

Weborientierte Ansätze für eine vereinfachte Herangehensweise an serviceorientierte Architekturen in der Landwirtschaft

Daniel Martini, Mario Schmitz und Martin Kunisch
Kuratorium für Technik und Bauwesen in der Landwirtschaft e. V. (KTBL)
Bartningstraße 49
D-64289 Darmstadt
d.martini@ktbl.de

Abstract: Entwicklungen in der Vernetzung von Computersystemen und in Sensortechnologien haben in der Landwirtschaft zu einer Flut an potenziell verfügbaren Daten aus verschiedenen Quellen geführt. Die Integration der Daten in Anwendungen hinkt dabei der Verfügbarmachung hinterher. Im Rahmen der agroXML-Entwicklung wurden verschiedene Komponenten einer Architektur entwickelt, die Vorgänge hierbei vereinfacht. Weborientierte Ansätze erlauben dabei eine schnelle Anpassung an sich ändernde Rahmenbedingungen und die Wiederverwendung von Daten in nicht vorhergesehenen Kontexten. Mit Hilfe eines überschaubaren Satzes von orthogonal aufeinander aufbauenden Technologien des World Wide Web Consortium werden Systeme dabei im Sinne einer service-orientierten Architektur lose miteinander gekoppelt.

1 Einleitung

Die ersten in den frühen Achtziger-Jahren des vorigen Jahrhunderts eingeführten Farm-Management-Informationssysteme ersetzen die bislang auf Papier geführte Dokumentation in Schlagkarteien und Buchführung. Ein- und Ausgabe erfolgten manuell bzw. visuell über Tastatur und Bildschirm oder Drucker. Daten wurden zentral an einem Platz in einer Datenbank auf dem Büro-PC gehalten.

Später gewann das Einlesen von automatisch erhobenen Daten – z. B. von der Milchmengenerfassung im Melkstand – über digitale Schnittstellen zunehmende Bedeutung. Anfalls ad-hoc programmiert, existieren hierfür inzwischen Standards.

Zunehmend spielten Prozesse auf dem Betrieb eine Kernrolle. Nachrichtenorientierte Systeme auf Basis von Remote Procedure Calls (RPC), wie sie teilweise in der industriellen Produktion verwendet werden, finden seitdem auch in der Landwirtschaft Anwendung. Im Gegensatz zur Industrie sind allerdings auch heute noch die meisten landwirtschaftlichen Unternehmen Kleinbetriebe. Das Nutzungsmuster von IT ist eher mit der Nutzung von Personalcomputern in Privathaushalten als mit dem typischer Unternehmens-IT vergleichbar.

In den letzten Jahren tauchten auch im Internet zunehmend Daten, die für die Landwirtschaft von Interesse sind, auf. Das umfasst beispielsweise Daten zu Tierarzneimitteln,

Zuchtbullen oder Bodendaten und Daten zu Sorten im Pflanzenbau. Eine einheitliche Bereitstellung erfolgt hier bislang nicht. Meist ist es notwendig, die gewünschten Daten interaktiv über den Webbrowser abzurufen.

Zu beobachten ist, dass – entgegen der „Incommensurate Scaling Rule“ ([SK09]) – die grundlegende Architektur von Informationssystemen für landwirtschaftliche Betriebe über die aufgezeigten Entwicklungen hinweg dieselbe blieb. Die vorherrschenden Systeme sind generell schwierig an die Bedingungen, wie sie in der Landwirtschaft vorkommen, anzupassen. Im Vergleich zur industriellen Produktion operiert die Landwirtschaft in einem nur begrenzt kontrollierten Umfeld. Beim Aufbau von Fabriken und Anlagen in der Industrie sind in aller Regel alle Parameter in gewissen Grenzen durch die zuständigen Ingenieure anpassbar. Durchsätze von Produktionsstraßen sind bekannt und Sensoren werden fest montiert. In der Landwirtschaft werden Produktionsraten stark von äußeren, unkontrollierbaren Faktoren bestimmt (z. B. Wetter). Sensoren sind oft beweglich, was deren ausgelieferte Datenmenge erhöht und für die Interpretation eine weitere Komponente an Variabilität und damit Komplexität hinzufügt. Relationale Modelle haben sich im Bereich der Züchtung zwar gut bewährt, da hier ein fester Satz von Kennzahlen und standardisierte Versuchspläne vorliegen, im Bereich der praktischen Produktion wachsen solche Modelle, wenn sie betriebsübergreifend funktionieren sollen, jedoch oft auf unhandliche Größenordnungen an, da die für den Einzelbetrieb sinnvollen Parameter von einer Reihe von Rahmenbedingungen abhängen.

Serviceorientierte Architekturen (SOA) wurden lange Zeit als Lösung dieses Problems propagiert. Sie sollten die notwendige lose Kopplung zur flexiblen Ausführung von Geschäftsprozessen in sich ändernden Umgebungen bereitstellen. In der praktischen Umsetzung blieb SOA in dieser Hinsicht weit hinter den Erwartungen zurück. In aller Regel wurden Paradigmen für das Design von geschlossenen Systemen auf offene Systeme in dynamischen Netzwerken übertragen. Wilde und Glushko führen aus, dass eine solche Übertragung von internen Daten- und Vorgehensmodellen auf Methoden des Datenaustausches im Internet nicht sinnvoll ist und nennen dieses Phänomen „web blindness“ ([WG08]).

Wie unten ausgeführt wird, kann in festgezurrtten Prozessmodellen mit einer Vielzahl von vordefinierten Nachrichten nur schwer auf neue Datenquellen und Abläufe eingegangen werden. Wenn Anpassungen vorgenommen werden, steigt häufig die Komplexität überproportional an. Um das volle Potenzial der an verschiedensten Stellen erhobenen und prinzipiell verfügbaren Daten voll auszunutzen, ist ein erhöhtes Maß an Flexibilität, Skalierbarkeit und Erweiterbarkeit notwendig.

Neuerdings hat sich daher die sogenannte weborientierte Architektur entwickelt, die dynamischere und flexiblere Integration von Informationen auf Basis einfacherer Technologien ermöglichen soll. Gleichzeitig bietet dieser Ansatz die Verallgemeinerbarkeit und Skalierbarkeit auf eine große Zahl an Teilnehmern innerhalb der SOA, die notwendig sind, wenn man betrachtet, dass alleine in Deutschland 360 000 landwirtschaftliche Betriebe existieren. Diese Methoden werden derzeit im Rahmen der Entwicklung von agroXML erprobt.

2 Architekturkomponenten

2.1 Dienste

Wie ausgeführt, muss ein Kommunikationsnetzwerk in der Landwirtschaft mit stetigen Änderungen und einer Vielzahl von Teilnehmern und datenliefernden Geräten und Sensoren zurechtkommen. Fielding [Fie00] führt aus, dass in einer solchen Umgebung „anarchische Skalierbarkeit“ notwendig ist. Im Detail führt er hierzu aus: „Clients cannot be expected to maintain knowledge of all servers. Servers cannot be expected to retain knowledge of state across requests“. In dem Fall ist der Server lediglich ein Informationslieferant, der den Zustand nicht kontrolliert. Der Anwendungszustand wird vom Client bestimmt. Hierdurch können eine Reihe von Optionen des Flusses durch eine Anwendung ermöglicht werden, sodass auf geänderte Rahmenbedingungen eingegangen werden kann.

In praktischen Implementationen wird ein sogenanntes ReSTful Design ([Fie00] und [RR07]) eingesetzt. Ein Prototyp für einen Anwendungsfall in der Landwirtschaft wurde in Martini et al. [MSFK09] beschrieben. Weitere Anwendungsbeispiele sind derzeit in Arbeit.

In ReSTful Webdiensten stellt der Server eine verteilte, persistente Speicherebene sowie Hinweise zur Navigation bereit, die von Clients für die Ausführung einer Vielzahl verschiedener Geschäftsprozesse genutzt werden können. Mögliche Methodenaufrufe werden auf das sogenannte Create-Read-Update-Delete (CRUD) Pattern, meist unter Verwendung von HTTP, reduziert. Clients müssen hierbei nur die grundlegenden HTTP-Anfragen kennen (GET, POST, abhängig von der Implementation PUT und DELETE [FGM⁺99]).

Mit den derzeit im Rahmen der Implementation von serviceorientierten Architekturen eingesetzten Technologien (vorwiegend Simple Object Access Protocol (SOAP, [ML07]) in Verbindung mit der Web Service Description Language (WSDL, [BL07]) zur Servicebeschreibung sowie weiteren Technologien zur Beschreibung von Geschäftsprozessen wie Business Process Modeling Language (BPEL) und Business Process Modeling Notation (BPMN)) erfolgt hingegen meist implizit eine Aufteilung der Kontrolle des Zustandes zwischen Client und Server. Umfangreiches, formal kodiertes Wissen über den Prozess ist dazu auf beiden Seiten notwendig. Updates des jeweiligen Zustandes werden über fest vordefinierte Nachrichten erzielt. Daher muss ein Client vor einer Interaktion über alle möglichen Methodenaufrufe auf der Serverseite informiert sein. Der Server muss meist außerdem den Zustand eines bestimmten Client innerhalb des Prozesses halten. Anstatt loser Kopplung führt dies in der Praxis zu einer – im Gegensatz zur in SOAen propagierten losen Kopplung – sehr engen Bindung zwischen Client und Server. Anpassungen des Prozesses, die mit Änderungen der Beschreibung der Webdienste und damit der WSDL-Dateien einhergehen, führen in aller Regel zur Notwendigkeit, Code sowohl auf Client- als auch auf Serverseite neu zu kompilieren. Die Anforderungen, was die vorausschauende Planung des Einsatzzweckes eines bestimmten Systems innerhalb der SOA anbelangt, sind daher hoch. Zwischenzeitlich existieren zwar Werkzeuge, die auch eine dynamische Bindung zulassen, diese entbinden allerdings nur auf der Clientseite vom Neukompilieren und zum Anderen stehen diese Werkzeuge nicht in allen Programmierumgebungen

zur Verfügung. Für ein skalierbares und robustes Netz, an dem Maschinen unabhängig von deren Leistungsfähigkeit und Betriebsumgebung teilnehmen sollen, ist es jedoch unumgänglich, möglichst keine Annahmen über die Beschaffenheit des Systems am anderen Ende einer Kommunikationsstrecke zu treffen ([Pad82]).

2.2 Repräsentation

Die Daten selbst können in einer Reihe von Formaten repräsentiert werden. Vorteil einer Repräsentation auf Basis der eXtensible Markup Language (XML) ist die gute Verfügbarkeit von Werkzeugen für die Verarbeitung in nahezu allen Programmiersprachen. XML wurde in einigen Standards des Agrarbereiches – z. B. dem ISOBUS, XML/ADED und agroXML – fest verankert. Allerdings hat XML auch Schwachpunkte (z. B. das ungünstige Verhältnis zwischen Tags und eigentlichen Daten), die in bestimmten landwirtschaftlichen Anwendungsszenarien zu Tage treten können.

agroXML wurde hierbei in letzter Zeit an die sich durch obige Architektur ergebenden Anforderungen angepasst. Zum Beispiel können mit XLink [DMO01] die notwendigen Verknüpfungen zwischen Ressourcen auf verschiedenen Servern erstellt werden. Objekte, die erwartungsgemäß verteilt abgespeichert werden können, wurden als globale Elemente im Schema deklariert, um die Erstellung als kleine, selbsterklärende XML-Instanzen zu ermöglichen. agroXML folgt außerdem gängigen Empfehlungen, die die Einbindung in weitere, andere XML-Vokabularien sowie die Erstellung von Mappinglayern, wie unten beschrieben, ermöglichen.

2.3 Semantik

Nützliche Dienste können bereits alleine durch oben beschriebene Bestandteile aufgebaut werden. Eine weitere Ebene kann jedoch semantische Annotationen bereitstellen und dadurch in einem Netzwerk verfügbare Ressourcen weiter mit Information anreichern. Diese Ebene wird für agroXML derzeit entwickelt. Sie besteht aus Beschreibungen der im XML Schema enthaltenen Konzepte mit dem Resource Description Framework (RDF, [KC04]). Verknüpfungen der Schemaelemente werden unter Verwendung der SAWSDL-Empfehlung des W3C [FL07] vorgenommen.

Hierdurch wird nicht nur eine präzisere Formalisierung bereitgestellt, sondern auch das Mapping in andere Ontologien ermöglicht. Derzeit wird ein Mapping in den AGROVOC Thesaurus der FAO (siehe <http://aims.fao.org>) erstellt. AGROVOC wurde ursprünglich als mehrsprachiges elektronisches Wörterbuch für die Landwirtschaft erstellt, um Indexing von Literatur zu ermöglichen. In neuerer Zeit wurde AGROVOC um eine Reihe von Relationen zwischen Begriffen ausgebaut. Das Mapping zwischen agroXML und AGROVOC eröffnet neue Möglichkeiten, Funktionalitäten wie die Nutzerführung durch agroXML-Instanzen oder das Zusammenführen von Information zu ermöglichen.

3 Schlussfolgerungen und Ausblick

Wie in Abschnitt 1 dargestellt, wird Landwirtschaft üblicherweise in einer hochgradig unkontrollierten Umgebung ausgeführt. Die Modellierung jedes möglichen Zweiges eines Prozesses führt daher zu sehr komplexen Strukturen. Die Kodierung von Prozesswissen aus den über das Netz ausgetauschten Datenpaketen herauszuhalten vereinfacht das zugrundeliegende System. Die vorgeschlagene Architektur ist mit einfachen Technologien auch auf mobilen Geräten umsetzbar. Eine HTTP Bibliothek und ein XML-Parser reichen aus, sodass auch Betriebe mit wenig leistungsfähiger IT-Ausstattung ein solches Netzwerk nutzen können.

Die Herausforderung im Datenmanagement in der Landwirtschaft ist nicht in einem Fehlen von Standardisierung, sondern viel mehr in schlechter Interoperabilität zwischen Standards zu sehen. Die bestehende Situation wurde in [MS09] beschrieben. Semantische Mappinglayer können hierbei Abhilfe schaffen. Hierfür ist es notwendig, ähnliche semantische Ebenen wie die oben beschriebene für weitere Agrarstandards zu erstellen und im Zusammenspiel zu erproben. Der Aufwand einer Umsetzung für das Agricultural Data Element Dictionary (ADED) beispielsweise würde sich nach heutigem Kenntnisstand in überschaubarem Rahmen halten, da wichtige Bausteine (Identifikation der Datenelemente, Label) bereits vorhanden sind.

Kritisiert wird am obigen Ansatz häufig, dass Methoden der Prozessdefinition und Bindung der Dienste fehlen. Wenn Implementationen betrachtet werden, wird jedoch deutlich, dass hierin auch seine Stärke liegt und solche Methoden nicht notwendig sind. Der Client kontrolliert die Interaktion mit dem Server. Dadurch wird es möglich, eine Reihe von Geschäftsprozessen auf ein und demselben Ressourcensatz auszuführen. Im oben erwähnten Demonstrator stellt beispielsweise eine Ressource einen Datensatz zu einem Schwein dar, eine weitere Ressource die Stammdaten des Betriebs, wieder eine andere Ressource die Zertifizierungsinformationen des Betriebs. Je nach Anforderungen des Prozesses können gezielt einzelne Informationen abgerufen werden. Da die Umsetzung relativ einfach ist, entsteht dabei oft ein Programmcode, der in der Lesbarkeit und Verständlichkeit einer WSDL-Datei in nichts nachsteht. Insgesamt kommt der Ansatz dem Ziel lose gekoppelter Systeme, einem Kerndogma serviceorientierter Architekturen, deutlich näher als Ansätze auf Basis von SOAP und WSDL ([Til09]).

Inhärent vorhanden in allen dynamischen Netzwerken ist die Schwierigkeit, referentielle Integrität zwischen verteilten Daten zu erzielen. So kann zum Beispiel im Internet nie garantiert werden, dass Verknüpfungen aufgelöst werden können. Hierfür gab es Lösungsvorschläge (z. B. [Kap95]), aber keiner hat seinen Weg in eine breite Umsetzung gefunden.

4 Danksagung

Wir danken dem Bundesministerium für Bildung und Forschung für die Förderung eines Teils der vorgestellten Arbeiten im Rahmen des iGreen Projektes unter dem Förderkenn-

zeichen 011A08005X.

Literatur

- [BL07] David Booth und Canyang Kevin Liu. *Web Services Description Language (WSDL) Version 2.0 Part 0: Primer*. World Wide Web Consortium, June 2007. <http://www.w3.org/TR/wsdl20-primer/>.
- [DMO01] Steve DeRose, Eve Maler und David Orchard. *XML Linking Language (XLink) Version 1.0*. World Wide Web Consortium, June 2001. <http://www.w3.org/TR/xlink/>.
- [FGM⁺99] Roy Thomas Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach und Tim Berners-Lee. *RFC2616: Hypertext Transfer Protocol – HTTP 1.1*, June 1999.
- [Fie00] Roy Thomas Fielding. *Architectural Styles and the Design of Network-based Software Architectures*. Dissertation, University of California, Irvine, 2000.
- [FL07] Joel Farrell und Holger Lausen. *Semantic Annotations for WSDL and XML Schema*. World Wide Web Consortium, August 2007. <http://www.w3.org/TR/sawSDL/>.
- [Kap95] F. Kappe. A Scalable Architecture for Maintaining Referential Integrity in Distributed Information Systems. *Journal of Universal Computer Science*, 1(2):84–104, 1995.
- [KC04] Graham Klyne und Jeremy J. Carroll. *Resource Description Framework (RDF): Concepts and Abstract Syntax*. World Wide Web Consortium, February 2004. <http://www.w3.org/TR/rdf-concepts/>.
- [ML07] Nilo Mitra und Yves Lafon. *SOAP Version 1.2 Part 0: Primer (Second Edition)*. World Wide Web Consortium, April 2007. <http://www.w3.org/TR/soap12-part0/>.
- [MS09] Daniel Martini und Mario Schmitz. Data Infrastructures in Agriculture – Attempts at Interoperability. In Jan L. G. Dietz, Hrsg., *KEOD 2009 – Proceedings of the International Conference on Knowledge Engineering and Ontology Development*, Seiten 287–292, Portugal, 2009. Institute for Systems and Technologies of Information, Control and Communication.
- [MSFK09] Daniel Martini, Mario Schmitz, Jürgen Frisch und Martin Kunisch. Enabling integration of distributed data for agricultural software applications using agroXML. In *EFITA conference '09 Proceedings*, Seiten 145–150, Wageningen, 2009. Wageningen Academic Publishers.
- [Pad82] Michael A. Padlipsky. *RFC871: A Perspective on the ARPANET Reference Model*, September 1982.
- [RR07] Leonard Richardson und Sam Ruby. *RESTful Web Services*. O'Reilly, Sebastopol, 2007.
- [SK09] Jerome H. Saltzer und M. F. Kaashoek. *Principles of Computer System Design – An Introduction*. Morgan-Kaufmann, 2009.
- [Til09] Stefan Tilkov. Der bessere Web Service? *Java Magazin*, 1:74–80, 2009.
- [WG08] Erik Wilde und Robert J. Glushko. Document Design Matters. *Communications of the ACM*, 51(10):43–49, 2008.