

Leicht Konfigurierbare Komponenten Kollaborativer Systeme (LK³S)

Prof. Dr. rer. nat. Uwe Schmidtman¹, Prof. Dr.-Ing. Clemens Westerkamp²,
Prof. Dr.-Ing. Jürgen Wübbelmann², Dipl.-Inf (FH) Martin Landwehr²,
¹Institut für Informatik, Automatisierungstechnik und Robotik,
²Fachhochschule Osnabrück, Laborbereich Technische Informatik,

Abstract: Das Projekt Leicht Konfigurierbare Komponenten Kollaborativer Systeme (LK³S [1], gefördert vom BMBF) befasst sich mit der Dezentralisierung von Entscheidungsstrukturen in der flexiblen Fertigung. Mit Industriepartnern wie Phoenix Contact, Schneider Electric, Beka Engineering und Willert Software Tools erarbeiten die Fachhochschulen Emden und Osnabrück ein teilweise dezentrales Automatisierungskonzept unter Nutzung von SW-Agententechnologie. Ausgehend von herkömmlichen Automatisierungsansätzen und -plattformen wird untersucht, wie durch automatisierte Vorortentscheidungen nach vorgegebenen und sich rasch ändernden Optimierungskriterien eine Flexibilisierung der Fertigung erreicht werden kann. Dabei wurde der Prototyp einer Laufzeitumgebung für eine agentenbasierte Fertigung geschaffen, der sowohl für leistungsstarke Industrierechner als auch für kleine eingebettete Systeme geeignet ist.

I. Motivation:

Die Globalisierung wird begleitet durch schnellen Daten- und Informationsaustausch im weltweiten Internet und dem preiswerten Gütertransport der Containerschifffahrt und führt somit zu einer ‚Flat World‘ [2]. Aus Sicht der etablierten Märkte wird der neue Weltmarkt durch Millionen neuer Konsumenten aber auch durch Millionen neuer Anbieter von Waren und Dienstleistungen bestimmt werden. Die laufende Transformation zu einem flachen, neuen Weltmarkt kann von Unternehmen erfolgreich bewältigt werden, wenn sie sich alle Ressourcen für neue innovative Produkte wie auch für neue Verfahren der Produktion erschließen [3,4]. Mit dem Begriff „kollaborative Arbeits-Umgebungen“ wird ein weites Forschungsgebiet bezeichnet, das optimale Strukturen und Methoden derartiger vernetzter Strukturen ergründen soll und das nahezu alle wissenschaftlichen Disziplinen anspricht. Das LK³S-Projekt befasst sich mit dem Teilaspekt der flexiblen Fertigung und versucht hierzu von unten nach oben (*bottom-up*) eine Basis zu schaffen, auf der eine flexible Fertigung aufbauen kann. Die zukünftige Fertigung muss flexibel auf individuell gestaltete Fertigungsaufträge reagieren können. Zu diesem Zweck sind in den letzten Jahren neue Konzepte auf Basis der Agententechnologie entwickelt worden, deren Brauchbarkeit teilweise an akademischen Modellen demonstriert wurden.

II. Grundlagen: Software-Agenten / FIPA / JADE

Das zunehmende Wachstum der intelligenten Fähigkeiten von Computern erlaubt die Entwicklung von persönlichen Assistenten/Agenten, die in der Lage sind, ihrem Benutzer Aufgaben abzunehmen. Es existiert noch keine allgemein anerkannte Definition von Agenten da der Begriff in einem weiten Feld der interdisziplinären Forschung und Entwicklung genutzt wird. Häufig wird jedoch die mehrfach überarbeitete Definition von Jennings und Wooldridge [5] verwendet:

„An agent is an encapsulated computer system that is situated in some environment and that is capable of flexible, autonomous action in that environment in order to meet its design objectives.“

Agieren Agenten in den virtuellen Welten von Computern und Netzwerken, werden sie Softwareagenten genannt.

Foundation for Intelligent Physical Agents (FIPA)

Um heterogene Agentensysteme in die Lage zu versetzen, sich gegenseitig zu lokalisieren und miteinander zu kommunizieren, werden Standards und Spezifikationen benötigt. Die FIPA [6] mit Sitz in Genf beschäftigt sich seit 1996 mit entsprechenden Richtlinien und Spezifikationen. Die FIPA ist seit 2005 eine IEEE Computer Society Standards Organization. Viele der Standards wurden in 2002 festgelegt und sind die Grundlage vieler heute bestehender Agentensysteme. Die FIPA hat dabei viele Aspekte von Agentensystemen festgelegt. Neben Festlegung der abstrakten Architektur von Agenten wurden hauptsächlich die Gebiete Agentenkommunikation, Agentenmanagement und Agentennachrichtentransport standardisiert [7]. Softwareagenten kommunizieren untereinander mittels Nachrichten. Die Kommunikationssprache der FIPA, die **Agent Communication Language (ACL)**, ist ein Standard der FIPA97-Spezifikation.

Java Agent DEvelopment framework (JADE)

Das Agentensystem JADE ist eines der weitest verbreiteten Agentensysteme derzeit [8]. Es setzt auf Java auf und kann somit auf jeder Plattform, die Java unterstützt, eingesetzt werden. Der Einsatz von JADE in der Forschung ist weit verbreitet und aufgrund der allgemeinen Akzeptanz wurde es auch in LK³S eingesetzt.

III. Das LK³S-Projekt

A) LK³S-Konzept

Mit dem Ziel, eine flexible Fertigung zu unterstützen, verfolgt das LK³S-Projekt ein Konzept, mit dem beispielsweise ein Anlageningenieur in die Lage versetzt werden soll, entscheidungsrelevante Teile des Systems komfortabel zu erstellen und sich auf seine Kernziele zu konzentrieren. Die LK³S-Konfigurationssoftware setzt dabei lediglich Basis-UML-Kenntnisse voraus. Mit Hilfe einer integrierten Simulation bietet sich die Möglichkeit, Fehlerquellen und Verbesserungspotential schnell zu erkennen und Fertigungs- und Logistikprozesse zu optimieren. Das typische Vorgehen zeigt Abbildung 1.

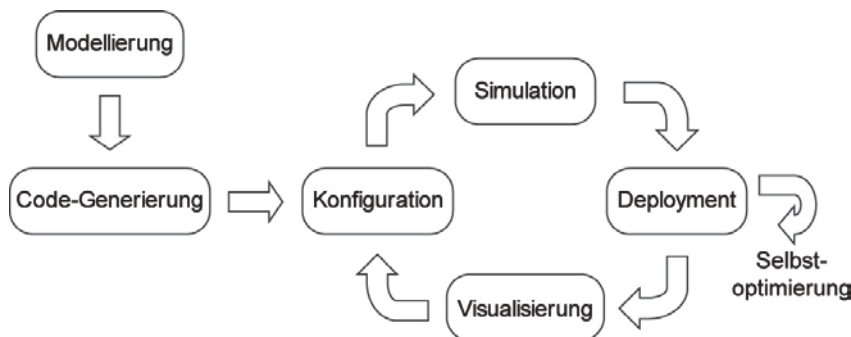


Abbildung 1: LK³S-Konzept und Workflow

Zur Konfigurierung und Generierung der entscheidungsfähigen Systeme werden UML-basierende Ansätze verwendet. Dabei wird schrittweise vom ersten Konfigurieren des Systems bis zur Quellcodegenerierung und Weiterverarbeitung vorgegangen. Für die Konfigurierung des Systems wurde im Rahmen des Forschungsprojekts ein UML-basiertes Werkzeug entwickelt, mit dem ein bestehendes, heterogenes System konfiguriert werden kann. Dazu werden verschiedene UML-Diagramme, schwerpunktmäßig Anwendungsfall- und Aktivitätsdiagramme, verwendet. Der Kerngedanke liegt an dieser Stelle bei der leichten Konfigurierbarkeit, so dass keine Programmierkenntnisse für eine Anpassung an das System benötigt werden. Dies bedeutet, dass der Anwender ein im LK³S-Projekt entwickelten Konfigurator verwendet.

B) LK³S-Architektur

Die LK³S-Plattform bietet verschiedene Software-Komponenten, die auf PCs, Soft-SPS, Embedded Systems, Stapler-Terminals etc. ausgeführt werden können. Abbildung 2 illustriert das Zusammenspiel.

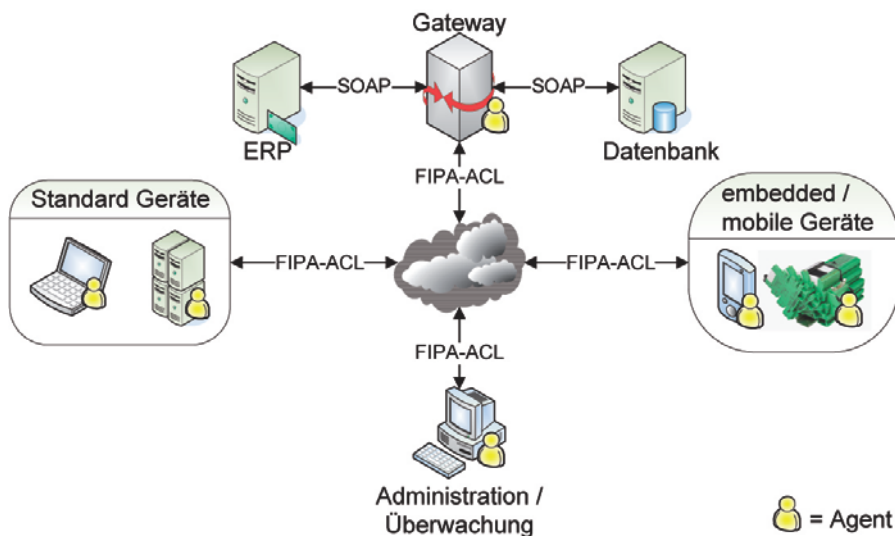


Abbildung 2: LK³S-Architektur

Jede Einheit, die durch einen SW-Agenten repräsentiert wird, kommuniziert auf Basis des FIPA-ACL-Standards. Im oberen Bereich sind die zentralen Systeme zu sehen. ERP-Systeme, Datenbanken/PPS etc. werden über Web Services an einem LK³S-Gateway angebunden. Im Rahmen des Projektes wurde ein Prototyp entwickelt, der das Zusammenspiel zwischen Intralogistik eines Zulieferers (Palettenlager) und einer digitalen Fabrik mit LK³S-Komponenten zeigt. Ziel des Projektes ist es aber auch, eingebettete Systeme in das Gesamtkonzept zu integrieren. In der Regel ist der Einsatz einer Java-Laufzeitumgebung auf einem eingebetteten System nicht möglich, so dass sich JADE als Agentensystem nicht im gesamten LK³S-Szenario eignet. Für eine nahtlose Integration von eingebetteten Systemen in die LK³S Plattform war es wichtig, dass die Interoperation zwischen JADE und Systemen mit eingeschränkten Ressourcen möglich ist [9]. Hierzu wurde in ANSI C ein Agentensystem entwickelt [10]. Das **LK³S Embedded Agent Framework (LEAF)** kann somit auch auf Geräten eingesetzt werden, für die keine Java-Ausführungsumgebung und damit auch kein JADE-Agentensystem verfügbar ist.

C) LK³S-Konfigurator

Der LK³S-Konfigurator enthält vordefinierte, an entscheidungsfähige Systeme angepasste UML-Bausteine, mit denen das System konfiguriert werden kann. Die Bausteine beschreiben unter anderem das Verhalten der einzelnen Systemkomponenten, sowie das Senden und Empfangen von Nachrichten. Des Weiteren können Entscheidungsfindungen dargestellt werden, die Faktoren wie Metriken enthalten. Das folgende Beispiel zeigt ein

Aktivitätsdiagramm in dem das Verhalten eines Agenten beschrieben wird (hier: Drehzahlsteuerung eines Motors auf einem eingebetteten System).

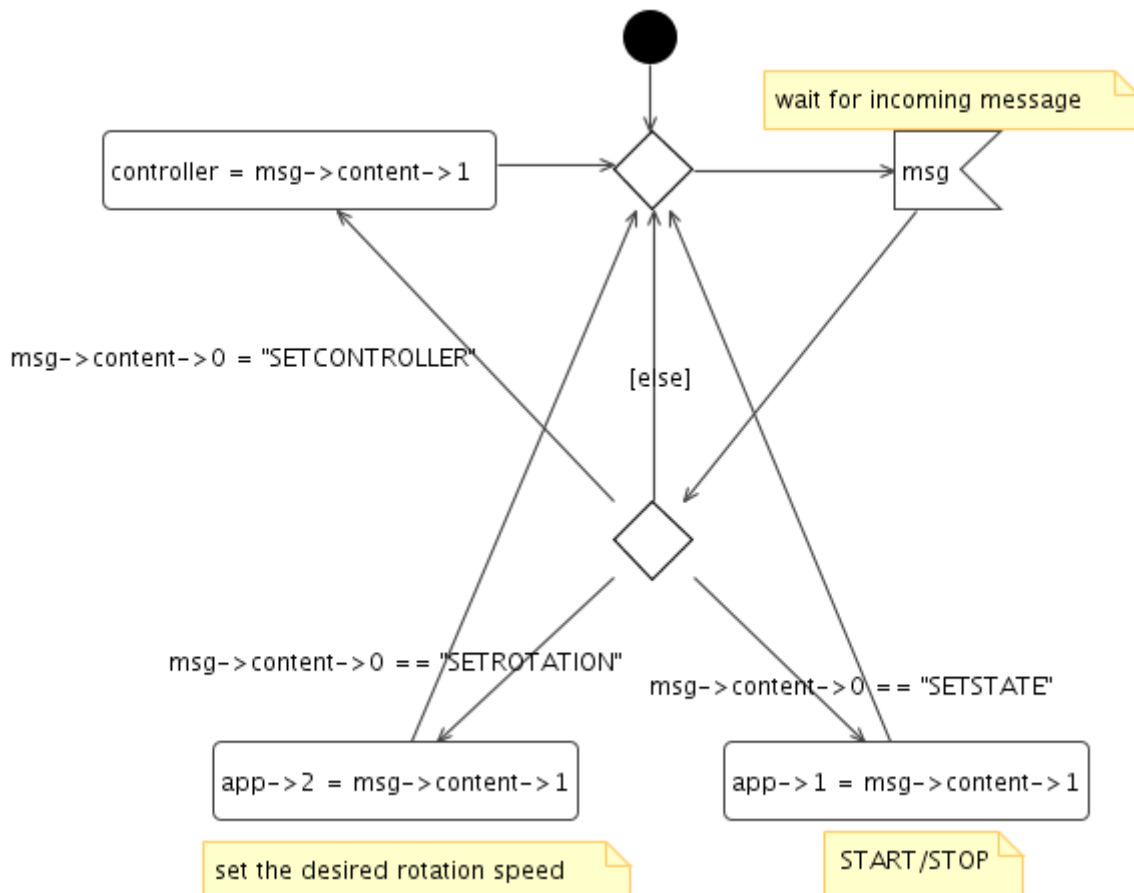


Abbildung 3: Aktivitätsdiagramm im LK³S-Konfigurator

Das in Abbildung 3 dargestellte Aktivitätsdiagramm beschreibt das Teilverhalten des Agenten für den Bereich des Empfangens und Senden von Nachrichten.

Der Inhalt einer Nachricht enthält Einträge für Befehle (im Beispiel gibt es drei verschiedene Befehle) und Parameter, mit denen die Motorapplikation konfiguriert und gesteuert werden kann. In anderen Aktivitätsdiagrammen werden weitere Verhaltensmuster des Agenten beschrieben, beispielsweise das Verhalten für die Drehzahlüberwachung oder die Vorgehensweise für das Senden eines Fehlerberichtes bei Überschreitung einer kritischen Drehzahlgrenze.

Aus dem Aktivitätsdiagramm erstellt der Konfigurator den in Abbildung 4 dargestellten XML-Code in einer für UML standardisierten Syntax. Der Wurzelknoten repräsentiert das gesamte Aktivitätsdiagramm und enthält als Kindelemente die Knoten des Diagramms. Diese Knoten

können vom Typ Start, Action, Send, Receive, Branch, Merge oder End sein. Abhängig von ihrem Typ enthalten die Knoten weitere Parameter. Außer dem End-Knoten haben alle Knoten Kindelemente, welche die Verbindungen zu Folgeknoten repräsentieren.

```

- <activity:Activity xmi:version="2.0" startNode="//@nodes.0">
  - <nodes xsi:type="activity.Start">
    <connection target="//@nodes.1"/>
  </nodes>
  - <nodes xsi:type="activity.Merge">
    <connection target="//@nodes.2"/>
  </nodes>
  - <nodes xsi:type="activity.Receive" content="msg" type="TMFORM" language="lk3s-easy">
    <connection target="//@nodes.3"/>
  </nodes>
  - <nodes xsi:type="activity.Branch">
    <connections expression="msg->content->0 == "SETSTATE"" target="//@nodes.4"/>
    <connections expression="msg->content->0 == "SETROTATION"" target="//@nodes.5"/>
    <connections expression="msg->content->0 == "SETCONTROLLER"" target="//@nodes.6"/>
    <connections expression="[else]" target="//@nodes.1"/>
  </nodes>
  - <nodes xsi:type="activity.Action" name="app->1 = msg->content->1">
    <connection target="//@nodes.1"/>
  </nodes>
  - <nodes xsi:type="activity.Action" name="app->2 = msg->content->1">
    <connection target="//@nodes.1"/>
  </nodes>
  - <nodes xsi:type="activity.Action" name="controller = msg->content->1">
    <connection target="//@nodes.1"/>
  </nodes>
</activity:Activity>

```

Abbildung 4: generierte Agenten-Konfiguration

Auf Basis einer aus dem Konfigurator erstellen XML-Beschreibungsdatei und jeweils für den entsprechenden Einsatz vorgefertigte Agenten-Templates können somit Agenten für das LK³S-Framework erzeugt werden.

D) LK³S Embedded Agent Framework (LEAF)

Für den Einsatz von Agenten auf Systemen mit beschränkten Ressourcen wurde ein eigenes Framework entwickelt. Um die Kompatibilität zu anderen Agentensystemen zu wahren, wurde für die Kommunikation zwischen den Agenten der IEEE FIPA-ACL Standard eingehalten. Somit ist eine nahtlose Integration von eingebetteten Systemen in die LK³S-Architektur gegeben.

Wie Agenten im JADE-Framework sind auch die LEAF-Agenten nicht echtzeitfähig, sie dienen allerdings zur Steuerung und Konfiguration echtzeitfähiger Komponenten innerhalb der Embedded Systeme.

Zur Erstellung von LEAF-Agenten gibt es verschiedene Möglichkeiten: Es ist möglich, Agenten in ANSI-C zu programmieren und zusammen mit dem Framework auf die Embedded Plattform zu übertragen. Dies bedeutet allerdings auch, dass dieser Agent fest mit dem Framework verbunden ist und somit zur Laufzeit nicht ausgetauscht oder verändert werden kann.

Eine andere Möglichkeit ist es, den Agenten mit dem LK³S-Konfigurator in UML zu beschreiben. Aus diesen UML-Diagrammen wird LEAF-spezifischer Bytecode generiert, der mit der LK³S- Plattform auf dem Embedded System ausgeführt werden kann. Dieser Bytecode wurde speziell für die Ausführung von Agenten entwickelt, da es für die sehr speziellen Randbedingungen und Anforderungen gerade im Bezug auf die FIPA-konforme Kommunikation keine bestehenden Alternativen gibt. Der Vorteil dieser Variante ist, dass Agenten auch zur Laufzeit zum Embedded System hinzugefügt werden können und nicht zusammen mit dem Framework kompiliert werden müssen.

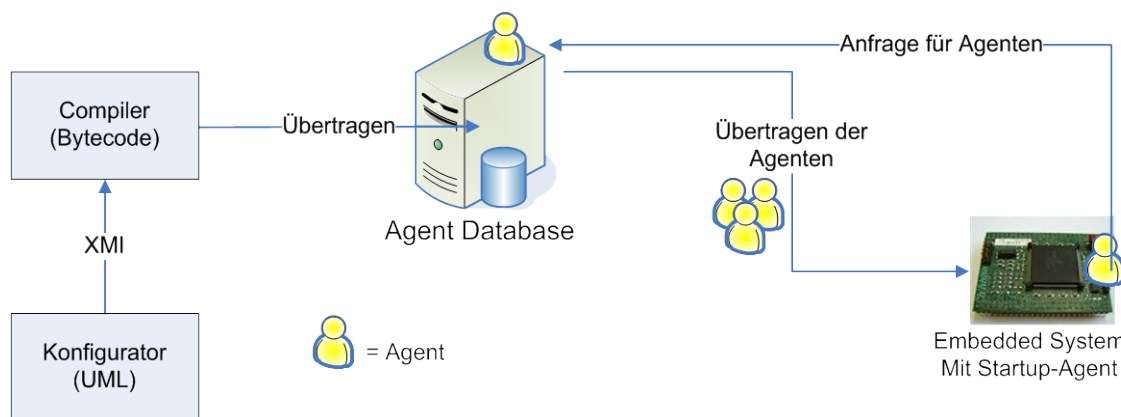


Abbildung 5: Workflow zur Erzeugung von LEAF-Agenten

Wie in Abbildung 5 zu sehen ist geschieht dies in folgenden Schritten.

- Der Agent wird in UML definiert und in einer XMI-kompatiblen XML Datei abgespeichert. XMI kann von fremden UML-Case-Tools weiter verarbeitet werden.
- Ein Compiler erzeugt den Bytecode für den Agenten.
- Dieser Bytecode wird in der Agentendatenbank gespeichert.
- Beim Start der LEAF-Plattform auf dem Embedded System fragt der immer vorhandene Embedded-Startup-Agent die vorhandenen Agenten beim zuständigen JADE-Agentensystem an.

- Der JADE-Agent lädt den/die Agenten aus der Agentendatenbank und sendet diese an den Startup-Agenten.
- Der Startup-Agent instanziiert daraus den/die Agenten für das Embedded System.

E) Prototyp einer Laufzeitumgebung

Der Prototyp der LK³S-Laufzeitumgebung wurde für ein Szenario entwickelt, in dem aus einem Palettenlager Komponenten für die Fertigung in einer digitalen Fabrik konfektioniert und für die Auslieferung vorbereitet werden. Bisher wurde dazu ein zentrales Lagerverwaltungssystem des Projektpartners Beka Engineering eingesetzt. Im neuen Konzept stellt jede Palette und jeder Gabelstapler eine eigenständig agierende und entscheidungsfähige Softwarekomponente dar. Alle Komponenten erhalten vom PPS Optimierungskriterien, nach denen sie ihr Verhalten ausrichten sollen. So könnte für einen bestimmten Zeitraum die Auslieferung auf optimalen Durchsatz optimiert sein. Bei veränderten Randbedingungen kann die Optimierung in Richtung bester Energieeffizienz umgesteuert werden

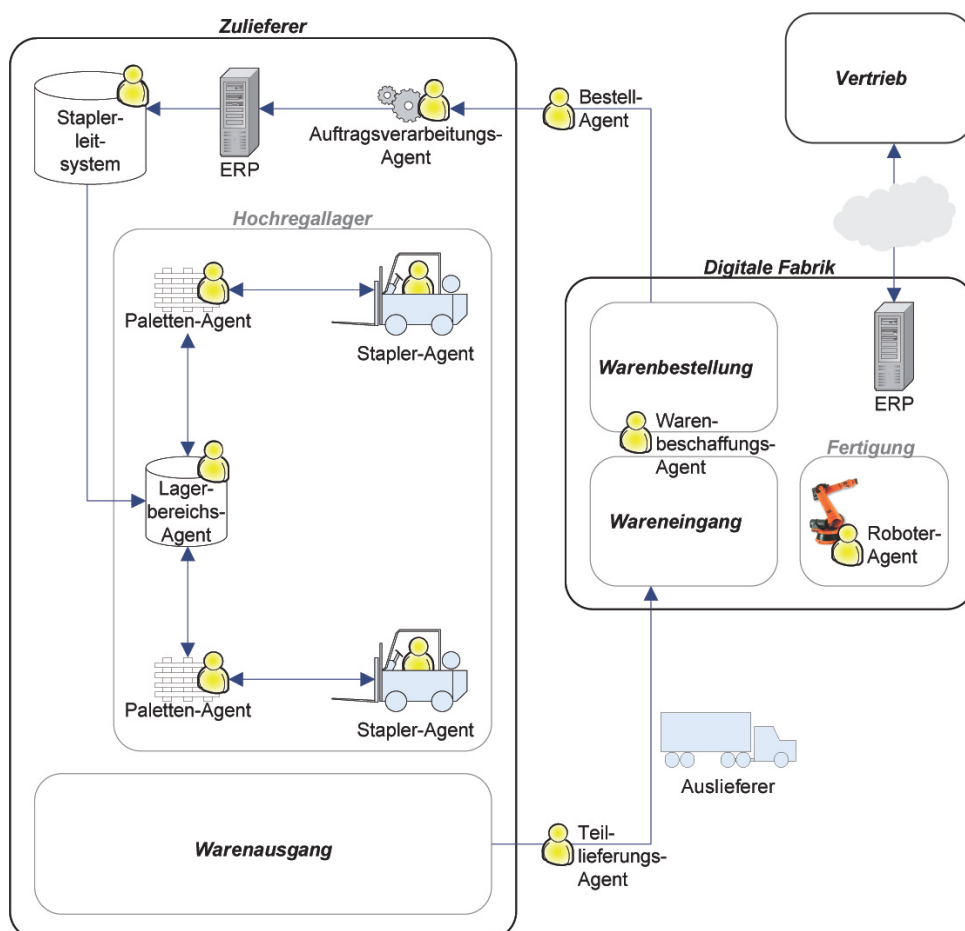


Abbildung 5: Zusammenspiel eines Zulieferers und einer digitalen Fabrik mit LK³S-Komponenten

Realisiert wurde das LK³S-Konzept bei einem Messeauftritt auf der SPS/IPC Drives 2009 in Nürnberg. Es wurde ein Szenario geschaffen, bei dem ein Kunde eines Fertigungsunternehmens ein maßgeschneidertes Produkt bestellen kann. Das Agentensystem der Digitalen Fabrik simuliert und optimiert die nötigen Fertigungsschritte, um ein Angebot zu erstellen. Dieses enthält Angaben zum Kaufpreis und zum Liefertermin. Ist der Kunde mit dem Angebot einverstanden, kann er bestellen. Alternativ kann er sich ein günstigeres oder früher lieferbares Produkt vom System zusammenstellen lassen. Hierbei kommen dann veränderte Optimierungskriterien des LK³S-Systems zum Einsatz. Nach Abschluss der Bestellung wird ein mobiler Bestellagent mit seinen Bestellinformationen zum Lieferanten geschickt und registriert sich in dessen Agentensystem.

Der Auftragsverarbeitungs-Agent beim Zulieferer nimmt die Bestellung entgegen. Im Palettenlager werden Materiallieferungen für die Fertigung in der digitalen Fabrik konfektioniert und für die Auslieferung vorbereitet. Ein gesamtes Auslagerungsszenario beinhaltet folgende Aktionen: Nachdem der Auftrag von der digitalen Fabrik beim Zulieferer eingegangen ist, wird dieser vom ERP System auf Machbarkeit geprüft. Das ERP-System erstellt ein Angebot mit Lieferdaten, welches zurück zur digitalen Fabrik gesendet wird. Nach der Bestätigung des Angebots werden durch das ERP System einzelne Teilaufträge an das Staplerleitsystem verteilt. Dieses leitet die Teilaufträge an die zuständigen Lagerbereichsagenten weiter. Diese informieren die auszulagernden Paletten, die sich wiederum Transportangebote von Gabelstaplern, hier in Form von Lagerrobotern, einholen. Die Paletten entscheiden sich anhand der Transportangebote für einen Gabelstapler und veranlassen die Auslagerung. Der Gabelstapler transportiert die Paletten zum Warenausgang, wo sie nun zum Transport bereit stehen.

Das gesamte oder ein Teil des bestellten Materials trifft zusammen mit einem Teillieferungsagenten am Wareneingang der digitalen Fabrik ein. Der Teillieferungsagent meldet sich bei dem Wareneingangsagenten an und es wird eine Qualitätsprüfung der Ware durchgeführt. Bei Mängeln kann der Teillieferungsagent mit entsprechenden Informationen an den Lieferanten zurückgesandt werden. Ist das benötigte Material eingetroffen, kann die Produktion begonnen werden. Das Endprodukt wird an den Warenausgangsagenten übergeben, da dieser das Produkt angefordert hatte. Es erfolgt eine Rückmeldung an das ERP-System und das Produkt kann nun an den Kunden ausgeliefert werden.

Eine erweiterte Version des Prototypen wird auf der Hannover Messe Industrie 2010 gezeigt.

Danksagung

Ein besonderer Dank gilt dem Bundesministerium für Bildung und Forschung als Sponsor des Projektes. Das BMBF fördert mit dem Programm FHprofUnt Forschung an Fachhochschulen mit Unternehmen unter FKZ 1717 B07. Ein weiterer Dank geht an die im Projekt beteiligten Firmen Phoenix Contact GmbH & Co.KG, Willert Software Tools GmbH, BeKa Engineering und Schneider Electric.

Referenzen / Literatur

- [1] Schmidtman, U.; et al., "Leicht Konfigurierbare Komponenten Kollaborativer Systeme" 6. Automatisierungstage 2008, Jan. 2008
- [2] Friedman, T. L.; The World is Flat, A Brief History of the Twenty-First Century. Farrar, Strauss and Giroux, N.Y., 2005.
- [3] ARC. Collaborative Manufacturing to Management Strategies. Tech. rep., ARC Advisory Group, Enterprise & Manufacturing Strategies for Industry Executives, Three Allied Drive, Dedham, MA 02026 USA, <http://ARCweb.com>, 11/2002.
- [4] ARC. Automation for the Discrete Industrie Worldwide Outlook. Tech. rep., ARC Advisory Group, Enterprise & Manufacturing Strategies for Industry Executives, Three Allied Drive, Dedham, MA 02026 USA, <http://ARCweb.com>, 10/2003.
- [5] Jennings, Nicholas R.;, Agent-Oriented Software Engineering, Springer Berlin/Heidelberg 2007
- [6] Foundation for Intelligent Physical Agents, <http://www.fipa.org>
- [7] Foundation for Intelligent Physical Agents, FIPA Abstract Architecture Specification, <http://www.fipa.org/specs/fipa00001/index.html>
- [8] Bellifemine, F. L.; Caire, G.; Greenwood, D; Developing Multi-Agent Systems with JADE, 2007
- [9] Proksch, P.; Konzeption und Entwicklung einer Ausführungsumgebung für aus UML generierte Agenten auf eingebetteten Systemen; Masterarbeit Fachhochschule Osnabrück 2010.
- [10] Proksch, P.; Westerkamp, C.; Wuebbelmann, J., Decision-Making Embedded Devices using FIPA-ACL Communication, 7th IEEE International Conference on Industrial Informatics, Cardiff 2009