WS 1996/97
Programmierpraktikum
Verteilte Objektsysteme

K.-P. Löhr
B. Bokowski
G. Brose

Anforderungsdefinition für ein objektorientiertes Framework "Cyberspace"

---

1 Einführung

Im Programmierpraktikum soll ein objektorientiertes Framework und eine Beispielanwendung "CyberWeb" entwickelt werden. Frameworks sind Klassenbibliotheken, mit denen bestimmte Klassen von Anwendungen entwickelt werden können. Das Framework "Cyberspace" ist für die Entwicklung von verteilten Anwendungen vorgesehen, in denen sich Personen in virtuellen Orten bewegen und dabei auf andere Personen und auf Gegenstände treffen können. Dieses Framework wird im folgenden beschrieben.

1.1 Cyberspace

Abstrakt gesehen ist ein *Ort* entweder eine *Region*, die wiederum aus anderen Orten besteht, oder ein nicht weiter strukturierter *Platz*. An *Plätzen* können sich *Gegenstände* und *Personen* befinden, die, wenn sie sich gleichzeitig an demselben Platz aufhalten, in bestimmter Weise interagieren können. Personen können dabei sowohl agieren wie reagieren, Gegenstände lediglich reagieren.

Sowohl Personen wie Gegenstände können von anderen Personen *inspiziert* werden. Inspizieren bedeutet, daß eine inspizierende Person von dem inspizierten Objekt eine Information über dieses Objekt erhält, beispielsweise den Namen des Objekts ("Karl–Heinz Rummenigge", "Baum"). Weitere Interaktionen sind z.B. die Unterhaltung zwischen Personen sowie verschiedenartige Manipulationen von Gegenständen.

Personen können ihren Aufenthaltsort wechseln, in dem sie den aktuellen Platz verlassen und einen weiteren Platz betreten, der aber von dem vorigen Platz aus erreichbar sein muß. Ein Platz ist von einem anderen Platz aus erreichbar, wenn er benachbart ist. Die Definition von Nachbarschaft ist hier bewußt offengelassen, um eine möglichst große Bandbreite von Erreichbarkeitsrelationen zu erlauben. Beispielsweise soll es möglich sein, zwei Plätze als benachbart zu betrachten, die durch eine Transportverbindung wie Aufzug, Treppe, Buslinie o.ä. verbunden sind.

Wenn Personen einen Platz betreten, werden sie zu *Beobachtern* dieses Platz, d.h. sie werden von allen Änderungen, die diesen Platz betreffen, informiert. Das beinhaltet insbesondere den Fall, daß andere Personen den Platz betreten oder verlassen, aber auch bestimmte andere Änderungen an Personen oder Gegenständen, die sich an diesem Platz befinden.[1] Auf diese Weise haben Personen zu jeder Zeit ein aktuelles Bild ihres Aufenthaltsortes.

Das beschriebene Modell virtueller Orte ist, wie es sich für Frameworks gehört, abstrakt genug, um für die Entwicklung einer ganzen Klasse von Anwendungen nützlich zu sein: denkbar sind "Hyper"–Strukturen wie z.B. HTML–Dokumente, Simulationsumgebungen, verteilte Computer Supported Cooperative Work (CSCW)–Anwendungen, z.B. verteilte Softwareentwicklungs–Umgebungen, und Mehrbenutzer–Spiele, z.B. *Multi–User Dungeons* (MUDs). Um die Tauglichkeit des entwickelten Frameworks zu testen, ist eine Beispiel– und Referenzanwendung zu erstellen, in unserem Fall ein interaktives Analogon zum World Wide Web.

1.2 Die Referenzanwendung "CyberWeb"

Unsere Anwendung betrachtet das WWW als eine Ausprägung der Klassen des Cyberspace–Frameworks. Dabei werden Web–Seiten zu Plätzen, Dokumententeile wie Textstücke und Bilder sind Gegenstände, und die Leser sind die Personen des Frameworks. Durch diese Sicht wird das herkömmliche WWW um eine ganze Reihe von Möglichkeiten erweitert, die mit einem geeigneten *Browser* genutzt werden können.

Durch die Eigenschaften des Frameworks, daß Plätze Beobachter der an ihnen versammelten Gegenstände und Personen sind, erhalten die Leser einer Seite Kenntnis von anderen Lesern, die sich zeitgleich mit ihnen "auf der Seite aufhalten". Leser von Seiten können also interagieren, z.B. sich miteinander über den Inhalt der Seite unterhalten.

Da die an einem Platz befindlichen Dinge nicht nur inspizierbar, in diesem Fall also lesbar, sind, sondern auch manipuliert werden können, können Personen

[1] Welche Änderungen das im Detail sind, hängt von der Definition des Platzes ab.

Aktionen jedweder Art zur Verfügung gestellt werden. Interessant ist z.B. die Realisierung sog. Blackboard–Strukturen, bei denen mehrere Personen über ein gemeinsames Objekt — einen Seiteninhalts–Gegenstand — kooperieren können. Web–Seiten können so beispielsweise zu Mehrbenutzer–Editoren werden.

Für das CyberWeb ist ein geeigneter Browser zu entwickeln, der es erlaubt, die Möglichkeiten des CyberWeb zu nutzen. Insbesondere muß die Kontaktaufnahme zwischen Lesern einer Seite und die Manipulation von Seiteninhalten, sofern jeweils vorgesehen, möglich sein. Es sollen außer speziellen CyberWeb–Seiten natürlich auch die vorhandenen Seiten des WWW lesbar sein.

2 Funktionalität

Im folgenden werden stichpunktartig die Anforderungen an das Framework skizziert. Die einzelnen Punkte sind noch zu verfeinern und in eine funktionale Spezifikation zu fassen.

- Verteilung: Orte, Personen und Gegenstände sind verteilbar (logisch und physisch)
- Transparenz: Verteilungseigenschaften sollen für den Anwendungsentwickler "weitestgehend transparent" sein.

3 Leistung

Steht nicht im Vordergrund, ein sauberer Entwurf ist zunächst wichtiger.

4 Einbettung, Entwicklung

Die Klassen des Frameworks sollen in CORBA IDL beschrieben werden, die Implementierung der Schnittstellen ist in Java basierend auf einer CORBA–Plattform (JacORB) vorzunehmen. (evtl. auch Orbix/C++, COM/VisualBasic o. VisualC++).

4.1 Erweiterbarkeit und Portabilität

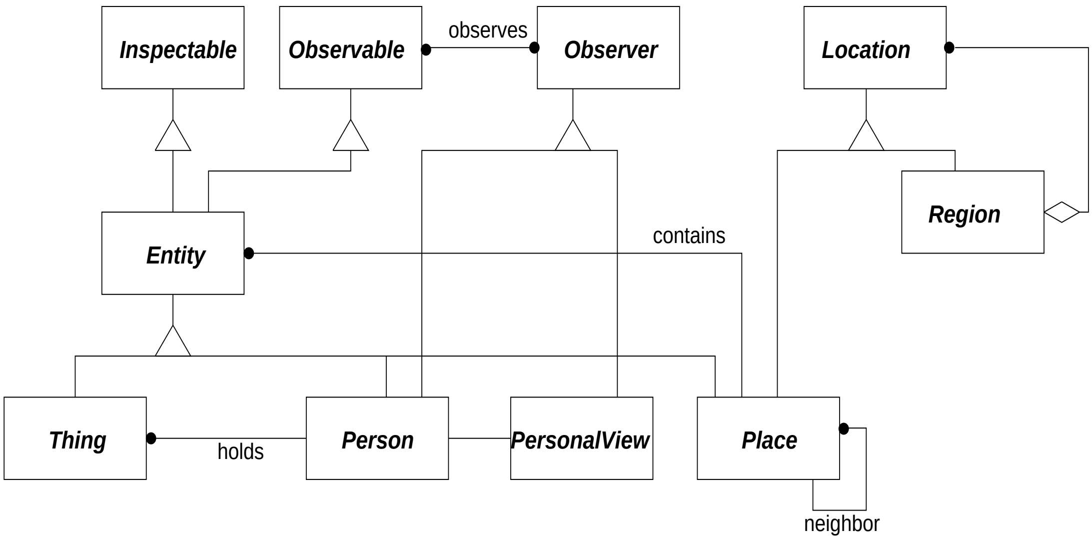
Durch die Verwendung von Java als Implementierungssprache ist Portabilität in hohem Maße gegeben. Erweiterbarkeit durch gute Dokumentation, guten OO–Entwurf (und Interaktionsprotokolle?).

4.2 Dokumentation

Ausführliche Dokumentation aller Klassen des Frameworks als annotierter IDL-Text und Objekt– und ggf. Interaktionsdiagramme.

5 Cyberspace Modell

Version 1.2.:



Änderungen seit V1.1:

- *Place* ist nun auch *Entity*.

- Neue Assoziation “holds“ zwischen *Person* und *Thing*.
- Neue Assoziation “contains“ zwischen *Place* und *Entity* statt der zwei alten Assoziationen zwischen *Place* und *Person* sowie zwischen *Place* und *Thing*.
- Neue Klasse *PersonalView* zur Darstellung des Blickfeldes einer Person.