Automotive Engineering im Enterprise-Domain-Network

Mathias Schraps[1], Torsten Ronneberger[1], Wolfgang Golubski[2]

[1]I/AEV-2, E, AUDI AG, 85045 Ingolstadt
[2]Westsächsische Hochschule Zwickau

(Mathias.Schraps, Torsten.Ronneberger)@audi.de
Wolfgang.Golubski@fh-zwickau.de

Abstract: Das Fahrzeugsystem wird aufgrund von Vernetzung in seiner Beschaffenheit und Entwicklung immer komplexer. Somit ist es zunehmend wichtiger das Wissen darüber zu managen, um weiterhin das Produkt und die Erzeugung dessen zu beherrschen. Das Enterprise-Domain-Network gibt einen konzeptuellen Ansatz, wie das Wissen benutzerfreundlich strukturiert aber trotzdem im Kontext gehalten wird. Darüber hinaus zeigt der Beitrag, wie das Enterprise-Domain-Network werkzeuggestützt realisiert wird.

1 Einleitung

Der Qualitätsunterschied bei Fahrzeugen wird nicht mehr nur anhand von Verarbeitungskriterien wie Spaltmaßen oder Materialien gemessen, sondern zunehmend an der Funktionalität des Fahrzeugs und der funktionalen Qualität. Aus diesem Grund werden innovative Technologien, wie Kommunikation und Datenverarbeitung in die aktuellen und zukunftsorientierten Automobile so integriert, dass sich die Fahrzeugsysteme zu hochvernetzten mechatronischen Systemen wandeln. Um diesen Trend realisieren zu können muss sich die Organisation der Fahrzeugentwicklung auf diesen Paradigmenwechsel einstellen. Die ausschließlich bauteilorientierten Organisationsstrukturen passen nicht mehr zu der Funktionsorientierung in der Entwicklung. Sie verkomplizieren zunehmend den Entwicklungsprozess und bremsen dadurch den Entwicklungsfortschritt. Daher ist es notwendig, das Fahrzeug in seinen Bestandteilen aus verschiedenen Perspektiven zu betrachten und mit neuen technologischen Möglichkeiten zu entwickeln.

Dem stetig steigenden Grad der Vernetzung im Produkt ist eine neue Herausforderung, der sich die Organisation und der einzelne Entwickler stellen muss. Ansätze aus dem Bereich des Semantic Web können dabei helfen, effizientere, flexiblere und transparentere Organisationsformen aufzubauen und die notwendig werdende Produktvernetzung in der Entwicklungsorganisation wiederzugeben. Mit Hilfe von semantischen Netzen auf Projekt- und Produktebene lassen sich die Vernetzungen rechnerintern erfassen, verarbeiten erweitern und bedarfsgerecht repräsentieren.

Der folgende Beitrag soll aufzeigen, wie Wissensverarbeitungstechniken den herkömmlichen Fahrzeugentwicklungsprozess unterstützen und den Umgang mit der zunehmenden Komplexität erleichtern. Es wird gezeigt, wie unter Verwendung einer

erschienen im Tagungsband der INFORMATIK 2011
Lecture Notes in Informatics, Band P192
ISBN 978-3-88579-286-4
weitere Artikel online:
http://informatik2011.de/519.html

kollaborativen Wissensplattform Konzepte der Wissensrepräsentation – speziell Ontologien – genutzt werden können, um Abhängigkeiten zu erfassen. Darüber hinaus bieten die Funktionalitäten sozialer Netzwerke eine Möglichkeit, die wachsende Informationsflut aufgrund der steigenden Individualität der Projektmitglieder zu unterstützen. Jeder Projektbeteiligte kann damit Informationen über sich und seinen Aufgabenkontext dezentral pflegen und für andere zur Verfügung stellen. Zur Strukturierung der Wissensdomänen (Unternehmen, Projekte, Produkte) wird in diesem Beitrag ein Enterprise-Domain-Network definiert und modelliert. Ein angepasstes Wiki und geeignete Inferenzmechanismen stellen die Ausgangsbasis für die Realisierung der Wissensplattform AEV-Fusion dar. Dazu werden im Detail folgende Schritte umgesetzt:

- Modellierung des Enterprise-Domain-Networks,
- Identifikation und Erfassung von generischen Klassen und Beziehungen,
- Abbildung der identifizierten Merkmale über Ontologien,
- Realisierung über ein semantisches Wiki,
- Anwendung von Inferenzmechanismen.

In diesem Beitrag werden zunächst aktuelle Herausforderungen der Automobilindustrie dargestellt und grundlegende Entwicklungen von Webtechnologien eingeführt. In Kapitel 4 schließt sich die Vorstellung des Enterprise-Domain-Networks an, das in einem nächsten Kapitel technisch umgesetzt wird. Abschließend folgen die Zusammenfassung der Ergebnisse sowie ein Ausblick für weitere Forschungsansätze.

2 Komplexität als aktuelle Herausforderung der Automobilentwicklung

Neben der permanenten Weiterentwicklung von Automobilen ist die Beherrschung der dabei resultierenden Komplexität eine der wesentlichsten Herausforderungen. Die Komplexität entsteht dadurch, dass Fahrzeuge aufgrund der zunehmenden Integration von Elektronik und Datenverarbeitungskomponenten funktionaler werden [WR11]. Die Funktionalität wird immer öfter mit Hilfe von Elektronik und Software realisiert, wodurch der Fahrzeughersteller neue Kerndisziplinen etablieren muss, um weiterhin das Fahrzeug als Gesamtsystem zu beherrschen. Durch die Funktionalität und der damit verbundenen Vernetzung von Fahrzeugkomponenten entstehen Querschnittsaufgaben, die nicht mehr in den klassischen bauteilorientierten Organisationsformen abgebildet werden können. Es entwickeln sich somit Matrixorganisationen. Zudem kommt eine stetig wachsende Variantenvielfalt und Derivatisierung dazu, die die Komplexität eines Fahrzeugsystems zusätzlich fördert. Die steigende Produktkomplexität schlägt sich somit auf die Organisation nieder, da Organisationen als Ordnungsmuster zur Beherrschung von Komplexität genutzt werden [Ba07].

Neben der Veränderung der Entwicklungsstruktur direkt beim OEM entstehen immer mehr Kooperationen in Form von unternehmensübergreifenden Netzwerken mit externen Partnern. Die Entwicklungsteams und -vorgänge bei der Fahrzeugentwicklung werden dadurch noch vielschichtiger und heterogener. Es wird somit zunehmend anspruchsvoller, die Zusammenhänge in den Projekten für alle Beteiligten so transparent darzustellen,

dass Projektmanagement und Projektbeteiligte effizient kommunizieren und die Entwicklungsaufgaben optimal koordiniert werden können.

Die Beherrschung der Komplexität des Fahrzeugproduktes ist somit der Schlüssel um eine bestimmte Produktqualität zu gewährleisten. Dabei ist die Komplexität keine objektiv wahrnehmbare Eigenschaft, sondern abhängig vom Beobachter eines komplexen Systems, dem Detaillierungsgrad der Beobachtung sowie der zur Beschreibung genutzten Sprache [Lu02]. Dieser Fakt wird im Weiteren aufgegriffen, um ein Konzept zum besseren Umgang mit der Komplexität vorzustellen. Der Begriff des Systems spielt dabei eine grundlegende Rolle. Er kann in seiner theoretischen Form in funktionales, strukturales und hierarchisches Systemkonzept dreigeteilt werden [Bu88]. Gerade im funktionalen und strukturalen Systemkonzept kommen die Wechselwirkungen bzw. Abhängigkeiten zu anderen independenten systembildenden Teilen zur Geltung, was wiederum dem Zustand der steigenden Fahrzeugvernetzung oder dem Wandel zu komplexeren Organisationsformen entspricht. Weiterhin lassen sich Systeme anhand von Komplexitätsfaktoren wie Vielzahl, Vielfalt, Vieldeutigkeit, Veränderlichkeit u.a. zu komplexen System einordnen.

Um diese Komplexitätsfaktoren zu managen und die Transparenz von (interdisziplinären) Entwicklungsprojekten für die Projektbeteiligten zu erhöhen, ist es essentiell, die darin enthaltenen Entitäten [Ba07], wie z. B. Mitarbeiter und Projekte, sowie deren Beziehungen zu erfassen und für eine benutzerfreundliche Repräsentation aufzubereiten. Mitarbeiter arbeiten nicht mehr nur ausschließlich an einem einzelnen Projekt oder haben in einem Projekt lediglich eine Rolle bzw. Verantwortlichkeit, sondern ihr Aufgabenbereich wird zunehmend individueller u.a. aufgrund ihres hohen Spezialisierungsgrades. Das bedeutet, die Eigenverantwortlichkeit wächst zunehmend. Auch wird es für Disziplinarvorgesetzte und andere Kollegen immer intransparenter, an welchen Projekten, Themengebieten und Aufgabenstellungen der einzelne Mitarbeiter tätig ist. Die zeitliche und räumliche Trennung von Kompetenzen und Wissen erschweren zudem die Zusammenarbeit, da die Präsenz der einzelnen Mitarbeiter nicht gegeben sein muss. Es ist deshalb notwendig, nicht nur Informationen über Personen, Projekte und Produkte zu sammeln und abzulegen, sondern diese bei der Speicherung in einen Zusammenhang zu setzen.

3 Web 3.0

Die Entwicklungen der letzten Jahre im Bereich sozialer Ansätze des Web 2.0 und der rasant steigenden Nutzerzahl dieser Systeme tragen dazu bei, dass zunehmend mehr Daten generiert werden. Die damit einhergehende Informationsflut bedeutet einen größeren Aufwand bei der Organisation dieser Daten. Um dem zu begegnen, wurden in [BHL01] die ersten Hauptgedanken zum Semantic Web formuliert. Demnach soll das bestehende Internet nicht ersetzt oder abgelöst werden, sondern vielmehr sollen bestehenden Inhalte durch Semantik angereichert und erweitert werden. Dies soll die Kommunikation zwischen Maschinen und die Nutzung von Informationen durch den Menschen verbessern [Hi08]. Die Daten müssen dazu mit einer definierten Bedeutung annotiert werden, wodurch diese um semantische Metainformationen erweitert werden.

Dies ermöglicht auch, Schlussfolgerungen oder Plausibilitätsprüfungen von Beziehungen vom Menschen auf Maschinen übertragen zu können. Der Mensch soll dadurch bei der Nutzung des Webs unterstützt werden, indem bspw. kontextbezogene Suchergebnisse geliefert werden. Das World Wide Web Consortium[1] (W3C) hat bereits Standards, wie beispielsweise RDF(S), OWL und SPARQL definiert. Diese sind in der Architektur des Semantic Webs[2] („LayerCake“) zusammengefasst und beschreiben den Aufbau der Technologien und beschlossenen Standards.

RDF(S)

Das Resource Description Framework[3] (RDF) bildet die Grundlage für das Semantic Web, da es eine Datenrepräsentation in Form von gerichteten Graphen ermöglicht. Diese bestehen aus mehreren sogenannten RDF-Tripeln, die in der Form Subjekt-Prädikat-Objekt notiert werden. Subjekt und Objekt repräsentieren hierbei jeweils einen Knoten und das Prädikat stellt die Kante dazwischen dar [St09] [Hi08]. Damit können Ressourcen in einer maschinenauswertbaren Form beschrieben werden. Die nachfolgende Abbildung zeigt einen solchen Graphen, der aus fünf Tripel besteht.

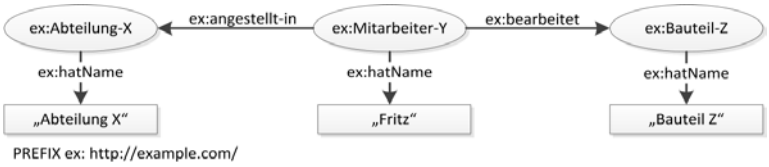

Abbildung 1 Ein RDF-Graph mit fünf RDF-Tripel

RDF Schema[4] (RDFS) stellt dem Benutzer ein Vokabular zur Verfügung, um beispielsweise Klassen, Eigenschaften und deren Vererbung zu modellieren und somit spezifisches Domänenwissen zu formalisieren. Deshalb wird RDFS auch als Wissensrepräsentations- oder Ontologiesprache für leichtgewichtige Ontologien verstanden. [Hi08]

OWL

Die Web Ontology Language[5] (OWL) ist eine weitere standardisierte Ontologiesprache, welche die Ausdrucksmöglichkeiten von RDF(S) um viele Sprachelemente erweitert, technisch aber auf der RDF-Syntax basiert. Dadurch ist es mit OWL möglich, sehr komplexe („heavyweight“) Ontologien zu modellieren. Der Umfang der unterstützten Sprachelemente und der damit verbundenen Ausdrucksstärke klassifiziert OWL in drei jeweils aufbauenden Teilsprachen: OWL Lite, OWL DL und OWL Full.

SPARQL

Mit SPARQL Protocol and RDF Query Language (SPARQL) lassen sich Anfragen an RDF-Datenbestände in Form von Graph-Mustern formulieren und auch Filterbedingun-

[1] www.w3.org/2001/sw
[2] http://www.w3.org/2007/03/layerCake.png
[3] http://www.w3.org/TR/REC-rdf-syntax/
[4] http://www.w3.org/TR/rdf-schema/
[5] http://www.w3.org/TR/owl2-overview/

ISBN 978-3-88579-286-4

gen oder Ausgabeformat des Suchergebnisses angeben. SPARQL ermittelt anhand dieser einfachen Graph-Muster Rückgabewerte, indem Teilgraphen des zugrunde liegenden RDF-Dokuments durchsucht werden. Manipulationen der RDF-Daten lassen sich über SPARQL/update (SPARUL) realisieren. [Hi08]

4 Methodischer Ansatz

Ein Entwickler ist im weiteren Sinne auch ein Wissensarbeiter, da er sich in seinem täglichen Aufgabenbereich einer Vielzahl von Wissensdomänen auseinandergesetzt sieht [Be06]. Das Spektrum reicht dabei u.a. von intensivem Projektmanagement, der eigentlichen Projektarbeit an verschiedenen Komponenten und Produkten, der Zusammenarbeit mit anderen Projektpartnern bis hin zu unternehmensinternem sowie übergreifendem Kontakt- und Beziehungsmanagement. Je nach Anwendungsbereich wird jedoch meist ein unterschiedlicher Detaillierungsgrad (Granularität) des aktuellen Wissens benötigt, um auf dessen Grundlage die richtigen Entscheidungen treffen zu können. Dafür stehen den Entscheidern viele verschiedene Datenbasen und Tools zur Verfügung, die z.T. umständlich zu bedienen sind, keine direkte Verknüpfung der Daten zulassen, für den eigentlichen Zweck zu umfangreiche Funktionen besitzen oder eben gerade die gewünschte Funktionalität nicht bieten. Einen Ansatzpunkt zum besseren Umgang mit Komplexität stellt der Einsatz von Ontologien dar.

4.1 Das Enterprise-Domain-Network

Zur Einteilung des Domänenwissens wird im Folgenden das Enterprise-Domain-Network (EDN) eingeführt. Es dient dazu, das komplexe Wissen bei der Entwicklung von Fahrzeugsystemen in einer ersten Stufe zu strukturieren bzw. zu clustern und legt den Grundstein für weiterführende Betrachtungen. Das Enterprise-Domain-Network sieht drei große Wissensdomänen vor (vgl. Abbildung 2).

Die Unternehmensdomäne (rot gekennzeichnet) beinhaltet vorrangig das personen- und unternehmensbezogene Wissen. Darunter fallen neben den Informationen, die der Mitarbeiter selbst über sich frei gibt, auch die verschiedenen Ausprägungen sozialer Beziehungen zwischen Mitarbeitern. Zudem umfasst diese Domäne auch Wissen über das Unternehmen, die verschiedenen Abteilungen und die damit verbundenen organisatorischen Strukturen der einzelnen Mitarbeiter.

In der Projektdomäne (blau gekennzeichnet) steht das projektspezifische Wissen im Vordergrund. Das bedeutet, dass hier interne Mitarbeiter mit externen Projektpartnern zusammentreffen und an gemeinsamen Aufgaben arbeiten. Die Mitarbeiter sind hier in einer Projektstruktur organisiert, wobei die Organisation parallel zur Unternehmensorganisation und nur temporär über die Dauer des Projektes existiert.

Die von den Mitarbeitern eines Projektes bearbeiteten Aufgaben spiegeln sich im Wissen der Produktdomäne wider (gelb gekennzeichnet). Hier spielen inhaltliche und fachliche Abhängigkeiten im Gegenstandsbereich des zugehörigen Projekts eine Rolle, wie z.B.

der Aufbau von Bauteilen, die Eigenschaften von Funktionen oder die Auswirkungen von Anforderungsänderungen.

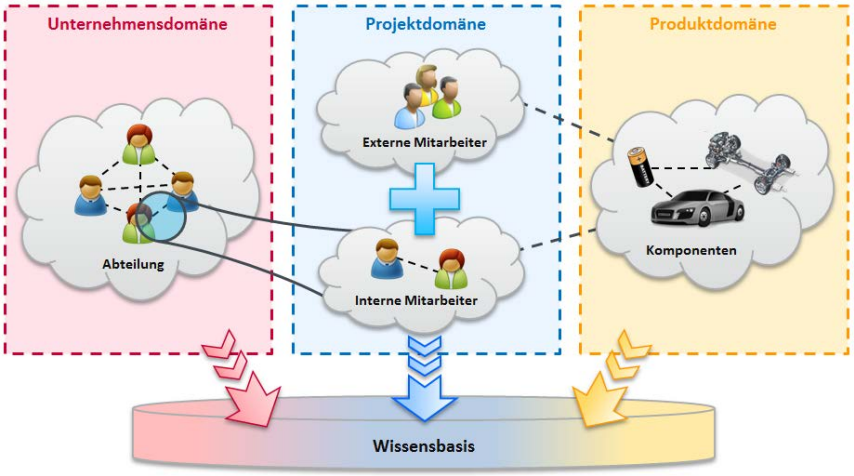

Abbildung 2 Enterprise-Domain-Network

Aus der Gesamtheit dieser sehr speziellen Informationen und Zusammenhänge erwächst ein komplexes Gebilde an kollektivem Wissen, das nutzbar gemacht werden muss, sodass jeder Mitarbeiter und damit auch das Unternehmen davon profitieren kann. Das heißt, das Wissen darf nicht aus dem Kontext gerissen werden und in heterogene Systeme einfließen, sondern es muss in einer gemeinsamen Wissensbasis integriert und weiter in Zusammenhang gebracht werden (vgl. Abbildung 2). Im weiteren Verlauf dieses Beitrags wird anhand des EDN's auf die Unternehmens- und Projektdomäne fokussiert.

4.2 Identifikation generischer Klassen und deren Beziehungen

Das EDN stellt ein konzeptuelles Modell dar, welches generische Beziehungsbestandteile (Klassen) in Relation setzt. Durch Abstraktion konkreter Instanzen gingen diese aus der Analyse der Projektlandschaft bei einem OEM hervor. Im weiteren Verlauf werden exemplarisch einige Klassen sowie deren Attribute und Beziehungen aufgegriffen und anhand dessen das Konzept erläutert.

Person

Personen sind die zentrale Entität in der Unternehmens- aber auch in der Projektdomäne, weil sie in beiden Bereichen eine zentrale aktive Rolle einnehmen. Im unternehmensübergreifenden Umfeld treten Personen als Repräsentanten ihrer Institution bzw. Abteilung auf und stellen für andere eine Kontaktmöglichkeit dar. Sie können daher im konkreten Fall entweder Mitarbeiter oder Geschäftspartner (externe Kontakte) sein.

Kompetenz

Eine Person kann verschiedene individuell ausgeprägte Fähigkeiten, Fertigkeiten oder Kenntnisse besitzen. Diese werden in der Klasse Kompetenz zusammengefasst. Das Ziel dabei ist es, die Kompetenzen der Mitarbeiter aufzuzeigen und effektiv zu nutzen.

Artefakt

Unter einem Artefakt wird allgemein sowohl alles vom Menschen Geschaffene wie auch (Teil-) Ergebnisse von Arbeitsschritten oder -prozessen verstanden. Speziell können Artefakte verschiedene Projekte, Projektbestandteile oder Projektgegenstände sein. Folglich können Personen an diesen Entwicklungs- oder Entstehungsprozessen beteiligt sein. Hier nehmen sie in Bezug auf ihre Tätigkeit eine bestimmte Rolle ein, um z.B. Verantwortlichkeiten für die Umsetzung eines Projektbestandteils festzulegen. Dies wurde durch die Beziehung *hat beteiligtes Artefakt* und der Assoziationsklasse *Rolle der Mitarbeit* modelliert (vgl. Abbildung 3).

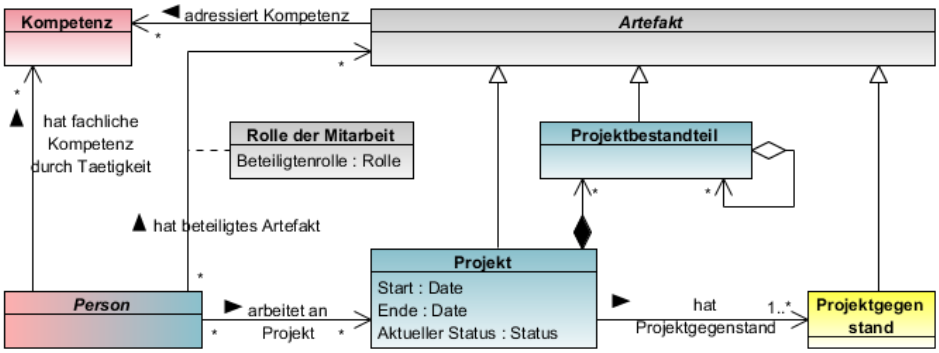

Abbildung 3 Generische Klassen und Beziehungen (Auszug)

Durch ihre Beteiligung an Artefakten sind sie implizit auch im entsprechenden Projekt involviert. Das bedeutet, sie arbeiten am Projekt selbst, in einem der Projektbestandteile oder an den damit verbundenen Projektgegenständen bzw. Komponenten. Die Relation *arbeitet an Projekt* kann damit aus der Tätigkeit an bestimmten Artefakten abgeleitet werden (vgl. Abbildung 3).

Rolle der Mitarbeit

Die Tätigkeiten an den verschiedenen Artefakten sind in Projekten stets mit spezifischen Rollen behaftet. Dies wurde mittels der Assoziationsklasse *Rolle der Mitarbeit* modelliert (vgl. Abbildung 3). Beispiele:

- Projektleiter bzgl. Projekte,
- Projektmitarbeiter bzgl. Projekte, Teilprojekte und Arbeitspakete,
- Bauteilverantwortlicher bzgl. Bauteile, Baugruppen und Komponenten sowie
- Funktionsverantwortlicher bzgl. Funktion

Diese sehr spezifischen Rollen wurden – in Anlehnung an das Prinzip der IMV-Matrix – in generische Rollen abgebildet. Das Attribut *Beteiligtenrolle* kann dazu folgende Werte annehmen: *Verantwortlich*, *Mitwirkung*, *Wird informiert*.

Projekt

Als wichtigster Objekttyp der Projektdomäne kann das Projekt selbst betrachtet werden. Es fungiert als Integrator von Personen, die aus verschiedenen Abteilungen und Unternehmen zusammenkommen können, um gemeinsam an den meist interdisziplinären Aufgaben zu arbeiten.

Ein weiteres generisches Merkmal von Projekten ist, dass sie jeweils in einer vorher geplanten Zeitspanne bearbeitet werden, die durch ein *Start*- und *Enddatum* festgelegt wird. Darüber hinaus ist es für die Projektverfolgung und -kontrolle wichtig, über den derzeitigen Fortschritt bzw. *aktuellen Status* informiert zu sein, damit die Verantwortlichen zeitnah reagieren können.

Abgesehen von organisatorischen Gesichtspunkten eines Projektes wird der eigentliche Inhalt – also das zu entwickelnde Produkt, System oder eine andere Problemstellung – durch die Klasse *Projektgegenstand* repräsentiert, wozu auch die entsprechende Relation *hat Projektgegenstand* existiert (vgl. Abbildung 3).

4.3 Anforderungen an die Architektur

Nachdem die generischen Entitäten (Klassen) und deren Relationen im Kontext der Fahrzeugentwicklung identifiziert und somit ein ontologisches Modell entwickelt wurde, kann eine entsprechende Architektur zur Instanziierung des Modells abgeleitet werden. Dabei müssen die praktische Umsetzung des EDN`s sowie die Aufnahme der generischen Entitäten und Beziehungen einige Prämissen erfüllen. Da die Fahrzeugentwickler Wissensquelle und gleichzeitig auch Wissenssenke sind, wird ihnen eine einfache und benutzerfreundliche Schnittstelle der Wissensakquisition, z.B. durch strukturierte Eingabefelder, zur Verfügung gestellt. Weiterhin steht ihnen eine Verarbeitungskomponente zur Verfügung, die die Klassen sowie deren binäre als auch n-äre Beziehungen aufnimmt und in einer standardisierten Form speichert. Ebenso soll die Komponente das semantische Netz für eine bedarfsgerechte Wiedergabe (möglichst in einer dynamischen Art und Weise) aufbereiten. Zum Schließen neuen Wissens aus den bestehenden Informationen mittels automatisierten Schlussfolgerungsmechanismen sowie zur Konsistenzprüfung dieses Wissens ist der Einsatz einer Inferenzmaschine erforderlich. Dazu wird eine Wissensbasis bereitgestellt, die die instanziierten Klassen und Relationen aufnimmt. Eine spezielle prototypische Umsetzung dieser Anforderungen wird im Weiteren vorgestellt.

5 AEV-Fusion als prototypische Toollösung

Als Basis des Zielsystems wurde Semantic MediaWiki^{+} eingesetzt. Dabei handelt es sich um eine Wissensplattform, die Web 3.0 Technologien beinhaltet. Es besteht aus einem

ISBN 978-3-88579-286-4

semantischen Wiki für Teams, die eine agile Wissensbasis für das gemeinsame Erstellen und Nutzen von Inhalten benötigen. Diese können manuell mit semantischen Metadaten annotiert werden. Darüber hinaus besteht die Möglichkeit, firmenspezifische Geschäftsregeln zu verwenden, um die Wissensbasis automatisch mit abgeleitetem Wissen zu ergänzen. Zudem lassen sich Daten von unterschiedlichen, externen Datenquellen in das Wiki integrieren. Im Folgenden werden die Systemarchitektur und das Zusammenspiel der einzelnen Komponenten näher erläutert (vgl. Abbildung 4).

5.1 Systemarchitektur

Der Controller besteht aus einem MediaWiki als Basissystem, welches durch die einzelnen Subkomponenten (Extensions) erweitert wird. Die beiden essentiellen Erweiterungen für die Arbeit mit semantischen Technologien sind *Semantic MediaWiki* und *SMWHalo*. Das MediaWiki selbst verfügt über eine relationale Datenbank zur Speicherung sämtlicher Wiki-Seiten, deren Inhalte und Einstellungen (vgl. Abbildung 4).

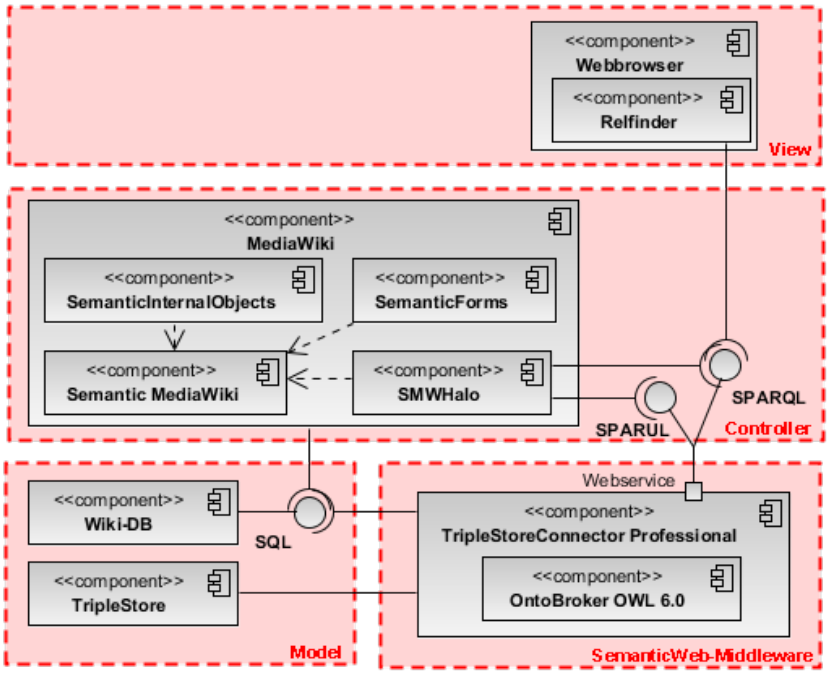

Abbildung 4 Systemarchitektur (vereinfacht)

Semantic MediaWiki (Anwendungsschicht)

Semantic MediaWiki[6] (SMW) zählt zu den bekanntesten semantischen Systemen vor dem Hintergrund des gemeinschaftlichen Wissensmanagements [KV09]. Dabei ist SMW

[6]http://semantic-mediawiki.org, http://www.ontoprise.de/en/solutions/smw-plus/

selbst eine Extension für MediaWiki. Mit Hilfe von dieser Erweiterung kann der Benutzer seine Inhalte mit expliziten, maschineninterpretierbaren Informationen annotieren. Dies trägt zum einen dazu bei, den direkten und schnellen Zugriff auf das Wissen zu gewährleisten. Zum anderen kann die Kennzeichnung mit Metainformationen bewirken, dass dieses externalisierte Wissen in verschiedenen Kontexten wieder verwendet werden kann, z.B. durch dynamische Auswertungen oder in Office-Anwendungen.

Für jedes als generisch identifizierte Merkmal muss jeweils eine separate Wiki-Seite (Property-Seite) erstellt werden, z.B. *Property:Projektstart*. Auf dieser Seite können nun weitere Typisierungen, Restriktionen oder Definitions- und Wertebereiche festgelegt werden, die sog. Property-Charakteristik. Diese Angaben dienen im Weiteren zur Modellierung von terminologischem Wissen (TBox) der Ontologie. Die SMW-Extension ist damit im vorliegenden System ein integraler Bestandteil zum Aufbau der Ontologie und zur Verwendung von semantischen Technologien. *Semantic MediaWik*i stellt dabei die Grundfunktionalität für viele nachfolgende Extensions bereit. Dies wird in Abbildung 4 durch Abhängigkeitsbeziehungen dargestellt.

SMWHalo bietet grafische Benutzerschnittstellen und stellt die Verbindung über einen Webservice zu einer Semantic-Web-Middleware her, die wiederum den Zugriff auf einen Ontologiespeicher (TripleStore) gewährleistet. Damit werden zum einen semantische Annotationen der erstellten Wiki-Seiten über den SPARUL-Endpoint in den TripleStore übernommen. Zum anderen werden über einen SPARQL-Endpoint entsprechende Abfragen an die Ontologie möglich, z.B. als Inline-Query.

Mit Hilfe der *SemanticForms*-Extension können Formulare für Eingabemasken entworfen werden und der Nutzer kann so seine Inhalte in strukturierter Form erfassen. Die erstellten Formulare werden zur Instanziierung der jeweiligen Wiki-Seiten bereitgestellt und dienen dazu, die Ontologie mit Faktenwissen (ABox) zu befüllen.

Im konzeptuellen Modell sind einige Assoziationsklassen enthalten (vgl. Abbildung 3). Zur Abbildung solcher n-ären Relationen dient die Erweiterung *SemanticInternalObjects* (SIO). Relationen mit einer Assoziationsklasse und mehreren Beziehungsattributen können damit instanziiert und intern repräsentiert werden. *SemanticInternalObjects* ist mit *SemanticForms* kompatibel, unterstützt Enumerationswerte und hinzukommende Beziehungsattribute können flexibel erweitert werden.

TripleStore-Connector (Middleware-Schicht)

Der *TripleStore-Connector* (TSC) ist eine Semantic-Web-Middleware, die zur Kommunikation zwischen Wiki und *TripleStore* dient und darüber hinaus Reasoning-Funktionalität bereitstellt. Der Zugriff wird über einen Webservice gewährleistet. Es werden dadurch zwei Endpoints bereitgestellt:

- SPARQL-Endpoint (<host>:<port>/sparql?) für lesende Zugriffe,
- SPARUL-Endpoint (<host>:<port>/sparul?) für schreibende Zugriffe.

Durch diese standardisierten Schnittstellen wird nicht nur die Verbindung zum MediaWiki über die SMWHalo-Extension, sondern außerdem auch für sämtliche externen Zugriffe hergestellt (vgl. Abbildung 4).

Der TSC Professional beinhaltet den OntoBroker OWL 6.0[7]. Es handelt sich dabei um eine skalierbare und sehr leistungsfähige Inferenzmaschine. Dieser besteht aus einer Variante des KAON2-Reasoners[8], welche RDF(S), OWL DL, SPARQL sowie Inferenzregeln in ObjectLogic[9] unterstützt. Anfragen an die Ontologie werden vom Webservice entgegengenommen und vom OntoBroker verarbeitet (vgl. Abbildung 4). Dieser greift über den jeweiligen Konnektor auf den TripleStore zu, wendet ggf. Inferenzmechanismen an und liefert schließlich das entsprechende Ergebnis zurück.

TripleStore & Relationales Datenbanksystem (Persistenzschicht)

Alle Wiki-Seiten inklusive der Inhalte, semantischen Annotationen und anderen Einstellungen werden primär in verschiedenen Tabellen einer relationalen Datenbank gespeichert (vgl. Abbildung 4). Beim Start der Semantic-Web-Middleware wird der TripleStore mit diesen aktuellen Daten initialisiert. Das heißt, dass im TripleStore ein neuer Graph angelegt wird, die Daten aus den Tabellen der relationalen Datenbank werden ausgelesen und im TripleStore als RDF-Tripel gespeichert. Damit enthält der TripleStore die gesamte Ontologie. Inferenzregeln werden beim Start ebenso geladen, sodass die Ontologie inferiert werden kann.

Webbrowser & interaktive Visualisierung (Präsentationsschicht)

Da es sich bei diesem Prototypen um eine Webapplikation handelt, wird die Benutzeroberfläche über einen Webbrowser dargestellt (vgl. Abbildung 4). Damit können Benutzereingaben in dafür vorgesehene Felder (*SemanticForms*) eingetragen werden. Darüber hinaus werden hier dem Anwender die Auswertungen, Analysen oder einfache Abfragen an die Ontologie dargestellt.

Zur interaktiven Visualisierung von Objekten und deren Beziehungen wurde der *RelFinder*[10] – eine Flash-Anwendung zur Darstellung von RDF-Datenbeständen – verwendet. Er kommuniziert ebenfalls über den SPARQL-Endpoint mit dem TSC und erhält so auch inferierte Relationen aus der Ontologie.

5.2 Inferenzmechanismen

Inferenzmechanismen werden von einer Inferenzmaschine (hier: OntoBroker OWL 6.0) zur Verfügung gestellt und können aufgrund der formalen Semantik von OWL und implementierten Inferenzregeln aus bereits vorhandenen Fakten neue Fakten berechnen. Darüber hinaus kann die Ontologie auch auf Inkonsistenzen geprüft werden. Zum einen kann die Ableitung neuen Faktenwissens durch die jeweils angegebene Property-Charakteristik beeinflusst werden, z.B. über:

[7] http://www.ontoprise.de/de/produkte/ontobroker/
[8] http://kaon2.semanticweb.org/
[9] ObjectLogic ersetzt ab OntoBroker 6.0 die vorher verwendete F-Logic.
[10] http://www.visualdataweb.org

- Symmetrische Eigenschaften (OWL-Class owl:SymmetricProperty),
- Inverse Eigenschaften (OWL-Property owl:inverseOf),
- Transitive Eigenschaften (OWL-Class owl:TransitiveProperty).

Zum anderen können benutzerdefinierte Schlussregeln erstellt werden, z.B. Berechnungsregeln (Calculation rules), Definitionsregeln (Definition rules) und Eigenschaftsverkettungsregeln (Property chaining rules). Dadurch können neue, explizite Relationen aus bereits vorhandenen Beziehungen geschlossen werden, sobald das in einer Regel definierte Graph-Muster von der Inferenzmaschine erkannt wird.

```
<rule name="Kompetenz durch Taetigkeit" type="Property chaining">
   ?PERSON[prop#Hat_fachliche_Kompetenz_durch_Taetigkeiten->?KOMPETENZ] :-
      ?SIO[prop#Beteiligte_Ressource->?PERSON]
      AND ?SIO[prop#Hat_beteiligtes_Artefakt->?ARTEFAKT]
      AND (
         ?SIO[prop#Hat_Beteiligtenrolle->"Verantwortlich"^^_string]
         OR ?SIO[prop#Hat_Beteiligtenrolle->"Mitwirkung"^^_string]
      )
      AND ?ARTEFAKT[prop#Adressiert_Kompetenz->?KOMPETENZ] .
</rule>
```

Abbildung 5 Eine Eigenschaftsverkettungsregel

Die Abbildung 5 zeigt im oberen Bereich einen Graphen bestehend aus mehreren RDF-Tripeln. Darunter ist eine Eigenschaftsverkettungsregel in ObjectLogic abgebildet, die für das Schließen der Beziehung *hat fachliche Kompetenz durch Taetigkeit* verantwortlich ist. Das bedeutet, eine Person (hier: *Mathias Schraps*) ist an einem Artefakt (hier: *AP9 Dokumentation und Reporting*) in einer entsprechend aktiven Rolle tätig. Dabei wird eine bestimmte Kompetenz zur Bearbeitung der Aufgaben unterstellt. Diese wird entsprechend adressiert (hier: *Wissensmanagement*). Die Inferenzmaschine erkennt das in der Regel definierte Graph-Muster und instanziiert daraufhin die grün gekennzeichnete Beziehung. Die ermittelte Kompetenz kann somit auf der Profilseite von *Mathias Schraps* angezeigt werden.

5.3 Benutzeroberfläche von AEV-Fusion

Projekte können erstellt, verwaltet oder strukturiert werden. Dazu werden vom jeweiligen Verantwortlichen alle als generisch identifizierten Informationen über ein Formular eingetragen, wie z.B. die Laufzeit, aktuelle Projektphase oder Projektziele. In diesem Zuge kann sogleich die erste Ebene der Strukturierung – also die Projektbestand-

teile – aufgenommen werden. Nach dem Abspeichern stehen die Informationen allen Projektbeteiligten zur Verfügung (vgl. Abbildung 6-Nr. 1).

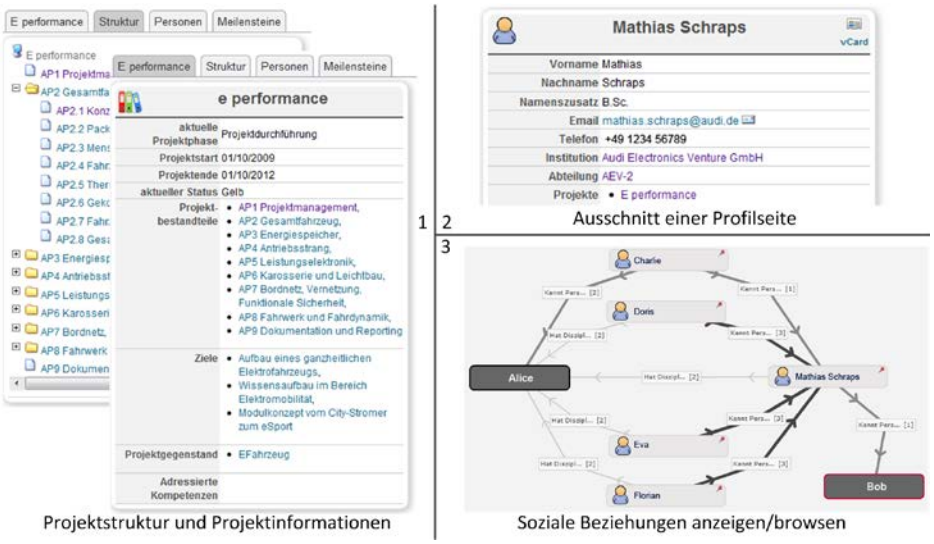

Abbildung 6 Prototyp AEV-Fusion

Der Ausgangspunkt für sämtliche Aktivitäten eines Benutzers ist dessen Profilseite, auf der sich jeder Mitarbeiter selbst darstellt, Tätigkeiten verwalten oder soziale Beziehungen herstellen kann (vgl. Abbildung 6-Nr. 2). Die instanziierten Tätigkeitsbeziehungen inklusive der angegebenen Rolle und genauen Tätigkeitsbeschreibung sind sowohl auf der Profilseite des Benutzers als auch auf der Wiki-Seite des Artefakts dargestellt. Diese sind dort auch für Dritte sichtbar und können somit Ausgangspunkt für eine Kontaktaufnahme sein. Zusätzlich werden hier vom Benutzer angegebene und vom System inferierte Kompetenzen dieser Person dargestellt. Damit wird die Suche nach einem geeigneten Experten unterstützt.

Hergestellte soziale Beziehungen können über die interaktive Visualisierungskomponente *RelFinder* dargestellt werden. Die gemeinsam genutzte Schnittstelle über den SPARQL-Endpoint sichert die konsistente Darstellung der Beziehungen sowohl auf der Profilseite als auch im *RelFinder*. Abbildung 6-Nr. 3 stellt ein Netz von sozialen Beziehungen dar und zeigt damit einen möglichen und auch kurzen Informationsweg von *Alice* (links) über *Mathias Schraps* (rechts) hin zu *Bob* (rechts unten) auf. Die Intensität der Beziehungen ist jeweils an der unterschiedlichen Stichstärke zu erkennen.

Dem Benutzer bieten sich durch *AEV-Fusion* noch weitere Einstiegsmöglichkeiten, sich über Projekte, deren Inhalte und Strukturen, zu entwickelnde Komponenten, beteiligte Partner oder Personen zu informieren. Durch die bereits erläuterten Zusammenhänge und die sich daraus ergebenden Verweise auf entsprechende Wiki-Seiten kann der Nutzer in den gewünschten Kontext hineinspringen und die für ihn interessanten Informationen abrufen. Dadurch wird dem Benutzer die Möglichkeit zur Exploration der

kollektiven Wissensbasis gegeben und unterstützt damit den Prozess der Internalisierung von Wissen nach NONAKA und TAKEUCHI [NT97].

6 Verwandte Arbeiten

Bei angestellten Nachforschungen wurde kein Ansatz entdeckt, der die Kombination EDN, Ontologien, Wiki und Web 3.0 verfolgt. Die Arbeiten von HEß, CHEN und SYLDATKE fokussieren die ontologiebasierte Entwicklung einer CAx-Architektur [HCS10]. Sie konzentrieren sich dabei ausschließlich auf Methoden der Fahrzeugentwicklung, wie Design, Engineering und Test. Ihre Arbeit wäre damit ausschließlich in die Produktdomäne des EDN einzuordnen.

In [Bu05] wird ein Ansatz vorgestellt mit Hilfe von Ontologien den Innovationsprozess zu unterstützen und zu beschleunigen. Zielrichtung dieser Arbeit ist allerdings vor allem das Managen von Innovationsideen, wobei aber auch Artefakte aus Projektdomäne des Enterprise-Domain-Networks eine Rolle spielen. Unser in diesem Beitrag vorgestellter Ansatz ist sehr viel stärker auf die Projektdomäne ausgerichtet und möchte ein in der Praxis verwendbares benutzerfreundliches Werkzeug anbieten.

Ontologien für Unternehmensstrukturen sind bereits in anderen Arbeiten, wie z. B. in [Ga03] vorgeschlagen worden. Die Modellierung der Unternehmensorganisation ist ähnlich zu unserem Enterprise-Domain-Network. Der Ansatz ist aber nur in einem studentischen Umfeld getestet worden.

7 Zusammenfassung und Ausblick

Die Organisation der Entwicklung technischer Systeme muss sich der Herausforderung der stetigen Evolution und der damit verbundenen zunehmenden Komplexität stellen. Daher werden konzeptuelle und technische Hilfsmittel zur Beherrschung der Komplexität benötigt. Das EDN als konzeptueller Ansatz in Kombination mit technischen Ansätzen des Wissensmanagements können dabei helfen, die Entwicklung von Fahrzeugsystemen strukturierter und transparenter zu gestalten. Dabei werden in einer Wissensbasis Informationen zum Unternehmen, den Projekten und den Produkten aufgenommen, in Relation zueinander gesetzt, mit inferierten Informationen angereichert und benutzerfreundlich sowie bedarfsgerecht wieder zur Verfügung gestellt. Somit können die Entwickler zielgerichtet Zusammenhänge aus einem komplexen Kontext wie der Fahrzeugentwicklung abrufen, um sie zu analysieren. Eine erste Evaluierung von AEV-Fusion lieferte darüber hinaus positives Feedback aller Evaluierungsteilnehmer.

Da neben der Unternehmens-, Projekt- und Produktdomäne auch die Prozesse die Fahrzeugentwicklung beeinflussen, wird eine Erweiterung des EDN's um diese 4. Domäne angestrebt. Ebenso ist zu untersuchen, in wie weit (semi-)formale Sprachen zur Beschreibung von Produktbestandteilen eingesetzt werden können, um daraus Ontologien zu erstellen, die zur Anforderungsanalyse und zur Quellcodegenerierung genutzt werden können.

8 Literaturverzeichnis

[Ba07] BANDTE, Henning: *Komplexität in Organisationen : Organisationstheoretische Betrachtungen und agentenbasierte Simulation.* Wiesbaden : Deutscher Universitäts-Verlag, 2007.

[Be06] BERNARDI, Ansgar ; HOLZ, Harald ; MAUS, Heiko ; VAN ELST, Ludger: *Komplexe Arbeitswelten in der Wissensgesellschaft. In:* PELLEGRINI, Tassilo; BLUMAUER, Andreas (Hrsg.): *Semantic Web : Wege zur vernetzten Wissensgesellschaft.* Berlin , Heidelberg : Springer, 2006, S. 27–45.

[BHL01] BERNERS-LEE, Tim ;HENDLER, James ; LASSILA, Ora: *The Semantic Web : a new form of Web content that is meaningful to computers will unleash a revolution of new possibilities.* In: *Scientific American* 284 (2001), Nr. 5, S. 35–43

[Bu05] BULLINGER, Hans-Jörg ; WARSCHAT, Joachim; SCHUMACHER, Oliver; SLAMA, Alexander; OHLHAUSEN, Peter: *Ontology-Based Project Management for Acceleration of Innovation Projects*. In: From Integrated Publication and Information Systems to Virtual Information and Knowledge Environments, LNCS 3379, Springer, 2005, S. 300-308.

[Bu88] BUTEWEG, Jörg (Hrsg.): *Systemtheorie und ökonomische Analyse : Ansätze einer neuen Denkweise vor neo-klassischem Hintergrund.* Pfaffenweiler : Centaurus-Verlagsgesellschaft, 1988 (Wirtschaftswissenschaften 45).

[Ga03] GARCIA, Ana Cristina Bicharra; KUNZ, John; EKSTROM, Martin; KIVINIEMI, Arto: Building a Project Ontology with Extreme Collaboration and Virtual Design & Construction, CIFE Technical Report #152, Stanford University, November 2003.

[HCS10] HEß, Claudia ; CHEN, Willy ; SYLDATKE, Thomas: *Business-oriented CAx Integration with Semantic Technologies Revisited*. In: Fähnrich, Klaus-Peter; Franczyk, Bogdan (Hrsg.): Informatik 2010 : Service Science - Neue Perspektiven für die Informatik. Beiträge der 40. Jahrestagung der Gesellschaft für Informatik e.V. (GI) Band 2 : GI, 2010 (LNI, 176), S. 115–120.

[Hi08] HITZLER, Pascal ; KRÖTZSCH, Marcus ; RUDOLPH, Sebastian ; SURE, York: *Semantic Web : Grundlagen.* Berlin, Heidelberg : Springer-Verlag, 2008.

[KV09] KRÖTZSCH, Markus ; VRANDEČIĆ, Denny: *Semantic Wikipedia. In:* BLUMAUER, Andreas; PELLEGRINI, Tassilo (Hrsg.): *Social Semantic Web : Web 2.0 - Was nun?* Berlin : Springer, 2009, S. 393–421.

[Lu02] LUHMANN, Niklas: *Einführung in die Systemtheorie.* 1. Aufl. Heidelberg : Carl-Auer-Systeme-Verl, 2002.

[NT97] NONAKA, Ikujiro ;TAKEUCHI, Hirotaka: *Die Organisation des Wissens : Wie japanische Unternehmen eine brachliegende Ressource nutzbar machen.* Frankfurt am Main : Campus Verlag, 1997.

[Sc05] SCHUH, Günther: *Produktkomplexität managen : Strategien Methoden Tools.* 2. Aufl. München, Wien : Hanser, 2005.

[St09] STUCKENSCHMIDT, Heiner: *Ontologien : Konzepte, Technologien und Anwendungen.* Berlin, Heidelberg : Springer-Verlag, 2009.

[WR11] WALLENTOWITZ, Henning (Hrsg.); REIF, Konrad (Hrsg.): *Handbuch Kraftfahrzeugelektronik : Grundlagen - Komponenten - Systeme - Anwendungen.* 2. Aufl. Wiesbaden : Vieweg+Teubner, 2011.

ISBN 978-3-88579-286-4