Noname manuscript No.
(will be inserted by the editor)

Realizing next generation web service driven industrial systems

Stamatis Karnouskos

the date of receipt and acceptance should be inserted later

Abstract Business continuity and agility form the core modus operandi of modern global enterprises. Complex business processes performed in highly distributed production systems need to be efficiently integrated with the shop-floor, which needs to be able to fully respond to dynamic adaptations and sophisticated interactions with the enterprise systems in a timely manner. As the new generation of industrial devices coming to the shop-floors features greatly improved storage, computing, and networking capabilities, the factory of the future transforms itself to a system of systems, where of large numbers of heterogeneous networked embedded devices dynamically exchange information, complement each-other's functionality and provide new innovative capabilities that satisfy the emergent dynamic business requirements. This new breed of networked embedded devices goes beyond simple passive roles e.g. being able to store and report information about themselves and their physical surroundings once queried, but execute complex computations and global logic locally, as well as dynamically adapt to fulfil goal-driven conditions. They communicate in an open interoperable way, form cooperative peer to peer networks and strongly interact with enterprise systems. This leads to highly modular, manageable and dynamic factories that will be able to adapt and optimize their behaviour to achieve the business goals pursued in a cross-layer collaborative way.

Keywords Service Oriented Architecture · enterprise systems · web services on devices · dynamic discovery · cooperating objects

S. Karnouskos
SAP Research, Germany (www.sap.com)
Vincenz-Priessnitz-str 1
Tel.: +496227-752515
Fax: +496227-7845799
E-mail: stamatis.karnouskos@sap.com

1 Introduction

The last decade has witnessed a deep paradigm shift on the shop-floor towards embedding Information and Communication Technologies (ICT) in all operational aspects of the factory in order to enhance its capabilities and qualitative aspects. As sophisticated networking and computation capabilities are embedded in devices used in manufacturing and process automation, it is possible to adopt modern software engineering approaches and benefit from them.

Our key message is that collaboration via an information centric infrastructure is of key importance for the future industrial systems. This cooperation has to be not only horizontal (e.g. among devices) but also vertical among devices, systems and enterprise services. Links should be created among them in an information-flat infrastructure that glues several layers directly without necessarily having to go to through the deployed hierarchies. This information driven interaction may enable rapid new application development and coupling of the real world as sensed by industrial devices to the enterprise systems. Throughout this paper we will show how such infrastructures can be realised, raise potential challenges and future directions and depict our experiences while implementing them.

1.1 Trends

Modern factory devices can provide their functionality as a collection of services that can be used by third entities for their purposes. This information-driven interaction empowers the vision of the so-called Internet of Things as described in Fleisch and Mattern (2005), according to which billions of devices interconnected will provide new innovative applications and systems as indicated by Karnouskos and Colombo (2011), that may significantly increase the efficiency of current systems.

Future shop-floors are expected to be service-oriented as envisioned in Colombo and Karnouskos (2009), where devices will offer their functionality as a service and collaborate as depicted in Camarinha-Matos and Afsarmanesh (2005). They are expected to be highly heterogeneous integrating not only shop-floor industrial-only devices as this is done today, but also a large number of modern IT devices that will be used by the employees and will need to be easily integrateable for better visibility and on-demand information acquisition. Enabled by software services, the Internet of Things provides for virtually infinite integration of sensors, actuators, microsystems, mechatronic systems, robots, mobile devices, etc.

The core idea behind the amalgamating the physical and virtual (business) world is to seamlessly gather any useful information about objects of the physical world and use the information in various applications during the object's entire life cycle. Collecting information and making it available, for example, about the objects' and goods' origin, location, movements, physical properties, usage history, and context, can help enterprises improve both existing intra- and inter-company business processes and also create new ones. Existing business processes may become more accurate since information taken directly from the point of action can be used to enhance decision-making procedures. The continuous evolution of embedded and ubiquitous computing technologies, in terms of decreasing costs and increasing capabilities, may even lead to the distribution of existing business processes not only to the network itself but also to "network edges" i.e. the advanced industrial networked embedded devices, and can overcome many limitations of existing centralized approaches.

The emerging approach is to create system intelligence by a large population of small and smart networked embedded devices at a high level of granularity, as opposed to the traditional approach of focusing intelligence on a few large and monolithic applications. This increased granularity of intelligence distributed among loosely coupled intelligent physical objects facilitates the adaptability and reconfigurability of the system, allowing it to meet business demands not foreseen at the time of design and providing real business benefits as we depict in Cannata et al (2010). It is expected that in the future business applications will heavily interact with the "real-world" via the optimal timely exploitation of the information offered by devices Karnouskos et al (2010). Monitoring and control (M&C) heavily depends on the integration of embedded systems, and is expected to grow from 188€ Bn in 2007, by 300€ Bn, reaching 500€ Bn in 2020 DECISION (2008). This will have a significant impact in several domains and especially in manufacturing and process industry.

1.2 Web Services on Devices

The use of the Service-Oriented Architecture (SOA) paradigm implemented through Web Service (WS) technologies is not new especially for high-level systems. However due to the powerful networked embedded devices we can now have the same capabilities at the device layer. Since any device can host web services as depicted in Mathes (2009); Moritz et al (2009); Karnouskos et al (2007, 2010) and make its functionality available via it, we can have integration of devices based purely on the composition of services i.e. information driven without having to focus on the devices as such and their specifics but rather only on the provided functionality. This has profound implications, especially when it comes to the interaction with enterprise systems as depicted in Yang et al (2010). Enterprise application and service designers can greatly benefit from real-world integration, however it is not feasible nor wished to engage towards tackling the heterogeneity and idiosyncrasies of devices but rather stay at abstract functional level i.e. that of services. This will enable third party service providers to provide much better and high performance implementation of their services (since they know the specifics of the device much better) and therefore enable a decoupled parallel evolution of devices and enterprise systems coupled only by standardized services; a win-win situation.

Web services on devices can act as a unifying technology, empowering all levels of the enterprise, from sensors and actuators to enterprise business processes. The benefits of service-orientation as shown in IBM (2008) are conveyed all the way to the device level, facilitating the discovery and composition of applications by re-configuration rather than re-programming. Dynamic self-configuration of smart networked embedded devices using loosely-coupled services provides significant advantages for highly dynamic and ad hoc distributed applications. The goal will require the definition of new integration concepts taking into account the emerging requirements of business applications and the explosion of available information from the device level. Of particular interest is the availability of near real-time event information, which will be used to specify new enterprise integration approaches for applications such as business activity monitoring, overall equipment effectiveness optimization, maintenance optimization, etc.

1.3 Cross-layer SOA-driven Collaboration

It is already demonstrated in detail in Karnouskos et al (2010); Colombo and Karnouskos (2009) that the future shop floor infrastructures can significantly benefit from service-oriented approaches, both in vertical (cross-layer) and horizontal communication. A vision of this is depicted in Fig. 1 where a service-enabled information driven collaboration is possible. We have pursued the realization of such a cross-layer infrastructure (as depicted in Fig. 2) within the SOCRADES project (www.socrades.eu). In such an infrastructure, new, rich services can be created by orchestrating and combining services from different system levels, i.e. services provided by enterprise systems, by middleware in the network such as the one detailed by Spiess et al (2009), and by advanced devices themselves.

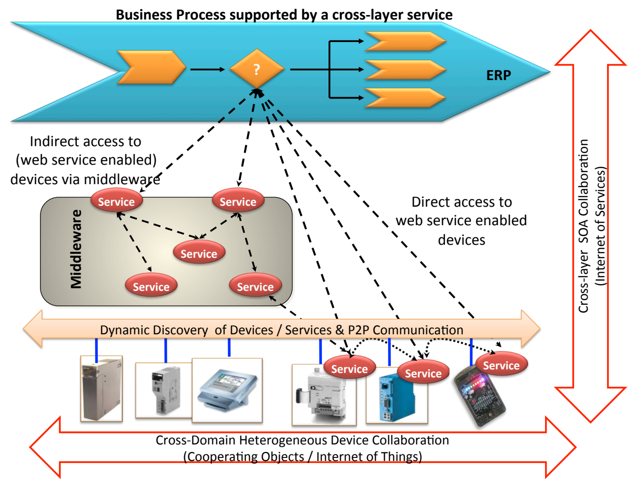

Fig. 1 Towards cross-layer collaboration

The composed services with complex behaviour can be created at any layer (even at device layer). In parallel, dynamic discovery and peer-to-peer (P2P) communication allows for easy identification of the device and its functionality. The trend is to clearly move away from proprietary connections between monolithic hardware and software systems towards more autonomous systems that interact in a more standardized, cooperative and open way. Entities (devices, applications, etc.) can not only communicate in a cross-layer way but heavily collaborate in mash-ups as envisioned by Colombo and Karnouskos (2009). Aggregated device-level services interact with higher-level business processes situated at the level of business applications – in particular Enterprise Resource Planning (ERP) systems – in order to demonstrate seamless integration of device level functionality into higher-order business application scenarios as demonstrated by Karnouskos et al (2010).

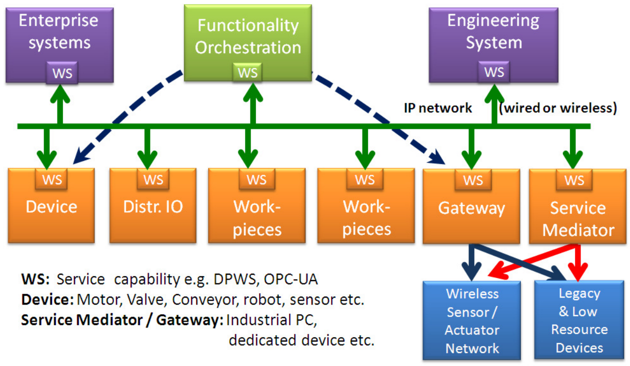

Fig. 2 SOCRADES Vision: SOA based integration of the future factory shop-floor

We have to point out that this approach flattens the currently hierarchical structures within factories, enabling for instance an Enterprise service to act as a "virtual device" on the shop floor and vice-versa a device to be directly integrated in an enterprise service. This is a very powerful and innovative concept that gives more flexibility to integrate on-demand in a lightweight manner, without having to go through the whole predefined hierarchies and painful integration vendor lock-ins.

The convergence of applications and products towards the SOA paradigm improves shop floor integration and transparency, thereby increasing reactivity and performance of the workflows and business processes commonly found in manufacturing and logistics. Events become available to any entity of the system as they happen, and business-level applications can exploit such timely information for purposes such as diagnostics, performance indications, or traceability. While these vertical collaborations are beneficial for business application software, new challenges arise: direct communication with devices can be error prone or unreliable, which must be considered when critical decisions, such as branches in a workflow, depend on it.

2 Amalgamating the Enterprise and shop floor landscapes

Significant effort has been invested into the integration of physical computing devices with standard enterprise software, such as Enterprise Resource Planning (ERP) systems. Planning a production order or creating a bill of materials in the ERP application is neither effective nor optimized, unless the shop floor is transparent to those applications. As an example, the manufacturing industry foresees enterprise applications to consider real time events on the shop floor to plan production, enhance customer relationship management, and have a healthy updated supply chain. This shop floor intelligence obtained in real time allows business to adapt to the market demand and forecast shop floor breakdowns in a timely fashion. Additionally as SOA approaches start to prevail, the introduction cycle of new applications could be significantly shorter. This could enable exchange of real time information across enterprises and trusted business partners, which will have an effect on the respective business decisions.

In order to realize the collaboration of the enterprise and shop floor systems three main activities have to be performed:

- Identification of the cooperating entities (systems): the identification of the collaborative automation units that are able to expose and/or consume services, for each production scenario in a defined production domain, e.g., electronics assembly, manufacturing, continuous process, etc. A collaborative unit can be a simple intelligent sensor or a part/component of a modular machine, a whole machine and also a complete production system.
- Building the system of systems: networking/bringing the entities together within a SOA or collaborative infrastructure, i.e., putting the units architecturally together, and
- Making the system working for reaching the production goal: collaborative behaviour of the systems for reaching common objectives, i.e., control objectives, production specifications, markets objectives, etc.

It is clear that cooperation would be beneficial for next generation production systems if it is done among devices as envisioned by Marrón et al (2011) in the shop floor and among the devices and the enterprise systems as advocated by Camarinha-Matos and Afsarmanesh (2005). In order to achieve that, several challenges need to be tackled and requirements to be met as already identified in (Karnouskos et al (2011)). Seamless cooperation and collaboration is necessary to realize adaptive production systems.

Key issue in all of these approaches is to be capable of integrating any device, without taking direct consideration of its device-dependent characteristics; hence we aim at hiding its heterogeneity and focus only on its functionality . As such, this abstract integration of devices based on web services can be realized as we have demonstrated; details can be found in Karnouskos et al (2007, 2010)). This may indeed result to significant benefits as we also show in an initial evaluation by Cannata et al (2010). Such benefits include cost reduction, improved performance and new opportunities but also potential drawbacks.

2.1 Tackling Device Heterogeneity with SOA

Current infrastructures are highly heterogeneous and a tremendous amount of effort has been invested in dealing with it. In practice this means to try to stick to specific vendors and products, and also invest in "glue" solutions that enable the cross-bridging of information. Several standards are used, but when investments are made with a 10 − 15 year horizon it is hard to predict the prevailing standards to use and invest accordingly. Today we still do not have long-lasting future-compatible solutions. This is critical as future factory shop-floor heterogeneity will increase drastically especially with respect to the networked embedded devices. However, using web services this heterogeneity can be hidden empowering information-driven integration approaches (and not device driven as done today).

2.1.1 Device Cartography

Device integration based on SOA has already been prototyped within the scope of the SOCRADES project. However today web services on devices, have some non negligible requirements on memory, computational power and storage. It might be that these do not pose a barrier in the future, however they pose a barrier in the short term and for devices with older technologies that can still be found on the majority of shop-floors. As such migration solutions need to be found, since the lifecycle of such infrastructures in some domains can be decades, and business continuity as well as return of investment (ROI) must be guaranteed.

As pointed out not all devices are expected to have in the future the resources to run web services natively. Even if that was feasible it might not really be reasonable (or offer any business advantages) in some cases as task-specific devices may assume a very specialized role, that can be fully accomplished with no advanced capabilities (e.g. a proximity sensor, and RFID tag etc.). Generally we can see different device categories in the future factory:

- *Passive non-electronic devices*: These are devices physically not capable of hosting web service technology as they do not feature any electronic capability. These passive devices can be monitored indirectly, e.g. via attached RFID or barcode tags. The tags themselves (e.g. a wireless sensor) or wrappers around them could make it possible for such devices to participate in the future factory e.g. as depicted in Wang et al (2011).

– *Resource constrained devices*: These devices have the power to communicate and process information. However, their resources are so limited that it would not be feasible or reasonable business-wise to deploy web services on them. In this situation, however, it seems very rewarding to connect them to gateway or service mediator that encapsulates the device's functionality and offers web services to the outside world (as depicted in Fig. 4). Typical such devices are today the RFID tags where the RFID reader is used as a Gateway depicting full-WS capabilities. Other lightweight alternatives might offer better integration for these devices e.g. via REST (Liu and Connelly (2008)).
– *WS-capable devices*: These devices have enough computing resources to retrieve, store, compute, and transmit information and can stand-alone participate in the future factory infrastructure.

2.1.2 Web Service driven Device Integration

Our focus for the future factory is on the service enabled devices. Business applications will need to access device data and state preferably always through (web) services. Although other connection operations exist, only the (web) service abstraction delivers a message-oriented asynchronous communication method truly independent from the underlying operating system and programming language. The basic need is to find at which level of the architecture those complex services are provided in a form that is ready for consumption by the enterprise system. The issue should not be discussed only bottom-up, i.e. derived from the level of functionality that automation devices can offer, but primarily top-down, i.e. defined by the data exchanged between the enterprise applications and the shop floor. The enterprise system needs direct access to timely and context specific information; as such many unnecessary details should be hidden and an abstract service should capture only the desired functionality. The last is possible usually as a composition of other more generic services.

A key issue is the integration of legacy devices, as any transition to the envisioned future factory will have a migration phase. Replacement of legacy infrastructure will come gradually and therefore transition approaches need to be defined. In parallel we must guarantee co-existence of legacy and future infrastructure as well as minimization of downtimes and media breaks. As such the only viable option is an evolutionary approach, where non-WS-enabled devices are software-updated to include a web service stack (if technically feasible), or be replaced one after the other, because they have reached the end of their lifetime, or new functionality on the physical level is required. In this way, WS-enabled devices can gradually replace the conventional systems of today.

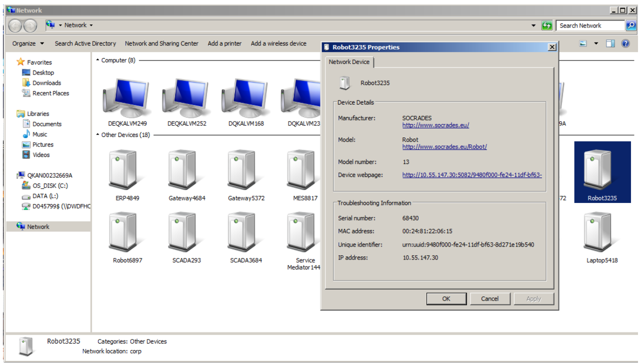

Fig. 3 Example of dynamic discovery and presentation of a Web Service enabled device in MS Windows

In Fig. 3 an example of a web service enabled device is depicted. Its functionality has been wrapped with web services (i.e. using the DPWS stack) and put on the network. Existing IT systems such as a computer hosting COTS Windows Vista/CE/7, can dynamically discover the device (due to WS-Discovery in DPWS profile), and see its data such as serial number, MAC address, IP address, model number, Unique ID (UUID), etc. Furthermore there is a standard way to access the functionality on the device and e.g. control it, or obtain its health status. Since now the device can provide this information in a standardized way via web services, other devices or services e.g. in a maintenance platform, can subscribe to the events it creates. The last depicts a clear paradigm shift towards an event-driven infrastructure, where information can be dynamically discovered and pushed to the interested parties only. This may have profound implications on a number of scenarios e.g. remote maintenance as depicted by Sekar et al (2011).

In Karnouskos et al (2007); de Souza et al (2008) we proposed an extensible integration architecture based on Web Services and capable also of supporting legacy and web-service enabled devices and products. The approach was realized with the help of infrastructure services implemented in a middleware which is described in detail in Spiess et al (2009), that would enable the device-to-business integration of a variety of devices and systems existing in factories today. Several prototypes following all these approaches have been demonstrated and evaluated some of which are depicted in Karnouskos et al (2010).

2.2 Migration to Service-driven Production Systems

Migrating towards a fully service based infrastructure may be the key to unleash the potential for sophisticated monitoring and control in the future heterogeneous factory shop floor. A gateway or a service mediator may be a viable approach for the migration phase of existing systems to the future ones. The concept behind them is depicted in Fig. 4. While gateway implementations may be straightforward, the more sophisticated one i.e. the service mediator, may serve us better in the longer run.

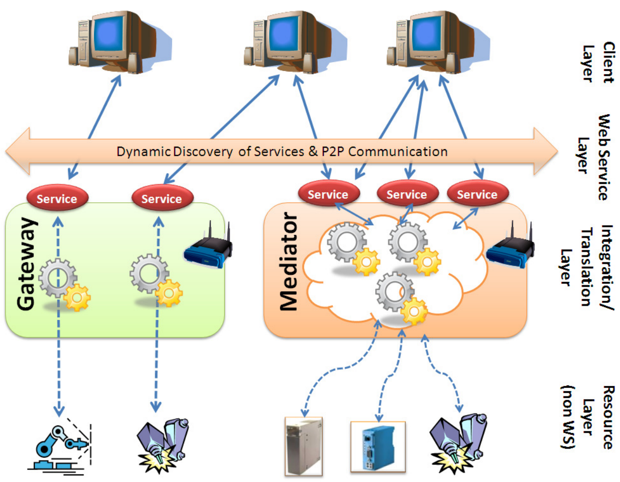

Fig. 4 Non-service-enabled device integration: Gateway vs. Service Mediator concepts

2.2.1 Gateway

A Gateway is a device that controls a set of lower-level non-service-enabled devices, each of which is exposed by the Gateway as a service-enabled device. This approach allows to gradually replace limited-resource devices or legacy devices by natively WS-enabled devices without impact on the applications using these devices. This is possible since the same web service interface is offered this time by the WS-enabled device and not by the Gateway. This approach is used when each of the controlled devices needs to be known and addressed individually by higher-level services or applications.

The Gateway approach requires some specific support e.g. from a DPWS implementation. Indeed, while a standard DPWS-enabled device is only required to store and manage its own discovery, description and hosted services metadata, a Gateway needs to support a multitude of devices. It is therefore necessary to introduce a registry for devices and hosted services that helps structure and manage the required information. When several instances of the same device type are present, the registry distinguishes between class- and instance-level information, both for devices and hosted services, so as to factor the information common to all instances and thus to save also resources e.g memory.

2.2.2 Service Mediator

Originally meant to aggregate various data sources (e.g. databases, log files, etc.), the Service Mediator components evolved and are now used to not only aggregate various services but possibly also compute/process the data they acquire before exposing it as a unified service. Service Mediators aggregate, manage and eventually represent services based on some domain specific semantics (e.g. using ontologies).

In our case the Service Mediator could be used to aggregate various non WS-enabled devices. In this way, higher level application could communicate to Service Mediators offering WS, instead of communicating to devices with proprietary interfaces. The benefits are clear, as we don't have the hassle of (proprietary) driver integration. Furthermore now processing of data can be done at Service Mediator level and more complex behaviour can be created, that was not possible before from the standalone devices.

2.2.3 Reality check: Gateway vs. Service Mediator

Service Mediators can be used instead of simple Gateways whenever we want to introduce some low-level semantics and multiplex functionality. Consider, for example a wireless sensor network monitoring temperature along a conveyor belt (shop floor). Such a network can be composed of tenths of temperature sensors, yet, the interesting service on the top floor is not the services offered by each and every sensor but rather the average temperature on the conveyor belt. A first prototype demonstrating this concept has been realized as depicted in Savio et al (2008).

As shown on Fig. 4 using Service Mediators introduces another level of abstraction and aggregation between the clients and devices. Thus, seen from the outside, there might not be significant difference between a Service Mediator and a composite service that relies on a set of service-enabled devices.

A Service Mediator is a device that controls a set of lower-level non-service-enabled devices that realise a process which is exposed as a service interface. Thus the individual lower-level devices are invisible outside of the Service Mediator. On the contrary the Gateway depicts as a service functionality, that can be directly related to a specific device; hence one can directly relate and identify an explicit device as the source of a service offered by the Gateway (which might not be possible in the Service Mediator).

Both the Gateway and the Service Mediator can host several services – limited only by their internal resources. Both approaches enable the functionality of the shop floor to be more accessible and tap into an event based infrastructure where devices (indirectly via their proxy) and functionality can be dynamically discovered as depicted by Edwards (2006) and used e.g. due to the WS-Eventing support of DPWS.

To have the two classes of systems (the SOA-enabled and the conventional ones) communicate with each other the new devices could come with a dual interface, providing both WS services as well as some other (e.g. vendor-specific/proprietary) protocol. This option is very useful for the migration phase. As soon as a significant part of the production environment has these dual-stack interfaces, the whole system could be re-configured and service interaction could be used as the single interaction method between devices. This switch to SOA-based control requires the remaining non-WS devices to be integrated in the new system. The preferred way of integration is to have proxy WS devices and services for the real devices. As a rule of thumb, the proxies should be as close as possible (in terms of network distance) to the real devices and should be instantiated at the lowest possible layer of the system.

Generally, the lower the level at which a non-WS-enabled device is wrapped into a WS-compliant web service, the more flexibly it can participate in the device SOA compositions. The lowest feasible level would be the PLC (assuming of course that no network adapter is on the device) that could have in addition to its cyclic, real-time, control part, a second part that hosts the WS stack (or just can speak *http* protocol in case of REST). This very low-level integration is however very costly and would require to re-design or introduce significant changes in PLC devices. Therefore it is practically preferable to add WS support using single-board computers or complete Gateways implemented on industrial PCs.

3 Demonstration and lessons learned

As we have discussed so far Web Services pose a promising approach towards tackling heterogeneity, and unleashing the full power of production system not only at individual level but in collaboration with other devices, and enterprise systems. We believe that the latest advances may enable us to import concepts and results from the IT domain such as service composition and apply it directly to the factory of the future e.g. service-based composition of devices and systems. This would further lead towards strong integration of physical and software based systems which may further stretch our capabilities as well as provide new innovations, mainly driven by cross-layer collaboration and on-demand information acquisition. We consider this a critical factor to move towards collaborative industrial systems as also depicted in Lee and Kim (2007).

3.1 Implemented Demonstration Scenario

In order to demonstrate the flexible integration and collaboration, we consider having a multi-site service-oriented enterprise in which the assembly of electromechanical components is performed in two geographically distributed assembly systems (both of them are similar to the system depicted in Fig. 5) and production orders could be allocated to different sites. This allocation/re-allocation of orders is done as an evaluation result of the best production facility available at the moment when a production request is made, or when - due to external factors - the production should be shifted to a different location (e.g. for performance reasons, maintenance etc.).

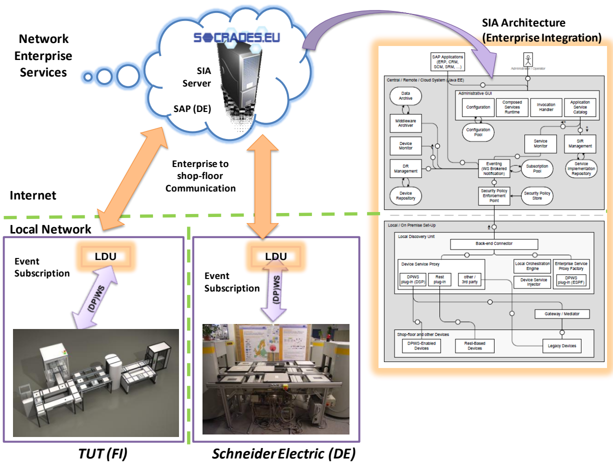

Fig. 5 Cross-location integration of service enabled device discovery and integration via SIA

The scenario depicted in Fig. 5 constitutes one of the trials designed, implemented and evaluated within the scope of the SOCRADES project by SAP, Schneider Electric and Tampere University of Technology. The scenario assumes that similar production facilities are available in remote locations (e.g. Schneider Electric in Germany or Tampere in Finland). The prototypes developed and hosted in Tampere (TUT) and Seligenstadt (Schneider Electric) represent two different companies that are linked with business relations. These companies are "inter-connected" via the Enterprise Applications that are hosted in Walldorf, Germany (SAP). All communication goes over the public Internet infrastructure while the in-cloud based enterprise services coordinate and mediate among the physical locations.

Both facilities provide electromechanical assembly capabilities as envisioned by the SOCRADES architecture; this means that the components of the production systems in these locations are abstracted/wrapped and perceived externally as one or more web services. At local level, each one of the facilities acts independently and can coordinate its service-enabled production system by using on-site tools. At global level, both facilities connect to a service-enabled ERP module provided by SAP, which is used for coordinating the production in the remote locations. A network application (named LDU) is downloaded over the Internet and once instantiated it immediately provides discovery of devices and services (via the device profile for web services- DPWS) on the local network and connection to the backend system via the SOCRADES Integration Architecture (SIA) middleware as we describe in detail in Spiess et al (2009). Different versions of the LDU can add-up functionalities, e.g., proxy also specific enterprise services at the local shop-floor, where they can be discovered and used by the devices and other services. LDUs provide a means for connecting and managing devices from different premises, without needing virtual private network connections to SAP premises.

The LDUs can discover local services and can interact directly with production execution systems exposed as WS. This is a typical example of hosted functionality on a network server at the provider side, where business services are being implemented/updated and the remote sites (in our trial Tampere and Seligenstadt) can interact over the network, with only minimal installations at their side (in order to interact with the business services). In this specific case, the software that interconnects each site is downloaded on the fly over the network via an one-click operation from a web browser (e.g. Firefox, Internet Explorer, Chrome etc.). It is critical that the device that instantiates the LDU has network access so that the discovery can succeed in identifying the necessary devices and services.

The different sites involved are collaborating between them via interactions that previously were not possible or would require significant implementation efforts. All communication is event-driven with the enterprise services acting as the main interaction point. This implies that status of services, dynamic information etc. are delivered via the pub/sub model to the interested parties and decision-making processes can be started as soon as situations arise. As can be seen, this is an event-based approach where all sites are notified about the necessary status of the production in the other side, and where the enterprise systems have full visibility on the production and can re-arrange orders in order to meet business goals. Hence our approach can act as enabling component to realise complex collaborative manufacturing e.g. as envision by Chae et al (2007).

3.2 Implementation Technologies

In order to demonstrate the Cross-layer service-driven collaboration and management of next generation production systems, we have implemented and demonstrated in the scope of SOCRADES project a prototype involving three different geographical locations that collaborate towards realising a virtual distributed shop-floor.

As depicted in Fig. 5, in order to achieve our goals we have

- implemented a middleware hosting infrastructure (web) services
- implemented selective enterprise services and exposed them as DPWS devices on the shop-floor
- exposed the functionality of shop-floor devices (e.g. sensors, robots etc.) as web services (implemented directly on-device or with the usage of gateways and service-mediators).

The developed middleware facilitates infrastructure services such as communication between shop-floors as well as integration of the production systems with enterprise services. All SIA components are independent and communicate through web services both with each other and to business systems. Therefore a networked device can connect to the SIA and directly participate in business processes while SIA hides the details of the underlying hardware.

As shown also on the right side of Fig. 5, SIA is split in two parts: a "local" part which runs on the on-site premises and features a Local Network Discovery and Management Unit (LDU) and is running at the local network that contains the devices to be integrated, and a central system (anywhere on the network or the Internet) that hosts enterprise-level applications. Many LDUs (e.g. one for each shop floor location) can connect to a central system. In the local subsystem at Device Layer there are several embedded devices that are running various services. SIA is able to interact with devices using several communication protocols.

SIA allows applications to subscribe to any events sent by the devices, offering a publish/subscribe component that supports WS-Notifications. It also offers buffered invocations of hosted services on devices that are only intermittently connected. When the device becomes available again a notification is sent to the client or the system caches an invocation message and delivers it when the device is ready to receive it.

On the device side, the implementation has been done by either implementing web services on the device or using service mediators and gateways in order to capture the functionality of several devices, including (partially only) the on-demand translation e.g. from OPC-UA to DPWS. All of the devices were discovered in a timely manner, despite of their different DPWS stack implementations that were used i.e. the WS4D (www.ws4d.org) and SOA4D (www.soa4d.org). This resulted in three different implementations i.e. WS4D (Java) and SOA4D (Java and C) coexisting in the trial.

3.3 Lessons Learned

The biggest issue that became apparent in the trial was that of interoperability. Although discovery was fully working, the same did not hold true for the event subscription. We solved this in some cases by utilizing a dual-stack approach i.e. implementing both SOA4D and WS4D stacks in a device in order to enable event subscription without problems from any client. Furthermore due to the stack-specific implementation of SOA4D, services offered by devices should be a priori known which was limiting. All in all, we were able to demonstrate seamless device discovery, integration on the enterprise system, interaction with the enterprise services and cross-layer collaboration and information flow. SIA has been proven in the trial an adequate way to provide cross-layer and timely SOA integration.

Integration of heterogeneous devices, especially their interaction with enterprise services in a timely manner, is a challenging research topic. Its impact in the future Internet infrastructure and its services is expected to be significant and critical to the success of a new generation of applications that will merge the real and the business world. However, we can report positive experiences in managing the infrastructure and especially with respect to asset management where devices are dynamically discovered and access to their metadata is immediate (even cross-location) reduces the need for static configurations and human error.

In traditional IT architectures, business process activities, applications, and data are locked independently and often incompatibly. Users have to navigate separate networks, applications, and databases to conduct the chain of activities that completes a business process. This absorbs an excessive amount of IT budget and staff time to maintain. Additionally, business demands enhancing reliability and real-time performance in wireless technology. Thus production systems require reconfigurability and flexibility in order to improve the efficiency of manufacturing products. The SOCRADES approach targets these business needs by utilizing the SOA paradigm at the device level, which enables the adoption of a unifying technology for all levels of the enterprise. This can enable a wide range of potential business opportunities but is also associated with challenges as we identify in Cannata et al (2010).

SIA allows enterprise applications to connect to devices for monitoring and management (soft control) using open standards. It features several advanced services in order to free business application developers from the complexity of interacting with a highly heterogeneous and unreliable infrastructure. It also helps discovering and keeping track of devices and can manage the embedded software if that is supported by the devices. Additionally, legacy devices can be included by providing proxy services for them. SIA is not build with a specific domain in mind, but in the scope of the SOCRADES project we have successfully demonstrated its capabilities in industrial automation domain. An overview of various other efforts to demonstrate the cross-layer architectural integration can be found in Karnouskos et al (2010), where we show how this can be applied in various industrial scenarios.

We have shown that the followed approach targets emerging business needs as depicted in Zayati et al (2010) by utilizing the SOA paradigm at the device level, which enables the adoption of a unifying technology for all levels of the enterprise. This can enable a wide range of potential business opportunities but is also associated with challenges which have already been identified in Cannata et al (2010), and Gao et al (2009).

Organizations that benefit most from adopting the full-blown SOA based approaches such as the one we have depicted here, have large and complex application portfolios with a vast quantity of point-to-point interfaces like manufacturing companies, since the more complex applications and their integration architectures become, the more risky it is to change them. With a high level of complexity, the impact of change cannot be assessed sufficiently, hence test cycles become longer and more defects occur during the production process. Thus the system change cannot keep pace with business changes and the organization can not maintain its competitive edge: the business cannot respond quickly enough to changes in market demand. This can be prevented when adopting SOA based architectures as we analyse in greater detail in Cannata et al (2010).

4 Conclusion

It is clear that we are heading towards an infrastructure where heterogeneity will be dominant and not all devices will have the capability of tapping directly to the future SOA dominated shop floor by implementing web services natively and provide 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 the business point of view. Therefore any approach proposed for the future industrial automation domain, has to make sure that all types of devices can be directly and indirectly integrated in a global communication infrastructure.

Ubiquitous, SOA-based device integration leading to interaction of devices with enterprise services in a timely manner is an important vision that creates substantial impact both from the perspective of research and industrial application. The paradigm of services on every layer of the network will influence the structure and operation of future factory. Dynamic service discovery and composition will enable a new generation of applications that will more closely couple physical environments and processes with the corresponding models in business software, which are their virtual counterparts. Collaboration will emerge as a key behaviour as already identified by Camarinha-Matos and Afsarmanesh (2005); Marrón et al (2011) and a new breed of industrial applications will be possible.

We have presented the imperatives and motivation for more dynamic and flexible production lines in the factory of the future. To ease legacy infrastructure transition to the SOA shop-floor we have shown how this can be realized via a Gateway or a Service Mediator approach. The overall architecture and its components have been designed while taking into account a wide set of requirements, as well as the existing service bridging concepts and technologies. With increasing integrated collaboration, more information exchange and cross-layer communication, business continuity can be achieved. The factory of the future will entail more dynamic and adaptive production equipment that will closely collaborate with each other through real-world services and adjust their behaviour dynamically. Future enterprise applications will heavily depend on these provided that security, scalability and real-time issues are adequately tackled.

Acknowledgements The author would like to thank the partners of the EU project SOCRADES (www.socrades.eu), for the fruitful discussions.

References

Camarinha-Matos LM, Afsarmanesh H (2005) Collaborative networks: a new scientific discipline. Journal of Intelligent Manufacturing 16:439–452, URL http://dx.doi.org/10.1007/s10845-005-1656-3, 10.1007/s10845-005-1656-3

Cannata A, Karnouskos S, Taisch M (2010) Evaluating the potential of a service oriented infrastructure for the factory of the future. In: 8th IEEE International Conference on Industrial Informatics (INDIN), pp 592 – 597, DOI 10.1109/INDIN.2010.5549675

Chae H, Choi Y, Kim K (2007) Component-based modeling of enterprise architectures for collaborative manufacturing. The International Journal of Advanced Manufacturing Technology 34:605–616, URL http://dx.doi.org/10.1007/s00170-006-0620-5, 10.1007/s00170-006-0620-5

Colombo AW, 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

DECISION (2008) Monitoring and control: today's market, its evolution till 2020 and the impact of ICT on these. European Commission DG Information Society and Media, URL http://www.decision.eu/smart/SMART_9Oct_v2.pdf

Edwards WK (2006) Discovery systems in ubiquitous computing. IEEE Pervasive Computing 5(2):70–77, DOI http://doi.ieeecomputersociety.org/10.1109/MPRV.2006.28

Fleisch E, Mattern F (2005) Das Internet der Dinge. Springer

Gao J, Yao Y, Zhu V, Sun L, Lin L (2009) Service-oriented manufacturing: a new product pattern and manufacturing paradigm. Journal of Intelligent Manufacturing pp 1–12, URL http://dx.doi.org/10.1007/s10845-009-0301-y, 10.1007/s10845-009-0301-y

IBM (2008) SOA in manufacturing guidebook. URL ftp://ftp.software.ibm.com/software/applications/plm/resources/MESA_SOAinManufacturingGuidebook.pdf

Karnouskos S, Colombo AW (2011) Architecting the next generation of service-based SCADA/DCS system of systems. In: 37th Annual Conference of the IEEE Industrial Electronics Society (IECON 2011), Melbourne, Australia.

Karnouskos S, Baecker O, de Souza L, Spiess P (2007) Integration of soa-ready networked embedded devices in enterprise systems via a cross-layered web service infrastructure. In: Proc. of the IEEE Emerging Technologies and Factory Automation Conference (ETFA), pp 293–300, DOI 10.1109/EFTA.2007.4416781

Karnouskos S, Savio D, Spiess P, Guinard D, Trifa V, Baecker O (2010) Real world service interaction with enterprise systems in dynamic manufacturing environments. In: Benyoucef L, Grabot B (eds) Artificial Intelligence Techniques for Networked Manufacturing Enterprises Management, ISBN 978-1-84996-118-9, Springer

Karnouskos S, Vilaseñor–Herrera V, Haroon M, Handte M, Marrón PJ (2011) Requirement considerations for ubiquitous integration of cooperating objects. In: 4th IFIP International Conference on New Technologies, Mobility and Security, 7–10 February 2011, Paris, France

Lee J, Kim K (2007) A distributed product development architecture for engineering collaborations across ubiquitous virtual enterprises. The International Journal of Advanced Manufacturing Technology 33:59–70, URL http://dx.doi.org/10.1007/s00170-006-0443-4, 10.1007/s00170-006-0443-4

Liu Y, Connelly K (2008) Realizing an open ubiquitous environment in a restful way. In: Proc. IEEE International Conference on Web Services ICWS '08, pp 96–103, DOI 10.1109/ICWS.2008.64

Marrón PJ, Karnouskos S, Minder D, Ollero A (eds) (2011) The emerging domain of Cooperating Objects. Springer, URL http://www.springer.com/engineering/signals/book/978-3-642-16945-8

Mathes M (2009) Time-constrained web services for industrial automation. PhD thesis, Philips Universität Marburg, Germany, URL http://archiv.ub.uni-marburg.de/diss/z2009/0466/

Moritz G, Zeeb E, Prüter S, Golatowski F, Timmermann D, Stoll R (2009) Devices profile for web services in wireless sensor networks: adaptations and enhancements. In: Proceedings of the 14th IEEE international conference on Emerging technologies & factory automation, IEEE Press, Piscataway, NJ, USA, ETFA'09, pp 43–50, URL http://portal.acm.org/citation.cfm?id=1740954.1740961

Savio D, Karnouskos S, Wuwer D, Bangemann T (2008) Dynamically optimized production planning using cross-layer soa. In: Proc. 32nd Annual IEEE International Computer Software and Applications COMPSAC '08, pp 1361–1365, DOI 10.1109/COMPSAC.2008.219

Sekar R, Hsieh SJ, Wu Z (2011) Remote diagnosis design for a plc-based automated system: 1-implementation of three levels of architectures. The International Journal of Advanced Manufacturing Technology pp 1–18, URL http://dx.doi.org/10.1007/s00170-011-3314-6, 10.1007/s00170-011-3314-6

de Souza LMS, Spiess P, Guinard D, Koehler M, Karnouskos S, Savio D (2008) Socrades: A web service based shop floor integration infrastructure. In: Proc. of the Internet of Things (IOT 2008), Springer

Spiess P, Karnouskos S, Guinard D, Savio D, Baecker O, Souza LMSd, Trifa V (2009) Soa-based integration of the internet of things in enterprise services. In: IEEE International Conference on Web Services, ICWS 2009 , Los Angeles, CA, USA, pp 968–975, DOI 10.1109/ICWS.2009.98

Wang B, Cao Z, Yan Y, Liu W, Wang Z (2011) Fundamental technology for rfid-based supervisory control of shop floor production system. The International Journal of Advanced Manufacturing Technology pp 1–19, URL http://dx.doi.org/10.1007/s00170-011-3358-7, 10.1007/s00170-011-3358-7

Yang J, Li J, Deng X, Xu K, Zhang H (2010) A web services-based approach to develop a networked information integration service platform for gear enterprise. Journal of Intelligent Manufacturing pp 1–12, URL http://dx.doi.org/10.1007/s10845-010-0477-1, 10.1007/s10845-010-0477-1

Zayati A, Biennier F, Moalla M, Badr Y (2010) Towards lean service bus architecture for industrial integration infrastructure and pull manufacturing strategies. Journal of Intelligent Manufacturing pp 1–15, URL http://dx.doi.org/10.1007/s10845-010-0401-8, 10.1007/s10845-010-0401-8