

Electronic Communications of the EASST Volume 17 (2009)



Workshops der Wissenschaftlichen Konferenz Kommunikation in verteilten Systemen 2009 in Kassel (WowKiVS 2009)

Kontextsensitive Konfiguration und Ausführung verteilter Geschäftsprozesse

Ch. Loeser, R. Trunko, Th. Steckel, K. Podratz, E. Georgiew, F. Swoboda

14 Pages

Kontextsensitive Konfiguration und Ausführung verteilter Geschäftsprozesse

Christoph Loeser
Siemens AG
Fuerstenalle 11
Paderborn
+49-5251-606 148
loeser@c-lab.de

Ralf Trunko
Institut für Angewandte
Informatik und Formale
Beschreibungsverfahren (AIFB)
Universität Karlsruhe (TH)
Karlsruhe
+49 720-608 4547
trunko@aifb.uni-karlsruhe.de

Thilo Steckel
CLAAS SFE GmbH
Muensterstr. 33
Harsewinkel
+49 5247-122 074
thilo.steckel@claas.com

Kevin Podratz
Forschungsinstitut für
Rationalisierung e.V.
RWTH Aachen
Pontdriesch 14-16
Aachen
+49-241-477 05-235
Pod@fir.rwth-aachen.de

Emanuel Georgiew
Siemens AG
Fuerstenalle 11
Paderborn
+49-5251-606 147
Emanuel.Georgiew@c-lab.de

Frieder Swoboda
CADsys GmbH
Systemhaus
Carl-von-Bach-Str. 3
Chemnitz
+49-371-4000 70 201
fswoboda@cadsys.de

Zusammenfassung:

In der Entwicklung moderner Informationssysteme gibt es einen Trend in Richtung adaptiver Systeme. Mittlerweile existieren diverse Ansätze auf dem Gebiet der Umgebungszintelligenz (Ambient Intelligence), in denen sich eine intelligente Umgebung an ihren Benutzer anpasst. Aktuelle Forschungsansätze untersuchen in diesem Zusammenhang auch Adaptivität in den Bereichen Geschäftsprozess- und Dienstleistungsmanagement. Ein Umsetzungskonzept für Adaptivität ist Kontextsensitivität. In diesem Beitrag wird ein holistischer Ansatz für die kontextsensitive Konfiguration und Ausführung von Dienstleistungen und der ihnen zugrundeliegenden Geschäftsprozesse beschrieben, der im Rahmen des vom Bundesministerium für Wirtschaft und Technologie (BMWi) geförderten Forschungsprojekts Robot to Business entwickelt wird. Die Integration von Kontext wird dabei auf drei unterschiedlichen Ebenen berücksichtigt (Dienstleistungen, Geschäftsprozesse, Technologie). Der Ansatz wird innerhalb des Projekts in zwei unterschiedlichen Anwendungsgebieten evaluiert (Landwirtschaft und IT Service Management), um sowohl die kontextsensitive Rekombinierbarkeit und Konfiguration von Leistungsmodulen (build time) als auch die kontextsensitive Ausnahmebehandlung in den zugehörigen Geschäftsprozessen (run time) nachzuweisen.

Keywords: Geschäftsprozessmodellierung, Prozessautomatisierung, Dienstleistungskonfiguration, Kontextsensitivität



1 Einleitung und Motivation

Die IT-gestützte Automatisierung von Geschäftsprozessen findet wachsende Verbreitung. Neben der Herausforderung, heterogene Systeme und IT-Landschaften in Geschäftsprozessen zu vereinen, wächst der Anspruch, auf unerwartete und dynamische Einflüsse in Geschäftsprozessen effektiv reagieren zu können. Adaptive Systeme ermöglichen automatisierte Anpassungen bzw. Reaktionen auf externe Einflüsse (Kontextsensitivität). Unter Kontext sind Umwelteinflüsse oder das Verhalten von Nutzern zu verstehen. Im Gegensatz zu Anpassungen, bei denen sich eine intelligente Umgebung ausschließlich an Anforderungen und Verhalten ihrer menschlichen Benutzer anpasst (z.B. bei Ambient Intelligence), beschäftigt sich der hier vorgestellte Ansatz mit Anpassungen auf der Ebene von Geschäftsprozessen. Dies beinhaltet sowohl die Konfiguration von Prozessen als auch deren Koordination zur Ausführungszeit basierend auf *Kontextsensitivität*. Der Begriff *Prozesskontext* sei hier definiert als die Menge aller Faktoren (Anforderungen, Umwelteinflüsse, Leistungskennzahlen etc.), die einen Einfluss auf die Konfiguration und/oder Ausführung eines Prozesses haben können. Bei der Verwendung von Prozesskontext wird zwischen Kontextsensitivität zur Konfigurationszeit ("build time") und zur Ausführungszeit ("run time") eines Prozesses unterschieden. Kontextsensitivität zur Konfigurationszeit eines Geschäftsprozesses bedeutet, dass die Konfiguration eines Prozesses einerseits abhängig von bestimmten Geschäftsanforderungen ist und andererseits von Ausprägungen relevanter Umweltfaktoren (initialer Prozesskontext). Kontextsensitivität zur Ausführungszeit eines Prozesses bedeutet, dass Faktoren überwacht werden, die einen Einfluss auf die Ausführung des Prozesses haben können. Somit ist es möglich, kontextsensitiv adäquate Ausnahmebehandlungen bei Auftreten von spezifischen Ereignissen (Ausnahmen, die ein Abweichen vom Standardablauf verursachen können) während der Ausführungszeit aufzurufen. Ein Kontextfaktor wird dabei durch eine *Kontextinformation* formal beschrieben. Zur Umsetzung von Kontextsensitivität müssen Regeln entwickelt werden, die jeweils Kontextinformationen mit den zugehörigen Prozessen bzw. Aktivitäten verknüpfen. Zur Konfigurationszeit werden *Konfigurationsregeln* genutzt, die innerhalb der Anwendung *Konfigurator* verwendet werden. Sie bilden Ge- und Verbote der Leistungs-, Prozess- und Ressourcenkombination ab. Zur Ausführungszeit werden *Interventionsregeln* verwendet. Für jede Kontextinformation wird ein gültiger Wertebereich definiert, innerhalb dessen der Prozess regelkonform abläuft. Sobald ein a priori definierter Toleranzbereich eines solchen Wertebereichs verletzt wird, wird eine entsprechende Interventionsregel aktiviert, durch die ein geeigneter *Ausnahmebehandlungsmechanismus* ausgelöst wird. Um das Abfangen solcher Ausnahmen während der Ausführung des Prozesses zu ermöglichen, müssen spezielle Reparatur-, Alarm- und Anpassungsmechanismen bspw. in Form von Web Services entwickelt werden.

2 Das Projekt Robot to Business

Die in diesem Beitrag vorgestellten methodischen Ansätze und technischen Implementierungen werden im Forschungsprojekt *Robot to Business – IT-gestützte Integration von semi-autonomen mobilen Maschinen und Prozessen in Geschäfts- und Service-Modelle (R2B)* entwickelt und umgesetzt. Das Projekt R2B hat zum Ziel, ein Vorgehensmodell inklusive der entsprechenden Technologiekomponenten zu entwickeln, auf deren Basis selbstorganisierendes Verhalten von semi-autonomen Systemen in dynamischen Umgebungen unterstützt wird (das

Projekt läuft von 10/2006 bis 03/2010). Basierend auf einer Methode zur Integration von Kontextsensitivität in das Prozessmanagement soll ein Ansatz zur kontextsensitiven, semi-autonomen Konfiguration und Anpassung von Geschäftsprozessen entwickelt werden. Der zu entwickelnde Lösungsansatz wird anhand von Anwendungsfällen aus unterschiedlichen Anwendungsdomänen validiert. Es werden ein landwirtschaftliches Anwendungsszenario (Grünfütterernte) und ein Szenario aus dem Bereich IT Service Management (flexible Wartung von IT-Infrastrukturen) betrachtet. Im erstgenannten Szenario kommt als zusätzliche Herausforderung die Integration des Backends hinzu, d. h. die Verknüpfung von Geschäftsprozessen mit "Maschinenprozessen", die auf mobilen Einheiten ablaufen (z. B. Grünfütter Erntemaschinen).

3 Stand der Wissenschaft

In der Informatik spielt Kontextintegration eine immer wichtigere Rolle. Man spricht hier von kontextsensitiven Anwendungen oder Diensten (*context-aware computing*), d. h. Anwendungen, die Kontextinformationen verwenden, um sich an die Situation ihres Benutzers oder ihrer Umgebung anzupassen. Grob gesagt bedeutet dies, eine Reihe von Variablen (Kontextinformationen) zu lesen, deren Werte zu verknüpfen und sie anschließend zu interpretieren. Diesen interpretierten Werten entsprechend können dann Aktionen ausgelöst werden. Für das Management von Kontextinformationen werden spezielle *Kontextmanagementsysteme* (auch: *Context Engines*) eingesetzt.

Eingesetzt werden kontextsensitive Dienste insbesondere im Rahmen der Bereitstellung von Umgebungszintelligenz (*Ambient Intelligence*, z. B. [1], [2]) oder beim Angebot standortbezogener Dienste (*location based services*). Neben dem Anwendungsgebiet *Smart Home* gibt es auch Anwendungsansätze zur Nutzung kontextsensitiver Dienste in der technischen Instandhaltung (z.B. [3]). Neuartig ist die Integration von Kontextsensitivität im Bereich des Geschäftsprozessmanagements. In [4] und [5] bspw. stellen die Autoren CAWE vor, ein kontextsensitives Prozessmanagementsystem basierend auf der Orchestrierung von Web Services und führen eine Rahmenarchitektur ein, welche die Einsatzmöglichkeiten für eine Workflow-Engine dahingehend erweitert komplexe Adaptionenregeln verarbeiten zu können. Der Ansatz wurde im Rahmen eines Szenarios aus dem Bereich der Gesundheitsversorgung evaluiert.

Mit der Verknüpfung von Workflow Engines mit Context Engines beschäftigten sich bspw. [6]. Aufgrund der wachsenden Verbreitung des Einsatzes von Sensoren und drahtloser Kommunikation existiert eine Lücke zwischen der Vielzahl von relativ einfachen Kontexten und den intelligenten Workflow-Systemen. Mit der Einführung unterschiedlicher semantischer Niveaus von Kontexten versuchen die Autoren, diese Lücke zu schließen.

Die Modellierung von Geschäftsprozessen umfasst die Identifizierung einer zugehörigen Menge von Metadaten. Jedoch hängt diese Menge von der spezifischen Notation ab (z.B. BPMN [7]). Dies führt zu einer eingeschränkten Wiederverwendbarkeit von Prozessmodellen aufgrund verschiedener inkompatibler (Notations-)Standards. In [8] wird eine Architektur vorgeschlagen, die die Daten und Metadaten eines modellierten Prozesses in einer semantischen Abstraktion kartografisch darstellt.



4 Modellierung

In diesem Abschnitt wird beschrieben, wie sowohl Geschäfts- als auch Maschinenprozesse identifiziert und modelliert werden. Es folgt eine Beschreibung, wie Dienstleistungen auf oberster Ebene kombiniert werden können, wie man diese kombinierten Dienstleistungen konfigurieren kann und diese Dienstleistungen letztendlich auf bereits entwickelte und wiederverwendbare Prozessmodule abgebildet werden.

Prozess-Ebene

Modellierung und Management von Prozessen sind ein Hauptaspekt des vorgestellten Ansatzes. Die Identifikation und formale Beschreibung von Prozessen ist notwendig, um einen Überblick über die in einer spezifischen Anwendungsdomäne relevanten Prozesse zu erhalten. Zudem werden hierdurch die gezielte Ausführung und Überwachung, das Management sowie die kontinuierliche Verbesserung dieser Prozesse ermöglicht. Der innovative Aspekt des hier beschriebenen Ansatzes liegt in der Entwicklung einer service-orientierten Prozessarchitektur, innerhalb derer jeder Prozess oder Teilprozess verknüpft werden kann mit vordefinierten Dienstleistungen, die in einer spezifischen Anwendungsdomäne angeboten werden (ein Prozess wird hierbei als *eine* Dimension einer Dienstleistung betrachtet).

Darüber hinaus integriert der vorgestellte Ansatz Prozesse und Kontextsensitivität sowohl zur "build time" (Prozesskomposition basierend auf Dienstleistungs-konfiguration) als auch zur "run time" (Ausnahmebehandlung während Prozessausführung). In diesem Zusammenhang wurde der gebräuchliche Top-Down-Ansatz zur Prozessmodellierung durch Identifikation, Beschreibung und Integration spezifischer Kontextinformationen erweitert. Der Begriff *Prozesskontext* wird hierbei definiert als die Menge aller Faktoren der Prozessumgebung, welche die Konfiguration und/oder Ausführung eines Prozesses beeinflussen können. Diese Faktoren können unternehmensintern oder -extern sein, manche können in ihren Ausprägungen beeinflusst werden (z. B. Nachtanken bei niedrigem Treibstoffstand), andere nicht (z. B. Wetterlage). Der Begriff Prozesskontext ist hier abweichend von Definitionen für Kontext wie der von [14] zu betrachten. Kontextsensitivität wird im Wesentlichen in Bereichen wie Ambient Intelligence oder Location Based Services eingesetzt, wo sich entweder eine intelligente Umgebung an eine Entität (Benutzer) anpasst oder einer Entität (Benutzer) situationsabhängig Dienste angeboten werden. Im Prozesskontext jedoch steht als Entität kein Benutzer, sondern ein Prozess im Mittelpunkt, der sich autonom an Veränderungen der für ihn relevanten Umweltbedingungen während seiner Ausführung anpasst.

- Der Ausgangspunkt ist die Entwicklung einer *Prozesslandkarte* (Abbildung 1 oben), die einen Überblick über die Prozesse innerhalb einer spezifischen Anwendungsdomäne gibt, basierend auf der Beschreibung von Leistungsmodulen. Die identifizierten Prozesse werden hierbei in *Prozessklassen* eingeteilt (z. B. Management-, Unterstützungs-, Kernprozesse). Diese Prozesslandkarte mit ihren Klassen bildet die oberste Ebene einer *Prozesshierarchie*, d. h. den höchsten Abstraktionsgrad.
- Im nächsten Schritt werden sogenannte *Input-/Output-Diagramme* erstellt, welche die Schnittstellen zwischen den verschiedenen Prozessen der Prozesslandkarte beschreiben.
- Im nachfolgenden Schritt beginnt die eigentliche Ablaufmodellierung, zu Beginn auf grobgranularer Ebene. Innerhalb der nächsten Schritte erfolgt eine schrittweise Verfeine-

ung der jeweiligen komplexen Aktivitäten der nächsthöheren Ebene als einzelne Prozesse. Diese Verfeinerung endet auf einer Ebene, auf der ein Prozess nur noch aus atomaren Aktivitäten besteht.

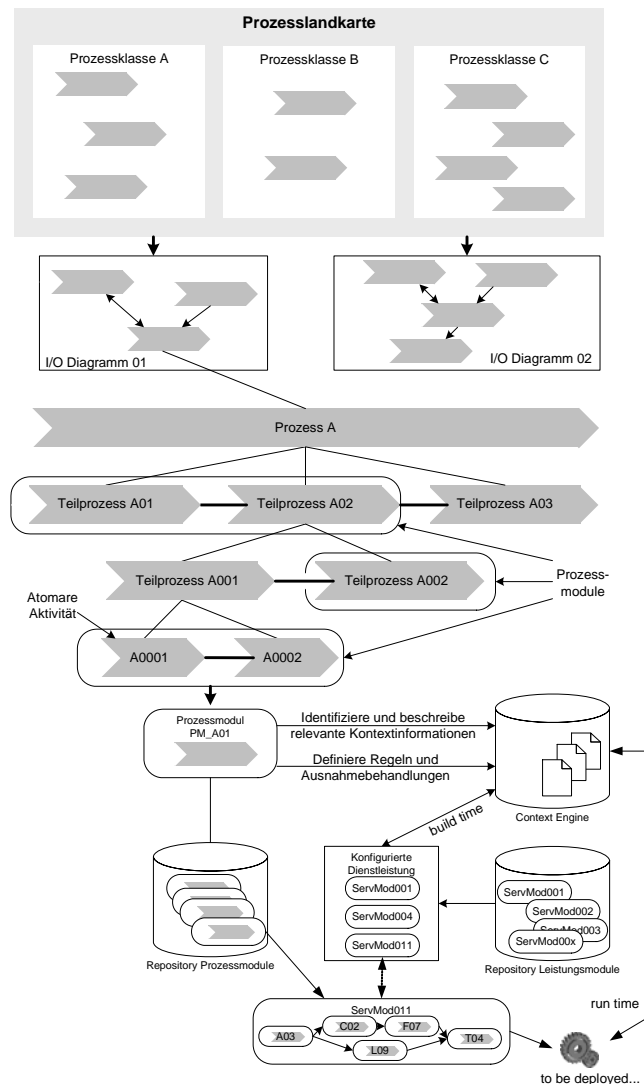


Abbildung 1 - Vorgehen bei der Prozessmodellierung

steht aus mindestens einer oder mehreren verknüpften Aktivitäten oder Teilprozessen, die eine spezifische Aufgabe erfüllen. Aktivitäten/ Teilprozesse eines Prozessmoduls können automatisierbar (z. B. Web Services), teilautomatisierbar oder manuell ausführbar sein. Die Verbindung dieser Prozessmodule mit Leistungsmodulen definiert die Schnittstelle mit der Dienstleistungsschicht: Jedes Prozessmodul und jedes Leistungsmodul verfügt über eine eindeutige Identifizierung (ID). Durch Verwendung dieser ID als Schlüsselattribut wird

Da im vorgestellten Ansatz WS-BPEL [9, 10, 11] für die Implementierung der Prozesse auf Technologie-Ebene verwendet wird, wurde beschlossen, für die graphische Beschreibung der Prozesse die Business Process Modeling Notation (BPMN) [7] einzusetzen.

Diese Entscheidung basiert auf folgenden Gründen: Erstens ist WS-BPEL eine XML-basierte Prozessmodellierungssprache, die für ungeschulte Anwender nicht intuitiv verständlich ist, wohingegen BPMN eine graphische Modellierungsnotation anbietet. Zweitens können (unter gewissen Bedingungen) BPMN-Diagramme in WS-BPEL Code transformiert werden. Somit wird die Zusammenarbeit zwischen Prozessmodellierung und Implementierung erleichtert. Drittens wurde dies auch im Hinblick darauf beschlossen, dass das Konzept von spezifischen Software-Lösungen unabhängig sein soll.

Nach Abschluss der Prozessmodellierung werden sogenannte *Prozessmodule* definiert, d. h. die modellierten Prozesse werden in wiederverwendbare Einheiten zerlegt, basierend auf ihren spezifischen Funktionalitäten. Ein Prozessmodul besteht



definiert, welche Prozessmodule notwendig sind für die Realisierung entsprechender Leistungsmodule. Einige Prozessmodule sind generisch und können daher von unterschiedlichen Leistungsmodulen verwendet werden. Die Konfiguration eines Leistungsmoduls durch eine Menge von Prozessmodulen basiert auf spezifischen, vordefinierten Konfigurationsregeln.

Wenn die Definition von Prozessmodulen abgeschlossen ist, wird mit der Integration des Kontextes auf Prozessebene begonnen. Für jedes Prozessmodul müssen die zugehörigen relevanten *Kontextinformationen* identifiziert und beschrieben werden, die Einfluss auf den Aufruf bzw. die Ausführung des jeweiligen Prozessmoduls haben können. Eine mögliche Beschreibung einer Kontextinformation beinhaltet Bezeichnung, Quelle, Format/Skala, Toleranzbereiche, Auswirkungen, Abhängigkeiten etc. Für jede Kontextinformation (oder Kombinationen verschiedener Kontextinformationen) müssen geeignete Ausnahmebehandlungen definiert werden. Hierdurch sollen mögliche Störfälle, die durch Erreichen oder Verletzen des Toleranzbereichs einer spezifischen Kontextinformation verursacht werden, vermieden bzw. kompensiert werden. Die Verknüpfung von Kontextinformationen mit ihren korrespondierenden Ausnahmebehandlungen erfolgt ebenfalls durch einen regelbasierten Ansatz. Die Anwender definieren hierfür spezifische *Event-Condition-Action-Regeln* im Sinne von "IF *Toleranzbereich von Kontextinformation x ist verletzt*, THEN *Ausnahmebehandlung y muss aufgerufen werden*". Für die Beschreibung von Prozesskontextinformationen soll hierbei eine Prozesskontext-Ontologie entwickelt werden, die für weitere Anwendungsdomänen erweiterbar sein soll.

Die Erfassung von Kontextinformation erfolgt durch unterschiedliche Technologien (z. B. Sensoren, Internet-Dienste). Das Verwalten der Kontextinformationen wird durch ein *Kontextmanagementsystem* geregelt. Diese Anwendung verfügt über Schnittstellen zu den unterschiedlichen Technologien, die zur Erfassung der Kontextinformationen eingesetzt werden. Innerhalb des Kontextmanagementsystems werden die unterschiedlichen Datenformate konvertiert in ein gemeinsames Format. Das System überwacht die erfassten Daten hinsichtlich der Einhaltung von Toleranzbereichen mittels einer *Rules Engine (Ablaufumgebung für Regeln)*. Wenn ein Toleranzbereich erreicht oder verletzt wird, ruft die Rules Engine eine geeignete Ausnahmebehandlung auf.

Sämtliche Schritte des vorgestellten Ansatzes müssen in enger Zusammenarbeit mit Experten aus der jeweiligen Anwendungsdomäne durchgeführt werden (z. B. mittels Interviews und Workshops), da nur diese über das notwendige Fachwissen hinsichtlich der zu modellierenden Prozesse und Dienstleistungen sowie über die entsprechende Erfahrung hinsichtlich des Einflusses spezifischer Kontextinformationen auf Prozesskonfiguration und –ausführung verfügen.

Dienstleistungsebene

Nach einem detaillierteren Blick auf die Prozessmodellierung fokussiert dieser Abschnitt nun die Ebene der Dienstleistungen. Dabei wird ein Konzept eingeführt, wie Dienstleistungen dekomponiert werden können in wiederverwendbare, konfigurierbare *Leistungsmodule*. Darauf basierend wird gezeigt, wie diese konfigurierten Leistungsmodule auf Prozessmodule abgebildet werden.

Zu Beginn werden alle Dienstleistungen in Basisleistungen, Leistungsspezifikationen und Zusatzleistungen dekomponiert und strukturiert. Die Anwendung *Konfigurator* ermöglicht es, Leistungsmodul auszuwählen und zu kompletten Dienstleistungen zu kombinieren. Die Konfigurationslogik definiert, welche Leistungsmodul verfügbar sind und wie sie jeweils miteinander kombiniert werden können. Die Verfügbarkeit eines Leistungsmodul hängt in erster Linie von der Verfügbarkeit einer oder mehrerer Ressourcen ab. Daher werden die Beziehungen zwischen Ressourcen (wie z.B. Maschinen, Mitarbeiter oder Endgeräte) und Leistungsmodul abgebildet. Dabei ist zu beachten, dass kein Leistungsmodul einzeln anwendbar ist. Es muss immer zumindest eine Kombination aus einer Basisleistung und einer Leistungsspezifikation vorliegen. Zusatzleistungen sind, wie ihre Bezeichnung schon sagt, ebenfalls nicht unabhängig, sondern mögliche Ergänzungen/Erweiterungen einer ausgewählten, spezifizierten Basisleistung. In diesem Zusammenhang gibt es eine Menge von Einschränkungen für die Kombination von Leistungsmodul.

Die Leistungsmodul und ihre zugehörigen Prozessmodul werden dabei in einer Weise entwickelt, welche die Rekombinierbarkeit der Modul ohne jegliche weitere Modellierung oder Implementierung gewährleistet.

5 Architektur und Technologie

An dieser Stelle wird zunächst auf die Architektur und die einzelnen Komponenten im R2B-Ansatz eingegangen. Im Anschluss daran folgt eine Beschreibung verschiedener System-schichten und wie diese aus der Modellierung abgeleitet wurden.

Architektur

Wie bereits erwähnt, ist ein wesentliches Ziel des Projektes die Integration von Geschäftsprozessen mit Maschinen-Verwaltungsprozessen. Bei der Entwicklung der Architektur wurden somit eine Management-Instanz und eine Member-Instanz definiert. Die Management-Instanz ist zentral im Backoffice angesiedelt. Die Member-Instanz repräsentiert mobile Einheiten. Die Kommunikation zwischen beiden Instanzen erfolgt (sofern durch Netzabdeckung möglich) über WLAN oder GPRS.

Abbildung 2 gibt einen Überblick über die Architektur. Im vorherigen Kapitel wurde erläutert, dass Kontextinformationen sowohl zur „build time“ als auch zur „run time“ betrachtet werden. Die bei der Prozessmodellierung in Abbildung 1 aufgeführte Context-Engine findet sich in Abbildung 2 daher mit der „build time“-Funktionalität in der Management-Instanz (als Rules-Engine im Konfigurator) und mit der „run time“-Funktionalität in der Member-Instanz.

Die durch den Konfigurator erzeugten konfigurierten Dienstleistungen werden in einer verteilten Datenhaltung abgelegt. Jede mobile Einheit ist somit in der Lage, die für sie bestimmten Aufträge respektive konfigurierten Dienstleistungen abzuarbeiten.

Auf den mobilen Einheiten bzw. der Member-Instanz interagieren zwei wichtige Komponenten miteinander: Die „run time“-Context Engine überwacht permanent Sensorenwerte und Aktorenzustände und überprüft diese auf Gültigkeit (d. h. ob sich die Werte in einem gültigen Intervall befinden). Parallel läuft auf der Member-Instanz eine BPEL-Engine. Diese besteht

aus einer „Process Execution“ Komponente und einer Kontrollkomponente bestehend aus „Process Monitoring“ und Master Prozess. Die „Process Execution“ Komponente kann (basierend auf einer konfigurierten Leistung) aus einem Repository Prozessmodule entsprechende BPEL-Prozesse laden und ausführen.

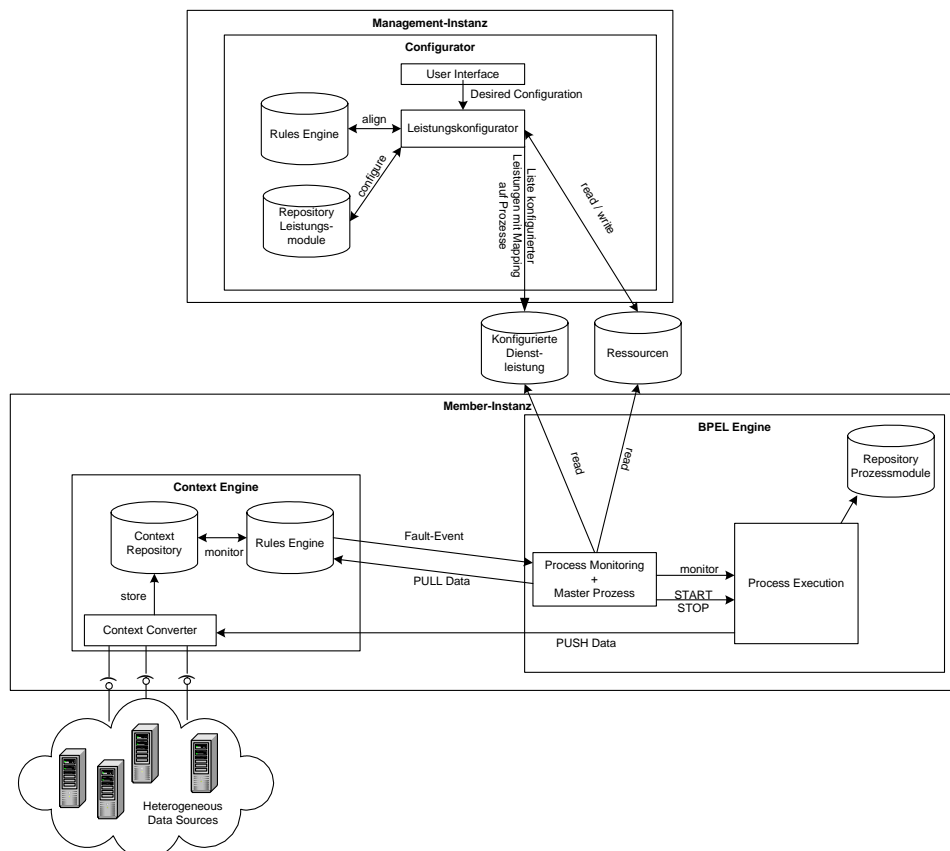


Abbildung 2 - Architektur mit Member und Management-Instanz

Die wichtigste Komponente auf der Member-Instanz ist der sogenannte *Master-Prozess*. Er ist dafür verantwortlich, dass Prozesse auf der BPEL-Engine durch Zugriff auf die zuvor abgelegten konfigurierten Dienstleistungen sowie auf die Ressourcen-Datenhaltung, instanziiert werden können. Die Auswahl der zu startenden Prozesse basiert ferner auf den Kontextinformationen, die der Master-Prozess von der „run time“-Context Engine über einen PULL-Mechanismus erhält. Dies umfasst sowohl „Rohdaten“ als auch bereits interpretierte Kontextinformationen. Beim Auftreten einer Ausnahme ist die Context Engine ihrerseits in der Lage, ein FAULT-Event an den Masterprozess zu senden, um eine geeignete Ausnahmebehandlung einzuleiten.

Über eine Process-Monitoring-Benutzerschnittstelle ist ein Benutzer in der Lage sowohl laufende oder abgebrochene Prozesse zu beobachten als auch neue Prozesse manuell zu starten.

Schichtenmodell

Innerhalb des folgenden Abschnitts werden die Systemschichten kurz erläutert. Diese basieren auf den vorgegebenen Ebenen der Modellierung. Wie bereits erwähnt, sind die mobilen Einheiten mit dem Backoffice über WLAN bzw. GPRS verbunden. Die Instanzen befinden sich in Abbildung 3 auf der linken Seite (Member-Instanz) mit zwei Schichten und auf der rechten Seite (Management-Instanz) mit drei Schichten.

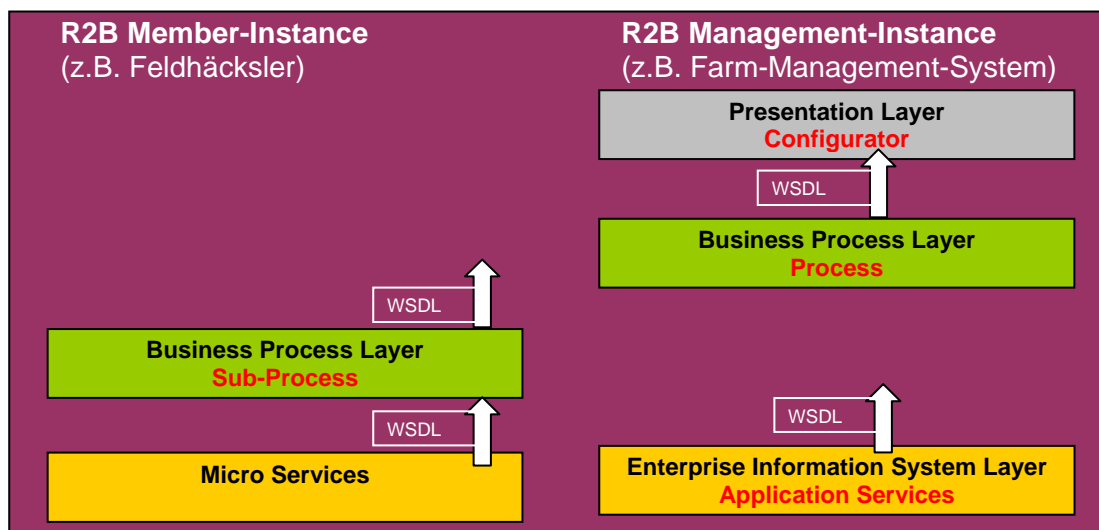


Abbildung 3 - Softwareschichtenmodell von Member- und Management-Instanz

Zunächst wird die Management-Instanz betrachtet. Auf oberster Ebene ist der Konfigurator angesiedelt. Er ist in der Lage, Leistungsmodule zu kombinieren und zu konfigurieren. Konfigurierte Dienstleistungen sind einzelnen BPEL-Prozessen zugeordnet. Letztere werden dann in der Schicht Business Process Layer zur Ausführung gebracht. Innerhalb des Enterprise Information System Layers sind Applikationen angesiedelt, die im Unternehmen bereits vorhanden sind. Diese Systeme werden mit WSDL-Schnittstellen ausgestattet und setzen somit eine Applikations-Integration um.

Ähnlich wie auf der Management-Instanz ist auch auf der Member-Instanz eine BPEL-Engine angesiedelt, um Prozesse auszuführen (BPEL-Engine auf Management-Instanz ist aus Gründen der Übersichtlichkeit nicht in Abbildung 2 dargestellt). Obwohl dieselbe Technik verwendet wird, ist diese Schicht dem Business Process Layer auf der Management-Instanz untergeordnet, da auf der mobilen Einheit Prozesse von sehr viel geringerer Komplexität ablaufen.

Ferner existiert auf der mobilen Einheit eine Schicht namens Micro-Services. Hier laufen Dienste die Funktionalität (über WSDL Schnittstellen) höher angesiedelten BPEL-Prozessen zur Verfügung stellen.

Für die Umsetzung der beiden untersten Ebenen wurde sowohl für die Management- als auch für die Member-Instanz ein Open Services Gateway Initiative-Framework (OSGi) [12] gewählt. Durch sogenannte Bundles besteht die Möglichkeit, Funktionalität "on Demand" zu installieren, zu starten, zu stoppen und ggf. durch eine neuere Version zu ersetzen. Aus der Menge der OSGi-Implementierungen fiel aus Gründen der Stabilität die Entscheidung für das Equinox-Framework [13].



6 Anwendungsfälle

Die oben beschriebene Vorgehensweise wird anhand zweier unterschiedlicher Anwendungsszenarien evaluiert. Im Szenario *Flexible Wartung von IT-Infrastrukturen* soll die Unterstützung von Service-Personal mittels kontextsensitiver Kommunikation erprobt werden. Durch die kontextsensitive Interaktion der beteiligten Systeme soll eine Verbesserung der Service-Qualität erreicht werden.

Das zweite Szenario *Grünfütterernte* ist in der Landwirtschaft/Landtechnik angesiedelt. Hier sollen sich Informationsflüsse und Prozesse mit Hilfe der dargestellten Vorgehensweise weitgehend selbst organisieren. Vereinfachte Bedienung und flexible Anpassung an bestimmte Situationen im Arbeitsablauf und eine Senkung von Stückkosten (€t Erntegut) werden hierbei angestrebt.

Im Gegensatz zum industriellen Umfeld können in der Landwirtschaft aufgrund von Umwelteinflüssen Prozesse nur eingeschränkt geplant und gesteuert werden. Mit Hilfe des beschriebenen Ansatzes sollen sich diese Prozesse jedoch durch Einbeziehung des relevanten Kontextes automatisieren lassen.

Das Prinzip wird nachfolgend an einem Beispiel beschrieben: Landwirte dokumentieren ihre durchgeführten Arbeiten in einem Schlagkartei-System, dem Arbeitstagebuch. Darin werden Daten zu den durchgeführten Arbeiten dokumentiert. Diese Daten bestehen aus der Arbeitsbeschreibung, Datum, Dauer, eingesetzte Maschinen, verbrauchte Mittel und geerntete Stoffe. In einer anschließenden Auswertung werden die Kosten für durchgeführte Maßnahmen auf das jeweilige Feld bzw. die jeweilige Frucht bezogen ermittelt. Daher sind die erfassten Daten eine wichtige Grundlage für die betriebswirtschaftliche Bewertung.

Typischerweise werden solche Daten manuell erfasst. Der Landwirt zeichnet sie mit Hilfe eines Notizbuches oder eines PDA auf und überträgt sie in das Schlagkartei-System. Aufgrund hoher saisonaler Arbeitsbelastung sind diese Aufzeichnungen häufig unvollständig und der Transfer in die Schlagkartei erfolgt stark zeitverzögert. Daraus resultiert häufig eine für die Unternehmensplanung und -steuerung unzureichende Datenbasis.

Der R2B-Ansatz automatisiert diese Dokumentationsarbeiten weitgehend. Dazu wird zunächst eine Dienstleistung konfiguriert, die folgende Bestandteile enthält:

- Basisleistung: z.B. Mähen
- Leistungsspezifikation: z.B. nach Fläche
- Zusatzleistung: z.B. Automatisierte Buchung der Arbeitserledigung

Die konfigurierte Leistung legt fest, welche Prozessdaten (i.d.R. Sensordaten) aufzuzeichnen und wie diese zu interpretieren sind. Auf dieser Grundlage werden Prozesse ausgewählt und von den ausführenden Maschinen abgerufen.

Wenn sich die Konfiguration zu einem späteren Zeitpunkt dynamisch ändert, etwa durch Hinzufügen bzw. Entfernen einer Ressource, so passt sich ebenfalls die Ausführung dynamisch zur Laufzeit an. Angenommen ein neues Verfahren *Häckseln* wird durch die geänderte Konfiguration aktiviert. Dann werden relevante Prozesse deployed und verhalten sich entsprechend dem ermittelten Prozesskontext. Ein wichtiger Prozesskontext ist die Schlagerkennung. Sobald die Maschine selbstständig feststellt, dass sie sich auf dem Feld befindet (durch den Vergleich der aktuellen GPS-Position mit einem bekannten Feld-Polygon (Grenzlinie)), wird die Aufzeichnung der entsprechenden Daten automatisch gestartet. Entsprechend festgelegter Kommunikationsregeln werden Daten zu einem Backend-System übermittelt. Dort werden die Prozessdaten interpretiert und ein Buchungsvorschlag erstellt.

Dieser Buchungsvorschlag enthält Informationen darüber, welche Fläche die Maschine in einem bestimmten Zeitraum bearbeitet hat. Dieser Wert wird in Beziehung zu den in der Schlagkartei hinterlegten Kostensätzen gesetzt. Der Landwirt verfügt nun über einen Arbeitsbericht, ohne Daten dafür manuell erfassen haben zu müssen.

Neben dem beschriebenen Beispiel sind zahlreiche weitere Anwendungen denkbar, die auf dieser technologischen Grundlage beruhen. Diese Anwendungen unterstützen nicht nur die Dokumentation, sondern auch Planung, Überwachung und Steuerung von landwirtschaftlichen Arbeitsvorgängen.

Die Möglichkeit zur Wiederverwendbarkeit und Rekombination von Prozessmodulen schafft die Basis für neue informationstechnische Dienste auch in anderen Anwendungsdomänen, z.B. im Baustellenbetrieb oder der Industrieautomation.

7 Demonstrator

Die im vorangegangenen Kapitel beschriebene Architektur und die prinzipielle Machbarkeit des Vorhabens wurden durch einen Demonstrator evaluiert. Abbildung 4 zeigt den Demonstrator, ausgestattet mit einem Industrie-PC: Siemens SIMATIC MicroBox, GPRS-Antenne (hinten), WLAN-Modul und Motorradbatterie (im Anhänger) für die Stromversorgung.



Abbildung 4 - Demonstrator mit installierten Softwarekomponenten auf einem Industrie-PC

Hierzu wurden zunächst mittels GPS zwei Flächen eingemessen (Feld und Hof) und die Polygone dem Fahrzeug verfügbar gemacht. Der Prozess „Flächen-Identifizierung“ prüft nun, ob die ermittelte GPS-Position im vorgegebenen Feld-Polygon liegt. Trifft das zu, wird die Aufenthaltsdauer (synonym für die Dauer der Arbeit) aufgezeichnet. Auf die gleiche Weise erfolgt die anschließende Identifikation des Hof-Polygons bei der Heimkehr des Fahrzeuges. Jedoch wird hier die Übertragung der zuvor erfassten Daten zum Backend ausgelöst und ein „Arbeitsbericht“ erstellt.



8 Literatur

- [1] Dey, A.; Abowd, G.; Salber, D.: *A Context-Based Infrastructure for Smart Environments*. In: GVU Technical Report GIT-GVU-99-39, Georgia Institute of Technology, 1999
- [2] Haryanto, R.: *Context Awareness in Smart Homes to Support Independent Living*, August 2005, Master-Arbeit.
- [3] Eikerling, H.; Benesch, M.; Berger, F.; Bublitz, S.: *Context-aware integration of Collaboration Services for the Support of Mobile Field Operations*, Challenges in Collaborative Engineering (CCE'07) Kraków, Poland, 2007
- [4] Ardissono, L.; Furnari, R.; Goy, A.; Petrone, G.; Segnan, M.: *A Framework for the Management of Context-Aware Workflow Systems*. In: Proceedings of the Third International Conference on Web Information Systems and Technologies (WEBIST 2007), S. 80-87, 2007
- [5] Ardissono, L.; Furnari, R.; Goy, A.; Petrone, G.; Segnan, M.: *Context-Aware Workflow Management*. In: Baresi, L.; Fraternali, P.; Houben, G. (Hrsg.): Proceedings of the Seventh International Conference on Web Engineering (ICWE 2007), LNCS 4607, S. 47-52, Springer, 2007
- [6] M. Wieland, M.; Kaczmarczyk, P.; Nicklas, D.: *Context Integration for Smart Workflows*, Sixth Annual IEEE International Conference on Pervasive Computing and Communications, 2008
- [7] Object Management Group: *Business Process Modeling Notation V1.1*, OMG Available Specification; 17.01.2008; <http://www.omg.org/docs/formal/08-01-17.pdf>
- [8] S. Dietze, S.; Gugliotta, A.; Domingue, J.: *Context-aware Process Support through Automatic Selection and Invocation of Semantic Web Services*, IEEE International Conference on Service-Oriented Computing and Applications (SOCA'07), 2007.
- [9] OASIS: *Web Services Business Process Execution Language Version 2.0 Specification*; 11.04.2007; <http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html>
- [10] Agrawal, A.; Amend, M.; Das, M.; Ford, M.; Keller, C.; Kloppmann, M.; König, D.; Leymann, F.; Müller, R.; Pfau, G.; Plösser, K.; Rangaswamy, R.; Rickayzen, A.; Rowley, M.; Schmidt, P.; Trickovic, I.; Yiu, A.; Zeller, M.: *WS-BPEL 2.0 Extensions for People (BPEL4People), Version 1.0*, Juni 2007, http://download.boulder.ibm.com/ibmdl/pub/software/dw/specs/ws-bpel4people/BPEL4People_v1.pdf
- [11] WS-Human Task: http://download.boulder.ibm.com/ibmdl/pub/software/dw/specs/ws-bpel4people/WS-HumanTask_v1.pdf
- [12] Open Services Gateway Initiative (OSGi) – <http://www.osgi.org/Release4/HomePage>
- [13] Equinox OSGi framework – <http://www.eclipse.org/equinox/>
- [14] Dey, A.: Understanding and Using Context. In: Personal and Ubiquitous Computing, Volume 5, Issue 1, S. 4-7, 2001

9 Danksagung

Die vorgestellte Arbeit wird von dem Bundesministerium für Wirtschaft und Technologie (BMWi) im Rahmen des Projektes „Robot to Business“ (R2B) gefördert. R2B ist Teil des SimoBIT Forschungsprogramms (Sichere mobile Informationstechnik in Mittelstand und Verwaltung; www.simobit.de) und steht unter der Trägerschaft des Deutschen Zentrums für Luft und Raumfahrt (DLR).