

Dokumentation Phase 2

WBA 2

Oliver Thesenvitz, Dennis Jaeger

30. April 2013

Inhaltsverzeichnis

1	Konzeptioneller Meilenstein - Kommunikationsabläufe und Interaktionen	3
1.1	Ressourcen	3
1.2	Synchrone Interaktion	4
1.3	Asynchrone Interaktion	4
1.4	Übermittelte Daten	4
1.5	Skizze - Interaktionen	5
2	Entwicklung XML	7
2.1	In XML nicht berücksichtigte Ressourcen	7
2.2	Umsetzung XML-Dateien	7

1 Konzeptioneller Meilenstein - Kommunikationsabläufe und Interaktionen

1.1 Ressourcen

Im Folgenden werden die in unserem Projekt identifizierten Ressourcen erläutert.

Tabelle

In „Tabelle“ wird die aktualisierte Rangliste aller Mannschaften nach jedem Spieltag abgelegt. Die Mannschaften werden absteigend anhand der erspielten Punkte geordnet. Neben der Reihenfolge sowie Namen der Mannschaften werden außerdem die geschossenen Tore sowie Gegentore und die daraus resultierende Tordifferenz angezeigt.

Spiel

Die aktuelle Begegnung einer Heim- gegen eine Gastmannschaft. Im Verlauf des Spiels werden optional die erzielten Tore bzw. das Endergebnis hinzugefügt.

Tor

Angabe eines erzielten Treffers in einem Fußball-Spiel. Zugehörig werden der Zeitpunkt des Tores (Spielminute) sowie der Torschütze gespeichert.

Spieltag

Es werden die zu diesem Spieltag zugehörigen Spielbegegnungen (*Spiel*), bestehend aus Heimmannschaft, Gastmannschaft und Ergebnis, abgelegt.

Liveticker

Echtzeit-Kommentare zu einem gegenwärtig laufendem Spiel. Diese werden als String (Text) von einem Kommentator eingegeben. Ein Kommentar hat folgende Bestandteile: Name der betroffenen Mannschaft sowie Spielereignis, beispielsweise einem Tor oder dem Ende einer Begegnung.

Mannschaft

In der Ressource *Mannschaft* werden die verschiedenen Fußball-Spieler zu einem Fußball-Verein zusammengefasst. Alternativ dazu wäre es möglich gewesen eine Ressource für die einzelnen Spieler zu entwerfen. Dadurch wären konkretere Angaben zu den einzelnen Spielern, beispielsweise die Spielerposition, realisierbar gewesen. Dieser Detaillierungsgrad ist aber in unserem Kontext nicht notwendig und würde nur die Komplexität der Anwendung erhöhen. Aus diesem Grund entscheiden wir uns für die Abbildung der Spieler in der Ressource *Mannschaft*.

Tippspiel

Die Benutzer der Anwendung können bis zu Anpfiff einer Spielbegegnung einen Tipp des Spielergebnisses einsenden. Nach Spielende wird der abgegebene Tipp mit dem realen Endergebnis verglichen. Richtige Tipps aller Benutzer können ebenfalls nach Beendigung des Spiels abgefragt werden.

1.2 Synchrone Interaktion

- Aktuelle Tabelle abrufen
- Spezielles Spiel abrufen
- Spieltag abrufen
- Liveticker eines Spiels starten
- Die zu folgenden Mannschaften auswählen
- Tipp für Tippspiel festlegen
- Tipp für Tippspiel ändern
- Tipp für Tippspiel löschen
- Eigenen Tipp mit Endergebnis des Spiels vergleichen

1.3 Asynchrone Interaktion

- Kommentare vom Liveticker abrufen
- Toralarm bei gefallenem Tor

1.4 Übermittelte Daten

Tabelle

- Mannschaftsnamen
- Punkte der Mannschaften
- Platzierung der Mannschaften
- Erzielte Tore
- Gegentore
- Tordifferenz

Spiel

- Ergebnis
- Tore
- Torschützen
- Heimmannschaft
- Auswärtsmannschaft

Tor

- Torschütze
- Mannschaft des Schützen
- Minute

Spieltag

- Spieltagsnummer
- Heimmannschaften
- Auswärtsmannschaften
- Ergebnisse

Liveticker

- Heimmannschaft
- Auswärtsmannschaft
- Aktuelles Ergebnis
- Kommentare

Livebenachrichtigung Mannschaft

- Spiel
- Tor (Torschütze / Minute)

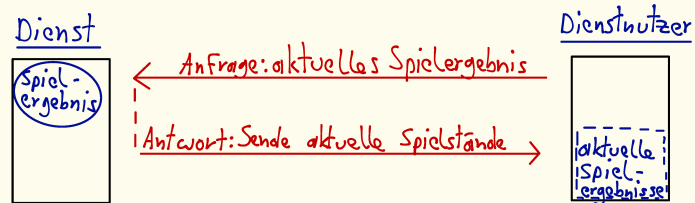
Tippspiel

- Name
- Tipp
- Ergebnis
- Spiel

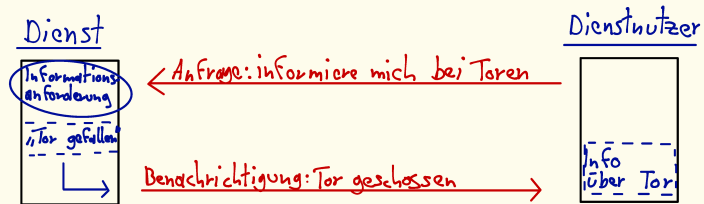
1.5 Skizze - Interaktionen

In der folgenden Skizze wurde anhand der Interaktion „Spielergebnis abfragen“ bzw. „Toralarm“ jeweils eine synchrone bzw. asynchrone Interaktion unserer Anwendung exemplarisch dargestellt. Da sich die synchronen bzw. asynchronen Interaktionen sehr ähneln genügt, hier die Abbildung einer Funktion jeder Art um die Interaktion zu verdeutlichen.

Beispiel synchrone Interaktion - aktuelle Spielergebnisse



Beispiel asynchrone Interaktion - Toralarm



2 Entwicklung XML

Für die im Projekt notwendigen XML-Dateien und der zugehörigen Schemata bedarf es einer Festlegung der Sichtweise. In der ersten Vorgehensweise können die XML-Dateien möglichst semantisch und frei von Redundanzen entwickelt werden. Die Vorteile dieser Auslegung ist durch geringere Redundanzen ein geringerer Speicherplatzbedarf. Des Weiteren sind durch die atomare Einteilung die jeweiligen Elemente kompakter und dadurch leichter lesbar.

Alternativ dazu kann die XML-Struktur anhand der späteren Programmlogik entwickelt werden. Dazu werden Überlegungen wie „welche Daten werden bei welchen Funktionen benötigt“ in den Vordergrund gestellt. Somit liegen, ähnlich der XML-Struktur in Phase1, alle relevanten Daten möglichst lokal beieinander. Die Vorteile sind eine einfacher Nachvollziehbarkeit des Gesamtprojektes in der entwickelten Software sowie eine verbesserte Maschinenlesbarkeit.

Wir werden im Folgenden mit Alternative 2 vorgehen und unsere XML-Struktur ähnlich der ersten Phase des Workshops mit möglichst vollständigen und größeren XML-Daten vorgehen. Die Entscheidung dazu beruht auf der besseren Nachvollziehbarkeit im Projektverlauf sowie der Übersichtlichkeit in den XML-Dateien.

Im Folgenden wird die Umsetzung der Ressourcen in XML-Dateien sowie die zugehörigen Schemata erläutert.

2.1 In XML nicht berücksichtigte Ressourcen

Tabelle

Da unsere Anwendung nur mit Beispieldaten gefüllt und nicht auf eine Bundesliga-Datenbank zugreifen wird haben wir uns gegen eine Darstellung der Ligatabelle entschieden. Diese würde keinen erkennbaren Mehrwert für das Projekt bedeuten und allein die Komplexität der Anwendung erhöhen. Eine Befüllung der Tabelle mit Beispieldaten würde den Datenumfang und damit die Komplexität erhöhen.

Spiel und Tor Die Ressourcen *Spiel* und *Tor* werden abstrahiert und in der Ressource *Spieltag* abgebildet. Dies ist für die weitere Entwicklung der Anwendung sinnvoller, eine einzelne Implementierung der Ressourcen ergibt in unserem Kontext keinen Sinn oder Mehrwert für das System. Mit der Entwicklung einzelner XML-Schemata für die Ressourcen *Spiel* und *Tor* würde lediglich die Komplexität erhöht ohne den Mehrwert zu erhöhen.

2.2 Umsetzung XML-Dateien

Spieltag

Die Ressource *Spieltag* wird in XML umgesetzt, um das synchrone Abfragen von aktuellen Spieltag-Ergebnissen zu realisieren. Dabei beziehen wir uns, da wir nur einen Beispiel-Datensatz verwenden, nur auf den aktuellen bzw. letzten Spieltag und nicht auf vergangene Spieltage. Im betreffenden XML-Schema werden u.A. Heim- und Gastmannschaft abgelegt. Diese Angaben könnten alternativ in einzelne XML-Files abgelegt werden. Eine zentrale Zusammenfassung in einer

XML-Datei erzielen hier aber eine bessere Nachvollziehbarkeit, eine Dezentralisierung würde eine größere Komplexität bedeuten.

Auch werden hier zu Gunsten der Lesbarkeit und Nachvollziehbarkeit eventuelle Redundanzen in Kauf genommen.

Da an jedem Spieltag alle Mannschaften antreten müssen finden bedingt durch die Anzahl der Mannschaften genau 9 Spiele statt. Diese Restriktion ist im XML-Schema abgebildet. Da in jeder Saison genau 34 Spieltage stattfinden ist das Element „Spieltagsnummer“ zwischen 1 und 34 beschränkt.

Liveticker

Aus den bei der Ressource *Spieltag* genannten Gründen wird auch hier aus logischen und funktionalen Gründen eventuell redundante Werte wie „Heimmannschaft“ und „Gastmannschaft“ abgelegt. Diese Werte werden hier semantisch zu einem Kommentar zugeordnet.

Im Schema sind sowohl Kommentare als auch Tore optional und ohne obere Schranke implementiert. Während eines Fußball-Spieles müssen keine Kommentare bzw. Tore entstehen, theoretisch aber könnten unendlich viele auftreten. Eine Begrenzung der maximalen Anzahl beider Elemente wäre möglich aber nicht sinnvoll.

Mannschaft

Der erste Entwurf des zugehörigen XML-Schemas hatte folgende Elemente:

- Name der Mannschaft
- Kader
- Spieler
- Spielerposition

Allerdings wurde diese Idee während einer Diskussion verworfen. Dieser Entwurf wäre inkonsistent zu unserer Vorgehensweise die XML-Dateien mit Sicht auf die spätere Implementierung hin zu entwerfen. Aus diesem Grund entwickeln wir eine neue XML-Datei die unserer bisherigen Vorgehensweise entspricht und semantisch besser zu der späteren Umsetzung der Anwendungslogik passt. Es werden die für die Funktion „Lieblingsmannschaft festlegen“ lediglich die Mannschaften sowie deren Abonennten gespeichert.

In der Bundesliga befinden sich zu jeder Zeit genau 18 Mannschaften. Diese Beschränkung haben wir ebenfalls im XML-Schema festgelegt. Jede der Mannschaften kann von unbegrenzt vielen Personen aboniert werden. Somit besteht auch keine Beschränkung der abonnierten Mannschften pro Person.

Tippspiel

Es werden der Name des Tippgebers, das getippte Ergebnis und das reale Endergebnis hinterlegt. Somit können die abgegebenen Tipps nach Ende eines Spiels auf Korrektheit überprüft werden, richtige Tipps können angezeigt werden. Auch der eigene Tipp kann überprüft werden. Alternativ dazu hätte eine Liste mit allen Tipps umgesetzt werden können. Allerdings wären dann die möglichen

Funktionen, die auf diese Liste angewendet werden können, stark beschränkt. Beispielsweise wäre die Überprüfung des eigenen Tipps nach Abgabe nicht mehr möglich, da die Werte keiner Person zugeordnet werden würden.