcoming challenge is the design of efficient interfaces to the WSNs and their tight integration into development environments such as Visual Studio.

The Research

The research focuses on the workflow from development via deployment to debugging. The research will investigate the arising challenges, such as the efficient design of interfaces to the WSN required by the necessary high level of abstraction, but on the other hand limited by the capabilities of real sensor hardware.

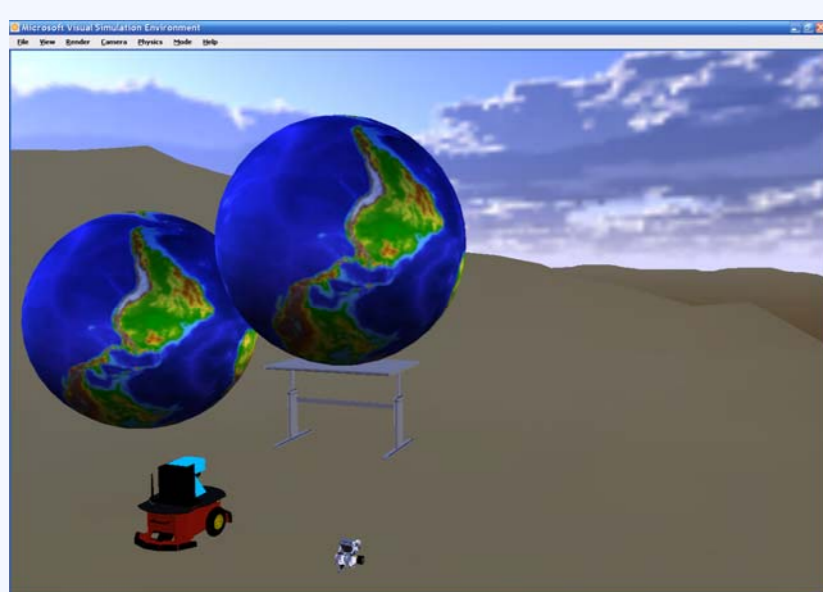
The primary goal is the accessibility of WSNs for every developer and researcher in order to enable new products, research methods and ideas from outside the computer science community.



ScatterWeb .NET SDK integrated into Visual Studio 2005: the development workflow

Current Stage

The first release of the ScatterWeb .NET SDK provides convenient access to the ScatterWeb WSN platform. The SDK was used to integrate the WSN into the Microsoft Robotics Studio and supports real as well as simulated sensor nodes.



Controlling a simulated robot using a real sensor node equipped with a three-axis accelerometer

The ScatterWeb .NET SDK won the BASTA .NET award 2006 in Germany for the best .NET contribution.

Expected Outcome

The ScatterWeb .NET SDK has to support the entire development workflow for the WSNs.

The behavior of the nodes needs to be extensible at run time by the use of an energy-saving virtual machine, the SDK should detect the changes in the functionality of the sensor nodes and adapt itself to the new system automatically.

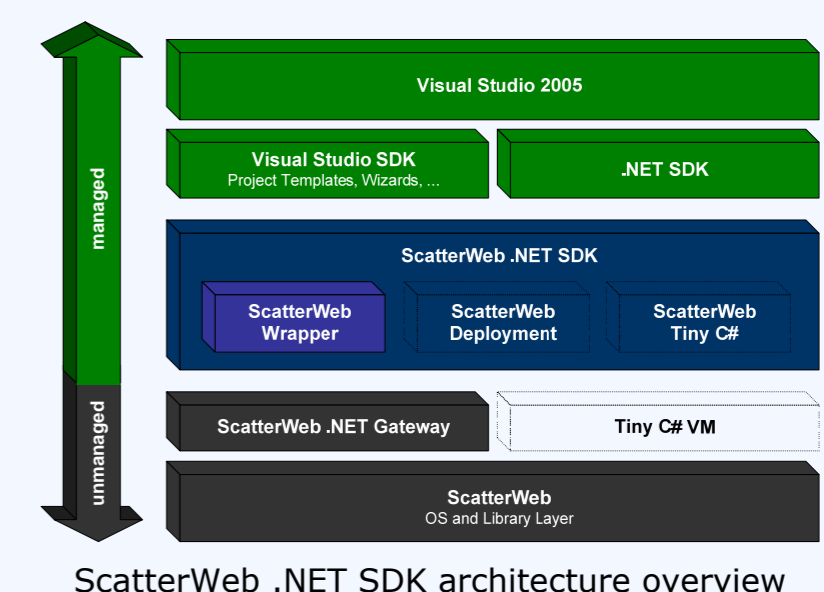
For the developer there should be no distinction between implementing for the distributed embedded platform and the end-user application.

The SDK would be the foundation of an easy to use and deploy "WSN toolbox" for researchers.

ScatterWeb .NET SDK

The SDK extends the .NET tools and architecture to Wireless Sensor Networks

- Seamless integration into Microsoft Visual Studio
 - Development, deployment and debugging
 - Visual Studio SDK: project templates and wizards



ScatterWeb .NET SDK architecture overview

- Easy access to sensor values, events and functions and well designed namespaces and interfaces
 - Well known programming model (events, methods, properties)
 - Support for IntelliSense and dynamic help

The ScatterWeb MSB-430 Platform for Wireless Sensor Networks

- Research platform developed by Freie Universität Berlin
- Commercially available through ScatterWeb GmbH

Core Features:

- Modular concept: sensors attached on demand
- 16-bit RISC microcontroller MSB430F1612, 868 MHz license-free radio Chipcon CC1020, SD-Card storage, Low power consumption 250 µA ... 115 mA



MSB430

```
foreach (Node node in networkManager.KnownNetwork)
{
    if (node.ID == MSB430)
    {
        MSB430 m = (MSB430)node;
        Console.WriteLine(m.Temperature);
        if (m.Temperature > 30)
        {
            m.RedLedOn = true;
        }
    }
}
```