der Bundeswehr
Universität München

Visualisierung von Fusionsmodellen

Bachelor Thesis von
Stephan Tzschoppe
1006374

UniBwM – IB 16/2009

Aufgabenstellung:
Prof. Dr. Stefan Pickl

Betreuung:
Dipl.-Inf. Marco Schuler

Institut für Theoretische Informatik,
Mathematik und Operations Research
Fakultät für Informatik
Universität der Bundeswehr München

Neubiberg
18.12.2009

Eidesstattliche Erklärung

Hiermit versichere ich, dass ich die vorliegende Arbeit selbständig angefertigt habe. Die aus fremden Quellen direkt oder indirekt übernommenen Gedanken und Zitate sind als solche kenntlich gemacht.

Es wurden keine anderen, als die in der Arbeit angegebenen Quellen und Hilfsmittel benutzt. Die Arbeit wurde weder einer anderen Prüfungsbehörde vorgelegt, noch veröffentlicht.

Neubiberg, 18. Dezember 2009 _____

Unterschrift

Zusammenfassung

Zusammenfassung der Bachelorarbeit

Inhaltsverzeichnis

1 Einleitung

1.1 Motivation

Wir leben in einer Zeit stetig voranschreitender Technologisierung. Dies äußert sich für jeden sichtbar auf vielerlei Art und Weise. Massenspeicher mit vor Jahren noch unvorstellbaren Kapazitäten, stetig wachsende Prozessorleistung und massiver Fortschritt in der Datenübertragung sind hier nur als Beispiele zu nennen. Solche Entwicklungen sind es, die das Sammeln, Verarbeiten und Speichern von riesigen Datenmengen erst ermöglichen. Diesen Fortschritt gilt es zu nutzen und auf mögliche Anwendungsfelder auszuweiten.

Betrachtet man ein Schlachtfeld, sei es im Rahmen einer Übung, einer kriegerischen Auseinandersetzung oder eines Konflikts, so werden auch hier Informationen gesammelt und ausgewertet. Man stelle sich folgende Situation vor: Drei eigene Panzer bewegen sich durch das Gelände. Plötzlich klären sie zwei feindliche Fahrzeuge auf. Jeder einzelne eigene Panzer setzt eine Meldung ab und beschreibt, was er sieht. Dies führt zu sechs Meldungen. Der S2[1]-Offizier muss nun aus diesen Meldungen ein Lagebild erstellen. Dabei gilt es aus der vorhandenen (teilweise redundanten) Information die zwei statt sechs feindliche Fahrzeuge zu erkennen.

Für das geschilderte Beispiel scheint es nicht notwendig, diese Informationsauswertung zu automatisieren. In der Realität hingegen [2] erfordert es viel Zeit, diese Arbeit zu erledigen. Und genau dies ist ein großes Problem, denn je älter die Lageinformation ist, umso weniger aussagekräftig ist sie. Entscheidungen, die darauf basierend getroffen werden, können daraufhin falsch oder unverhältnismäßig sein. Um diesen Missstand zu

[1] Der S2-Offizier ist verantwortlich für die Militärische Sicherheit, Militärisches Nachrichtenwesen mit Aufklärung und Zielfindung, elektronisch Kampfführung und eben die Wehrlage

[2] Ein Panzerbataillon der Bundeswehr umfasst zum Beispiel ungefähr 40 Kampfpanzer

beseitigen, gilt es, den S2-Offizier bei seiner Arbeit technisch zu unterstützen.

1.2 Ziel der Arbeit

Eine technische Unterstützung kann in unterschiedlichen Abstufungen geschehen. So können die separaten Meldungen zu einer großen Meldung zusammengefasst werden. Dies ist aber nicht sonderlich hilfreich, wenn zum Beispiel aus vielen einzelnen Datensätzen einfach eine zusammenhängende Datensammlung erstellt wird. Denn wenn diese in einem textuellen Format vorliegt, ist sie von einem Menschen nur schwer zu verstehen. Weiterhin können die auftretenden Redundanzen nicht einfach erfasst, geschweige denn überhaupt genutzt werden.

Aus diesem Grund möchte ich mich in dieser Arbeit mit Möglichkeiten auseinandersetzen, eine menschenlesbare Sicht auf multisensorische Daten zu erstellen. Zum Einen sollen Visualisierungsmöglichkeiten aufgezeigt werden. Weiterhin soll bei unterschiedlichen Ansätzen, eine Bewertung dieser vorgenommen werden.

Ergebnis dieser Betrachtungen wird ein Framework zur Darstellung multisensorischer Daten. Dieses verwende ich dann prototypisch dazu, die eingangs erwähnte Problemstellung des S2-Offiziers zu bearbeiten und ihm eine, für seine Lageerstellung hilfreiche, Sicht auf die eingehenden Meldungen zu geben.

Die Erweiterbarkeit des Frameworks soll durch dessen Verwendung zur Darstellung von NDP [ABK] Daten gezeigt werden.

1.3 Aufbau der Arbeit

2 Theoretische Grundlagen

2.1 Multisensorische Daten

2.2 Fusion

2.3 Aggregation

2.4 Visualisierungsmöglichkeiten

3 Design und Implementierung eines Visualisierungsprototypen

In diesem Kapitel werden der Entwurf und die Implementierung eines Visualisierungsframeworks beschrieben. Dazu werden die einzelnen Komponenten des Frameworks beschrieben und ihr Zusammenspiel erläutert. Basierend auf dem Wissen über die Bestandteile können die Möglichkeiten der Erweiterbarkeit und der Individualisierbarkeit skizziert werde. Die Grundlage hierfür stellt zum einen eine prototypische Implementierung einer Visualisierung einer militärischen Lage und des weiteren der Versuch, Daten des NDP [ABK] dreidimensional darzustellen.

Abschließend werden interessante Aspekte der Implementierung aufgezeigt, die neben den umgesetzten Implementierungsentscheidungen auch andere Wege aufzeigen sollen.

Dem Leser, der sich vorrangig für den entstandenen Prototyp interessiert, dem sei der Abschnitt 3.2 empfohlen. Die Details des Frameworks sollten für das Verständnis des Funktionsumfangs und die Bedienung keine Rolle spielen, können aber bei Bedarf nachgeschlagen werden.

3.1 Design eines Visualisierungsframeworks

Das Ergebnis dieser Arbeit soll nicht nur eine potentielle Visualisierungs-Umgebung sein, sonder ein Framework, das zum einen die Möglichkeit bietet, mit geringem Aufwand Daten anzeigen zu können und auf der anderen Seite aber umfangreiche Erweiterungsmöglichkeiten zulässt. Wie das Framework konkret entworfen und umgesetzt wurde, soll im Folgenden dargestellt werden.

3.1.1 Eingabedaten und Datenmodell

Grundlage für die weitere Arbeit sollen Eingabedaten sein, die bestimmte Voraussetzungen erfüllen. Diese gestellten Bedingungen werden im Folgenden kurz beschrieben. Darauf basierend wird das Datenmodell erläutert, in das die Quelldaten überführt werden sollen.

Eingabedaten

Um Daten dreidimensional visualisieren zu können, müssen diese bestimmte Voraussetzungen erfüllen. Diese sollen hier kurz aufgezeigt werden.

Identifizierbarkeit Unabhängig von der Art der Visualisierung ist es unabdingbar, dass jedes einzelne Datum identifizierbar ist. Aus diesem Grund sollte in den Quelldaten bereits eine eindeutige Benennung vorliegen. Sollte das nicht der Fall sein, muss dieser Misstand beim Importieren der Daten spätestens behoben werden. Probleme, die sich ergeben können, wenn diese Bedingung nicht beachtet wird, sind zum einen, dass keine aussagekräftigen Berechnungen auf den importierten Daten durchgeführt werden können. Auch ist eine Markierung eines gezielten Datensatzes unmöglich.

Lokalisierbarkeit Aus dem Ziel, die Eingabedaten im dreidimensionalen, kartesischen Raum darzustellen ergibt sich eine ganz logische Voraussetzung: Es sollte mindestens für jede der drei Dimensionen eine Eigenschaft der Daten existieren, die eine Positionierbarkeit möglich macht. Zwar ist es genauso möglich, die Daten auf einer Linie anzuordnen und somit nur eine Positionierungseigenschaft vorauszusetzen oder sich analog auf den zweidimensionalen Raum zu beschränken. Der daraus resultierende Informationsverlust muss aber in Kauf genommen werden. Die Voraussetzungen an den Typ der für die Lokalisierung herangezogenen Eigenschaften sind nicht sehr streng. Zwar ist es zielführend, wenn es sich hier um kontinuierliche oder zumindest diskret ganzzahlige Größen handelt. Ist dies nicht der Fall, so müssen diese lediglich zur Berechnung der Anzeigekoordinaten mit einer geeigneten Abbildungsvorschrift umgerechnet werden.

mittelbar geeignet	*unmittelbar geeignet*
IP [ABK] Adressen	Entfernungsangaben
MAC [ABK] Adressen	Ortsangaben (Länge/Breite)
Zeichenketten allgemein	Zeitangaben

Tabelle 3.1: Beispiele für unmittelbar und mittelbar geeignete Größen

Weitere Eigenschaften Die Darstellung von Daten im dreidimensionalen Raum, lediglich basierend auf der Position und dem Namen, bietet noch keine hohe Anschaulichkeit. Um die in [Verweis, Visualisierungsmöglichkeiten] dargestellten Visualisierungen auch verwenden zu können, braucht es weitere Eigenschaften, die in optische Information umgewandelt werden kann.

Die Art der Daten die für solche zusätzliche Veranschaulichung verwendet kann, muss nicht genau spezifiziert werden. Trotzdem gibt es Typen, die sich für bestimmte Visualisierungsmöglichkeiten besser eignen. Als einfaches Bespiel seien hier Aufzählungsdatentypen genannt, die sich gerade zu anbieten, in Objekte in unterschiedlichen Farben darzustellen. Dabei können auch unterschiedliche Enumerationen gleichzeitig visualisiert werden, indem pro Eigenschaft eine Farbe verwendet wird, die je nach Wert der Eigenschaft in unterschiedlichen Helligkeiten dargestellt wird. Der Kreativität sind hier keine Grenzen gesetzt.

Datenmodell

Das im Rahmen dieser Arbeit entwickelte Datenmodell setzte den Grundstock für das später entworfene Framework und hat zum Ziel, durch generische Gestaltung in der Lage zu sein, unterschiedlichste Quelldaten ohne Anpassung speichern zu können.

Umgesetzt wurden diese Anforderungen durch die Trennung in eine Datensammlung und deren Daten auf der einen, und Eigenschaften auf der anderen Seite.

Datensammlung mit Daten Wie in 3.1 dargestellt, besteht das wesentliche Datenmodell aus zwei Klassen.

DataSet Die Klasse DataSet kapselt die einzelnen Datensätze in Form einer Liste. Diese Klasse ist die zentrale Datenstruktur, die für alle weiteren Prozesse die notwendi-

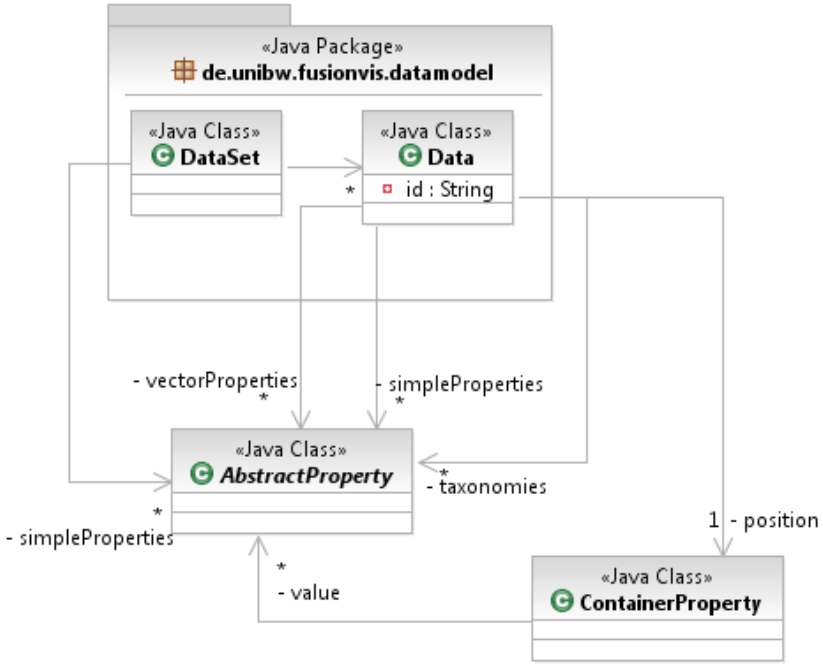

Abbildung 3.1: Übersicht des Datenmodells

gen Informationen bereithält. Sie besteht im Wesentlichen aus zwei Bestandteilen. Das erste ist die angesprochene Liste der gespeicherten Daten. Weiterhin kann sie Eigenschaften erfassen, die nicht einem bestimmten Datum zu Eigen sind, sondern global für die gesamte Datensammlung gelten. Das benutzen dieser Eigenschaften ist aber fakultativ um eine hohe Flexibilität zu gewährleisten. Die Typisierung der Eigenschaften wird weiter unten dargestellt.

Data Die Klasse Data Kapselt ein einzelnes Datum. Wie bereits in 3.1.1 dargelegt, sind die notwendigen Bestandteile eines Datensatzes ein Bezeichner, der innerhalb einer Datensammlung eindeutig sein muss, sowie eine Positionsangabe. Diese ist mithilfe einer zusammengesetzten Eigenschaft festgehalten. Das Eigenschaftssystem wird weiter unten noch detailliert beschrieben.

Zusätzlich zu diesen beiden obligatorischen Angaben erfasst die Data-Klasse weiterhin drei Listen von Eigenschaften.

- Einfache Eigenschaften
- Zusammengesetzte Eigenschaften

- Taxonomien

Die einfachen Eigenschaften sind in der Lage textuelle und Numerische Merkmale eines Datensatzes zu erfassen. Sie sind Grundlage die spätere Visualisierung. Die Zusammengesetzten Eigenschaften ermöglichen das Ablegen von vektoriellen Größen, wie zum Beispiel einer Blickrichtung, Geschwindigkeiten, oder Beschleunigungen usw. Auch ist es möglich mit ihrer Hilfe baumartig strukturierte Eigenschaften zu erfassen. Die Taxonomien umfassen eine Liste möglicher Klassifikationen, die einem Datensatz zu Eigen sein können. Auch ihre Angabe ist nicht zwingend erforderlich.

Der Aufbau einer Data-Klasse kann in 3.1 nachvollzogen werden.

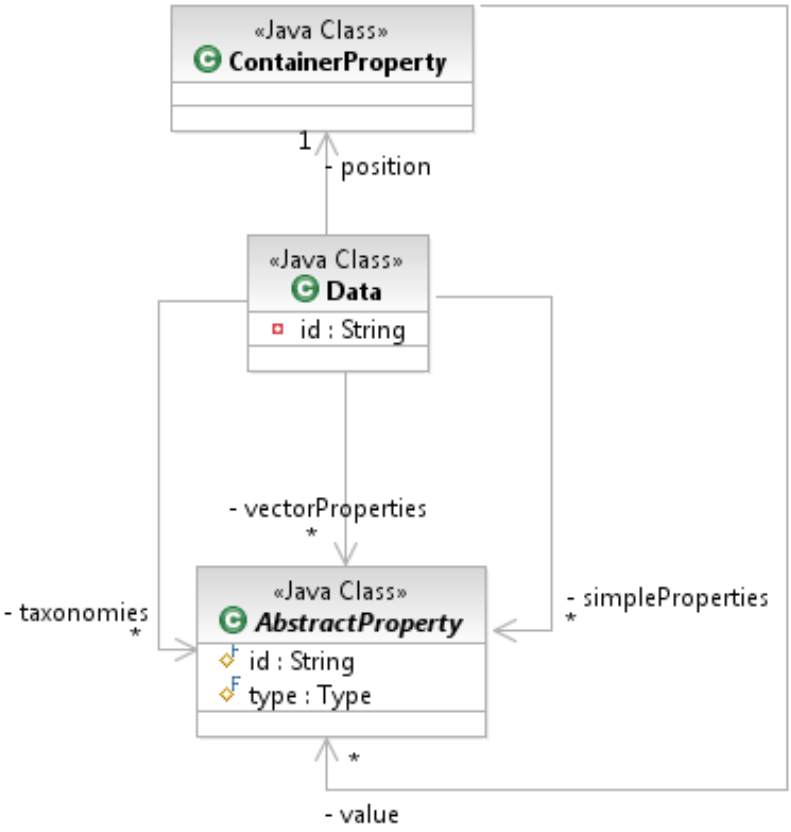

Abbildung 3.2: Aufbau der Klasse Data

Eigenschaften Die Eigenschaften der Datensammlung, wie auch die der Daten an sich, sollten so generisch aufgebaut sein, dass sie für möglichst viele unterschiedliche Anwendungsgebiete ohne Änderung und Anpassung übernommen werden können.

Beim Entwurf fiel auf, dass die auftretenden Eigenschaften nach zwei Kriterien zerfallen. Auf der einen Seite kann nach Dimension der Eigenschaften unterschieden werden. So kann zum Beispiel gespeichert werden, ob die Einheit in einem Schlachtfeldsimulator feindlich, freundlich oder neutral ist. Weiterhin ist es wäre es möglich, eine Gewichtsangabe zu speichern. In beiden fällen handelt es sich um eine einfache, weil eindimensionale, Eigenschaft.

Im Gegensatz gibt es zusammengesetzte Eigenschaften wie vektorielle Größen. In diesen Bereich fallen auch Gliederungsinformationen, oder Unterstellungsverhältnisse. Eine zusammengesetzte Eigenschaft besteht damit entweder aus einfachen, oder wiederum aus zusammengesetzten Eigenschaften. Diese Unterscheidung führte im Entwurf zur Auswahl des Composite-Patterns [Cite Gang of Four] für die Modellierung des Sachverhalts.

Das zweite Kriterium, in das die Informationen der Daten zerfallen, ist der Typ dieser. Um einen Kompromiss zwischen einem kompakten Datenmodell und einem breiten Spektrum unterstützter Typen zu gewährleisten, fiel die Entscheidung auf folgende Datentypen:

- Boolean
- Char
- Date [ANM]
- Float
- Integer
- String

Um die grundlegenden Bedürfnisse zu stillen, hätte auch eine Auswahl eines Fließkommadatentyps, mit dem sich auch ganze Zahlen darstellen lassen, und ein Zeichenkettendatentyp, mit dem alle anderen Informationen gespeichert werden können, ausgereicht. Eine noch kleinere Teilmenge, die nur den String-Datentyp umfasst, wäre unter Ausnut-

zung von programmiersprachenspezifischen Typumwandlungen auch denkbar gewesen. Diese beiden Möglichkeiten wurden aber mit Hinblick auf die komfortablere Handhabung verworfen.

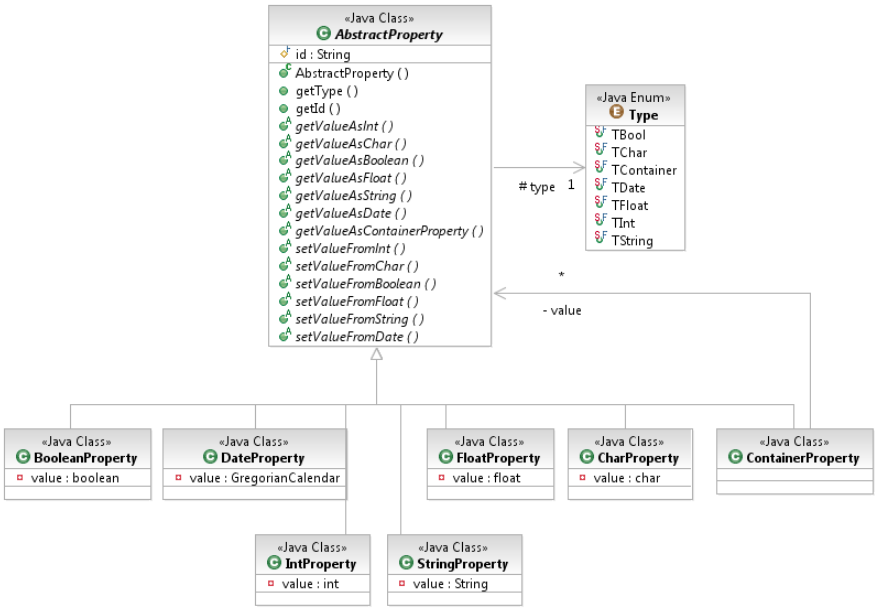

Abbildung 3.3: Eigenschaftssystem im Datenmodell

Der resultierende Entwurf ist in 3.3 dargestellt.

3.1.2 Übersicht des Frameworks

3.1.3 Beschreibung der Komponenten

Importer

Anbindung an XML

Instanzieren des Datenmodells

Mapper

Viewer

3.1.4 Zusammenspiel der Komponenten

Visualisierungsprozess

Importer und Mapper

Viewer und Importer

Viewer und Mapper

3.2 Vorstellung der prototypischen Implementierung (Visualisierung einer militärischen Lage)

3.3 Erweiterbarkeit und Individualisierbarkeit

3.4 Fallstricke und interessante Aspekte der Implementierung

3.4.1 Mapper

Streckung der Eingabedaten auf den Projektionsbereich

Entfernungsberechnung mithilfe der Haversine Formel

Informationsverlust durch Projektion eines Kugelabschnitts auf eine Ebene

3.4.2 Viewer

Mousepicking

Bewegungskegel

4 Verwendete Technologien

In diesem Kapitel werden die wichtigsten Technologien, die im Laufe der Arbeit verwendet werden, vorgestellt und erläutert.

4.1 Java

4.2 XML

4.3 jMonkeyEngine

4.4 Google Code

5 Fazit und Ausblick

5.1 Bewertung

5.2 Weiterführende Arbeit

5.3 Fazit

Literaturverzeichnis

Abbildungsverzeichnis

Listings