Reconfigurable, Ubiquitous, Networked Embedded Systems and their use in Emergency Response Applications

Lesley Hanna*, Konstantinos Koumpis*, Janne Riihijärvi# and Petri Mähönen#

*Department of Computer Science, University College London, Gower Street, London, WC1E 6BT, United Kingdom, email: {l.hanna/c.koumpis}@cs.ucl.ac.uk

#Department of Wireless Networks, RWTH Aachen University, Kackerstrasse 9, D-52072, Aachen, Germany, e-mail: {jar/pma}@mobnets.rwth-aachen.de

Introduction

The aim of the RUNES project is to enable the creation of large-scale, widely distributed, heterogeneous networked embedded systems that interoperate and adapt to their environments. Only around 2% of microprocessors end up in conventional PCs [1], the rest can be found in a huge variety of devices as diverse as mobile phones, cars, washing machines and scientific instruments By linking the capabilities of the embedded systems in the multitude of devices available and creating networks which can reconfigure and change their functionalities, it is possible to create a technology which has the capability to change the way we live, work and do business. Since the software fabric of networked embedded systems is ad hoc, this is a challenging task. Power, computational capabilities, and communication bandwidth limitations demand new approaches in software design. Embedded software components must be designed and reused at an abstraction level independent of the underlying hardware platform, operating system, and even programming language.

Application areas

Previous work by the RUNES partners has identified a number of application areas where the technology can be used currently or in the short-term future [2]. These include industrial automation, 'smart' buildings and utility monitoring, examples of which can already be seen in commercial applications. Another example worth highlighting is that of medical monitoring which can be used to make a variety of physiological measurements (eg heart rate, blood pressure, blood sugar, oxygenation level) and therefore enables constant, automated monitoring of old or chronically ill people in any location via a Body Area Network (BAN). The technology allows vulnerable people the opportunity to stay in their own homes rather than enter residential care and also offers one way of addressing the care needs of older people when the average age of the population of Europe is climbing rapidly.

However, the application area selected by the RUNES project as the most exciting and the one with the greatest potential, is that of disaster and emergency response. Disasters and emergencies can take many forms and exist on many scales. An earthquake, a health epidemic, an air crash and a house fire all have different timescales and involve varying numbers and types of emergency workers, but all generate a need for fast, accurate information on the scale and nature of the problems and effective communication strategies at a time when the infrastructure to provide information and communication has been compromised or destroyed and human access is limited. The example of 'fire in a road tunnel' was selected to demonstrate how reconfigurable wireless networked embedded systems could be utilised to assist emergency workers,

victims and incident commanders to reduce risk, save lives and limit the damage and disruption arising from an emergency situation.

The project has examined the uses of the technologies developed within the project, firstly by considering how a road tunnel fire might be dealt with in the future and then by demonstrating the most important possibilities identified. The vision for how the technology will be utilised has been summarised in a video produced by the consortium [3]. The demonstration not only shows how the technology will operate, but also addresses fundamental issues such as interoperability and robustness which must inevitably be a part of the technology achieving its full potential.

Tunnel fire scenario definition

The scenario developed by the consortium considers the operation of a fictitious road tunnel on a major transport route in the year 2012. The tunnel is old, but alternative routes are much longer so it is the preferred route for goods vehicles, but some cargoes are refused entry and compliance is monitored via obligatory RFID tagging on heavy goods vehicles. Ventilation is adapted automatically according to the vehicles and goods within the tunnel. In the scenario several vehicles are involved in a collision, including a tanker loaded with vegetable oil which spreads its load on the carriageway. A small fire started after the collision spreads to the oil and a larger fire results producing not only heat but dense smoke.

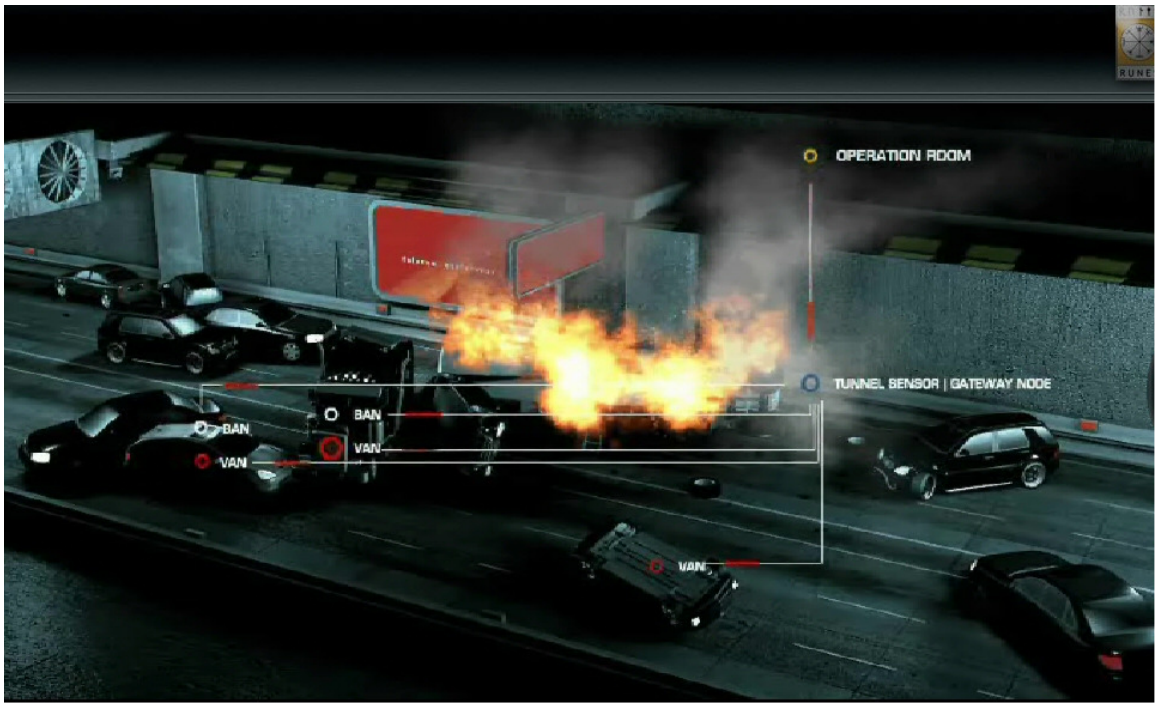

Figure 1: The RUNES project is looking at how reconfigurable wireless networked embedded systems can assist those involved in road tunnel fires

The automatic fire detection system in the tunnel detects the fire and immediately the tunnel operators will summon the emergency services and close the tunnel to further traffic. At this point the tunnel operators need to supply as much information as possible

to the emergency services personnel, specifically the location and nature of the crash, the number of people and vehicles involved and the access and condition of the area they will enter. Just as this information is required video surveillance cameras will become useless because of the dense smoke from the fire and conventional wired systems in the tunnel are likely to start to fail because of the heat.

In the above scenario, the technologies developed in the project enable the tunnel operators and emergency services personnel to interrogate the sensors which remain operative within the tunnel in order to obtain information on conditions in advance of fire fighting and rescue process. Wireless operation means that data can still be obtained even if the wired infrastructure has been destroyed. Combining data from existing tunnels sensors (temperature, obscuration, air quality etc) and those in which can be accessed from vehicles and personnel involved in the accident (eg accelerometers used to trigger air bags will indicate that the vehicle has been involved in a collision, engine or cabin temperature readings may denote the spread of the fire) will give a much clearer picture of the conditions in the affected area. It should be noted that some rescuers bravely entered the Mont Blanc tunnel during the catastrophic fire there in 1999 without significant data on the conditions they were likely to meet, and were themselves trapped when the oxygen levels in the tunnel became too low for their vehicle engines to operate.

Other data available to emergency response teams may come from people with medical problems who already have BANs. This shows not only where casualties are located who may require priority evacuation, but also the areas where other survivors may be found.

Figure 2: The fire fighters of the future will be equipped with interconnected devices to provide more data about the hazards they face and to monitor their own safety during rescue work

Once rescue workers enter the affected area their own devices will be added to the network. Remote controlled vehicles or robots may be deployed to some areas in order to provide sensor readings in areas where more data are needed. Rescue workers, fire fighters in particular, may be equipped with BANs of their own allowing the command and control to locate and monitor the condition of all their personnel. Enhanced

communications will also enable plans and records to be accessed to identify escape routes, retrieve medical records and easily locate any information needed to assist the rescue process as well as contacting remote experts on medical, engineering or safety matters.

The value of robot and other autonomous vehicles is potentially very significant, and other initiatives also exist to examine how they might best be deployed such as ROBOGAT [4]. As well as acting as a sensor platform or distributing sensor devices in areas of interest, robots can act as communications gateways and when suitably equipped become actuators, taking an active part in the rescue work.

Demonstration of RUNES technology

The RUNES project is committed to demonstrating how the systems will work in the real world, and as a result the partners have collaborated on a series of practical demonstrations of reconfigurable wireless networks based on the fire in a road tunnel scenario. In a tunnel fire situation, requirements would be constantly changing with devices being destroyed and new devices and personnel being added, so networks would be required to reconfigure continuously. Under changing conditions the requirements upon the networks would also vary, for example higher temperatures might require increased reporting frequency, which has implications for the middleware.

For the technology to be useful under real-world conditions, the system must be robust and not require significant intervention or technical knowledge from any of the users. Devices must autoconfigure and autoreconfigure when changes occur.

Capabilities demonstrated so far include:

- Wired to wireless failover due to destruction of part of the wired tunnel infrastructure network
- Establishment of a network based around a mobile gateway carried by a fire fighter to enable fire fighters to receive data from all available sensor nodes in the vicinity and transmit data from their own sensors
- Loss of sensor node to network and consequent rerouting of data traffic
- Loss of gateway node and consequent rerouting through an alternative gateway

RUNES Architecture

Design of an embedded systems architecture capable of catering for the scenarios discussed above is certainly a challenging task. Devices present in the networks are very heterogeneous in their capabilities, and extremely resource-constrained devices are commonplace. The overall RUNES architecture should scale from small sensor and actuator nodes to large emergency communications networks consisting of thousands of nodes, some of which with very high computational capabilities.

To design such an architecture the RUNES consortium has adopted a three-phase approach. First, a generic reference architecture has been specified, followed by more

specific “sub-architectures” for different application domains. During this work detailed requirements gathered in the early stages of the project have been used to guide and to prioritise the work. These requirements are publicly documented in [5]. The third, presently ongoing phase of the design work is the validation procedure based on the demonstrators described here, combined with simulations and software assurance methods. In the following section we shall briefly discuss the overall and specific RUNES architectures, focussing on the demonstrators targeting the emergency scenario selected.

The generic RUNES architecture developed provides the overall framework in which specific embedded networking solutions can be realised. In this framework different functionalities present in networked embedded systems are grouped on four abstraction levels shown in Figure 3. At a certain level this higher level RUNES architecture can be seen as a template that can be used to “compile” the final fixed architecture. The general architecture defines the basic abstraction levels and generic interfaces, for specific networks some more fixed technology solutions are needed and we have place holders in our generic architecture for those. When the specific requirements and additions are put in, the end result is more specific and fixed architecture which is ready to be used for real deployment projects. We shall now discuss the content of the architectural levels in more detail.

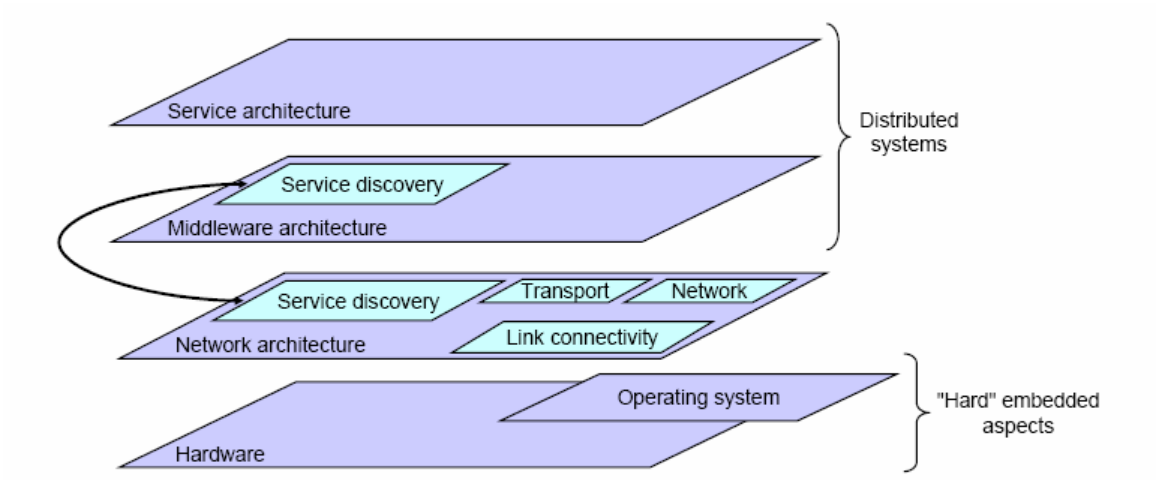

Figure 3: Architectural levels in RUNES

The hardware and platform level forms the foundation of the higher abstraction levels with specific sub-architectures for platform hardware, operating systems and so on. For example, in the hardware sub-architecture three categories of device are defined: the lowest level is that of sensor nodes which have the capacity to make measurements and forward the data to the network. There are generally expected to be resource-poor and therefore capable of running strictly limited middleware components. At the next level of functionality we have sensor routing nodes. They are considered to be less resource-constrained and thus capable of running more advanced software components. Sensor routing nodes are able to act as cluster heads for sensor nodes, communicate with gateway nodes, and are likely to be able to carry out a certain amount of processing. The final category of device is the gateway node. The gateway node is more powerful

and is expected to run a full set of middleware components, connect the network to other networks, potentially over IP, and have the capability to store and process data from the entire network. The gateways through their middleware provide a strong abstraction capability, and in principle application programmers can often consider sensors just like web-services as the technical complexities are hidden behind the middleware components.

The middleware and application functionalities can be realized to a large extent in operating system independent fashion, and the overall RUNES architecture is consequently operating system independent. The present RUNES *demonstrator* architecture contains provisions for three operating systems. Sensor nodes run the Contiki operating system, which has been substantially extended during the course of the project. The sensor routing nodes used in the demonstrator, having more hardware resources available, run a more powerful OS. FreeRTOS has been chosen to serve this purpose. The gateway nodes developed in the project are finally operating with embedded version of the Linux OS.

On top of the hardware and platform level resides the network level with basic communications functionalities typically relatively tightly integrated with the operating system. Finally, the middleware level offering a highly flexible component-based framework for managing software on the nodes and the service level at which the distributed applications themselves reside are situated above the network level. The RUNES middleware is based around a component model supported by a minimal runtime API. The component model is independent of language and the components themselves can be implemented in different ways dependent on the device they are to be run on. This model gives the flexibility to allow the middleware to work with different OSs and device capabilities. A more comprehensive description of the middleware can be found in reference 6.

The framework outlined above is used mainly for reference and organisation purposes. Given that suitable primitives are offered by the network level, for example, complete communications protocols can be realized entirely within the flexible middleware or the middleware can offer simple “wrappers” to give access to networking protocols implemented outside the middleware framework. The advantage of this dual approach is that application developers only have a single entity (the middleware runtime) to interact with on any platform, and the programming interfaces used for this interaction are platform and operating system independent. Both approaches are present in the current RUNES system, and in the demonstrators described. Additionally, some functions such as service discovery naturally use elements residing on multiple levels (such as neighborhood discovery on network layer coupled with probabilistic broadcast implemented within the middleware).

Some constraints or “fixed points” have to be specified for this generic architecture to make it usable for practical applications. In the RUNES project a number of standard interfaces are being specified, fixing the format of the entry points to any component offering, for example, capability to access sensor readings, influence actuators, access platform information (such as operating system details and available memory) or information about the communications subsystems. As an example on the extent of the interface definition work RUNES is closely collaborating with the European GOLLUM-project which is defining a generic interface for accessing link-layer information in a technology-independent manner. In addition to specifying a small collection of well-

defined interfaces constraints are also placed on types of protocols supported, as well as packet formats used. These types of restrictions ensure that different nodes utilizing RUNES technologies are able to interoperate, while still retaining the flexibility offered by the component-based middleware to dynamically add new protocols and other functionality dynamically to optimize the interaction further or react to changes in the network.

Details of the RUNES demonstrator

The RUNES demonstrator is designed to show how the architecture will operate in a real world situation. The demonstration starts with traffic running normally in a tunnel equipped with sensors as part of the infrastructure. The sensors used in the demonstration are TMote Sky motes with a gateway node produced especially for the project by one of the industrial partners. When fire breaks out as a result of the collision, the fire is detected by the nearest sensor, and this information is conveyed to the tunnel control room. Since use of real flames is not practical, the fire is modelled using light intensity with a certain threshold denoting fire, and a greater intensity indicating destruction of the sensor node and resulting in its removal from the network.

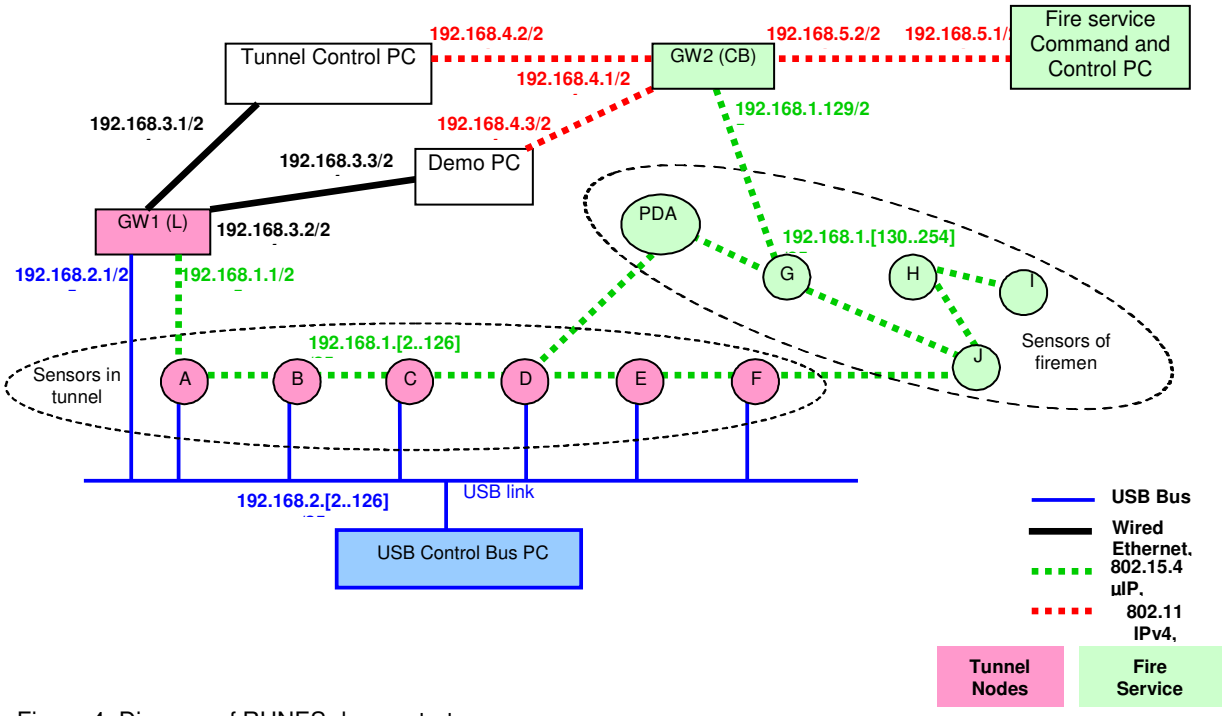

Figure 4: Diagram of RUNES demonstrator

The emergency services are summoned, and the first fire fighters arrive at the scene. Each is equipped with a sensor node (again a TMote Sky mote) and they bring with them a portable gateway with which they can set up their own network. The gateway is again one which has been developed especially for RUNES, but by a different project partner and using a different operating system, showing how heterogeneity is addressed. The failure of one of the gateways is demonstrated, showing how the network can be reconfigured to allow use of an alternative gateway. The development of the fire and the response of the fire fighters results in variable levels of danger, and

dynamic code upload is used to allow the firemen's sensor nodes to respond to the situation by changing the way data are broadcast back to the emergency control centre.

In the real world networks can be expected to be extremely heterogeneous and the various stakeholders (eg tunnel operators, fire fighters, medical workers, command centre personnel and victims) all have different requirements from the system. The RUNES demonstrator shows how this variability can be successfully accommodated in a dynamic environment. Further information on the demonstrator can be found in RUNES public deliverable D7.3.1 [7].

Concluding Remarks

The RUNES technology is ideal for addressing issues associated with: gaining an understanding of the environment resulting from natural or man-made disasters; for assisting the command authorities in understanding the situation of people and assets in the emergency zone; and for allowing tasks to be carried out to locate or rescue victims, in many cases using systems that are already present at the scene and that can be used for purposes for which they were not originally designed. The demonstrator system outlined here enables the provision of services unattainable only a few years ago, and which we expect to play a significant role in disaster mitigation and management as well as in the field of networked embedded systems in general.

Acknowledgement

RUNES is an integrated project supported by the European Commission Framework 6 programme and involves 21 partners from 9 countries.

References

[1] J. Turley, 'The Two Percent Solution', Embedded Systems Design, Dec. 2002, available from http://www.embedded.com/story/OEG20021217S0039

[2] RUNES deliverable D8.1.2 'Proceedings of the First Industry Forum' available from www.ist-runes.org/public_deliverables.html

[3] For further details or to download the video visit www.ist-runes.org/scenario.html

[4] Laura Celentano, Bruno Siciliano and Luigi Villani, 'A robotic system for fire fighting in tunnels', Proc. 2005 IEEE Int. Workshop on Safety, Security and Rescue Robotics, Kobe, Japan, pp253-8, June 2005.

[5] RUNES deliverable D1.2 'Requirement and Constraint Analysis' available from www.ist-runes.org/public_deliverables.html

[6] P. Costa, G. Coulson, C. Mascolo, G.P. Picco and S. Zachariadis, 'The RUNES middleware: A reconfigurable component-based approach to networked embedded systems, in 16th IEEE Int. Symposium on Personal Indoor and Mobile Radio Communications (PIMRC05), Berlin, Germany, September 2005.

[7] RUNES deliverable D7.3.1, 'Small Scale Deployment Specification' available from www.ist-runes.org/public_deliverables.html