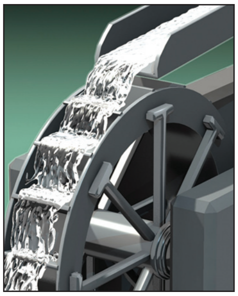

Plug-and-Play Virtual Factories

Stefan Schulte • *Vienna University of Technology*

Dieter Schuller and Ralf Steinmetz • *Technische Universität Darmstadt*

Sven Abels • *Ascora*

Although virtual factories are a well-established approach to managing distributed, cross-organizational manufacturing processes, corresponding end-to-end IT support is still lacking. Service-oriented virtual factories use and extend well-known methods and technologies from service-oriented computing, the Internet of Things, and business process management to establish, manage, monitor, and adapt virtual factories in a plug-and-play-like fashion.

In today's globally connected economy, manufacturing companies face increasing pressure to quickly adapt to market demands and minimize time-to-market for new products. To preserve competitiveness under these conditions, efficiency and fast adaptation in manufacturing processes are key.

One approach that's gained considerable momentum is to combine several independent factories into virtual factories. Cross-organizational manufacturing processes in virtual factories are directed toward a specific goal — that is, producing a particular product. Processes have a modular structure and can contain several building blocks, or *process steps*, that different partners carry out at different locations. Following the virtual factory approach, manufacturing processes are modeled and executed as if they're being carried out in a single factory. For this, IT support is essential to success.

Although some approaches exist for establishing virtual factories, current solutions would benefit significantly from functionalities besides partner-finding and factory-building. These include addressing virtual factories on a deep technical level (albeit one hidden from users) and providing the means to easily adapt such factories to changing requirements and circumstances. The different building blocks of a virtual factory must be connected, and organizations must exchange information across borders. Still, no standardized technologies provide the necessary end-to-end integration that would represent the idea that "In a real virtual factory, the network is the factory."[1]

Virtual factories become real only if companies meet requirements for creating and executing flexible manufacturing processes. Process design, optimization, execution, maintenance, and monitoring must be supported in a plug-and-play fashion. At present, manufacturers are missing the appropriate real-time information that would let them assess process status — for example, no information is available in their IT systems about the status of certain product parts. The problem here is the missing interoperability between business partners' IT systems as well as the potential loss of information due to prevailing data silos. Instead of receiving information about the complete supply chain, process owners can often only monitor a particular part of the process. This lack of knowledge and control also prevents scalability and flexibility in manufacturing because corresponding processes must be adapted manually rather than reacting automatically to, for example, production delays in one part of a virtual factory.

 1089-7801/12/$31.00

Being process- and communication-driven, virtual factories resemble common aspects of service-based workflows. Thus, using concepts, technologies, and methods from service-oriented computing (SOC) and business process management (BPM) seems to be a natural approach. Manufacturing processes can be modeled from individual manufacturing services from different business partners; this corresponds directly to how we compose software and human-provided services in service-based workflows. Here, we investigate this concept of *service-oriented virtual factories* in more detail.

Roles

Comparable to the famous service-oriented architecture (SOA) triangle,[2] which defines the essential roles *service requester*, *service provider*, and *service registry*, we can categorize the parties involved in service-oriented virtual factories into three roles:

- *Service providers* offer a particular physical service in terms of factory capacity.
- *Service consumers* use services that other parties offer.
- The *virtual factory broker* controls and governs the virtual factory. The broker needs a corresponding knowledge base (repository) regarding services that providers offer.

Organizations involved in virtual factories might assume different roles simultaneously. Service consumers can use provider-offered services, add value, and re-offer the value-added services, hence acting as providers themselves. Furthermore, in most cases, one service consumer within a virtual factory will be the broker — that is, the "owner" of the virtual factory. Furthermore, the broker could also be a service provider.

Conceptual Overview

Using Web service technologies and SOAs as a communication means and architectural approach within the manufacturing domain isn't novel. However, service-oriented virtual factories take this approach one essential step further, going beyond merely applying technologies for communication and information exchange. Rather, they provide a holistic approach in which services represent all manufacturing steps. SOAs let companies describe, provide, and use software- and human-provided services that encapsulate certain, well-defined functionalities and can be composed into service-based workflows; likewise, service-oriented virtual factories represent single manufacturing steps from different stakeholders as physical services. A virtual factory broker can use these services to model manufacturing processes, assembling products based on the skills and outputs of real factories from different business partners.

Process Definition

Process definitions in service-oriented virtual factories are blueprints — that is, a process template is composed of abstract process steps. Rather than identifying concrete physical services during process design time, the virtual factory broker defines these steps based on necessary skills and technical requirements. Brokers can reuse and easily adapt processes to create new manufacturing processes and therefore products by generating new process templates from existing ones. Furthermore, a broker can model different variants of the same product as manufacturing processes by exchanging the actual physical services prior to or during process execution (late binding).

Apart from the functional requirements that single services must fulfill, a process template also contains information about nonfunctional requirements for single process steps or the complete manufacturing process. Nonfunctional requirements can include necessary quality-of-service (QoS) levels and cost limits.

Process Execution

To generate process instances from process templates, the virtual factory management software (see Figure A in the sidebar) automatically chooses concrete physical services in terms of factory capacities for the previously defined abstract process steps. It can do this either on a long-term basis, if the broker wants to use the same service providers in every process instance, or ad hoc, if it's best to choose service providers individually for each process instance. The second approach is more dynamic, and two instances of the same process might differ substantially. In contrast to short-term virtual factories, long-term ones are quite stable, and corresponding process instances will be similar with regard to their underlying physical services.

In either case, the virtual factory broker can choose suppliers and their services manually or integrate them (semi-)automatically based on descriptions of a supplier's capabilities and offered services. Unlike with common service-based workflows, the system might invoke several physical services for the same process step. This would be necessary if a single supplier couldn't provide the production capacities needed for a particular step.

Runtime selection and service replacement allows the system to react to unexpected events — such as production downtimes at a particular factory or an increase in market demand — by exchanging or adding new factory capacities in terms of physical services.

Research Challenges

Even though service-based workflows and service-oriented virtual factories are based on common principles and concepts, we must address

Example Service-Oriented Virtual Factory

Figure A shows the most important aspects of a service-oriented virtual factory as presented within the conceptual overview and based on a simplified example from the automotive industry:

A manufacturing process is at the core of the virtual factory run by JD-Company (in other words, this company is the virtual factory broker). For simplicity's sake, the figure shows a sequential manufacturing process. However, processes can incorporate arbitrary workflow patterns such as parallel splits, exclusive choices, or loops.[1]

To instantiate the process, the virtual factory management software assigns physical services in terms of factory capacities to each defined process step. These services must meet previously defined functional and nonfunctional requirements. The software conducts monitoring by integrating corresponding data from partners' existing IT systems (supplier 2) or getting data from smart objects (indicated by the sensor nodes depicted for supplier 1 and JD-Company). Finally, service-level agreements (SLAs) exist for each physical service.

In this example, JD-Company produces car seats by setting up a manufacturing process based on the different parts — the frame, upholstering, and cover. During process definition, the company defines part production as well as the final assembly as abstract process steps. Prior to process execution, the virtual factory broker selects corresponding physical services from a service registry and assigns them to the single process steps.

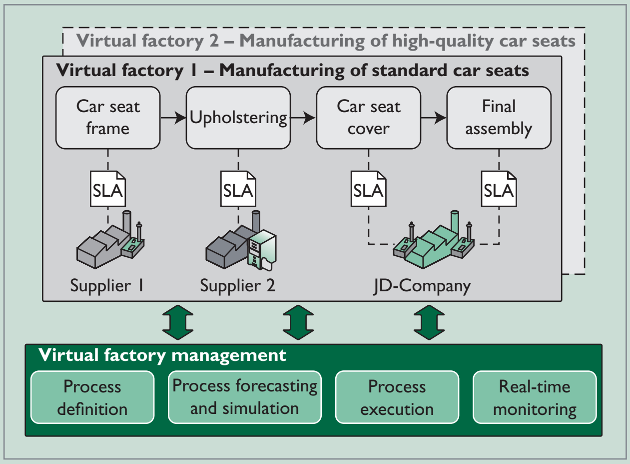

Figure A. A service-oriented virtual factory.

In the example, external suppliers fulfill the first two process steps, while JD-Company carries out the latter two.

Several virtual factories can exist simultaneously, with interdependencies among the different virtual factories in which a company is involved.

Reference

1. W. van Der Aalst et al., "Workflow Patterns," *Distributed and Parallel Databases*, vol. 14, no. 1, 2003, pp. 5–51.

important research challenges for virtual factories to achieve widespread adoption. These topics include both methodologies and software implementations, which are indispensable if we're to realize virtual factories and give stakeholders a plug-and-play experience.

Virtual Factory Description Formats

As a fundamental step, virtual factory brokers must first model real-world enterprises. Instead of using proprietary business object models, brokers should use semantically enriched, standardized descriptions of companies, including their capabilities and products, for data exchange and process descriptions. Among the most important data entities that virtual factory brokers must model and handle are manufacturing processes and physical services. As is generally the case in SOC, a service provider can model services for service-oriented virtual factories by specifying their functionality — that is, the outcome of the physical service — and also describing nonfunctional aspects, such as the maximum number of products a service can produce within a certain time frame (throughput), lead time to initiate production, overall production time, and cost limits.[3]

In manufacturing services, the product description and quality characteristics are very important. Current service description formalisms will need adjustments before they can describe physical services. In addition, common service description formats don't take into account detailed information about service providers. Instead, these formats simplify supplier characteristics to attributes such as reputation or experience and assess them on a numerical scale. In service-oriented virtual factories, such a simplification isn't sufficient: supplier and production facility information must be more fine-grained and must incorporate functional and nonfunctional competences, quality standards, environmental certifications, physical locations, regulations to which the partners are subject, and so on. Moreover, whereas software-based services usually have static descriptions, those of physical services can change over time.

Process Execution, Forecasting, and Simulation

To instantiate an executable process instance from a process template, a virtual factory broker must select physical services for each process step, either manually or semi-automatically (through the virtual factory management software). In the latter case, physical services selection occurs in three steps: first, discovering services that are suitable according to each process step's functional and nonfunctional requirements; second, negotiating service-level agreements (SLAs) between service providers and the virtual factory broker; and finally, selecting one or more specific services. Although current discovery mechanisms are often restricted to information coming from the actual service description, virtual factory management software must also consider supplier information when choosing physical services.

Once the system has selected services for each process step, it must conduct forecasting and simulation for this specific service composition. A virtual factory broker can simulate the impact of integrating different partners and their factory capacities into the manufacturing process's overall outcome as regards cost, delivery time, or scalability of production capacities. Using simulation and forecasting facilities, the enterprises involved should be able to calculate the cost-benefit curves of various alternatives.

Simulation and forecasting shouldn't be restricted to a single process or virtual factory, respectively, but must consider the impact a specific process instance can have on the existing process landscape. Only if we consider the interdependencies between several processes can we optimize single processes as well as the whole process landscape.

For example, consider the scenario described in the "Example Service-Oriented Virtual Factory" sidebar. Assume that three factories (instead of the single supplier 1 that Figure A depicts) provide skills for the first process step and can each handle a maximum of 1,000 seat frames per day. The virtual factory broker can calculate which service allocation is most cost-efficient if the factory itself must manufacture 2,500 or 3,500 car seats per day and assuming that a certain percentage of frames don't pass quality control. The broker can also forecast how the process would change by adding another factory and assess the corresponding risks (such as delays in the delivery process). The broker and its partners should also receive information regarding the interdependencies between different virtual factories. For example, the physical services the broker provides could lead to missing capacity for other manufacturing processes.

Process Adaptation

Certain events can hamper process execution and lead to delays in manufacturing. Virtual factory management software must identify such events early to initiate countermeasures in a timely manner.

Whenever the difference between planned and factual process execution is so severe that a virtual factory can't meet production deadlines, the broker must adapt the process instance. If shifting a physical service from one business partner to another is possible, the actual abstract process definition need not be altered, although nonfunctional aspects of the service might be affected. If this isn't possible, it might be necessary to change the process template by either adding additional process steps or relaxing the functional and nonfunctional requirements for the corresponding step or the whole process. Of course, this in turn could have additional effects. Thus, the broker should forecast and simulate all changes in a process instance or template prior to execution using the virtual factory management software.

A process execution framework should automatically adapt process instances and reflect all changes; if templates require adjustment, the system must also compose a thorough process template version management.

Integrating Process Status Data

Real-time monitoring should automatically collect, preprocess, and make process status information available to the process owner. This includes information about both the product (including parts and raw materials) and manufacturing utilities, delivery status, and so on. To prevent information overflow, the virtual factory management software should use *key performance indicators* (KPIs) to monitor and govern processes.

Data from several sources could be helpful in identifying events that might indicate a severe deviation between expected and actual process behavior. Obviously, existing IT systems such as enterprise resource planning (ERP) or supply chain management (SCM) are valuable data sources. To give the virtual factory broker information about the complete process, the virtual factory management software must integrate data from these systems. However, such systems usually only provide data already available within a certain company.

Another valuable data source is the Internet of Things (IoT). Resources required for manufacturing processes, intermediate goods, and manufactured goods are often enriched with identification, sensor, and communication technologies, such as RFID tags. Up-to-date, real-time status information from so-equipped "smart objects" is then transmitted using (wireless) sensor networks and provided through well-defined interfaces.

Whether information is coming from existing IT systems or the IoT, actual communication with such data sources isn't as much of an issue, and researchers have already proposed corresponding solutions.[4] However, the ability to integrate these data items into the process- and service-based perspective applied in service-oriented virtual factories is still an open issue. Instead of being preprocessed and reported in a device-centric way, IoT data should be presented as process-driven.

Ultimately, we need new methods and technologies for connecting diverse technologies and aggregating the data from them. In the best case, it shouldn't matter to the virtual factory broker whether the process to be executed and monitored is company-internal or cross-organizational because the virtual factory management software closes all information gaps and integrates data seamlessly.

Plug-and-play virtual factories help enterprises move beyond existing operational limitations with technologies and methodologies that can establish and execute cross-organizational manufacturing processes. Using the ideas presented here, enterprises will be able to manage cross-organizational manufacturing processes as though they were being carried out within a single company.

As part of the Adaptive Virtual Enterprise Manufacturing Environment (Adventure) project (www.fp7-adventure.eu), we will further investigate service-oriented virtual factories. Adventure combines methods and technologies from SOC, BPM, and the IoT to meet the research challenges we've outlined here. The project will provide a prototypical, open source virtual factory management framework that companies can connect to their existing ERP and SCM systems through standardized interfaces.

Acknowledgments

This work is part of the Adaptive Virtual Enterprise Manufacturing Environment (Adventure) project, which is funded as an EU FP7 Specific Targeted Research Project.

References

1. D.M. Upton and A. McAfee, "The Real Virtual Factory," *Creating Value in the Network Economy*, Harvard Business School Press, 1999, pp. 69–89.
2. K.D. Gottschalk et al., "Introduction to Web Services Architecture," *IBM Systems J.*, vol. 41, no. 2, 2002, pp. 170–177.
3. P. Leitner, W. Hummer, and S. Dustdar, "Cost-Based Optimization of Service Compositions," *IEEE Trans. Services Computing*, to appear, 2012.
4. P. Spiess et al., "SOA-Based Integration of the Internet of Things in Enterprise Services," *Proc. IEEE Int'l Conf. Web Services*, IEEE Press, 2009, pp. 968–975.

Stefan Schulte is a postdoctoral researcher in the Distributed Systems Group, Institute of Information Systems, at the Vienna University of Technology. His primary research focus is on the application of service-oriented computing within the manufacturing and transport industries. Schulte has a PhD in computer science (Dr.-Ing.) from Technische Universität Darmstadt. Contact him at s.schulte@infosys.tuwien.ac.at; www.infosys.tuwien.ac.at/staff/sschulte/.

Dieter Schuller is a PhD candidate in computer science and heads the Service-Oriented Computing research group in the Multimedia Communications Lab at Technische Universität Darmstadt. His research focuses on QoS-aware service composition. Schuller has a diploma (MSc equivalent) in information systems from Technische Universität Darmstadt. Contact him at dieter.schuller@kom.tu-darmstadt.de; www.kom.tu-darmstadt.de.

Ralf Steinmetz is a full professor in and heads the Multimedia Communications Lab at Technische Universität Darmstadt. His research focuses on distributed computing and seamless multimedia communications. He's the principal investigator for the Adventure project. Contact him at ralf.steinmetz@kom.tu-darmstadt.de; www.kom.tu-darmstadt.de.

Sven Abels is the CEO of the research-driven software company Ascora. His research and business interests are in smart business solutions offering end-to-end integration of IT systems for customers in the health and manufacturing industries. Contact him at abels@ascora.de; www.ascora.de.