

Towards the Real-Time Enterprise: Service-based Integration of Heterogeneous SOA-ready Industrial Devices with Enterprise Applications

Stamatis Karnouskos* Dominique Guinard* Domnic Savio*
Patrik Spiess* Oliver Baecker* Vlad Trifa*
Luciana Moreira Sá de Souza*

* SAP Research, (e-mail: stamatis.karnouskos@sap.com)

Abstract It is expected that millions of embedded devices and machines empowered with Internet technologies will be able to communicate, collaborate and offer their functionality as a service. At the shop floor, this creates new opportunities for more dynamic environments where timely usage of the monitoring information is coupled with control and in full collaboration with enterprise systems. We focus on demonstrating our efforts towards such cross-layer composition for the future service-enabled factory.

Keywords: Enterprise-wide Information System, Extended Enterprise, Enterprise Interoperability, Service Oriented Architectures

1. INTRODUCTION

We see a paradigm shift on the shop-floor as Information and Communication Technologies (ICT) are being used at an increasing rate. As high performance micro-controllers are embedded on devices commonly used in the industrial and automation domains, collaborative networks and services hosted on them, promise the emergence of new applications on the shop floor. It is expected that this trend will go on, while in parallel the number of devices will explode.

As we are moving towards the "Internet of Things" as depicted by Fleisch and Mattern (2005), millions of devices will be interconnected, provide and consume information available on the network and cooperate. As these devices need to interoperate, Service Oriented Architectures (SOA) seems to be a promising solution. The world market for technologies, products, and applications alone that are related to the Internet of Things will increase significantly from €1.35 billion to more than €7.76 billion in 2012, with average annual growth rates of almost 50% (sap (2008)).

In a real-world SOA, each device offers its functionality in a service-oriented method, while in parallel it is possible to discover and invoke new functionality from other services on-demand. These devices can be considered as a set of intelligent system entities which act like conglomerate of distributed, autonomous, intelligent, pro-active, fault-tolerant and reusable units Colombo and Karnouskos (2009). They co-operate to form a new dynamic infrastructure that is able to provide a better insight of its individual components to the higher levels in the plant IT Landscape. They flexibly react to dynamic business changes that can influence the production plan on the shop floor.

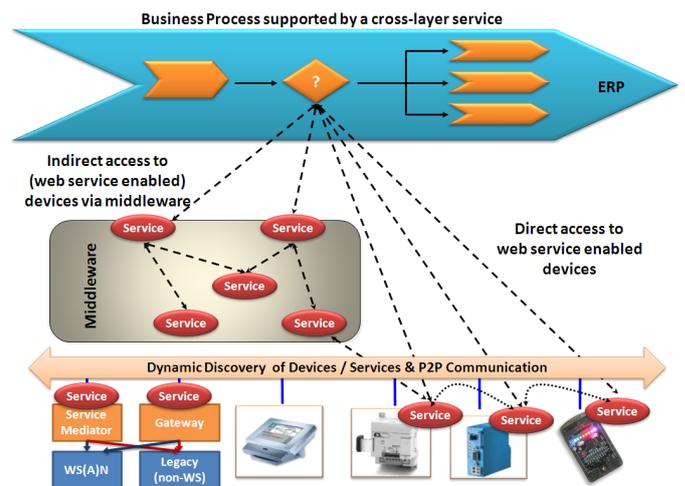


Figure 1. Vision of cross-layer service composition. The services provided by embedded devices on the shop-floor are orchestrated either through a middleware or directly and used as the building bricks of dynamic business processes

As demonstrated in previous work e.g. by Jammes and Smit (2005); de Souza et al. (2008); Priyantha et al. (2008); Karnouskos et al. (2007), future shop floor infrastructures can adopt and benefit from service oriented approaches. Furthermore this can be really applied in horizontally and vertically e.g. as depicted in Figure 1 and pursued within the SOCRADES project (www.socrades.eu). In these infrastructures, new functionalities can be introduced by combining services in a cross-layer form i.e. services based on the enterprise system, on the network, and hosted on the device itself. These services allow orchestration and service composition that provides new integration scenarios across the shop floor. Sophisticated services can

be created at any layer (even at device layer) considering the functionality of other entities that can be provided as a service. In parallel, dynamic discovery and peer-to-peer (P2P) communication allows for the device's functionality to be dynamically discovered and optimally exploited. It is clear that we are moving away from isolated stand-alone hardware and software solutions towards more cooperative models.

The convergence of solutions and products towards the SOA paradigm enhances the shop floor by better integration, increased transparency and thereby contributes to the improvement in reactivity and performance of workflows and business processes, commonly found in manufacturing, logistics and others. This leads to information being available as it happens, on an event-driven basis, and business-level applications can exploit such information for diagnostics, performance indications, traceability, etc. These future vertical collaborations would be vital for business application software which would consider error prone or critical units on the workflow while making decisions

In this paper, we start by discussing different approaches to integrate industrial systems and machines to enterprise systems and further describe the chosen approach. Our core contribution follows with the description of two concrete prototypes that serve as a proof-of-concept for our proposed integration architecture Karnouskos et al. (2007); de Souza et al. (2008).

2. INTEGRATING HETEROGENEOUS DEVICES

Devices and machines are poorly visible to enterprise business applications like the ERP (Enterprise resource Planning) Software. Planning a production order or creating a bill of materials in the ERP application is neither effective nor optimized when the shop floor is not transparent. Hence device manufacturers end up writing custom drivers or provide standard connectors that talk only to its counterparts on the shop floor. We propose to leverage this integration gap on the shop floor by extending the service-oriented approach on devices. The integration concept proposed in this section focuses on an overall architecture that brings the advances of service-oriented approaches on devices to the enhancement of business processes and the current enterprise systems.

Several efforts have explored the integration of real-world and enterprise services e.g. Marin-Perianu et al. (2007); Edwards (2006). However, the protocols used do not offer uniform interfaces across the application space and are thus complicated to integrate with traditional enterprise applications. Alternative shop floor integration technologies based on web services like OPC UA (opc (2008)) are emerging. At the time of writing this paper, advanced features like dynamic device discovery, eventing and notification mechanisms are only on conceptual level in OPC UA specifications. Further more the web service part of OPC UA is optional and no implementations are available. Hence OPC UA based clients have to be installed on all systems that would need to consume shop floor data from a device hosting the services (Jammes et al. (2009)).

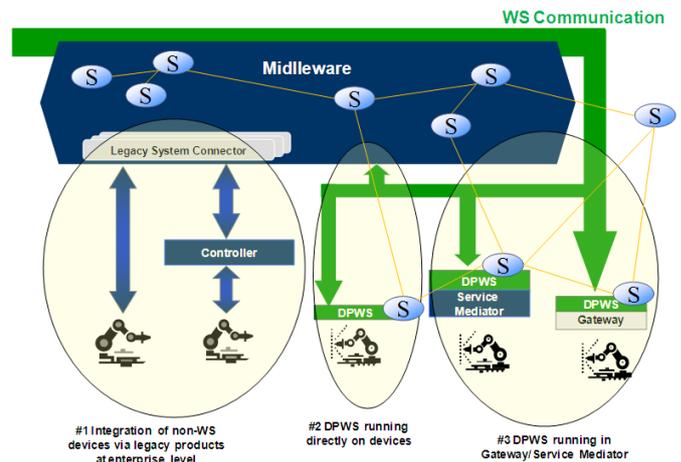


Figure 2. Integration approaches to couple legacy and emerging device infrastructure to enterprise systems

Other shop floor integration standards in the semiconductor industry are available for example the SEMI Equipment Communications Standard / Generic Equipment Model (SECS/GEM) Communication protocols between a host computer and semiconductor manufacturing equipment (Cimetrix (2009)). The Photovoltaic Equipment Communication Interfaces (PVECI) is based on the SECS/GEM and more oriented towards the photovoltaic industry (pve (2009)). Clients have to be implemented that understand the protocols and communicate using interfaces defined in the standards. The biggest disadvantage is that only one client can interact with a server in a session. Multiple sessions are not possible. None of these technologies provide web service capabilities - a paradigm we consider vital for flexibility, agile and ease of integration between the shop floor and enterprise applications.

To ensure interoperability across all systems, recent work has focused on applying the concept of Service Oriented Architecture (SOA), in particular Web Services standards (SOAP, WSDL, etc.) directly on devices. Implementing WS-* standards on devices presents several advantages in terms of end-to-end integration and programmability by reducing the needs for gateways and translations between the components. This enables the direct orchestration of services running on devices, with high-level enterprise services, e.g. offered by Enterprise Resource Planning (ERP) applications.

In Karnouskos et al. (2007); de Souza et al. (2008) we propose an extensible integration architecture based on Web Services and capable also of supporting legacy products. There are basically three directions we follow as shown on Figure 2:

- Integration via legacy products: several tools are available to the market today, so we rely to them for providing the connectivity (as it is done today).
- Web Service (WS) enabled devices: these have the computational power and communication capabilities to run natively the stack that provides their functionality as a set of Web Services. These devices can be directly discovered and interact with each other.
- Integration via Gateways and Service Mediators: legacy or resource-scarce devices have their function-

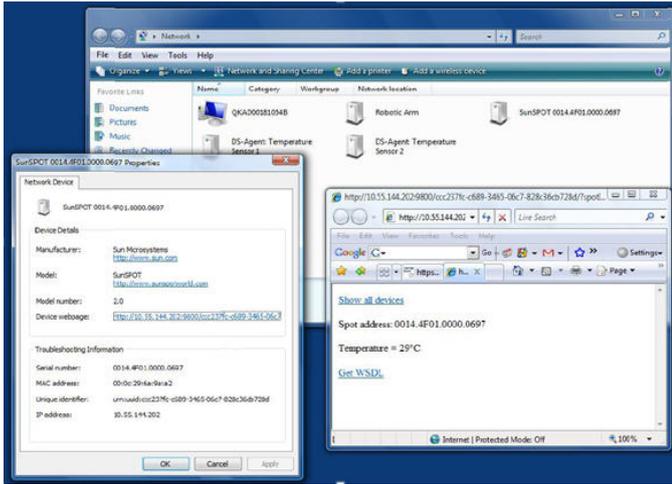


Figure 3. Dynamic discovery of devices built in Microsoft Windows Vista

ality wrapped as a Web Service at a higher layer, e.g. a control point (gateway approach) or their functionality is aggregated/composed and a new service depending on (possibly many devices) is created (mediator approach).

In order to embed web services in devices we selected the Devices Profile for Web Services (DPWS) Chan et al. (2005), which is a collection of web service standards targeted towards devices. Initially the home automation was scope of this standard but lately efforts such as the SOCRADES project have shown its applicability to the automation (Jammes and Smit (2005)) world. DPWS advances previous dynamic discovery (Edwards (2006)) concepts such as Jini (www.jini.org) and UPnP (www.upnp.org) to integrate devices into the networking world and make their functionality available in an interoperable way. DPWS is an effort to bring a web services on the embedded devices taking into consideration its constrained resources. Several implementations exist in Java and C (e.g. www.ws4d.org, www.soa4d.org), while Microsoft has also included a DPWS implementation (WSDAPI) by default in Windows Vista and Windows Embedded CE.

3. PROTOTYPE IMPLEMENTATION

We consider some common use cases with respect to shop floor and ERP integration. Machines on the shop floor are usually monitored for their status. This information could represent completion of an order, number of work pieces produced or even failure of a machine, which is vital for consecutive production plans. To gather such information, we use the DPWS and implement web services on experimental (wireless) sensors like SunSPOTs (www.sunspotworld.com), Phidgets www.phidgets.com, as well as common devices on the shop floor like Programmable Logic Controllers (PLC) and Radio-frequency identification (RFID) readers. We integrate them to ERP System components like the SAP MII (SAP Manufacturing Integration and Intelligence).

An asset management scenario is built using such a setup. It offers the possibility to dynamically identify discover, easily integrate and manage the devices. The business

value of such direct device visibility, and access to data is significant. Another interesting scenario we have investigated is the collaboration of services coming from the devices, and those available across the network that could be brought together in meaningful composite applications. Two implementations are presented here, showing that:

- simple event-based monitoring and limited (soft / high-level) control can indeed be applied, and
- complex device integration can be easily realized using dynamic Web Service composition in a mashup manner.

These two use cases have common characteristics, and we focus on:

- **Smart Devices:** Manufacturing devices, sensors, PLCs, IT devices etc are the actors forming an "Internet of Things" in the factory as well as outside of the factory. They all offer web service interfaces, either directly or through the use of gateways or service mediators. Through these interfaces they offer functional services (e.g. start/stop, swap to manual/automatic mode) or status information (e.g. power consumption, mode of operation, usage statistics, etc.).
- **Business Logic and Visualization Services:** In our prototypes, the business logic services are supported by a service composition engine and a GUI using a visualization toolkit. An operator can use existing tools to create the business rules. Via visualization tools the plant-floor status and the overall process execution can be depicted in detail. As an example the operator can instantiate and use a set of widgets such as gauges and graphs to monitor in real time the status of production machines and associate it with the respective orders or business effects e.g. if a malfunction happens.
- **Enterprise Applications:** This is the place of high-end business software such as ERP or PLM (Product Life cycle Management). The idea at this level is to visualize processes rather than the machines executing the processes. This layer is connected to the plant-floor devices through the other layers. As such it can report machine failures and plant-floor information on the process visualization and workflow. Furthermore, business actions (e.g. inform customers about a possible delay) can be executed based on this timely information.

3.1 Overheating Monitoring and Dynamic Management Decision

In a production line we need to monitor a robotic gripper for overheating as this would cause further malfunctions. As shown in Figure 4, a SunSPOT wireless sensor node is attached to the gripper and checks continuously the temperature at the relevant location right before the gripper starts its operation, e.g., open or close. A PLC controls the robotic gripper and offers its available functionality as several web services; the same holds true for the wireless sensor. In SAP MII we have modeled the business logic (shown in Figure 5) which takes decisions based on the input delivered via the data arriving from the web services hosted by the robotic gripper and the wireless sensor. The operation of both devices generates events,

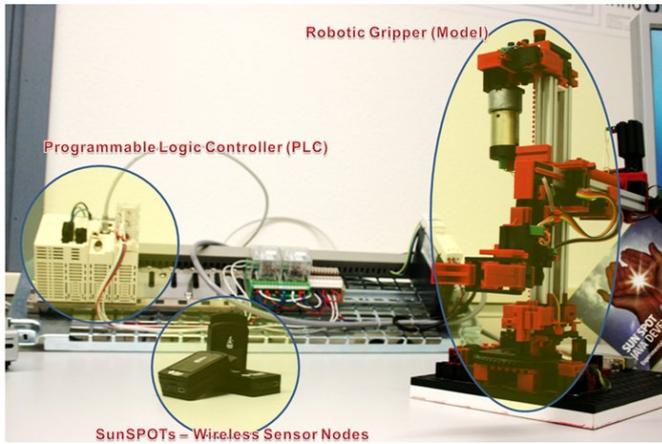


Figure 4. Overheating monitoring via wireless sensor and ERP-supported control of a process

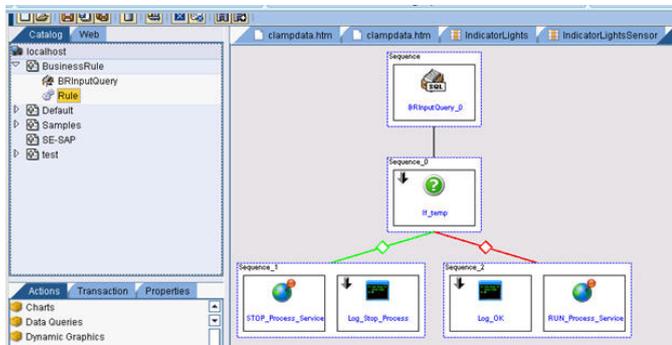


Figure 5. Modeling business logic in SAP MII

which are picked up and consumed by SAP MII as we extended the tool to subscribe to these.

In our scenario, during normal operation the robot gripper should not exceed a specific temperature limit. If the threshold is exceeded, a business rule triggers necessary countermeasures e.g. it stops the gripper in order to prevent damage and invokes appropriate enterprise-level services. This includes visualizing the stopping of the gripper and possible delays in production, e.g. by changing the color in the management view of the factory as depicted in Figure 7. If the resulting order delay is critical, a notification is generated for the key account manager, about the fact that an order for his client is in danger of missing the deadline. The timely information about production delays has the potential to prevent lost sales, which are more likely to occur if customers do not receive their order in time and are not informed timely and reliably about it. The management cockpit of Figure 7 is another example of web service composition at a higher layer where we integrate the real time status of the factory with Google Maps in order to visualize the overall effect of a single failure that results from an order delay.

Apart from the high level view for the manager, there is also other visualization information for the operator. A simple gauge fed with the temperature data provided by the wireless sensor can be depicted also in real-time via the SAP MII tool. For this purpose, he uses manufacturing

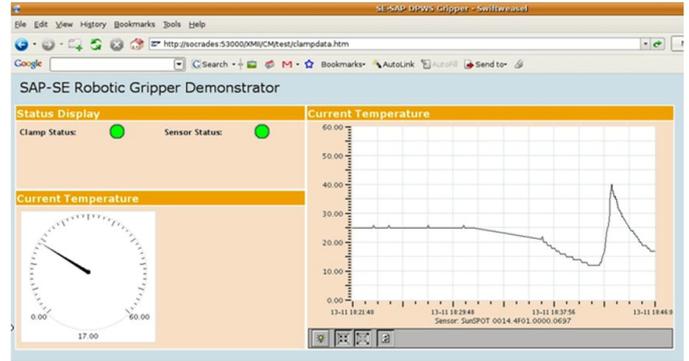


Figure 6. Live reporting in SAP MII of the current shop-floor status based on events received from the devices

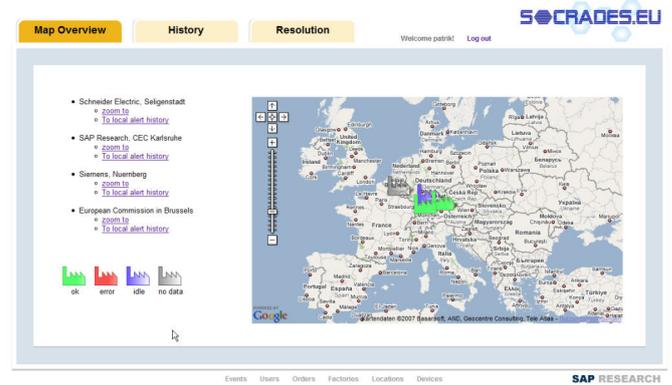


Figure 7. High-level factory view in the management cockpit

intelligence software and displays the gauge on a screen situated close the robot. Finally, the sales manager can also benefit from the service-oriented architecture paradigm, as e.g. the output of the business rule is connected to an ERP system which provides up-to-date information about the execution of the current orders. Whenever the process is stopped because the rule was triggered, an event is sent to the ERP system through its Web Service interface. The ERP system then updates the orders accordingly and informs the clients of a possible delay in the delivery.

The main goal of our prototype was to show:

- Web service based integration.
- Cross-layer composite services.
- Enterprise control and visualization via Web Services.
- Dynamic workflow for cross-layer alert resolution.
- Dynamic customer relationship management based on information from embedded-devices and machines.

3.2 Web mashup of heterogeneous SOA-devices and services

Today, integrating devices in applications requires not only advanced knowledge of the device, its configuration, and the way it connects, but also the installation of highly specialized software that glues the data (often in proprietary format) with the functionality required by each application. Such integrations are costly, application specific and create islands of isolation at the shop floor. As a result, it is extremely hard for enterprise service developers to enrich service functionality with real time data coming from the shop floor. The SOA concept, has

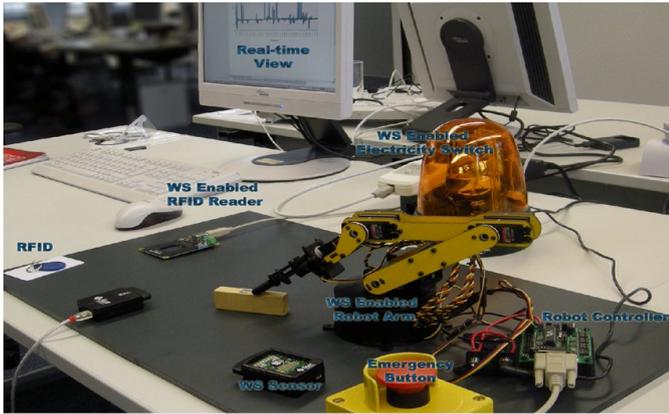


Figure 8. WS enhanced/wrapped heterogeneous Devices - RFID readers, Sensors, Robot, IP electrical Switch etc

proven very successful for gluing heterogeneous systems, and if the same would be applicable for devices this would be a significant step forward in the direction of coupling the real-world and the business world.

To demonstrate the concept we have taken several devices (some of them IP-(Internet Protocol)enabled) and wrapped their functionality with web services. More specifically as depicted in Figure 8 we have:

- An RFID reader: RFID tags as they appear are read by the reader, which raises events showing info with respect to the tag. Each tag is considered to be integrated with a product and serves as a token that links it with the business information (e.g. order).
- A robotic arm: The functionality of the robotic arm controller has been wrapped, e.g. grab or move.
- An electricity IP switch: An alarm lamp has been attached in an electricity IP switch, which offers the on/off functionality as a web service.
- A wireless sensor-controlled emergency button: in the IO pins of the SunSPOT wireless sensor, the emergency button has been attached. The press or release of it is captured by the SunSPOT, and a WS-event is generated.
- A wireless sensor for vibration monitoring: The capability of SunSPOTs to measure acceleration is used for vibration monitoring via the sensor mounted on the robotic arm. This monitors the transportation conditions of the product by the robot to make sure it adheres to the quality guidelines for the specific product.

As production parts come to the factory shop floor, they are equipped with an RFID tag. As they reach the processing point, the data is picked up by the RFID reader. An event is generated by the reader with all the necessary data, e.g. RFID number, and placed on the network. The robot gets notified by this event and picks up the production part. By matching data from the enterprise system and the RFID tag, it knows how to further process the part.

In parallel, the SunSPOT mounted on the robot monitors the vibration and if this exceeds a specific threshold an event is raised to immediately stop the process. The same

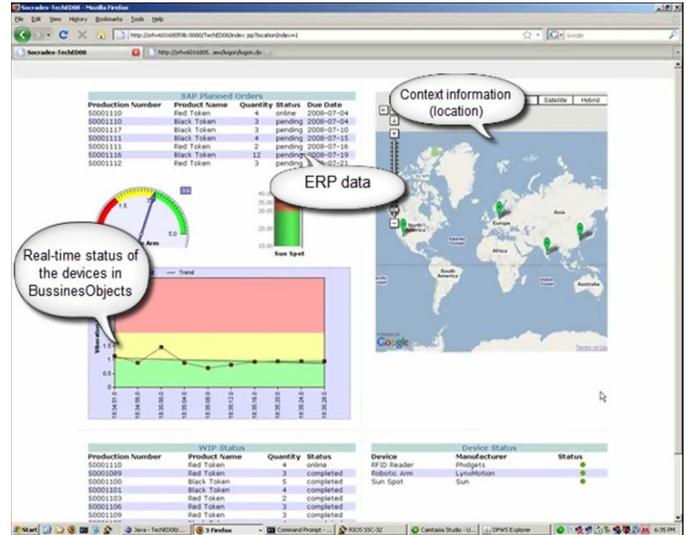


Figure 9. Real-time reporting via enterprise visualization tools

holds true for the emergency switch, if an operator at the shop floor for any reason wants to immediately stop the process, s/he presses this switch. The result is captured by the wireless sensor and an event is raised to immediately stop the robot.

Once an event occurs, the devices process it and react accordingly. As such, the robot picks the emergency shut-down event and immediately stops its action. Also the IP electricity switch turns on the emergency light once it receives the event. At a higher level, there is an enterprise application such as shop-floor reporting realized by the SAP product Business Objects. There we have a real-time monitoring of the shop-floor status as the application also subscribes to all events raised. The plant manager can immediately see (Figure 9) the status of the ERP orders, the production progress, the device status, as well as have a global view on all factories and the possible side-effects of a production line delay due to shop-floor device malfunctions.

All of the communication is done via web service technologies and the realization of this specific scenario was made possible by having a composition of the available functionality that all devices and applications expose as a service. In legacy systems integration of a new device or reassignment of its role would result in reconsidering how the device integrates with other devices and how they control it. However with the SOA approach described, a new scenario is possible by modifying the orchestration of the services already available.

The main goal of this prototype was to demonstrate:

- Easy integration of functionality of heterogeneous devices.
- Embed real-world device functionality in existing enterprise service modeling tools.
- Cross-layer alerts in an event based infrastructure.
- Visualization via existing enterprise reporting tools.

4. FINAL THOUGHTS AND FUTURE WORK

We are heading towards industrial infrastructures where heterogeneity will be dominant. One possible direction to enhance interoperability and cross-layer collaboration is to use service oriented approaches. As an example we have demonstrated the effects of using web services on enterprise system, network and the device itself. However, even in a future infrastructure where more powerful devices with respect to memory, computation, and communication will be available at large numbers, not all devices will have the capability of hosting natively Web Services, e.g. running the DPWS stack, and providing their functionality as a service to the others. In fact, the last might not only be infeasible due to technological constraints, but it also might not make sense from a business point of view. Therefore, any approach proposed for the future manufacturing domain has to make sure that several types of devices can be directly and indirectly integrated in a global communication infrastructure.

Effective integration of devices into business processes can move beyond simple monitoring (although this can be greatly enhanced) towards soft control of high level functionality relevant to the business processes associated with the respective task. Data provided by devices can play a more active role and influence directly and in real-time the flow of business processes running on an enterprise system. For example, at a decision point where different subsequent process steps are possible, information provided by device-level services on-demand can affect dynamically the flow of the corresponding business process.

If the device layer is directly accessible from enterprise systems (and vice-versa), task-specific steps can be delegated to the devices, thereby partially distributing the processing of info to the devices, the network, and the enterprise system, and all would work in a collaborative manner. Putting the process execution at the place of action (e.g. device) has profound implications that can lead to more sophisticated approaches. Wrapping the existing device functionality can be an intermediate step towards moving from the legacy infrastructure to a more powerful and fully IP-based one (at least at gateway level) as described in the prototypes above.

In this paper, we demonstrated an integration architecture which supports several types of devices via protocol plugin concept e.g. web services. Future work will include a thorough evaluation of the whole process and specifically parts of the architecture, focusing on performance, scalability and usability. We are currently setting up several trials within the SOCRADES project where the exposed concepts can be evaluated in realistic environments. Furthermore, with the advances of Web technologies, we currently explore and evaluate alternative ways e.g. OPC-UA, RESTful services (Guinard and Trifa (2009)) of integrating real-world devices into enterprise in order to allow for even more flexibility for non-critical services and applications.

ACKNOWLEDGEMENTS

The authors would like to thank the European Commission and the partners of the European IST FP6 project "Service-Oriented Cross-layer infRAstructure for

Distributed smart Embedded devices" (SOCRADES - www.socrades.eu), for their support.

REFERENCES

- (2008). Opc unified architecture (opc-ua). URL <http://www.opcfoundation.org/UA>.
- (2008). Toward a european strategy for the future internet. URL http://www.europeansoftware.org/documents/SAP_WP_FutureInternet.pdf.
- (2009). Pv-eis equipment interface specification standards. URL www.semi.org.
- Chan, S., Kaler, C., Kuehnel, T., Regnier, A., Roe, B., Sather, D., Schlimmer, J., Sekine, H., Walter, D., Weast, J., Whitehead, D., and Wright, D. (2005). Devices profile for web services. Microsoft Developers Network Library.
- Cimetrix (2009). Secs/gem semi standards overview. Internet. URL <http://www.cimetrix.com/gemintro.cfm>.
- Colombo, A.W. and Karnouskos, S. (2009). Towards the factory of the future - a service-oriented cross-layer infrastructure. *in the book ICT Shaping the World, A Scientific View, ETSI, John Wiley and Sons Ltd*, ISBN: 9780470741306.
- de Souza, L.M.S., Spiess, P., Guinard, D., Koehler, M., Karnouskos, S., and Savio, D. (2008). Socrades: A web service based shop floor integration infrastructure. In *Proc. of the Internet of Things (IOT 2008)*. Springer.
- Edwards, W.K. (2006). Discovery systems in ubiquitous computing. *IEEE Pervasive Computing*, 5(2), 70–77.
- Fleisch, E. and Mattern, F. (2005). *Das Internet der Dinge*. Springer, 1 edition.
- Guinard, D. and Trifa, V. (2009). Towards the web of things: Web mashups for embedded devices. In *2nd Workshop on Mashups, Enterprise Mashups and Lightweight Composition on the Web (MEM 2009)*. Madrid, Spain.
- Jammes, F., Smit, H., Mensch, A., Harrison, R., and Kirkham, T. (2009). Use of web services for next-generation automation systems. URL <http://epubs.cclrc.ac.uk/work-details?w=49594>.
- Jammes, F. and Smit, H. (2005). Service-oriented paradigms in industrial automation. *IEEE Transactions on Industrial Informatics*, 1, 62–70.
- Karnouskos, S., Baecker, O., de Souza, L.M.S., and Spiess, P. (2007). Integration of soa-ready networked embedded devices in enterprise systems via a cross-layered web service infrastructure. In *Proc. ETFA Emerging Technologies & Factory Automation IEEE Conference on*, 293–300. doi:10.1109/EFTA.2007.4416781.
- Marin-Perianu, M., Meratnia, N., Havinga, P., de Souza, L., Muller, J., Spiess, P., Haller, S., Riedel, T., Decker, C., and Stromberg, G. (2007). Decentralized enterprise systems: a multiplatform wireless sensor network approach. *Wireless Communications, IEEE*.
- Priyantha, N.B., Kansal, A., Goraczko, M., and Zhao, F. (2008). Tiny web services: design and implementation of interoperable and evolvable sensor networks. In *Proc. of the 6th ACM conference on Embedded Network Sensor Systems*, 253–266. ACM, Raleigh, NC, USA. doi: 10.1145/1460412.1460438. URL <http://portal.acm.org/citation.cfm?id=1460438>.