

Publiziert in: *Alt et al. (Hrsg.), Tagungsband 15. Interuniversitäres Doktorandenseminar Wirtschaftsinformatik der Universitäten Chemnitz, Dresden, Freiberg, Halle-Wittenberg, Jena und Leipzig, Leipzig, 2011, S. 1-10.*

## Model Driven Logistics Integration Engineering

Robert Kunkel

Universität Leipzig, Grimmaische Straße 12, 04109 Leipzig

kunkel@wifa.uni-leipzig.de

**Abstract:** Der Logistikdienstleistungssektor ist durch arbeitsteilige sowie kurz-, mittel- und langfristige Zusammenarbeit gekennzeichnet. Insbesondere Fourth Party Logistics (4PL) stehen permanent vor der Aufgabe unterschiedliche Logistikdienstleister und damit auch deren Informationssysteme ad-hoc und medienbruchfrei in unternehmensübergreifende Informationsflüsse zu integrieren. Dieser Beitrag stellt verschiedene logistikspezifische Integrationsvarianten, einen modellgetriebenen Integrationsansatz sowie ein Lösungskonzept auf Basis der Logistik Service Engineering & Management (LSEM)-Plattform vor.

### 1 Einleitung

Das Managementkonzept und Unternehmensmodell Fourth-Party Logistics (4PL) wurde 1996 vom Beratungsunternehmen Accenture erstmals benannt und definiert. Ein Logistikdienstleister vom Typ 4PL fasst Dienstleistungsangebote verschiedener anderer Logistikdienstleister (LD) zu einer komplexen Gesamtdienstleistung zusammen und koordiniert die Arbeit in den Lieferketten, ohne eigene logistische Ressourcen wie Fuhrpark oder Lagerhaus zu besitzen (vgl. [Bauknight/Miller 2011]). Ein 4PL stellt seine angebotenen Dienstleistungen somit ausschließlich aus den Angeboten von sogenannten Second und Third Party Logistics (ausführende und spezialisierte Logistikunternehmen) zusammen. Als Integrator und Koordinator von Lieferketten stehen dem 4PL ausschließlich Informationssysteme und logistisches Domänenwissen zur Verfügung.

Informationen und der mit ihnen verbundene Informationsfluss sind ein wesentlicher Teil der Logistik. Durch die effiziente Bereitstellung von Informationen zur richtigen Zeit kann der Güterfluss in einer Lieferkette deutlich beschleunigt werden (vgl. [Luo 2010]). In sämtlichen Phasen der Zusammenarbeit, ob Verhandlung, Erstellung und Erbringung von Dienstleistungen oder bei der Abrechnung, müssen Informationen zwischen den LD und dem 4PL ausgetauscht werden. Alle anfallenden Informationen müssen dabei zur Planung, Steuerung und Kontrolle der Lieferkette vom 4PL verarbeitet werden. Um diese Informationsflut beherrschen zu können, wird ein Softwaresystem benötigt, welches die anfallenden Informationen sammelt, aggregiert und verarbeitet. Eine mögliche Lösung dieses Problems wird durch die Logistik Service Engineering and Management (LSEM) Plattform bereitgestellt, welche sich an den Anforderungen des Geschäfts- und des Softwarelevels orientiert (vgl. [Klinkmüller et al. 2011]). Integrationsplattformen, wie die LSEM-Plattform, bieten dem 4PL die Möglichkeit eigene Anwendungssysteme und die seiner Partner einzubinden, um die anfallenden Informationen effizient, performant und über Regeln automatisierbar zu verarbeiten. Im Gegensatz zur innerbetrieblichen Integration oder der Integration zwischen langfristig kooperierenden Unternehmen, erfordern die Spezifika der Logistikbranche verschiedene Arten der Integration. Neben der Teilnahme an mittel- bis langfristigen Kontrakten, muss

es den LD ermöglicht werden, sich auf Basis einzelner Aufträge an den Lieferketten des 4PL zu beteiligen. Zusätzlich verbessert dies einerseits die Akzeptanz der Plattform bei LD und andererseits vergrößert sich die Gruppe der zur Verfügung stehenden LD für den 4PL. Zur Integration von Anwendungssystemen der LD in die Plattform des 4PL müssen neben der Vollintegration (Vollintegration bezeichnet in diesem Beitrag die komplette Integration sämtlicher benötigter Funktionalitäten eines Anwendungssystems) weitere Integrationsarten, welche sich einerseits im Umfang und andererseits im zeitlichen Aufwand zur Einrichtung unterscheiden, zur Verfügung stehen (vgl. [Kunkel et al. 2011] und [Kunkel/Ludwig/Franczyk 2012]).

In diesem Beitrag werden die Spezifika der Integration von LD und deren Anwendungssysteme in eine logistische Plattform betrachtet. Kapitel 2 adressiert den Integrationsansatz, die zugrunde liegende Architektur der LSEM-Plattform, logistikspezifische Integrationsvarianten und die Umsetzung der Varianten mit Hilfe modellgetriebener Verfahren. Abschließend wird in Kapitel 3 ein Fazit gezogen und ein kurzer Ausblick gegeben.

## **2 Integrationsansatz**

Dem 4PL als Integrator von Lieferketten stehen ausschließlich Informationssysteme und logistisches Domänenwissen zur Verfügung, um komplexe Gesamtdienstleistungen von verschiedenen LD zusammenzustellen und die Arbeit in den Lieferketten zu koordinieren. Im nachfolgenden Abschnitt wird die grundlegende Architektur des Integrationsystems vorgestellt; Der Abschnitt 2.1 beschreibt den Aufbau der Integrationsplattform im Detail. Abschnitt 2.2 stellt logistikdomänenspezifische Integrationsarten vor, während im Abschnitt 2.3 auf die Umsetzung dieser durch modellgetriebene Verfahren eingegangen wird.

### **2.1 Architektur**

Die LSEM-Plattform stellt eine Integrationsplattform für einen 4PL dar. Sie wurde innerhalb des Forschungsprojekts Logistik Service Bus (vgl. [Franczyk 2010]) entwickelt und baut auf dem Paradigma der Service-Orientierung auf, das sowohl auf geschäftlicher als auch auf technischer Ebene angewendet wird. Grundsätzlich gliedert sich die LSEM-Plattform in zwei Dimensionen. Zum einen umfasst die Laufzeitumgebung grundlegende Komponenten für die Bereitstellung und die Integration von Softwaresystemen und erlaubt auf dieser Basis die Komposition von Informationsflüssen und das Erstellen neuer Anwendungen. Zum anderen enthält die Werkzeugumgebung Komponenten, die das Konfigurieren und das Arbeiten mit der Laufzeitumgebung ermöglichen. Diese Systeme erlauben dem 4PL die Planung, Steuerung und Kontrolle der Lieferketten. Die in Abbildung 2-1 dargestellte Architektur der LSEM-Plattform (vgl. [Klinkmüller et al. 2011]) baut auf der S3-Referenzarchitektur (vgl. [Arsanjani 2007]) auf und erweitert diese.

Die vorliegende Arbeit bezieht sich auf die Schichten der zweiten Dimension, welche grundlegende Funktionalitäten für die Erstellung von Integrationskomponenten bereitstellen. Hierbei werden die Integration Schicht, welche typische Enterprise Service Bus Eigenschaften umfasst und somit zentrale Komponenten für das Bereitstellen und Integrieren von Anwendungen bereitstellt, und die Information Architecture Schicht, welche ein kanonisches Datenformat für den Datenaustausch auf der Plattform defi-

niert, benötigt. Eine ausführlichere Beschreibung der LSEM-Plattform und deren Schichten befindet sich in [Klinkmüller et al. 2011].

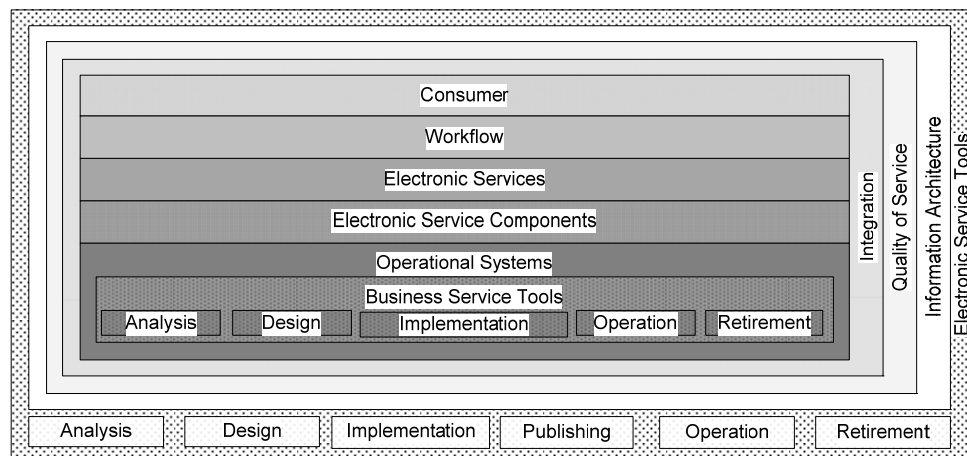


Abbildung 1: Die Architektur der LSEM-Plattform (aufbauend auf [Arsanjani 2007] und [Klinkmüller et al. 2011])

## 2.2 Integrationsvarianten

Die Zusammenarbeit des 4PL mit verschiedenen LD bedingt, wie in den vorherigen Kapiteln dieses Beitrages dargestellt, die Integration von Anwendungssystemen der LD in die zentrale LSEM-Plattform. Im Gegensatz zur innerbetrieblichen Integration oder der Integration zwischen langfristig kooperierenden Unternehmen, erfordern die Spezifika der Logistikbranche verschiedene Arten der Integration. Neben der Teilnahme an mittel- bis langfristigen Kontrakten, muss es einem LD ermöglicht werden, sich auf Basis einzelner Aufträge an den Lieferketten des 4PL zu beteiligen. Weiterhin wird zur Integration der Anwendungssysteme der LD eine Integrationsumgebung benötigt, welche die Integration nach adaptiven Verfahren (vgl. [Thränert/Kühne 2008], [Erl 2009] und [Kunkel et al. 2011]) ermöglicht. Die Adaption ermöglicht dabei eine nicht-invasive Integration, welche die innere Struktur des Systems und die vorhandenen Schnittstellen nicht verändert. Hierdurch müssen die zu integrierenden Anwendungssysteme nicht angepasst werden.

Zur schrittweisen Integration von Anwendungssystemen der LD wird somit eine Integrationsumgebung benötigt, welche sowohl die unterschiedlichen Integrationsvarianten unterstützt als auch die Anwendungssysteme durch ein adaptives Integrationsverfahren mit Schnittstellen ausstattet. Diese Integrationsumgebung muss hierbei individuell für jeden LD erstellt, konfiguriert und bereitgestellt werden und übernimmt anschließend sämtliche Kommunikationsaufgaben zwischen dem LD und dem 4PL. Die hierzu benötigten Komponenten werden entweder durch die LSEM-Plattform bereitgestellt oder durch diese entwickelt. Durch die zweite Dimension der LSEM-Plattform, welche für die Bereitstellung und die Integration von Softwaresystemen zuständig ist, werden Integrationsumgebungen entwickelt, welche sich in die erste Dimension der LSEM-Plattform einordnen und für die zu integrierenden Anwendungssysteme Integrationslogik und Services erzeugen. Nähere Erläuterungen zur Umsetzung und zur theoretischen Konzeption der Komponenten sind [Klinkmüller et al. 2011], [Hohpe/Woolf/Brown 2010] und [Erl 2009] zu entnehmen. Daten und Informationen können in jedem Anwendungssystem in unterschiedlichen, oftmals proprietären Formaten und Modellen vorliegen. Zur Lösung dieses Problems werden in der Integrationsumgebung Message Translator eingesetzt, welche das vorliegende Format der LD in das kanonische Format

der LSEM-Plattform und zurück übersetzen. Für Anwendungssysteme sollte transparent bleiben, welches Ziel eine gesendete Nachricht hat und über welche Stationen, z.B. Message Translator, sie gesendet wird. Diese Entkopplung von Sender und Empfänger wird über Message Channel erreicht. Zur Weiterleitung von Nachrichten aufgrund des Inhaltes werden Content-based Router eingesetzt, welche über definierte Regeln den Empfänger bestimmen.

Ziel dieses Integrationsvorgehens ist die Integration von verschiedenen LD an die Logistikplattform des 4PL mit Hilfe einer Integrationsplattform. Der Einsatz einer solchen Integrationsplattform ermöglicht es dem 4PL über einheitliche Services mit den LD zu kommunizieren. Hierzu werden auf der LSEM-Plattform für verschiedene Rollen (z.B. Lager, Transport und Umschlag) abstrakte Services definiert, welche durch die LD implementiert und bereitgestellt werden müssen. Weiterhin können für die Services Übertragungstypen und Kommunikationseinstellungen festgelegt werden. Die implementierten Services der Integrationsumgebung können nun von der LSEM-Plattform aufgerufen werden und die Services der LSEM-Plattform können durch die Integrationsplattform genutzt werden. Dies ermöglicht das Umsetzen von One-Way - (z.B. dem Senden von Avis in beide Richtungen) und Request-response - Operations (z.B. Anfrage Transportauftrag). Die ausführliche Beschreibung von Rollen, abstrakten Services, Typen und Kommunikationseinstellungen erfolgt in Abschnitt 2.3. In Abbildung 2-1 sind drei Integrationsvarianten dargestellt, bei welchen sich von links nach rechts jeweils der Integrations- und Automatisierungsgrad, jedoch auch der Integrationsaufwand erhöhen. Die Varianten basieren auf der Arbeit von [Kunkel et al. 2011] und [Kunkel/Ludwig/Franczyk 2012]. Bei allen drei Varianten übernimmt die Integrationsumgebung die Implementierung und Bereitstellung der abstrakten Services sowie die Kommunikation mit der LSEM-Plattform über Message Channel. Variante I stellt den ersten Schritt im Integrationsprozess dar. Die Kommunikation und der Informationsaustausch werden vollständig von der Integrationsumgebung übernommen. Die Integrationsvariante I soll dem LD ad-hoc und ohne weitere Integrationsaufwände und Integrationskosten bereitgestellt werden. Diese Variante ermöglicht es den LD sich kurzfristig an Kontrakten des 4PL zu beteiligen ohne Kosten für die adaptive Integration seiner Anwendungssysteme aufwenden zu müssen. Zur Interaktion zwischen LD und der Integrationsumgebung steht ein Rich Client zur Verfügung, welcher die Kommunikation von Menschen mit den Services der Integrationsumgebung durch Human-Service Interaction (HSI) ermöglicht (vgl. [Shneiderman 1997] und [Dix 2003]). Sämtliche Serviceaufrufe werden dem LD als Aufgaben in menschenlesbarer Form bereitgestellt.

Die in Variante I bereitgestellte Integrationsinfrastruktur ermöglicht es nun in den weiteren Schritten die Anwendungssysteme des LD schrittweise zu integrieren. Neben der kurzfristigen Zusammenarbeit können die LD bei mittel- bis langfristigen Kooperationen den Integrationsgrad erhöhen und somit den Informationsaustausch schrittweise automatisieren. Hierzu können, wie in Variante II dargestellt, einzelne Komponenten des Anwendungssystems adaptiert und die so gewonnenen Schnittstellen mit der Integrationsumgebung integriert werden. Beispielsweise könnte in einem ersten Schritt die Übermittlung von Auftragsanfragen automatisiert werden. Durch das Bereitstellen entsprechender Message Translator und Router werden die Informationen direkt in das Anwendungssystem übertragen. Der Rich Client übernimmt für die integrierten Services anschließend Monitoringfunktionalitäten, für nicht integrierte Services wird weiterhin die Kommunikation über HSI zur Verfügung gestellt.

Die Variante III stellt die Vollintegration des Anwendungssystems dar. Sämtliche Services sind hierbei über entsprechende Message Channel, Translator und Router mit den für das Anwendungssystem bereitgestellten Schnittstellen verbunden. Der Rich Client übernimmt hierbei ausschließlich das Monitoring der Integrationsumgebung. LD können mit Hilfe der dargestellten Varianten je nach Art der Zusammenarbeit über den Integrationsgrad entscheiden, während der Integrationsfortschritt für den 4PL transparent ist, da bei jeder Variante die gleichen Schnittstellen der Plattform unterstützt werden.

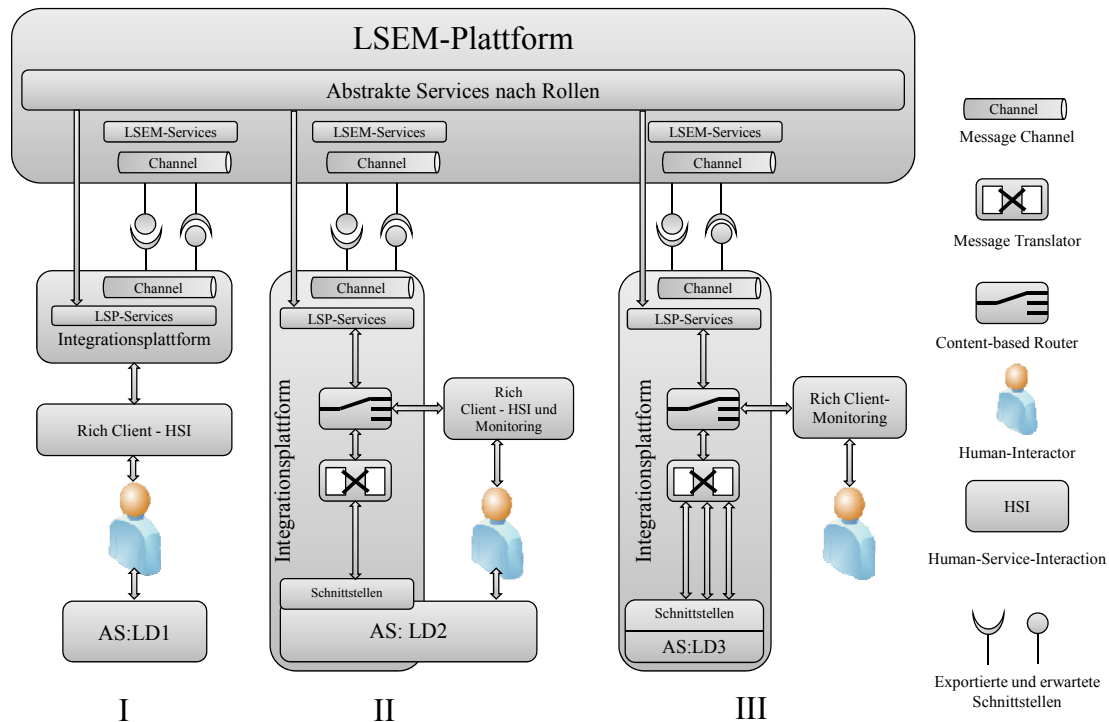


Abbildung 2: Integrationsvarianten (Vgl. [Kunkel et al. 2011] und [Kunkel/Ludwig/Franczyk 2012])

### 2.3 Modellgetriebene ad-hoc Integration

Die im vorhergehenden Kapitel dargestellten Integrationsarten ermöglichen einem 4PL und den LD, neben mittel- und langfristigen Kontrakten, die Zusammenarbeit in kurzfristigen Kontrakten. Die ad-hoc Zusammenarbeit wird durch die effiziente Bereitstellung der Integrationsumgebung und die Serviceimplementierung durch HSI in Integrationsvariante I ermöglicht, ohne dass für den LD weitere Integrationsaufwände und -kosten entstehen. Für diese effiziente ad-hoc Bereitstellung einer Integrationsumgebung eignen sich vor allem modellgetriebene Verfahren, da diese besondere Stärken aufweisen, wenn mehrere verwandte Produkte entwickelt werden und Teile sich wiederverwenden lassen (vgl. [Stahl et al. 2007] und [Thränert/Kühne 2008]). Mit Hilfe von modellgetriebenen Ansätzen lässt sich die Gesamtkomplexität dieses Integrationsansatzes beherrschen, da Modelle das System auf einer höheren Abstraktionsstufe beschreiben und sich die Komplexität somit auf das Modell selbst (Modellierung durch den 4PL als Logistikdomänenexperte) und auf die Abbildung des Modells auf die Zielsprache, also die Generierung und Interpretation (Modelltransformationen innerhalb der LSEM-Plattform), aufteilen lässt. Beide Teile können getrennt voneinander bearbeitet werden, wodurch beim Modellieren keine technischen Details der Implementierung bekannt sein müssen (vgl. [Stahl et al. 2007]).

### 2.3.1 Integrationsmodell

Als Ausgangspunkt zur Modellgetriebenen Integration werden vom Domänenexperten, in diesem Fall dem 4PL, die in Abbildung 2-2 dargestellten Modelle beschrieben. Diese gliedern sich in die fünf Teilmodelle Rolle, Workflow, abstrakter Service, Typen und

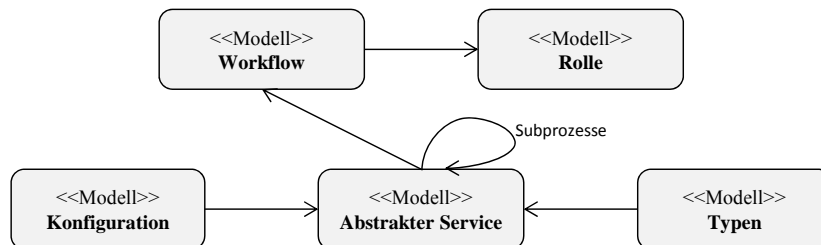


Abbildung 3: Integrationsmodell (in Anlehnung an [Kunkel et al., 2011] und [Kunkel/Ludwig/Franczyk 2012])

Konfiguration. Sämtliche Modelle werden auf technischer, elektronischer Ebene beschrieben. Die Definition dieser Modelle ermöglicht es innerhalb der LSEM-Plattform Prozesse zu erstellen und darin in Workflows zusammengefasste (abstrakte) Services der LD zu nutzen, ohne sich bereits auf einen ausführenden LD festlegen zu müssen. Dies wird ermöglicht da jeder beteiligte LD mit Hilfe der Integrationsumgebung die gleichen für ihn definierten Services implementiert.

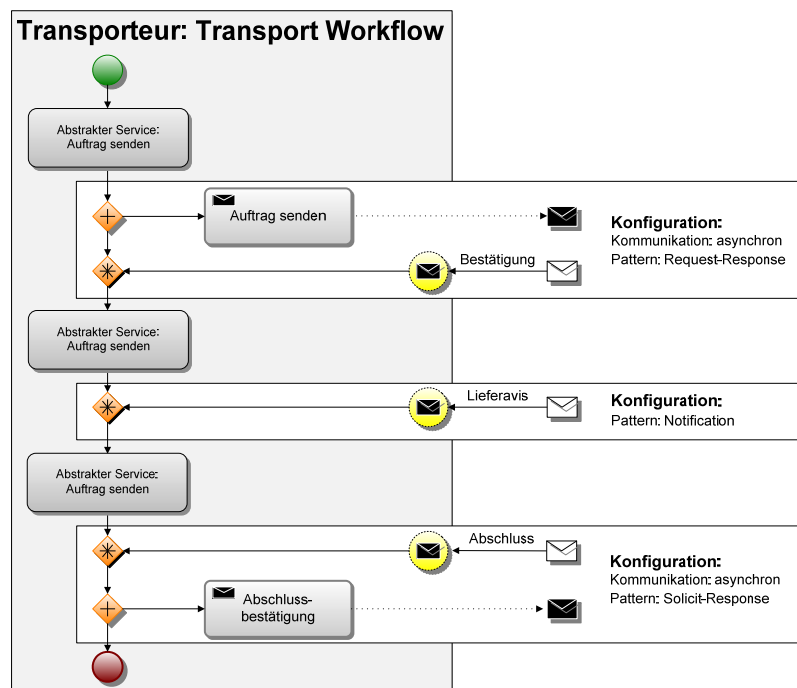


Abbildung 4: Anwendungsbeispiel - Transport Workflow

Im ersten Schritt legt der 4PL Rollen (z.B. Transport- oder Lageranbieter) fest. Diese dienen der Zuordnung zu definierten Workflows, welche durch einen LD implementiert werden müssen. Das Workflow-Modell fasst mehrere abstrakte Services in eine inhaltlich abgeschlossene, zeitlich und sachlogisch zusammenhängende Folge zusammen.

Die Definition eines abstrakten Services enthält die zu implementierenden Serviceoperationen. Für die einzelnen Operationen werden die zu übertragenden Datentypen ausgewählt oder neu modelliert. Die einzelnen Typen werden hierbei mit Hilfe der Extensible Markup Language (XML) beschrieben und können in verschiedenen abstrakten Services wiederverwendet werden. Im Konfigurationsmodell werden Kommunikationsparameter definiert. Beispielsweise werden hier synchrone und asynchrone Aufrufe und Webservice-Kommunikationsarten wie z.B. One-Way-, Request-Response-, Solicit-Response- und Notification unterschieden (vgl. [Weerawarana et al. 2001]). Abbildung 2-3 verdeutlicht das Integrationsmodell am Beispiel vom Transport Workflow, welcher der Rolle Transporteur zugeordnet ist. In der Abbildung werden mehrere abstrakte Services mit verschiedenen Konfigurationen definiert.

Das Integrationsmodell dient im weiteren Vorgehen als Ausgangspunkt zur modellgetriebenen Erstellung einer Integrationsumgebung. Dabei werden je nach Rolle sämtliche Workflows beim LD als konkrete Services implementiert und innerhalb der Integrationsumgebung zur Verfügung gestellt. Dies ermöglicht dem 4PL die Planung von Abläufen durch vordefinierte Workflows und die spätere Instantiierung eines Workflows mit den konkreten Prozessen eines LD.

### **2.3.2 Modelltransformation zur Generierung der Integrationsumgebung**

Zur ad-hoc Bereitstellung der Integrationsumgebung werden Modelltransformationen auf Basis des in Abbildung 2-2 dargestellten LSEM-Integrationsmodells verwendet. Die auf der LSEM-Plattform bereitgestellten Modelle können nun durch modellgetriebene Verfahren in der Integrationsumgebung verwendet werden, um einerseits die Quellmodelle in konkretere Zielintegrationsmodelle (in Abbildung 2-4 im unteren Bereich mit <<Modell>> dargestellt) zu transformieren und andererseits durch Interpretation und Generierung lauffähigen Code zu erzeugen (in Abbildung 2-4 im unteren Bereich mit <<generiert>> dargestellt).

Nach der Auswahl der zu konkretisierenden abstrakten Services über das Rollenkonzept werden in der Integrationsumgebung konkrete Services mit wohldefinierten Schnittstellen und Typen erzeugt. Die so generierten Service-Instanzen (konkreter Service) sind durch die LSEM-Plattform nach der Registrierung aufrufbar und können zur Laufzeit auf der LSEM-Plattform die abstrakten Services, welche in den Prozessen des 4PL verwendet wurden, ersetzen und die Prozesse somit in konkrete, lauffähige Prozesse umwandeln.

Als erster Schritt werden dabei in Abhängigkeit zur gewählten Rolle ein oder mehrere abstrakte Services mit dazugehörigem Typenmodell ausgewählt und in ein Instanzen- und Kompositionsmodell überführt. Für jeden abstrakten Service wird das definierte Konfigurationsmodell in ein Bereitstellungsmodell innerhalb der Integrationsumgebung überführt. Diese Modelle definieren die zu generierenden Services und deren Bereitstellung innerhalb der Integrationsumgebung. Somit können anschließend die konkreten Services, deren Schnittstellen und Typen generiert und mit Hilfe der erstellten Konfigurationen innerhalb von Service-Containern bereitgestellt werden. Das vorgestellte Integrationsmodell unterstützt alle Arten der in Abbildung 2-1 vorgestellten Integrationsvarianten. Zur Anbindung an die Plattform des 4PL steht somit eine Integrationsumgebung zur Verfügung, welche nach der Bereitstellung sämtliche Kommunikation über definierte Services zwischen dem 4PL und den LD übernimmt, wobei noch keinerlei Integration mit den Anwendungssystemen der LD stattfand.

Um die ad-hoc zur Verfügung stehende Integrationsvariante I umsetzen zu können werden daher, wie in Abbildung 2-4 dargestellt, von den konkreten Services, deren Schnittstellen und Typen sowie der erzeugten Konfiguration eine Serviceimplementierung in Form einer Human-Service Interaction generiert. Die Integrationsumgebung mit der Integrationsvariante I kann durch die eingesetzten modellgetriebenen Verfahren unmittelbar zur Verfügung gestellt werden und erlaubt die Interaktion mit den erzeugten Services und die Darstellung der übertragenen Informationen. Um die Varianten II und III im weiteren Verlauf der Zusammenarbeit des 4PL mit dem LD zu unterstützen, werden einzelne Human-Service Interactions durch Integrationslogik ersetzt. Diese Integrationslogik beinhaltet die manuelle Adaption des Anwendungssystems zur Erzeugung von Schnittstellen sowie die manuelle Implementierung der benötigten Message Translator und Router. Durch das schrittweise Ersetzen der Human-Service Interactions durch Integrationslogik kann das Anwendungssystem nach und nach integriert werden.

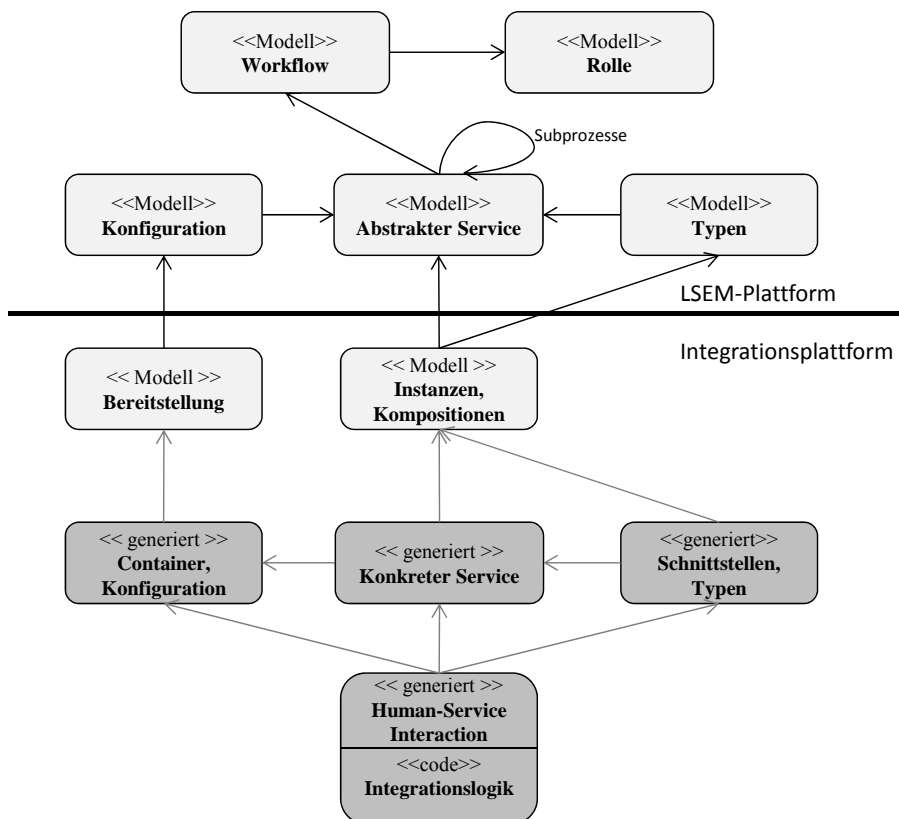


Abbildung 5: Quell- und Zielintegrationsmodelle (in Anlehnung an [Kunkel et al. 2011] und [Kunkel/Ludwig/Franczyk 2012])

### 3 Zusammenfassung und Ausblick

Der in diesem Beitrag vorgestellte Integrationsansatz bedient die im Logistikdienstleistungssektor vorzufindenden arbeitsteiligen Zusammenarbeiten und bietet eine Möglichkeit neben mittel- und langfristigen Kooperationen auch die kurzfristige Zusammenarbeit mit Hilfe von modellgetriebenen Verfahren zu ermöglichen. Der Informationsaustausch lässt sich durch diesen Ansatz aus Sicht des 4PL unabhängig vom Integrationsgrad und -tiefe, insbesondere bei kurzfristigen Kontrakten durch die ad-hoc In-



tegration, automatisieren. Die vorgestellten logistikspezifischen Integrationsvarianten sowie der Integrationsansatz basieren auf der Logistik Service Engineering & Management (LSEM)-Plattform.

In den nächsten Schritten der Forschungsarbeit sollen die Integrationsvarianten II und III näher betrachtet werden. Hierbei sollen semantische Datenintegrationsverfahren zum Mapping der Datenformate der LSEM-Plattform und der Anwendungssysteme betrachtet werden. Außerdem soll die Sicherheit der zu übertragenden Daten im unternehmensübergreifenden Einsatz betrachtet werden und in einer Lösungskomponente von der Integrationsumgebung bereitgestellt werden. Weitere zu bearbeitende Themen stellen somit die Transparenz des Datenverkehrs, die Datenhoheit auf Seiten der LD sowie Sicherheitsmechanismen dar. In einem weiteren Schritt soll das Konzept anhand von Fallstudien validiert und auf Praxistauglichkeit überprüft werden. Bei dieser Überprüfung sollen zeitliche und aufwandmäßige Kenngrößen zur Bewertung der verschiedenen Integrationsansätze ermittelt werden. Schließlich soll der vorgestellte Ansatz durch eine geeignete Werkzeugunterstützung, zur Erstellung und Transformation von Modellen sowie zur Generierung von Code, erweitert werden.

## Literatur

- Arsanjani, A., et al.: S3: A Service-Oriented Reference Architecture, in: IT Professional 9, S. 10-17, 2007.
- Bauknight, D., Miller, J., Fourth Party Logistics: The Evolution of Supply Chain Outsourcing, in: CALM Supply Chain & Logistics Journal, 2011.
- Dix, A., et al., Human-Computer Interaction, Prentice Hall, Harlow, 2003.
- Erl, T., SOA Design Patterns, Prentice Hall, Upper Saddle River, 2009.
- Franczyk, B., Logistik-Service-Bus - Projekt, 2010, <http://www.lsb-plattform.de>, 11.11.2011.
- Hohpe, G., Woolf, B., Brown, K., Enterprise integration patterns: Designing, building, and deploying messaging solutions, Addison-Wesley, Boston, 2010.
- Klinkmüller, C., et al., The Logistics Service Engineering & Management Platform: Operations, Architecture, Implementation, in: 14th International Conference on Business Information Systems, Poznań, 2011.
- Kunkel, R., et al., Modellgetriebene Integration von Logistik-Informationssystemen in die LSEM-Plattform, 41. GI-Jahrestagung Informatik 2011, Die Rolle von Plattformen für Unternehmensökosysteme, Berlin, 2011.
- Kunkel, R., Ludwig, A., Franczyk, B., Modellgetriebene ad-hoc Integration von Logistikdienstleistern - Integrationsansatz und Prototyp, MKWI 2012, Informationssysteme in Logistik und Verkehr, Braunschweig, 2012 (unveröffentlicht).
- Luo, Z., Service science and logistics informatics. Innovative perspectives, Premier reference source, Business Science Reference, Hershey Pa, 2010.
- Shneiderman, B., Designing the User Interface: Strategies for Effective Human-Computer Interaction, Addison-Wesley Longman Publishing Co., Boston, 1997.
- Stahl, T., et al., Modellgetriebene Softwareentwicklung. Techniken, Engineering, Management. dpunkt Verlag, Heidelberg, 2007.

Thränert, M., Kühne, S., Model-Driven Integration Engineering zur Integration betrieblicher Anwendungssysteme, in: Fähnrich, K.; Kühne, S.; Thränert, M. (Hrsg.), Model-Driven Integration Engineering. Universität Leipzig Pressestelle, Leipzig, 2008.

Weerawarana, S., et al., Web Services Description Language (WSDL) 1.1., 2001, <http://www.w3.org/TR/wsdl>, 11.11.2011.