

Webservices in der Produktion

Nikhil Bakshi

ETH Zürich, April 2008

1 Übersicht

*Nothing is constant but change*¹. Dieses alte Zitat hat bis zum heutigen Tag kaum an Bedeutung eingebüsst - im Gegenteil: Ständig kommen neue Technologien auf den Markt, Kundenwünsche verändern sich rasant und der globale Wettbewerb nimmt stetig zu.

Um zu überleben, müssen Firmen rasch auf solche Veränderungen reagieren können. Die dazu benötigte Flexibilität fehlt aber in den meisten Fertigungsanlagen und Produktionssystemen. Diese werden oft monolithisch entwickelt und programmiert, sodass Änderungen kostspielig und zeitintensiv sind.

Dieser Bericht beleuchtet das Problem und bespricht Lösungsansätze, die im Rahmen der europäischen Forschungsprojekte SIRENA [1] und SOCRADES [2] erarbeitet wurden. Bei beiden Projekten spielt der Einsatz von Webservices eine zentrale Rolle, um die Flexibilität von Produktionssystemen zu erhöhen. Das SIRENA-Projekt (Service Infrastructure for Realtime Embedded Networked Applications) wurde von 2003 bis 2005 durchgeführt und war ein Kooperationsprojekt zwischen mehreren Universitäten und Industriepartnern. Das Ziel bestand darin, ein dienstgetriebenes Kommunikationssystem von Geräten im Automationsbereich vorzuschlagen. Das SOCRADES-Projekt (Service-Oriented Cross-layer Infrastructure for Distributed Smart Embedded Devices) läuft seit 2006 und baut auf die Resultate von SIRENA auf.

Der Bericht ist wie folgt gegliedert: Kapitel 2 stellt das Problem dar und Kapitel 3 führt Webservices als Lösungsidee ein. Die Kapitel 4 und 5 erweitern diese Grundidee mit einer Middlewareschicht und mit Ontologien, um den Einsatz in grösseren Firmen zu ermöglichen. Kapitel 6 gibt zum Schluss eine Zusammenfassung und einen Ausblick.

2 Problemstellung

Um die heutige Problematik konzeptionell zu verstehen, betrachten wir zunächst ein vereinfachtes Beispiel. Abbildung 1 visualisiert dieses Szenario.

¹ Heraklit von Ephesos, griechischer Philosoph, ca. 540 - 480 v. Chr.

Das Produktionssystem eines Orangensaftherstellers besteht aus einem Fließband und zwei Maschinen. Die Erste füllt Orangensaft in die vorbeikommenden 1-Liter-Flaschen ab, die Zweite versieht die Flaschen mit Deckeln. Der Fließbandmotor und die Maschinen können über einfache Signale via Datenbus angesteuert werden.

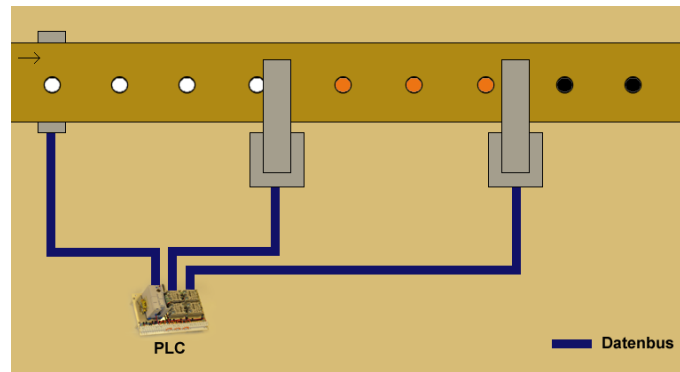


Abbildung 1. Vereinfachtes Produktionssystem eines Orangensaftherstellers

Auf einem PLC (Programmable Logic Controller) befindet sich die gesamte Steuerungslogik. Ein PLC ist ein speicherprogrammierbares Steuerelement, welches mit Assembler-ähnlichen Sprachen programmiert wird. In unserem Beispiel ist der PLC für das Bewegen des Fließbands, das korrekte Abfüllen des Orangensafts und das Verschliessen der Flaschen zuständig.

Nach einem Monat bemerkt die Firma, dass die Haltbarkeit des Orangensafts verlängert werden könnte durch das Anbringen einer Abdeckfolie unter dem Deckel. Zudem soll die Anlage auch 1/2-Liter-Flaschen abfüllen können, um so in den Markt eines Konkurrenten einzudringen.

Um diese Änderungen umzusetzen, müssen zusätzliche Maschinen zwischen den beiden Bisherigen geschaltet und der PLC neu programmiert werden. Aufgrund der tiefen Abstraktionsebene des PLC-Codes und der zunehmenden Komplexität des Systems ist eine solche Neuprogrammierung sehr aufwändig. In den folgenden Abschnitten wird ein Lösungsansatz diskutiert und erweitert.

3 Mehr Flexibilität dank Webservices

Im Rahmen des SIRENA-Projekts wird vorgeschlagen [3], alle Geräte mit kleinen Rechnern auszustatten und zu einem Netz zusammenzuschliessen. Abbildung 2 zeigt die neue Anordnung am Beispiel des Orangensaftherstellers. Die Datenbusse wurden durch Ethernetkabeln und der PLC durch einen zentralen Server

ersetzt. Nun sind alle Komponenten per IP-Adresse erreichbar und die Kommunikation soll über Webservices stattfinden.

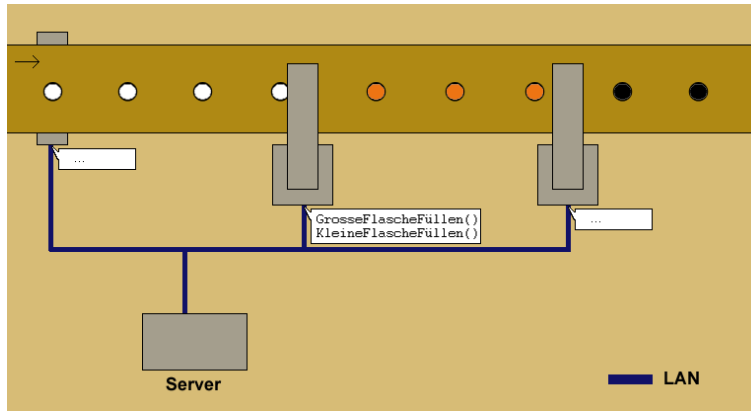


Abbildung 2. Verbessertes Produktionssystem eines Orangensaftherstellers

3.1 Was ist ein Webservice?

Ein Webservice ist ein Softwaredienst, der in einem Netz eindeutig identifizierbar ist und der eine klare Schnittstelle nach aussen anbietet. So könnte in unserem Beispiel die Abfüllmaschine die Dienste `KleineFlascheFüllen()` und `GrosseFlascheFüllen()` anbieten.

Im Vergleich zu einer PLC-getriebenen Architektur erlauben es uns Webservices, das Abstraktionsniveau stark zu erhöhen. Statt Maschinen auf einer tiefen operationellen Ebene zu steuern (z.B. Roboterarm drehen, Motor anschalten), werden sie nun auf funktionaler Ebene angesprochen (z.B. Flasche füllen, Flasche verschliessen). Dies erhöht die Flexibilität des Systems.

Im Zusammenhang mit Webservices spielen folgende Kerntechnologien eine wichtige Rolle.

- WSDL (Web Service Description Language): Diese XML-basierte Sprache wird eingesetzt, um angebotene Dienste zu beschreiben. Dabei werden Methodennamen, Aufrufparameter, Rückgabewerte und Datentypen spezifiziert.
- SOAP: Dieses Protokoll baut auf HTTP auf und wird für den Nachrichtentransport eingesetzt. SOAP-Nachrichten sind XML-basiert und sorgen für Plattformunabhängigkeit. Jede SOAP-Nachricht besteht aus einem Header-Abschnitt, der Adressierungsinformationen enthält, und einem Body-Abschnitt, der die eigentlichen Daten enthält.

- WS-Security: Unter diesem Begriff sind mehrere Protokolle gruppiert, die Verschlüsselungs- und Autorisierungsmechanismen anbieten. Bei sensiblen Daten kann es Sinn machen, den Nachrichtenverkehr zu verschlüsseln. Je nach Einsatzszenario können zudem Autorisierungsregeln für Dienste aufgestellt werden, so dass Unbefugte keinen Zugriff erhalten.

Gemäss [3] bringt die Verwendung von Webservices zwei wichtige Vorteile mit sich.

- Autonomie: Alle Komponenten sind autonom und beinhalten gekapselte Funktionalität. Die Schnittstelle und die Implementierung sind dabei sauber getrennt.
- Interoperabilität: Komponenten können über unterschiedliche Plattformen und Programmiersprachen hinweg kommunizieren.

Heutzutage werden Webservices bereits erfolgreich bei Software-Projekten eingesetzt. Damit sie einfacher bei Projekten mit kommunizierenden Geräten eingesetzt werden können, wurde die Erweiterung DPWS (Device Profile for Web Services) erstellt.

3.2 Device Profile for Web Services (DPWS)

DPWS fügt die Protokolle WS-Discovery und WS-Eventing zu den Kerntechnologien von Webservices hinzu. Zweck dieser Protokolle ist es, die Komponenten eines Systems weiter zu entkoppeln.

- WS-Discovery: Mittels Broadcast können andere Geräte und ihre angebotenen Dienste im Netz gefunden werden. Dadurch kann sich ein neu angeschlossenes Gerät automatisch in ein bestehendes System integrieren und benötigt weniger manuelle Intervention. Bei einer hohen Anzahl von Geräten ist ein Broadcast allerdings nicht sehr effizient; dieses Problem wird in Abschnitt 4 behandelt.
- WS-Eventing: Mittels Publish/Subscribe kann sich ein Gerät die Ereignis-Nachrichten eines anderen Gerätes abonnieren. Dieses Modell eignet sich besonders gut für industrielle Szenarien (beispielsweise könnte das Fließband sich weiterbewegen, sobald die Abfüllmaschine eine volle Flasche meldet).

Nebst DPWS gibt es auch andere dienstgetriebene Technologien, die sich für die Kommunikation zwischen Geräten einsetzen liessen. Einige Beispiele sind OSGI (Open Service Gateway Initiative), JINI (Java Intelligent Network Infrastructure) und UPnP (Universal Plug and Play). Eine Analyse [4] kam aber zum Schluss, dass sich DPWS momentan am besten dafür eignet. Untersucht wurden dabei Kriterien wie Plattformunabhängigkeit, Skalierbarkeit, Sicherheit, Marktakzeptanz und Geräteunterstützung.

4 Einsatz in Firmen: Middleware-Erweiterung

Die bisher vorgestellten Techniken reichen aus, um Produktionssysteme als solches zu realisieren. Um aber das volle Potential der eingeführten Webservices in Firmen auszuschöpfen, werden wir eine Erweiterung benötigen. Dazu soll zunächst die typische Situation und Problematik in grösseren Firmen verstanden werden.

Praktisch alle grosse Firmen setzen ERP-Software (Enterprise Resource Planning) ein, um Finanzen, Lagerhäuser, Personal und andere Ressourcen zu verwalten. ERP-Systeme sind höchst komplex und sind oft mit grossen Einführungskosten und langen Umstellungszeiten verbunden. Die Firma SAP ist ein dominanter Hersteller von solchen Systemen.

Allerdings sind ERP-Systeme meist nicht sehr stark mit den darunter liegenden Produktionssystemen integriert. Einerseits ist dies auf unterschiedliche Kommunikationsprotokolle zurückzuführen, andererseits auch auf die logische Kluft zwischen der betriebswirtschaftlichen und technischen Welt.

Das SAP-Forschungslabor in Zürich hat dieses Integrationsproblem in Angriff genommen und mehrere Publikationen wie [5, 6] im Rahmen des SOCRADES-Projekts dazu veröffentlicht. Die vorgeschlagene Lösung führt eine neue Middleware-Schicht zwischen dem ERP-System und den Webservice-basierten Produktionsgeräten ein. Abbildung 3 veranschaulicht dieses Konzept. Ziel der Middleware ist es, als Vermittler zu fungieren und erweiterte Funktionalität anzubieten. Die direkte Kommunikation ist weiterhin möglich, aber nicht empfohlen.

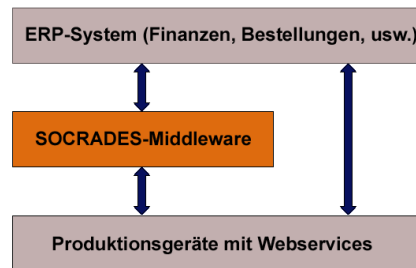


Abbildung 3. Middleware-Schicht zwischen einem ERP-System und Produktionsgeräten

Die Middleware beinhaltet unter anderem folgende Kernkomponenten.

- Zentrales Geräte- und Dienstverzeichnis: Damit das ERP-System nicht selber mit dem WS-Discovery Protokoll alle Dienste ausfindig machen muss, wird

ein Verzeichnis der zur Verfügung stehenden Geräte und Dienste angeboten. Dieses Verzeichnis löst auch gleich das Effizienzproblem von Broadcast bei einer hohen Anzahl von Geräten.

- Ereignisverwaltung und -filterung: Das ERP-System kann Interesse an geschäftsrelevanten Ereignistypen bekunden. So kann es beispielsweise bei der Überhitzung von Maschinen alarmiert werden. Kunden, die von potenziellen Lieferengpässen betroffen sind, können sofort benachrichtigt werden.
- Logging: Ereignisse und Fehlermeldungen der Middleware können zentral aufgezeichnet und archiviert werden. Später können diese Daten zu Analysezwecken wieder aufgerufen werden.

Die Umsetzung der Middleware ist momentan in Gange, nachdem ein Prototyp für einzelne Anwendungsszenarien gebaut wurde.

5 Verbesserte Semantik: Erweiterung mit Ontologien

Für den produktiven Einsatz in Firmen eignet sich die zuvor eingeführte Middleware sehr gut, da eine grosse Anzahl von Geräten effizient verwaltet werden kann. Allerdings können manchmal komplexe Interaktionsszenarien auftreten, bei denen die WSDL-Schnittstellenbeschreibungen zu primitiv sind. Ein Beispiel wäre die Herstellung eines Produktes mit vielen Einzelteilen und vielen Abhängigkeiten zwischen den verschiedenen Produktionsschritten. Eine Arbeit [7] von der Tampere Universität in Finnland schlägt vor, Ontologien als Erweiterung einzuführen.

Vereinfacht gesagt, ist eine Ontologie eine strukturierte Repräsentation von einem Wissensgebiet. OWL (Web Ontology Language) ist eine bekannte Spezifikationsprache für Ontologien. In unserem Fall wollen wir die Produkte, die Prozesse und die zur Verfügung stehenden Maschinen mit Ontologien darstellen. Abbildung 4 zeigt stark vereinfachte Ontologien, die beispielsweise bei einer Automobilfabrik eingeführt werden könnten.

- Produktontologie: Hier werden die Einzelteile eines Autos spezifiziert und in einer Hierarchie angeordnet. Somit wird klar, welche Teile produziert werden müssen, um ein Auto herzustellen.
- Prozessontologie: Hier werden die Prozesse und Subprozesse definiert, die zur Herstellung jedes Einzelteils nötig sind. Als Beispiel sind die Schritte *Formen*, *Lackieren* und *Trocknen* aufgelistet, die bei der Türherstellung durchgeführt werden müssen.

- Maschinenontologie: Hier werden alle Produktionsgeräte aufgeführt, die zur Verfügung stehen. Jedes Gerät kann ein oder mehrere Prozesse aus der Prozessontologie ausführen. Details zu jeder Maschine wie Lage, Status und Wartungsstand können auch als Teil der Ontologie abgespeichert werden.

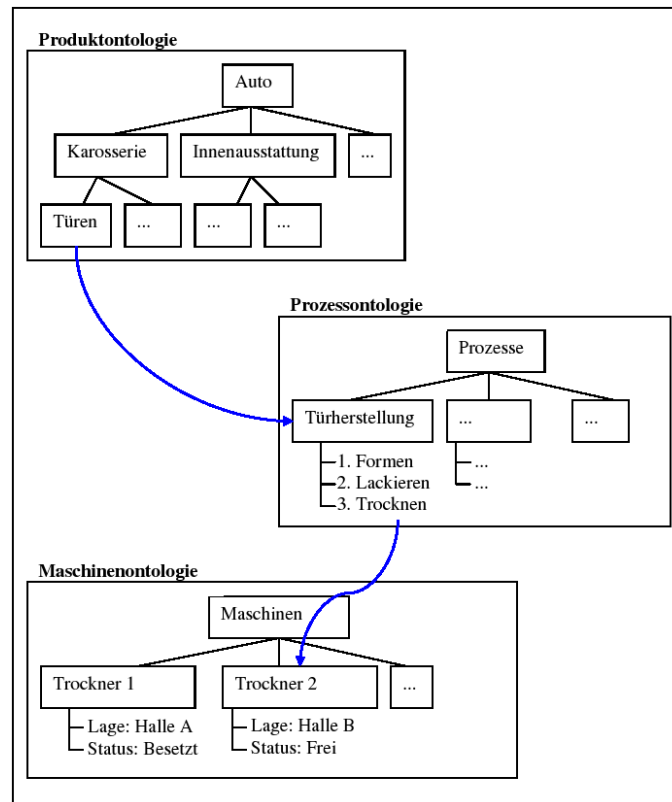


Abbildung 4. Vereinfachte Ontologien einer Automobilfabrik

In der Praxis ist die Erstellung von detaillierten Ontologien recht zeitintensiv. Aber bei grossen Projekten lohnt sich der Aufwand und bringt zwei nützliche Vorteile mit sich.

- Automatisierung: Dank dieser zusätzlichen Semantik können die Komponenten intelligenter reagieren. Das System kann zum Beispiel zur Laufzeit die Produktionsreihenfolge automatisch umstellen, falls gewisse Maschinen gerade ausgelastet sind. Zur Optimierung der zeitlichen Koordination kann auch die Lage der Geräte innerhalb der Fabrik berücksichtigt werden.

- Flexibilität: Änderungen an den Ontologien sind einfacher umzusetzen als Änderungen am Server oder an den Webservices. Ontologien bieten eine geeignete Speicherstruktur für solche Abhängigkeiten und Fakten. Dies bedingt aber, dass industrieweite Standards für Ontologien definiert und eingehalten werden.

6 Zusammenfassung und Ausblick

In diesem Bericht haben wir die fehlende Flexibilität von heutigen Produktionssystemen anhand eines Beispiels studiert und Webservices als geeigneten Lösungsansatz betrachtet. Als Erstes wurden Webservices auf Geräten eingeführt und die DPWS-Spezifikation erläutert. Konzeptionell wurde die Steuerungslogik, die sich vorher zentral auf einem PLC befand, nun grösstenteils auf die Geräte verteilt.

In grossen Firmen macht aber eine zusätzliche Middleware-Schicht Sinn, um Produktionssysteme mit ERP-Systemen zu integrieren und die Komplexität von vielen Geräten zu verwalten. Das SAP-Forschungslabor in Zürich setzt sich unter anderem mit solchen Themen auseinander. Auch Ontologien wurden als Möglichkeit studiert, um Produkte, Prozesse und Maschinen strukturiert darzustellen. Die zusätzliche Semantik erlaubt es, interessante Laufzeitanpassungen vorzunehmen, ist aber aufwändig zu definieren.

Es ist spannend zu beobachten, dass sich der Einsatz von Webservices nicht auf Produktionsanlagen beschränkt. Mit der zunehmenden Leistungsfähigkeit und Miniaturisierung von Rechnern wird es in der Zukunft immer einfacher sein, Webservices auf jeglichen Geräten laufen zu lassen. Im Bereich der Heim-elektronik sind schon erste Audioanlagen, Fernseher und Beleuchtungssysteme präsentiert worden, die untereinander über Webservices kommunizieren können und die zentral von einer einzigen Fernbedienung gesteuert werden können. In diesem Bereich setzt sich DPWS immer stärker als Standard durch.

Literatur

- [1] Das SIRENA-Projekt. <http://www.sirena-itea.org> (Zugriff am 14. April 2008)
- [2] Das SOCRADES-Projekt. <http://www.socrades.eu> (Zugriff am 14. April 2008)
- [3] Jammes, F., Smit, H.: Service-Oriented Paradigms in Industrial Automation. IEEE Transactions on Industrial Informatics, Feb. 2005, Volume 1, Issue 1, 62-70
- [4] Bohn, H., Bobek, A., Golatowski, F.: SIRENA Service Infrastructure for Real-time Embedded Networked Devices: A service oriented framework for different domains. International Conference on Networking, International Conference on Systems and International Conference on Mobile Communications and Learning Technologies, Apr. 2006, 43

- [5] Karnouskos, S., Baecker, O., De Souza, L., Spiess, P.: Integration of SOA-ready Networked Embedded Devices in Enterprise Systems via a Cross-Layered Web Service Infrastructure. IEEE Conference on Emerging Technologies and Factory Automation, Sep. 2007, 293-300
- [6] De Souza, L., Spiess, P., Koehler M., Guinard, D., Karnouskos, S., Savio, D.: SOCRADES: A Web Service based Shop Floor Integration Infrastructure. Internet of Things Conference, Mar. 2008
- [7] Delamer, I., Lastra, J.: Loosely-coupled Automation Systems using Device-level SOA. IEEE International Conference on Industrial Informatics, Jun. 2007, Volume 2, 743-748