Entwurf von Informationsintegrationssystemen auf der Basis der Merkmalsmodellierung

Susanne Busse

Technische Universität Berlin
Computergestützte Informationssysteme
sbusse@cs.tu-berlin.de

Johann-Christoph Freytag

Humboldt-Universität zu Berlin
Institut für Informatik
freytag@dbis.informatik.hu-berlin.de

Abstract. Informationsintegrationssysteme bieten einen integrierten Zugriff auf eine Menge autonomer Datenquellen. Ihr Entwurf erfordert die Abwägung zwischen einer Vielzahl sich zum Teil widersprechender Anforderungen von Nutzern und Datenbereitstellern. Es sind jedoch Typen von Integrationssystemen bekannt, die grob verschiedene Integrationsvarianten beschreiben.
In diesem Papier wird gezeigt, wie die Merkmalsmodellierung aus dem Bereich der Produktlinien benutzt werden kann, um Integrationsvarianten systematisch zu beschreiben. Durch ein Matching der Anforderungsspezifikation eines geplanten Systems und der Definition existierender Integrationsvarianten kann so konkret die Wahl eines Integrationsansatzes unterstützt werden. Der Ansatz ermöglicht somit den Entwurf eines Systems als auch deren Dokumentation.

1 Einleitung

Informationssysteme setzen meist auf eine Reihe verschiedener autonomer und damit heterogener Datenquellen auf, deren Daten gesammelt und integriert werden müssen. Wir bezeichnen diese Systeme als *Informationsintegrationssysteme (IIS)*. Beim Entwurf eines IIS müssen sowohl die gewünschten funktionalen und nicht-funktionalen Anforderungen späterer Nutzer also auch die Charakteristika der zu integrierenden Daten und Datenquellen berücksichtigt werden. Die dabei auftretende Vielfalt führt zu einem entsprechend großem Spektrum existierender IIS, das von föderierten Datenbanksystemen bis hin zu Suchmaschinen reicht, die auf sehr unterschiedlichen Integrationsansätzen beruhen.

Eine Hilfestellung zur Orientierung bieten Diskussionen wichtiger Entscheidungskriterien und 'Trade-Offs'. Als Beispiele seien hier nur die Diskussionen bzgl. semantischer Heterogenität in [Hu97], bzgl. des Umgangs mit der unterschiedlichen Strukturiertheit der unterliegenden Daten in [Ha03], bzgl. von Qualitätsaspekten in [BP04] und [NFL04] oder die Klassifikation in [PBC00] genannt. Neben der Betrachtung spezifischer Aspekte haben sich außerdem verschiedene Typen von IIS herausgebildet, deren Beschreibung zur Wahl eines geeigneten Integrationsansatzes herangezogen werden, etwa mediatorbasierte Informationssysteme ([Wi97]), Data Warehouses ([In96]), Peer Data Management Systeme ([Ha03]), Portale ([Ma02]) oder Information Retrieval (IR)-basierte Systeme wie Suchmaschinen ([Ar01]). In [LC03] werden IIS im Bereich der Bioinformatik diskutiert, was

einen Eindruck der vielfältigen Herausforderungen beim Entwurf eines IIS gibt.

Die genannten Arbeiten bieten zwar einen guten Ausgangspunkt zur Orientierung beim Entwurf eines IIS, ihre Einbindung in den Entwicklungsprozess ist jedoch unzureichend. Weder existiert eine gemeinsame Basis, die die für IIS relevanten Kriterien systematisiert, den verschiedenen Phasen innerhalb des Entwicklungsprozesses zuordnet, noch kann eine adäquate Dokumentation getroffener Entscheidungen erfolgen, die Entwurf und Anforderungsspezifikation in Bezug setzen und so die Nachvollziehbarkeit des Entwurfs gewährleisten würde.

Genau an dieser Stelle setzt unsere Arbeit an. Wir zeigen in diesem Papier, wie die Merkmalsmodellierung für den Entwurf von IIS benutzt werden kann. Die Merkmalsmodellierung, eingeführt im Rahmen der FODA-Methode[1] ([Ka90]), wird im Bereich von Produktlinien ([CN01]) und der generativen Softwareentwicklung ([CE00]) verwendet. Ein Merkmalsmodell beschreibt die variablen und gemeinsamen Merkmale der Produkte einer Produktlinie oder Systeme einer Systemfamilie. Das Modell kann zur Anforderungsspezifikation eines zu entwickelnden Produkts verwendet werden und ist Basis für die anschließende Generierung eines Entwurfs oder einer konkreten Konfiguration des Produkts entlang der für die verschiedenen Typen von IIS definierten Referenzarchitekturen.

Dieses Vorgehen lässt sich im Rahmen der modellgetriebenen Softwareentwicklung in jeder Phase einsetzen: Das Merkmalsmodell beschreibt die Möglichkeiten der in der jeweiligen Phase zu treffenden Entscheidungen und steuert die Richtung des Übergangs in die nächste Entwicklungsphase. Es dient damit der Verknüpfung von Anforderungen bzw. Entscheidungsalternativen und der späteren Lösung. Entgegen der bisherigen textuellen Dokumentation bietet die formale Basis und Werkzeugunterstützung der Merkmalsmodellierung so einen erheblich besseren Rahmen in Bezug auf Nachvollziehbarkeit und Änderbarkeit von Softwarelösungen.

Wir wollen hier konkret die Wahl eines passenden Typs von IIS im Rahmen des Entwurfs betrachten, indem eine Unterstützung bzgl. folgender Fragestellungen angeboten wird:

1. Welcher der bekannten Integrationsansätze bietet sich für das geplante IIS an?
2. Welche spezifischen Herausforderungen müssen noch gemeistert werden?

Nach einer Einführung in die Merkmalsmodellierung (Kapitel 2) modellieren wir dazu die Merkmale von IIS im Allgemeinen sowie – im Sinne einer Spezialisierung – spezifischer Typen von IIS (Kapitel 3). Die Merkmale beziehen sich dabei insbesondere auf die für einen Nutzer sichtbaren Eigenschaften der Informationssuche. Als spezifische Typen betrachten wir exemplarisch mediatorbasierte und IR-basierte Integrationssysteme. Auf die Merkmalsmodelle aufsetzend zeigen wir dann, wie ein Matching der Typdefinitionen mit einer Anforderungsspezifikation eines IIS erfolgen kann, um die Wahl eines IIS-Typs zu erleichtern sowie konkrete Schwierigkeiten aufzeigen zu können (Kapitel 4). Dazu ziehen wir exemplarisch den Entwurf des IIS für das ’European Migration Networks’ heran.

1. FODA = Feature-Oriented Domain Analysis

Beispiel: Das 'European Migration Network'

Der Aufbau des 'European Migration Network' (EMN)[1] wurde von der Europäischen Kommission initiiert mit dem Ziel, die Zusammenarbeit von Organisationen und Individuen im Bereich Migration und Asyl auf europäischer wie nationaler Ebene zu unterstützen. In diesem Kontext wurde 2004/05 ein Informationssystem entwickelt, das die Daten der sogenannten 'National Contact Points' (NCPs) der beteiligten Staaten integriert, um eine vergleichbare und umfassende Informationsbasis für den Bereich zu schaffen.[2] Das Informationssystem umfasst Kontaktdaten der Mitglieder des Netzwerks, Publikationen, Gesetzestexte sowie Urteile, Statistiken, Presseinformationen sowie Beschreibungen relevanter Informationsquellen, etwa online verfügbarer Bibliothekskataloge. Eine detailliertere Darstellung ist in [Bu05] zu finden.

Das EMN-Informationssystem bietet sich hier als Beispiel an, da sowohl eine Umsetzung als mediatorbasiertes IIS als auch als IR-basiertes IIS denkbar ist: Ein mediator-basiertes System kommt der geforderten engen Integration entgegen, wohingegen ein IR-basiertes System im Sinne einer Suchmaschine beste Möglichkeiten bietet, der Heterogenität seitens der NCPs zu begegenen.

2 Merkmalsmodellierung

Die Merkmalsmodellierung (auch Feature Modellierung genannt) wurde als Bestandteil der FODA-Methode ([Ka90]) eingeführt und wird heute vor allem zum Variantenmanagement im Bereich der Produktlinienentwicklung eingesetzt. Eine Übersicht über existierende Methoden ist in [CE00] zu finden. [AC04] beschreibt ein Werkzeug für die Merkmalsmodellierung.

2.1 Grundlagen der Merkmalsmodellierung

Ein Merkmalsmodell beschreibt die gemeinsamen und variablen Eigenschaften von Systemen einer Systemfamilie – der Domäne. Ein *Merkmal* (engl.: *feature*) ist eine Eigenschaft eines Produkts oder allgemeiner eines Konzepts, die für mindestens einen Beteiligten relevant ist, etwa für den Nutzer eines Informationssystems.

Ein Merkmalsmodell besteht aus einem Diagramm, das die Merkmale hierarchisch anordnet, und einem Glossar, das die Merkmale dokumentiert. Abbildung 1 zeigt ein Beispiel eines Merkmalsdiagramms in der hier verwendeten, an [Ka90] angelehnten Notation. Die hierarchische Beziehung zwischen Merkmalen spiegelt eine Verfeinerung der Beschrei-

1. www.european-migration-network.org/
2. Das Projekt wurde initiiert von der GD Justiz, Freiheit und Sicherheit (GD JLS) der Europäischen Kommison. Die Koordination im Rahmen der Vorbereitungsphase zum Aufbau des EMN erfolgte vom Berliner Institut für Vergleichende Sozialforschung e.V. und der Fachgruppe CIS / TU Berlin.

bung wider. Die Art der Beziehung wird dabei nicht weiter unterschieden – sowohl eine Spezialisierung, Komponentenbeziehung oder Instanzbildung im Sinne der konzeptionellen Modellierung ist denkbar. Erweiterungen der Merkmalsmodellierung beinhalten eine entsprechende Unterscheidung, [Cz05] führt darüber hinaus auch Kardinalitäten ein.

Wir werden jedoch ausschließlich die Basiskonzepte der Merkmalsmodellierung verwenden, um den eigentlichen Fokus der Merkmalsmodellierung zu verdeutlichen: die Beschreibung von *Variabilität*, d.h. die Beschreibung der möglichen Varianten in der betrachteten Domäne. Variabilität findet sich in einem Merkmalsdiagramm in zweifacher Form: Jedes Merkmal beschreibt eine gemeinsame Eigenschaft (ist obligatorisch) oder eine optionale Eigenschaft. Zum anderen können Merkmale zu Alternativen gruppiert werden, wobei xor- und or-Gruppen unterschieden werden. Neben diesen direkt im Diagramm ausgedrückten Abhängigkeiten zwischen den Merkmalen können Abhängigkeiten auch zusätzlich textuell ergänzt werden, wobei insb. zwei Beziehungen betrachtet werden: *Requires* beschreibt die Abhängigkeit eines Merkmals von einem anderen, *Excludes* den gegenseitigen Ausschluss.

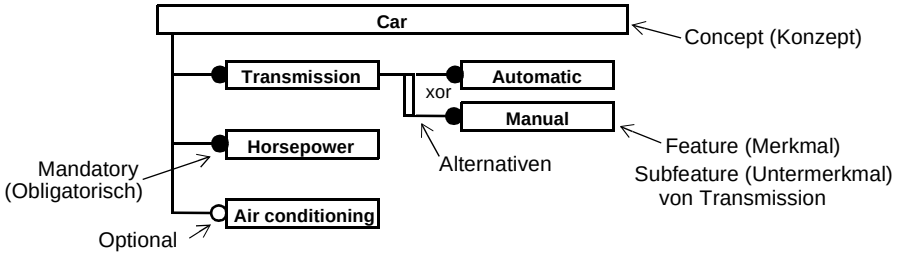

Abb. 1: Beispiel eines Merkmalsdiagramms ([Ka90])

Ein konkretes Produkt, im obigen Beispiel ein konkretes Auto, lässt sich entlang eines Merkmalsmodells beschreiben, indem die tatsächlichen Eigenschaften des Produkts ausgewählt („gebunden“) und die in dem Exemplar nicht auftretenden Eigenschaften gestrichen („gelöscht“) werden. Die Variabilität wird dabei entfernt. Bei der Bindung müssen natürlich die in dem Merkmalsmodell ausgedrückten Abhängigkeiten berücksichtigt werden. So kann ein Merkmal nur dann gewählt werden, wenn auch das Vaterelement gewählt wurde, ein obligatorisches Merkmal *muss* ggf. gewählt werden usw.

Sind alle variablen Eigenschaften entweder gebunden oder gelöscht, spricht man von einer *Konfiguration*, die genau ein Exemplar der Produktfamilie beschreibt. Dieser Prozess lässt sich natürlich auch in Schritte gliedern, wobei bei jedem Schritt einige Entscheidungsalternativen getroffen werden, andere jedoch noch offen bleiben („undecided“). Eine Konfiguration kann bezogen auf das zugrundeliegende Merkmalsmodell beschrieben werden, indem für jedes Merkmal die Bindung angegeben wird. Es ist jedoch auch möglich, eine Konfiguration selbst als Merkmalsmodell darzustellen – ggf. ohne jegliche Variabilität.

Betrachten wir insbesondere den Prozess der Konfiguration, lässt sich ein Merkmalsmodell daher auch dahingehend interpretieren, dass es eine Menge möglicher Konfigurationen beschreibt, wobei jede Konfiguration die in dem Merkmalsmodell definierten Bedin-

gungen bzgl. der Bindung von Merkmalen erfüllt. Ein Zwischenschritt in dem Konfigurationsprozess wird auch *Spezialisierung* eines Merkmalsmodells genannt. Eine Spezialisierung beschreibt damit eine Untermenge der ursprünglichen Produktfamilie, da die Menge der möglichen Konfigurationen kleiner wird.

Wenden wir die Merkmalsmodellierung auf Informationsintegrationssysteme an, so werden wir die Klasse von IIS als ein Merkmalsmodell beschreiben, spezifische Typen von IIS, wie mediatorbasierte IIS, als Spezialisierung davon und die vollständige Beschreibung eines existierenden oder geplanten IIS als Konfiguration. Um die Beschreibung eines IIS-Typs noch etwas feiner angeben zu können, erweitern wir dabei die Charakterisierung eines Merkmals um die Angabe 'common' (üblich) bzw. 'uncommon' (unüblich), um besonders häufige bzw. seltene Merkmale zu identifizieren.

2.2 Formalisierung von Merkmalsmodellen

Ein Merkmalsmodell definiert hierarchisch angeordnete Merkmale eines Konzepts sowie deren Variabilität. Entsprechend definieren wir ein Merkmalsmodell *FM* als ein Tupel

FM = (Concept, Features, FeatureGroups,
name, subfeatureOf, memberOf,
variability, groupVariability, Constraints)

mit der in der folgenden Tabelle beschriebenen Bedeutung.

Element	Beschreibung
Concept	ein Konzeptelement, das die beschriebene Familie bezeichnet – die Wurzel des Diagramms
Features = *SolitaryFeatures* ∪ *GroupedFeatures*	eine Menge von Merkmalen – alleinstehende (wie Air conditioning) oder gruppierte (Automatic, Manual)
FeatureGroups	eine Menge von Merkmalsgruppen
Funktionen zur Abbildung der Merkmalshierarchie	
name: Features ∪ { *Concept* } → Σ^+	zur Beschreibung des Namens der Elemente
subfeatureOf: *Features* → *Features* ∪ { *Concept* }	zur Ermittlung des Vaters eines Merkmals
memberOf: *GroupedFeatures* → *FeatureGroups*	zur Ermittlung der Merkmalsgruppe eines gruppierten Merkmals
Variabilitätselemente und dadurch beschriebene Konfigurationsbedingungen	
groupVariability: *FeatureGroups* → { *or, xor* }	zur Definition der Variabilität einer Merkmalsgruppe
variability: *Features* → { *mandatory, optional* }	zur Definition der Variabilität eines Merkmals
Constraints = *ImplicitConstraints* ∪ *ExplicitConstraints*	der Menge von Konfigurationsbedingungen, die explizit angegeben wurden oder sich implizit aus dem Merkmalsdiagramm ergeben

Tabelle 1: Formalisierung von Merkmalsmodellen

Konfiguration und Spezialisierung

Eine **Konfiguration** bezieht sich auf ein Merkmalsmodell. Es benennt das konkret beschriebene Produkt (die konkrete Ausprägung des Konzepts oder der Wurzel des Merkmalsmodells) und gibt eine Bindung für die Merkmale des Merkmalsmodells an (wobei natürlich die durch das Merkmalsmodell beschriebenen Constraints erfüllt sein müssen). Formal beschreiben wir eine Konfiguration *Config* eines Merkmalsmodells *FM* als ein Tupel

Config = (Concept, name, FM, binding)

wobei *Concept* wiederum ein Konzeptelement ist, *name* die namensgebende Funktion, *FM* ein Merkmalsmodell mit den Merkmalen *Features* und *binding* eine Funktion

binding: Features $\rightarrow$ { *bound, removed* }

die angibt, ob ein Merkmal bei der Konfiguration gebunden oder gelöscht wurde.

Analog lässt sich eine **unvollständige Konfiguration** *IncompleteConfig* definieren als

IncompleteConfig = (Concept, name, FM, incompleteBinding)

wobei lediglich die bindende Funktion auch ein Offenlassen der Bindung erlaubt:

incompleteBinding: Features $\rightarrow$ { *bound, removed, undecided* }

Eine **Spezialisierung** *Specialization,* definiert als ein Tupel

Specialization = (Concept, name, FM, specBinding)

lässt wiederum als Erweiterung der bindenden Funktion auch die Spezifikation von 'common' bzw. 'uncommon'-Merkmalen zu:

specBinding: Features $\rightarrow$ { *bound, common, uncommon, removed, undecided* }

3 Klassifikation von Informationsintegrationssystemen

Die hier vorgestellte Klassifikation von Informationsintegrationssystemen (IIS) versteht sich nicht als neue Definiton der Merkmale von IIS, sondern stützt sich auf existierende Klassifikationen und Diskussionen spezifischer Aspekte. Insbesondere seien dabei die einführende Diskussionen in [LC03] sowie die Klassifikation wissensbasierter IIS in [PBC00] genannt. Diese führt ebenfalls eine Präzisierung von Eigenschaft ein, die einem Merkmalsmodell ähnelt, bezieht sich allerdings nur auf einen Teil der hier betrachteten IIS. Bezüglich spezifischer Aspekte sind insbesondere die Übersicht in [Hu97] sowie die Diskussion von Datenqualitätsaspekten ([SMB05]) zu nennen.

Ebenfalls zu erwähnen ist, dass die Klassifikation nicht alle Aspekte von IIS detailliert abbilden soll, sondern auch eine Strukturierung von Eigenschaften hinsichtlich ihrer Relevanz im Rahmen des Entwicklungsprozesses eines IIS anstrebt. Die hier vorgestellte Klassifikation konzentriert sich daher auf die Eigenschaften, die für die Wahl eines bestimmten

Typs von IIS relevant sind. Dies sind insbesondere die für den Nutzer sichtbare Funktionalität und Qualität bei der Informationssuche sowie die Merkmale der dem IIS zugrundeliegenden Daten und Datenquellen.

Andere Merkmale, etwa die Charakterisierung einer Materialisierung von Daten zur Beschleunigung der Anfragebearbeitung oder die Verfeinerung der Suchfunktionalität hinsichtlich der Anfragesprachen, die speziell bei mediatorbasierten IIS zum Einsatz kommen (vgl. etwa eine entsprechende Klassifikation in [Wa01]), würden Bestandteil einer Verfeinerung für eine spätere Entwicklungsphase sein. Auch die Beschreibung der hier exemplarisch betrachteten Typen – mediator-basierte und IR-basierte IIS – sowie die Anforderungen des EMN-IIS werden anschließend auf dieser Abstraktionsebene diskutiert.

3.1 Informationsintegrationssysteme im Allgemeinen

Da die Beschreibung des gesamten Merkmalsmodells für IIS den Rahmen dieses Papiers sprengen würde, geben wir im folgenden einen Überblick über das gesamte Modell und beschreiben dann detaillierter die Elemente, die sich auf die Funktionalität der Informationssuche beziehen.

Für ein Informationsintegrationssystem sind die folgenden Unterscheidungsmerkmale von besonderer Bedeutung und daher Bestandteil des Merkmalsmodells (vgl. Abbildung 2):

- Welche Akteure oder *Rollen* sind an dem System beteiligt und vereinen Akteure typischerweise mehrere Rollen in sich? Grob lassen sich hier Nutzer, Datenanbieter sowie Anbieter des Integrationsdienstes unterscheiden. In einem Peer Data Management System ist ein Nutzer dabei typischerweise auch selbst Datenanbieter.
- Welche *Prozesse* werden von dem IIS unterstützt? In jedem Falle werden dies Prozesse im Rahmen der Informationsermittlung sein. Darüber hinaus lassen sich jedoch auch Managementprozesse identifizieren, etwa zur Pflege unterstützender Thesauri oder zur Integration von Datenquellen.
- Welche Art von *Daten* werden von dem IIS angeboten? Wir unterscheiden hierbei zum einen, ob es sich um ein reines Metainformationssystem handelt (etwa ein Bibliothekssystem ohne direkte Zugriffsmöglichkeit auf die Publikationen) oder nicht. Zum anderen charakterisieren wir die Art der Daten näher: sind es strukturierte, semi-strukturierte oder unstrukturierte Daten und sind die Datenobjekte miteinander verknüpft oder nicht? Da die Charakterisierung des Strukturierungsgrads und der Repräsentation von Daten in gleicher Art und Weise auch bei den Daten der unterliegenden Quellen von Bedeutung sind, wurden diese Merkmale in einem separaten Merkmalsmodell beschrieben (vgl. Abbildung 3).
- Welche *Form der Interaktion* steht dem Nutzer für die Informationsermittlung zur Verfügung? Wir unterscheiden hierbei zunächst Systeme, die eine Navigationsmöglichkeit bieten von solchen, die eine Schnittstelle zur Suche anbieten. Bei der Suche sind insbe-

sondere Stichwortsuchen von Anfragen in einer Anfragesprache, wie SQL, zu unterscheiden. Letztere basieren stets auf einem Schema, das das Vokabular für die Suche definiert. Im Falle der Stichwortsuche muss dies nicht der Fall sein: Suchmaschinen etwa unterstützen in der Regel ausschließlich eine Suche mit freien Stichworten. Für Systeme, die eine Suchmöglichkeit bieten, sollten die Semantik der Suche sowie das zu erwartende Suchergebnis genauer charakterisiert werden. Wir werden auf diese Elemente später detaillierter eingehen.

- Wird ein *kontrolliertes Vokabular* für die Suche verwendet, kann weitergehend unterschieden werden, in welcher Form dies definiert ist. Die hier vorgenommene Unterscheidung lehnt sich an das in [Mc02] definierte Ontologie-Spektrum an.

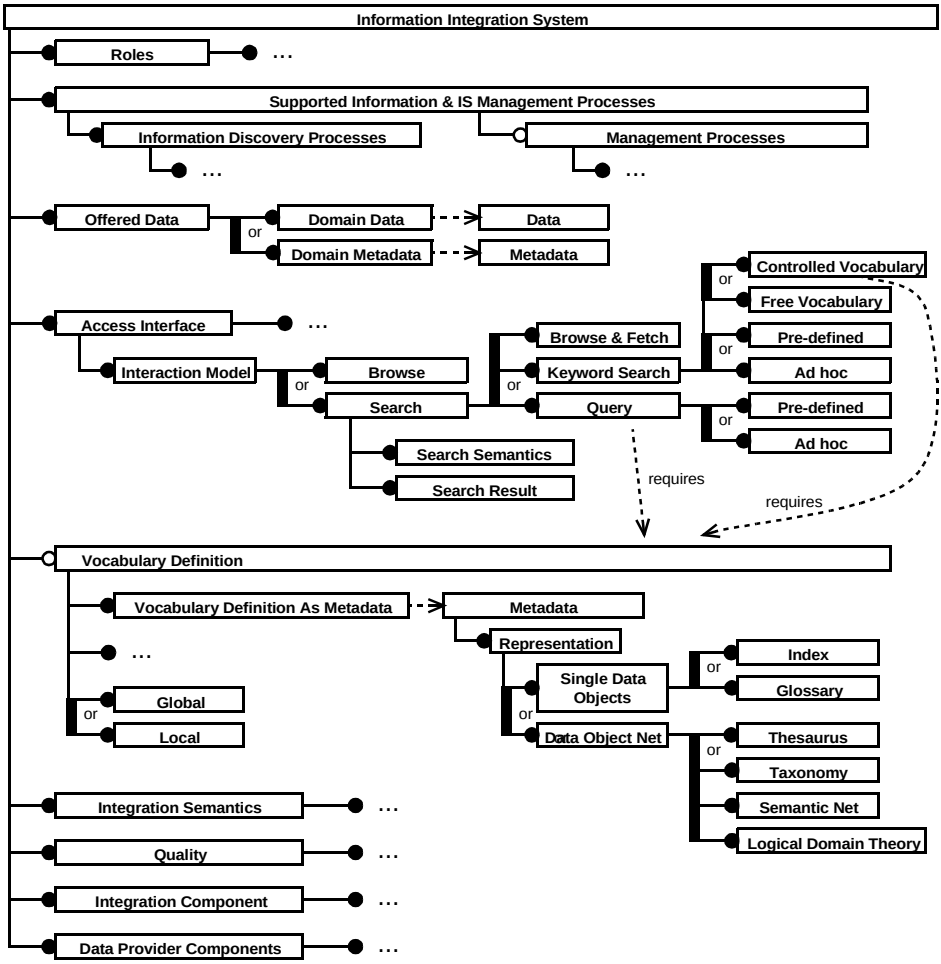

Abb. 2: Merkmalsmodell von IIS – Überblick

- Wie erfolgt die *Integration* von Daten verschiedener Quellen? Dabei wird zunächst eine einfache Sammlung von Daten (wie sie etwa bei Suchmaschinen erfolgt) und die Integration mit der Fusion und Verknüpfung von Datenobjekten (z.B. in mediatorbasierten IIS) unterschieden. Bei der Integration kann weitergehend charakterisiert werden, inwieweit Konflikte bei der Integration betrachtet und ggf. gelöst werden.

- Welche *Qualität* haben die vom IIS gelieferten Daten und das Integrationssystem selbst? Da die Qualität eine herausragende Bedeutung für die Bewertung des IIS insgesamt hat, wurde dieser Aspekt separat aufgenommen. Es werden hier die für den Nutzer relevanten Qualitätsaspekte beschrieben, wobei verschiedene existierende Klassifikationen herangezogen werden können. Wir haben uns zunächst auf die in [SMB05] diskutierten Qualitätsattribute gestützt.

- Welche Merkmale haben die *Integrationskomponente* sowie die *datenbereitstellenden Komponenten*? Hierbei spielen insbesondere die Arten der Schnittstelle der Datenquellen sowie die von ihnen gelieferten Daten eine Rolle.

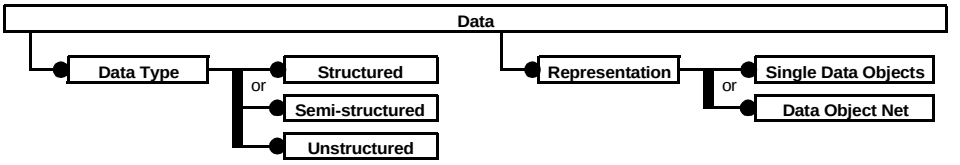

Abb. 3: Merkmalsmodell von IIS – Daten

Bei der detaillierteren Betrachtung der Suche sollte zunächst die Semantik der Suche charakterisiert werden, wobei folgende Varianten auftreten können (vgl. Abbildung 4):

- Manche Systeme führen eine *Modifikation der Anfrage* durch, um die Genauigkeit oder den Recall der Anfrage zu erhöhen. Auch eine Personalisierung durch Berücksichtigen von Kontextwissen über den Nutzer sind dabei denkbar.

- Der *Suchraum* kann sich stark unterscheiden: So decken Web-Suchmaschinen grundsätzlich nur einen unvollständigen Teil des Webs ab, wohingegen mediatorbasierten IIS potentiell alle Datenquellen berücksichtigen.

- Ein weiteres Merkmal ist die Art, wie der Vergleich von Datenbasis und Anfrage erfolgt: ist dabei eine exakte Übereinstimmung notwendig oder lediglich eine angemessene Nähe der Datenwerte? Diese Unterscheidung ist sowohl in syntaktischer wie auch semantischer Richtung sinnvoll. Information Retrieval Methoden auf der Basis des Vektorraummodells führen bspw. eine semantisch approximierte Suche durch.

- Das Ergebnis einer Suche kann weitergehend in der Ergebnismenge (vollständig, die top-k Ergebnisse oder unvollständig?) sowie in der Anordnung der einzelnen Ergebnisse unterschieden werden.

Für den Nutzer unmittelbar sichtbar ist das Suchergebnis selbst, wobei wir zum einen charakterisieren, welche Art von Daten er dabei erhält und zum anderen, in welcher Form die Ergebnismenge repräsentiert ist.

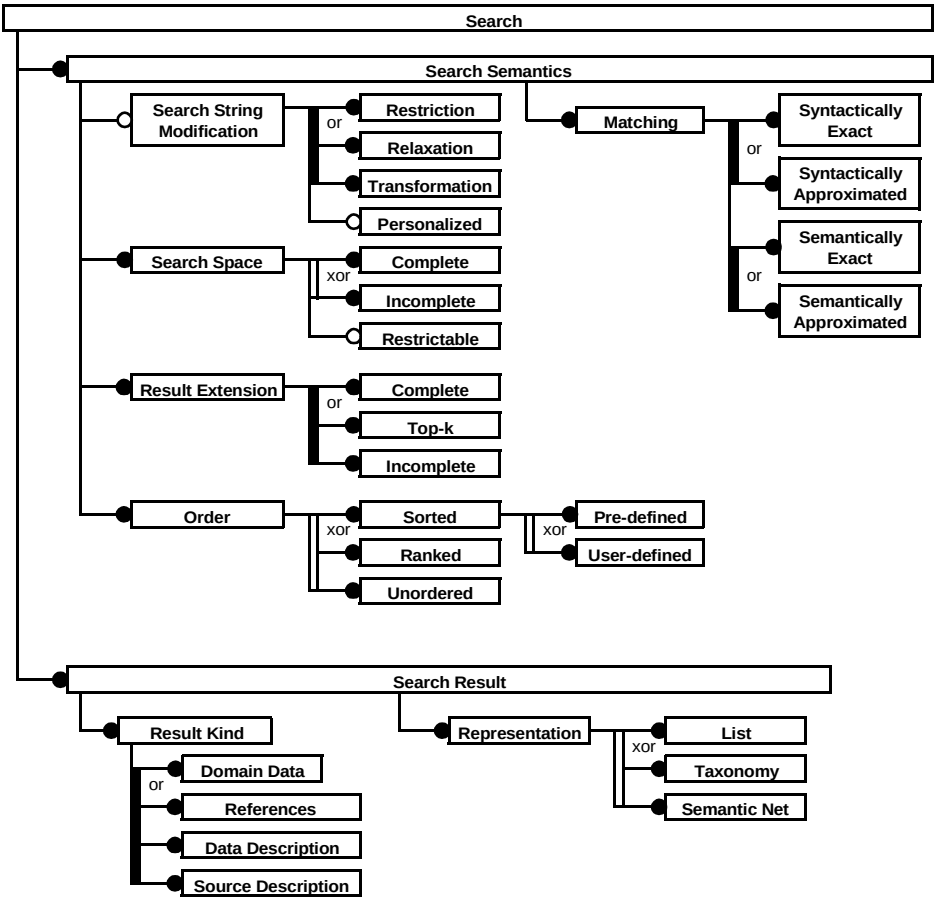

Abb. 4: Merkmalsmodell von IIS – Suche

3.2 Spezielle Typen von Informationsintegrationssystemen

Von den eingangs genannten Typen von IIS betrachten wir exemplarisch mediatorbasierte und Information Retrieval-basierte Systeme, da sie sich bzgl. ihrer Eigenschaften bei der Suchfunktionalität stark unterscheiden.

Ein mediatorbasiertes Informationssystem ([Wi97], [Kn01], [Bu02]) bietet einen integrierenden lesenden Zugriff auf eine dynamisch änderbare Anzahl heterogener Datenquellen. Die Integration ist für den Nutzer transparent – der Mediator übernimmt die Ermittlung benötigter Daten aus den Datenquellen und die quellübergreifende Integration. Dem Nut-

zer wird ein globales Schema sowie eine Anfragesprache zur Verfügung gestellt. Mediatorbasierte IIS verfolgen einen virtuellen Integrationsansatz, d.h. es wird erst bei der Anfragebearbeitung auf die Datenquellen zurückgegriffen. Zur Integration werden Wrapper eingesetzt, die die Datenquellen kapseln und für die Lösung technischer und logischer Heterogenität verantwortlich sind. Sie bieten dem Mediator in der Regel eine eingeschränkte Anfrageschnittstelle, die bei der Integration spezifiziert und bei der Anfragebearbeitung des Mediators ausgewertet wird.

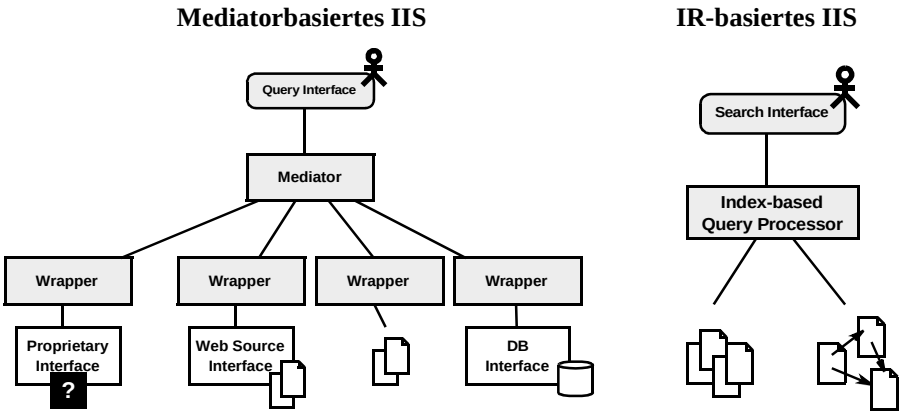

Abb. 5: Mediator- und IR-basierte IIS

Sind mediatorbasierte Systeme vor allem für strukturierte bzw. semistrukturierte Daten geeignet, addressieren IIS, die auf Techniken des Information Retrieval ([BR99]) zurückgreifen, vor allem unstrukturierte Datenquellen. Suchmaschinen im World Wide Web sind typische Vertreter dieses IIS-Typs ([Mi04]). Die dem zugrundeliegenden Textdokumente werden indiziert und dem Nutzer auf dieser Basis eine Stichwortsuche angeboten. Die Stichworte können dabei in der Regel frei gewählt werden. Manchmal wird dem Nutzer jedoch auch ein Vokabular, das zur Indizierung verwendet wurde, angeboten, etwa in Form eines Thesaurus oder einer Taxonomie. Die auf dem Index basierende Suche liefert (im Gegensatz zum mediatorbasierten IIS) approximierte Ergebnisse. [Ha05] stellt verschiedene Formen der Approximation vor, die in existierenden Bibliotheken zum Einsatz kommen. Das Ergebnis einer Suche in einem IR-basierten IIS besteht aufgrund der Approximation in der Regel aus einer mit einem Rankingverfahren sortierten Liste von Referenzen.

Abbildung 5 zeigt den typischen Aufbau mediatorbasierter und IR-basierter IIS im Vergleich. Eine präzise Beschreibung, etwa mit Hilfe der UML, bildet die Basis der Referenzarchitektur, die als Entwurfsgrundlage bei Wahl des jeweiligen Typs generiert werden würde. Für die Wahl ist die Definition entlang des Merkmalsmodells für IIS erforderlich. Im folgenden werden die charakteristischen Eigenschaften präzisiert. Es wird dazu die Bindung bei der jeweiligen Spezialisierung angegeben. Nicht aufgeführte Merkmale bleiben unverändert (’undecided’).

	Mediatorbasiertes IIS	IR-basiertes IIS
Offered Data	Data Type: removed: Unstructured	Data Type: removed: Structured
Access Interface – Interaction Model	bound: Search – Query – Ad hoc	bound: Search – Keyword Search – Ad hoc
Vocabulary Definition	bound bound: Global, removed: Local Representation: common: Data Object Net	optional bound: Global, removed: Local
Search Semantics		
Search String Modification	uncommon	optional
Search Space	bound: Complete uncommon: Restrictable	
Matching	common: • Semantically Exact	common: • Syntactically Approximated • Semantically Approximated
Result Extension	bound: Complete	common: Top-k
Order	removed: Ranked uncommon: Sorted – Pre-defined	bound: Ranked
Search Result	bound: • Result Kind – Domain Data • Representation – List	bound: Result Kind – References common: Representation – List

Tabelle 2: Definition mediator- und IR-basierter IIS

3.3 Anforderungsspezifikation für das 'European Migration Network'

Das Merkmalsmodell für IIS kann auch zur Spezifikation der Anforderungen für ein konkretes oder geplantes IIS, wie das EMN-IIS, herangezogen werden. Die variablen Elemente des Merkmalsmodells können – vergleichbar mit einem Fragenkatalog – gewählt oder ausgeschlossen werden. Offen gelassene variable Merkmale sind zu einem späteren Zeitpunkt zu betrachten. Wir gehen zunächst also immer von einer unvollständigen Anforderungsspezifikation aus.

Die folgende Tabelle zeigt die wesentlichen Anforderungen an das IIS des EMN. Dem Nutzer soll eine Stichwortsuche angeboten werden, wobei auch auf ein global vorhandenen Thesaurus zurückgegriffen werden kann. Neben der Stichwortsuche soll eine Navigationsmöglichkeit über Länder und Informationstyp (Kontakt, Publikation etc.) angeboten werden. Die Suche soll alle passenden Informationen in vergleichbarer Form ermitteln. Daher wird auf eine Integration (nicht nur Sammlung) der Informationsobjekt wert gelegt, bei der insb. mit der Multilingualität umgegangen werden muss. Die Daten (wie übrigens auch die Datenquellen) sind sehr heterogen. Es sind im EMN sowohl strukturierte Daten als auch unstrukturierte Dokumente vorhanden.

	EMN
Offered Data: Domain Data	Data Type: Structured, Semi-structured, Unstructured Representation: Data Object Net
Access Interface – Interaction Model	• Search – Browse & Fetch • Search – Keyword Search – Controlled Vocabulary, Free Vocabulary – Ad hoc
Vocabulary Definition	bound, Details: • Global • Representation: Data Object Net – Thesaurus
Search Semantics	removed: Search String Modification Search Space: Complete, removed: Restrictable Matching: Semantically Exact Result Extension: Complete Order: Sorted – Pre-defined (nach Land und Informationstyp)
Search Result	Result Kind: Domain Data, References Representation – List

Tabelle 3: Anforderungsspezifikation für das 'European Migration Network'

4 Merkmalsbasierte Wahl eines Integrationsansatzes

4.1 Vergleich von Merkmalsspezifikationen

Die Wahl eines Integrationsansatzes und der damit verbundenen Referenzarchitektur soll durch Vergleich einer Anforderungsspezifikation mit den Merkmalsspezifikationen verschiedener Typen von IIS unterstützt werden. Allgemein wird dabei eine Matching-Funktion

match (FM-Spec, FM-Conf): record (set (MatchResult), Similarity)

verwendet. Sie erhält als Parameter

- eine Spezialisierung **FM-Spec** eines Merkmalsmodells **FM** (im Falle von IIS eine Spezialisierung, die einen bestimmten Typ von IIS beschreibt) und
- eine Anforderungsspezifikation **FM-Conf**, d.h. eine ggf. noch unvollständige Konfiguration des gleichen o.g. Merkmalsmodells **FM** (im Falle von IIS etwa die Anforderungsspezifikation für das EMN-Informationssystems)

und liefert als Ergebnis

- eine Menge von Vergleichsergebnissen **MatchResult** sowie
- eine quantifizierte Aussage **Similarity** zur Ähnlichkeit mit einem Wert $\in$ [-1, 1]

Der Matchingalgorithmus besteht im wesentlichen aus zwei Teilen: Zum einen aus dem direkten Vergleich der Bindung eines Merkmals, aus dem auch ein detailliertes Vergleichsergebnis **MatchResult** resultiert, zum anderen aus der Traversierung des

gesamten zugrundeliegenden Merkmalsmodells mit der Aggregation der Einzelvergleiche zu einem zusammenfassenden Ähnlichkeitswert.

Vergleich der Bindung eines einzelnen Merkmals

Ein Vergleichsergebnis **MatchResult** beschreibt das Ergebnis des direkten Vergleichs eines einzelnen Merkmals und dient der Anzeige eines detaillierten Vergleichsergebnisses innerhalb eines Werkzeugs. Jedes **MatchResult** beinhaltet das Merkmal selbst, eine textuelle Beschreibung sowie eine Einordnung in definierte Ergebnisklassen und deren Kategorisierung als Information, Warnung oder Konflikt. Tabelle 4 zeigt die möglichen Vergleichsergebnisse mit der Benennung ihrer Ergebnisklasse. Ein '+' fällt dabei in die Kategorie einer Information, ein '–' in die einer Warnung und ein '– –' in die eines Konflikts.

Anforderungsspez. / Spezializierung	**bound**	**removed**	**undecided**
bound	++ (Match)	– – (Mismatch)	– (Potential Mismatch)
common	+ (Common)	– (Uncommon)	– (Potential Uncommon)
undecided in changed FeatureGroup	– (Uncommon)	+ (Common)	– (Potential Uncommon)
undecided	+ (Possible)	+ (Possible)	+ (Possible)
uncommon	– (Uncommon)	+ (Common)	– (Potential Uncommon)
removed	– – (Mismatch)	++ (Match)	– (Potential Mismatch)

Tabelle 4: Vergleich von Merkmalen aus Anforderungsspezifikation und Spezialisierung

Zu beachten ist die unterschiedliche Bewertung des 'undecided' bei gruppierten Merkmalen einer Spezialisierung, d.h. in der Definition eines Typs von IIS: Wurde kein Merkmal der Gruppe bei der Spezialisierung als häufig oder immer auftretendes Merkmal ausgewählt, so gehen wir davon aus, dass alle Alternativen gleichermaßen unterstützt werden. So unterscheidet ein MBIS bspw. nicht, ob in dem System Daten oder Metadaten als 'Domain Data' verwendet werden. Wurde hingegen eines der Merkmale bevorzugt (bound oder common), so bedeutet dies in der Regel, dass die anderen noch möglichen Merkmale zwar denkbar sind, aber nicht gesondert unterstützt werden. Die Wahl ist daher eher ungewöhnlich (Ergebnisklasse 'Uncommon'). So ist es in einem IR-basierten IIS zwar möglich auch eine Navigationsmöglichkeit zum Datenbestand anzubieten ('Browse & Fetch'), aber nicht durch die für ein IR-basiertes IIS charakteristischen Konzepte.

Für die definierten Ergebnisklassen ist auch eine Quantifizierung als Wert zwischen -1 und +1 definiert (vgl. Tabelle 5), die später die Basis für die Nähe (**Similarity**) bildet und die

Passgenauigkeit des Integrationsansatzes quantifiziert. Potenzielle Konflikte, die sich aufgrund einer noch unvollständigen Anforderungsspezifikation ergeben, werden dabei von der Betrachtung ausgeschlossen.

++ (Match)	+1	– (Uncommon)	-0.5	– (Potential Uncommon)	—
+ (Common)	+0.5	– – (Mismatch)	-1	– (Potential Mismatch)	—
+ (Possible)	+0.5				

Tabelle 5: Quantifizierter Vergleich von Merkmalen

Vergleich des gesamten Merkmalsmodells

Der direkte Vergleich zweier Bindungen eines Merkmals bildet die Basis des Matchingalgorithmus. Der Gesamtalgorithmus bestimmt, welche Merkmale verglichen werden und welches Maß für quantifizierte Angaben sinnvoll ist. Das Ergebnis über die Ähnlichkeit von Anforderungsspezifikation (einer ggf. unvollständigen Konfiguration) und der Definition eines IIS-Typs (definiert als Spezialisierung im Sinne der Merkmalsmodellierung) beinhaltet eine Sammlung detaillierter **MatchResults** sowie eine quantifizierte Angabe der Ähnlichkeit. Dieser Wert liegt wiederum zwischen -1 und +1. Ein negativer Wert bedeutet, dass der Integrationsansatz eher ungeeignet, ein positiver, dass er geeignet ist.

Der Matchingalgorithmus beinhaltet grob die im folgenden beschriebenen Schritte (vgl. die Beschreibung in Pseudocode in Abbildung 6, die auf der Formalisierung von Merkmalsmodellen in Kapitel 2 beruht):

1. Startpunkt ist die Wurzel der Anforderungsspezifikation, die im folgenden traversiert wird. Die Orientierung an der Anforderungsspezifikation ermöglicht den Umgang mit unterschiedlich weit fortgeschrittenen Konfigurationen.
2. Für jeden Knoten wird die Ähnlichkeit des Teilbaumes mit diesem Knoten als Wurzel ermittelt: zum einen die Menge von detaillierten **MatchResults**, zum anderen die Quantifizierung als ein Wert zwischen +1 und -1.

   Der quantitativen Berechnung liegen folgende Annahmen zugrunde:

   a) Variable Eigenschaften, die in der Anforderungsspezifikation nicht betrachtet wurden (’*undecided*’) werden nicht berücksichtigt.
   b) Die betrachteten Untermerkmale eines Merkmals gehen *gleichberechtigt* in die Berechnung ein.
   c) Es wird beim Vergleich nur dann weiter ins Detail gegangen, d.h. die Untermerkmalsbäume werden nur dann weiter betrachtet, wenn der *Vergleich* des Merkmals selbst *positiv* verlaufen ist. Ein negatives Ergebnis bedeutet, dass die Spezialisierung dieses Merkmal als eher untypisch definiert hat. Untermerkmale werden daher dort nicht weiter spezifiziert, so dass ein Vergleich nicht sinnvoll wäre.

Wir berechnen daher die Ähnlichkeit des (Teil)baums mit dem Wurzelmerkmal **f** als Produkt des Einzelvergleichs von **f** und der Summe der anteiligen Ähnlichkeit der relevanten Untermerkmalsbäume von **f.**

```
match (FM-Spec, FM-Conf):
  return match(FM-Conf.Concept, FM-Spec, FM-Conf)

match (f, FM-Spec, FM-Conf):
  // Vergleich des betrachteten Knoten: qualitativ sowie quantitativ
  nodeMatch := nodeMatch(f, FM-Spec, FM-Conf);
  nodeSimilarity := nodeSimilarity(f, FM-Spec, FM-Conf);

  // weitere Untersuchung sinnvoll? (Punkt c)
  if (nodeSimilarity < 0)
     return (nodeMatch, nodeSimilarity);
  else
     // Initialisierung für beide Ergebniselemente
     subMatches := { };
     sumSubSimilarity := 0;

     // detaillierte Vergleichsergebnisse für 'undecided' Merkmale sammeln
     undecidedSubfeatures := { sub ∈ Features | subfeatureOf(sub) = f ∧
                                   incompleteBinding(sub) = undecided  }
     for each sub ∈ undecidedSubfeatures do
        subMatches := subMatches ∪ nodeMatch(sub, FM-Spec, FM-Conf);

     // Traversierung der relevanten Untermerkmale (Punkt a)
     relevantSubfeatures := { sub ∈ Features | subfeatureOf(sub) = f ∧
                                   incompleteBinding(sub) ≠ undecided  }
     // Gleichberechtigung der Unterbäume (Punkt b)
     weightSubfeature := 1 /  | relevantSubfeatures |
     for each sub ∈ relevantSubfeatures do
        // Vergleich des Unterbaums
        subMatchRecord := match(sub, FM-Spec, FM-Conf);
        subMatches := subMatches » subMatchRecord [ 1 ];
        sumSubSimilarity := sumSubSimilarity +
                            (weightSubfeature * subMatchRecord [ 2 ]);
     // Gesamtergebnis ermitteln
     return (nodeMatch » subMatches, nodeSimilarity * sumSubSimilarity);
  end if;

nodeMatch (f, FM-Spec, FM-Conf)
  gibt das detaillierte Vergleichsergebnis zurück, das sich aus dem
  direkten Vergleich von f entsprechend der Tabelle 4 ergibt.

nodeSimilarity (f, FM-Spec, FM-Conf)
  gibt den Ähnlichkeitswert zwischen -1 und 1 zurück, der sich aus dem
  direkten Vergleich von f entsprechend der Tabelle 5 ergibt.
```

Abb. 6: Algorithmus zum Vergleich von Anforderungsspezifikation und Spezialisierung

Bei der Ermittlung der detaillierten Vergleichsergebnisse **MatchResult** gelten die gleichen Abbruchsbedingungen bei der Traversierung. Allerdings werden die Ergebnisse von nicht betrachteten Merkmalen in der Anforderungsspezifikation ('undecided') noch mit aufgenommen, um potentielle Konflikte zu identifizieren.

4.2 Vergleich am Beispiel des 'European Migration Networks'

Es soll der beschriebene Algorithmus zum Vergleich einer Anforderungsspezifikation mit einem definierten Typ eines IIS anhand des EMN verdeutlicht werden. Exemplarisch wird dazu der Ausschnitt des Merkmalsmodells für IIS, das sich auf das Suchergebnis bezieht, herangezogen und die Anforderungsspezifikation für das EMN mit der Definition IR-basierter IIS verglichen. Abbildung 7 zeigt den entsprechenden Ausschnitt. Annotiert wurden die Werte für die **nodeSimilarity** an den Knoten sowie die Gewichtung der jeweiligen Unterbäume (unter der Annahme ihrer Gleichberechtigung). Daraus ergibt sich eine Ähnlichkeit für den Unterbaum 'Search Result' von 7/16, der anzeigt, dass ein IR-basiertes System bzgl. dieses Merkmals geeignet ist (positiver Wert).

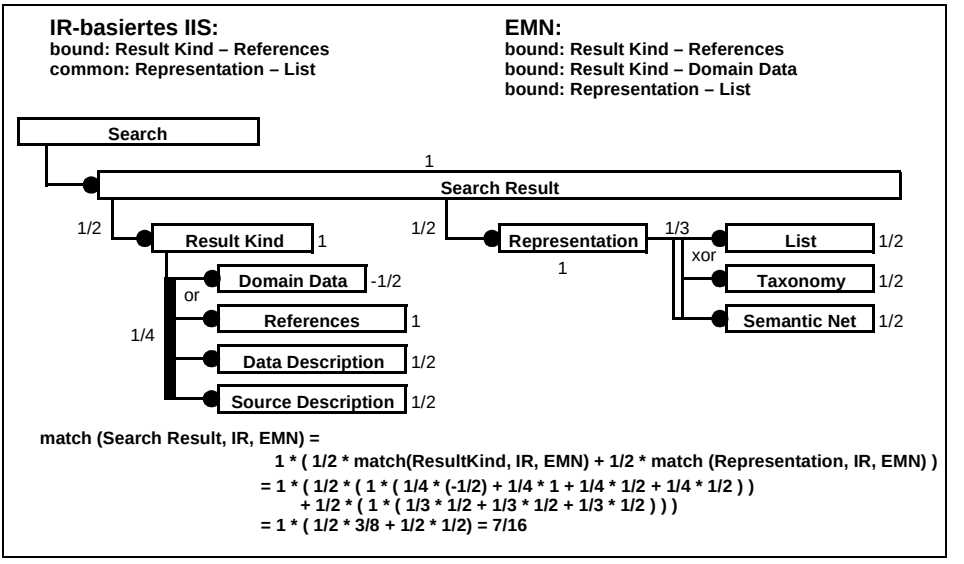

Abb. 7: Beispiel eines Vergleichs

Bei dem Vergleich der Ergebnisse des Matching der EMN-Anforderungsspezifikation mit mediatorbasierten IIS auf der einen und IR-basierten IIS auf der anderen Seite zeigt sich insgesamt eine geringfügig bessere Passgenauigkeit des mediatorbasierten Ansatzes. Ausschlaggebend sind dabei die Anforderungen des EMN-Systems hinsichtlich der Semantik der Suche: die geforderte semantisch exakte Suche aller passenden Ergebnisse trifft die Intention mediatorbasierter Systeme.

Der Algorithmus zeigt durch die detaillierten Vergleichsergebnisse jedoch auch die Unzulänglichkeit eines MBIS: die gewünschte Form der Interaktion des Nutzers mit dem System – die Navigationsmöglichkeit sowie eine Stichwortsuche – passt sehr viel besser zu einem IR-basierten Ansatz. Hier ergibt sich ein 'Match' beim Vergleich mit dem IR-Typ, dagegen ein 'Uncommon' bei dem MBIS-Typ.

Der tatsächlich vorgenommene Entwurf des EMN-IIS folgt einer materialisierten Form der Mediation, da diese insbesondere auch den Anforderungen bzgl. der gewünschten engen Kopplung (der Integrationssemantik) entgegen kommt. Eine der größten Herausforderungen war bei der Entwicklung die Realisierung einer Stichwortsuche auf dieser Basis – wie es die hier vorgenommene Analyse auch bestätigt.

5 Diskussion

Ausgangspunkt für die hier vorgestellte Arbeit ist die Beobachtung, dass der Entwurf eines Informationsintegrationssystems (IIS) weitgehend manuell erfolgt, obwohl eine Vielzahl von Diskussionen bzgl. der 'Trade-Offs' sowie Beschreibungen der spezifischen Merkmale und Architektur bestimmter Typen von IIS existiert. Wir haben hier gezeigt, wie die Merkmalsmodellierung aus der generativen Softwareentwicklung genutzt werden kann, um diese Lücke zu schließen. Es wird mit dem hier gezeigten Ansatz folgendes erreicht:

- Das Merkmalsmodell für IIS schafft mit der Präzisierung der Eigenschaften von IIS eine geeignete Basis sowohl für den Vergleich und die Dokumentation konkreter Systeme als auch für die präzise Abgrenzung verschiedener Typen von IIS. Es bietet eine geeignete Grundlage für eine strukturierte Anforderungsspezifikation und zeigt bereits wichtige Abhängigkeiten zwischen Eigenschaften von IIS auf.
- Die Merkmalsmodellierung bietet den formalen Rahmen, um die Entwicklung eines IIS, etwa im Rahmen eines Werkzeuges, konstruktiv zu unterstützen. In diesem Sinne wurde hier gezeigt, wie durch den Vergleich von Anforderungsspezifikation und Definition eines IIS-Typs die Wahl eines Integrationsansatzes und die Identifizierung möglicher Problemstellungen unterstützt werden kann. Insbesondere die Menge der identifizierten Konflikte des Vergleichs ermöglicht eine detaillierte Analyse und Planung des folgenden Entwurfs.

  Die quantifizierte Aussage zur Ähnlichkeit kann einen ersten Hinweis auf die Passgenauigkeit eines IIS-Typs geben, sollte allerdings nicht überbewertet werden. So sind die hier vorgenommenen Annahmen, etwa über die Gleichberechtigung aller Merkmale auf allen Detaillierungsebenen, in weiteren realen Beispielen zu überprüfen und ggf. anzupassen.

Die hier begonnene Richtung zu einer durchgängigen Methode für die modellgetriebene Entwicklung und Evolution von IIS soll in folgenden Arbeiten weiterverfolgt werden. Insbesondere sollen dazu

- weitere Typen von IIS, etwa Portale und Peer Data Management Systeme, definiert werden und für eine weitergehende Analyse quantifizierter Aussagen genutzt werden;
- das Merkmalsmodell zur Unterstützung späterer Entwicklungsphasen verfeinert werden, etwa für die Wahl geeigneter Datenmodelle und Anfragesprachen, aber auch existierender Algorithmen zur Realisierung der Merkmale;
- weitere Algorithmen zur Unterstützung der Entwicklung, etwa zur Generierung eines Rahmens für die Architektur eines IIS, definiert und im Rahmen eines Werkzeugs angeboten werden.

Literaturverzeichnis

[AC04] M. Antkiewicz, K. Czarnecki, *FeaturePlugin: feature modeling plug-in for Eclipse*, in: Proceedings of the 2004 OOPSLA Workshop on Eclipse Technology Exchange, ACM Press, pp. 67-72, 2004.

[Ar01] A. Arasu, J. Cho, H Garcia-Molina, A. Paepcke, S. Raghavan, *Searching the Web*, ACM Transactions on Internet Technology, Vol. 1, No. 1, pp. 2-43, 2001.

[BP04] M. Bouzeghoub, V. Peralta, A *Framework for Analysis of Data Freshness*, in: Proceedings of the 2004 International Workshop on Information Quality in Information Systems (IQIS '04), pp. 59-67, ACM Press, 2004.

[BR99] R.A. Baeza-Yates, B.A. Ribeiro-Neto, *Modern Information Retrieval*, Addison-Wesley, 1999.

[Bu02] S. Busse, *Modellkorrespondenzen für die kontinuierliche Entwicklung mediatorbasierter Informationssysteme*, Dissertation, TU Berlin, Logos Verlag, 2002.

[Bu05] S. Busse, G. Goldbeck, K. Grunwald, T. Kabisch, R.-D. Kutsche, M. Neiling, and M. Stübin, *The European Migration Network – Challenges in Federated Information Systems Development for the Federation of European States*, in: Proc. of the 8th Int. Conf. Informatics 2005, pp. 37-46, 2005.

[CE00] K. Czarnecki, U.W. Eisenecker, *Generative Programming – Methods, Tools, and Applications*, Addison-Wesley, 2000.

[CN01] P. Clements, L. Northrop, *Software Product Lines: Practices and Patterns*, Kluwer, 2001.

[Cz05] K. Czarnecki, S. Helsen, U.W. Eisenecker, *Formalizing Cardinality-based Feature Models and their Specialization*, Software Process: Improvement and Practice, Vol. 10, No. 1, pp. 7-29, 2005.

[Ha03] A. Halevy, O. Etzioni, A. Doan, Z. Ives, J. Madhavan, L. McDowell, I. Tatarinov, *Crossing the Structure Chasm*, Proc. 1st Biennial Conf. on Innovative Data Systems Research (CIDR 2003), 2003.

[Ha05] V. Hassler, *Open Source Libraries for Information Retrieval*, IEEE Software, Vol. 22, No. 5, pp. 78-82, IEEE, 2005.

[Hu97] R. Hull, *Managing Semantic Heterogeneity in Databases: A Theoretical Perspective*, Proc. ACM Symp. on Principles of Databases Systems PODS'97, 1997.

[In96] W. Inmon, *Building the Data Warehouse*, John Wiley & Sons Inc., 1996.

[Ka90] K.C. Kang, S.G. Cohen, J.A. Hess, W.E. Novak, A.S. Peterson, *Feature-Oriented Domain Analysis (FODA) Feasibility Study*, Technical Report CMU/SEI-90-TR-21, Software Engineering Institut, Carnegie Mellon University, Nov. 1990.

[Kn01] C.A. Knoblock, S. Minton, J.L. Ambite, N. Ashish, J. Muslea, A.G. Philpot, S. Tejada, *The Ariadne Approach to Web-based Information Integration*, Int. Journal of Cooperative Information Systems (IJCIS), Vol. 10, No. 1-2, pp. 145-169, 2001.

[LC03] Z. Lacroix, T. Critchlow (eds.), *Bioinformatics – Managing Scientific Data*, Morgan Kaufmann, 2003.

[Ma02] A. Maedche, S. Staab, R. Studer, Y. Sure, R. Volz, *SEAL – Tying Up Information Integration and Web Site Management by Ontologies*, IEEE Data Engineering Bulletin, Vol. 25, No. 1, pp. 10–17, 2002.

[Mc02] D.L. McGuinnes, Ontologies Come of Age, in: D. Fensel, J. Hendler, H. Liebermann, W. Wahlster (eds.), *Spinning the Semantic Web: Bringing the World Wide Web to its Full Potential*, MIT Press, 2002.

[Mi04] M. Michalowski, J.L. Ambite, C.A. Knoblock, S. Minton, S. Thakkar, R. Tuchinda, *Retrieving and Semantically Integrating Heterogeneous Data from the Web*, IEEE Intelligent Systems., Vol. 19, No. 3, pp. 72-79, May/June 2004.

[NFL04] F. Naumann, J.-C. Freytag, U. Leser, *Completeness of integrated information sources*, Information Systems, Vol. 29, No. 7, pp. 583-615, 2004.

[PBC00] N.W. Paton, C.A. Goble, S. Bechhofer, *Knowledge based information integration systems*, Information and Software Technology, Vol. 42, No. 5, pp. 299-312, Elsevier Science Publishers, 2000.

[SMB05] M. Scannapieco, P. Missier, C. Batini, *Data Quality at a Glance*, Datenbank Spektrum, Heft 14/2005, pp. 6-14, Aug. 2005.

[Wa01] H. Wache, T. Vögele, U. Visser, H. Stuckenschmidt, G. Schuster, H. Neumann, S. Hübner, *Ontology-Based Integration of Information – A Survey of Existing Approaches*, in: H. Stuckenschmidt (ed.), IJCAI-01 Workshop 'Ontologies and Information Sharing', pp. 108-117, 2001.

[Wi97] G. Wiederhold, *Mediators in the Architecture of Future Information Systems*, in: M.N. Huhns, M.P. Singh (eds.), Readings in Agents, pp. 185-196, Morgan Kaufmann, 1997.