

Pflichtenheft  
zum  
Prototypen des Forschungsprojekts  
Architektonische Komplexe

Dipl.-Ing. Norbert Paul

4. März 2008

**Zusammenfassung**

Im von der Deutschen Forschungsgemeinschaft geförderten Forschungsprojekt *Architektonische Komplexe* wurde die Entwicklung eines Prototypen für ein Planungssystem auf Basis der algebraischen Topologie versprochen. Der Prototyp sollte ermöglichen, einfache architektonische Planungsaufgaben zu bearbeiten. Dies ist das Pflichtenheft für diesen Prototypen.

## Inhaltsverzeichnis

|          |                                    |          |
|----------|------------------------------------|----------|
| <b>1</b> | <b>Zielbestimmung</b>              | <b>2</b> |
| 1.1      | Musskriterien . . . . .            | 3        |
| 1.2      | Wunschkriterien . . . . .          | 4        |
| 1.3      | Abgrenzungskriterien . . . . .     | 5        |
| 1.4      | Anwendungsbereiche . . . . .       | 5        |
| 1.5      | Zielgruppen . . . . .              | 5        |
| 1.6      | Betriebsbedingungen . . . . .      | 5        |
| <b>2</b> | <b>Produktumgebung</b>             | <b>5</b> |
| 2.1      | Software . . . . .                 | 5        |
| 2.2      | Hardware . . . . .                 | 6        |
| 2.3      | Orgware . . . . .                  | 6        |
| 2.4      | Produktschnittstellen . . . . .    | 6        |
| <b>3</b> | <b>Produktfunktion</b>             | <b>6</b> |
| 3.1      | Login . . . . .                    | 6        |
| 3.2      | Komplexauswahl . . . . .           | 6        |
| 3.3      | Darstellung . . . . .              | 6        |
| 3.4      | Zoom/Pan . . . . .                 | 6        |
| 3.5      | Arbeitsebene . . . . .             | 6        |
| 3.6      | Zeichnen . . . . .                 | 7        |
| 3.7      | Grundrisslayout . . . . .          | 7        |
| 3.8      | Versionierung/Bearbeiten . . . . . | 7        |
| 3.9      | Maßstäblichkeit . . . . .          | 7        |
| 3.10     | Schnitte . . . . .                 | 7        |
| 3.11     | Konsistenz . . . . .               | 7        |

|           |  |           |
|-----------|--|-----------|
| 3.12      | Detaillieren . . . . .                   | 7         |
| 3.13      | Sachdaten . . . . .                      | 8         |
| <b>4</b>  | <b>Produktdaten</b>                      | <b>8</b>  |
| 4.1       | Raumprogramm . . . . .                   | 8         |
| 4.2       | Raumbuch . . . . .                       | 8         |
| 4.3       | Bauteilverzeichnis . . . . .             | 8         |
| 4.4       | Katalog . . . . .                        | 8         |
| 4.5       | Komplexe, Morphismen . . . . .           | 8         |
| 4.6       | Versionen . . . . .                      | 8         |
| 4.7       | Geometrie . . . . .                      | 8         |
| 4.8       | Projekte . . . . .                       | 9         |
| 4.9       | Benutzer . . . . .                       | 9         |
| 4.10      | Sitzungsdaten . . . . .                  | 9         |
| 4.11      | Sachdaten . . . . .                      | 9         |
| <b>5</b>  | <b>Produktleistungen</b>                 | <b>9</b>  |
| <b>6</b>  | <b>Benutzerschnittstelle</b>             | <b>9</b>  |
| 6.1       | Sitzung . . . . .                        | 9         |
| 6.2       | Befehle . . . . .                        | 9         |
| 6.3       | Arbeitsebene . . . . .                   | 10        |
| 6.4       | Darstellung . . . . .                    | 10        |
| <b>7</b>  | <b>Qualitätsanforderungen</b>            | <b>10</b> |
| <b>8</b>  | <b>Nichtfunktionale Anforderungen</b>    | <b>10</b> |
| <b>9</b>  | <b>Testszenarioszenarien / Testfälle</b> | <b>10</b> |
| <b>10</b> | <b>Entwicklungsumgebung</b>              | <b>10</b> |
| 10.1      | Software . . . . .                       | 10        |
| 10.2      | Hardware . . . . .                       | 11        |
| 10.3      | Orgware . . . . .                        | 11        |
| <b>A</b>  | <b>Glossar</b>                           | <b>11</b> |

## 1 Zielbestimmung

Das Ziel des Prototypen ist, dem Antrag entsprechend, den Nutzen von algebraischer Topologie für architektonische Produktdatenmodelle zu illustrieren. Mit dem Prototypen soll es möglich sein, eine einfache architektonische Aufgabe zu bearbeiten. Daraus ergibt sich, dass die üblichen Rollen der Leistung, die eine Software zu erbringen hat, einerseits und der Implementierung dieser Leistung andererseits nicht klar voneinander getrennt sind. Im Projekt soll ja letzten Endes untersucht werden, ob gewisse *Vorgaben* bei der Implementierung (Komplexe und algebraische Topologie) für die Entwicklung von architektonischen Planungssystemen nützlich sind.

Ein Ziel des Prototypen ist, einen gewissen Minimalumfang an Planungssoftware unter Zuhilfenahme der algebraischen Topologie abzudecken.

## 1.1 Musskriterien

Für die architektonische Aufgabe ist ein Beispielentwurf und eine beispielhafte Umplanung eines bestehenden Gebäudes vorgesehen. Die Umplanung ist wichtig, da ihr im Projektantrag besondere Bedeutung beigegeben wurde. Der Prototyp muss dazu folgende Leistungen in jedem Fall erfüllen:

**Arbeitsschritte der Planung** Die im Folgenden aufgeführten Arbeitsschritte müssen vom Prototypen unterstützt werden:

- **Raumprogramm.** Spezifikation einer Entwurfsaufgabe. Das Verzeichnis aller Räume und Raumnutzungen (mit Raumgröße, eventuell auch Höhe) des zu entwerfenden Gebäudes.
- **Volumenadjazenz.** Es kann festgelegt werden, welche Räume des Raumprogramms (Volumen) nahe beieinander liegen bzw. direkt miteinander verbunden sein sollen. Ergebnis ist ein (eventuell gewichteter) Volumenadjazenzgraph.
- **Grundrisslayout.** Aus dem Volumenadjazenzgraphen (Konzeptskizze) wird eine Verteilung der Räume festgelegt. Das Ergebnis ist eine Entwurfsskizze.
- **Detaillierung.** Ausarbeitung der Elemente in der Entwurfsskizze.
- **Schnitte.** Ableitung von Schnitten und Grundrissen (als Sonderfall von Schnitten) aus dem Modell. Idealerweise sind diese editierbar. Festlegung von Bearbeitungsebenen im interaktiven Dialog.
- **Dokumentation.** Planungsverlauf und -ergebnis sind geeignet zu dokumentieren. Dies betrifft den Austausch von Planungsdokumenten und die Versionen des zugrundeliegenden Planungsmodells. Diese sind vor Veränderungen und Verlust zu schützen.

Aus der im Antrag besonders hervorgehobenen Planung im Bestand ergeben sich folgende zusätzliche Arbeitsschritte:

- **Bestandsaufnahme.** Eingabe eines Bestandsgebäudes mit weniger Detaillierungsstufen. Dieses Bestandsgebäude wird als unveränderliche Planungsgrundlage fixiert.
- **Strategisches Konzept.** Geht häufig einer Umplanung voraus, wird aber vom Prototypen nicht unterstützt.
- **Veränderung und Umplanung.** Abbruch am aufgenommenen Bestand dürfen nicht die Dokumentation der aufgenommenen Planungsgrundlage verfälschen sondern modifizieren eine Kopie desselben oder sind Maßnahme („Patch“) am Bestandsgebäude.

**CAD** Es muss möglich sein, Planzeichnungen in der Art einer CAD anzufertigen. Ein präzises geometrisches Konstruieren ist eventuell nicht erforderlich.

**Projektdateien** Es muss eine einfache Benutzer- und Projektverwaltung geben. Projektdateien wie Raumbuch, Raumprogramm und Volumenadjazenz sind ebenfalls zu halten.

**Geometrie** Eine Festlegung der geometrischen Realisierung der Komplexe. Dabei ist es ausreichend, sich auf Polyeder zu beschränken (woraus folgt, dass es prinzipiell ausreichend ist, die räumlichen Koordinaten der 0-Zellen (Knoten, Vertices) zu speichern).

**Detailierungsstufen** Die Entwurfsskizze und die Realisierung der konstruktiven Elemente werden ausgearbeitet. Dabei ist im Sinne des Projektantrags die Skizze als Dokument zu erhalten und mit der Ausarbeitung in Beziehung zu setzen (Morphismen). Zonierungen von Gebäuden sind gewissermaßen „inverse“ Detaillierungen.

**Maßstäblichkeit** Es muss also möglich sein, mehrere Detaillierungsstufen eines Projekts zueinander konsistent zu halten. Dabei kann eine feste Anzahl von Detaillierungsstufen statisch modelliert werden. Die Darstellung der Elemente ist vom Maßstab abhängig. Derart statische Modellierungsstufen können sein:

1:1000 - Volumenkörper

1:200 - Skizze

1:100 - Vereinfachte Detaillierung

1:50 - Vollständige Detaillierung

**Formalität** Die Implementierung muss eng an die im Projekt zugrundegelegte Theorie der algebraischen Topologie angelehnt sein. Der Bezug zur Mathematik ist im Quellcode stets genau zu dokumentieren.

## 1.2 Wunschkriterien

Die Erfüllung dieser Kriterien durch den Prototyp wird angestrebt.

**Benutzeranmeldung** Nur eingetragene Benutzer dürfen ein Projekt bearbeiten.

**Versionierung** Für die verschiedenen Versionen einer Planung wird in Analogie zu CVS eine Versionsgeschichte angelegt (mit Differenzplänen).

**Bestandsaufnahme** Eine Version der Planzeichnung kann als *dokumentierter Bestand* ausgezeichnet werden und ist dann prinzipiell nicht mehr durch Planung veränderbar. Jede Veränderung am Bestand erzeugt dann modifizierte Kopien des dokumentierten Bestands. Abzulegen sind dann nur die Differenzpläne.

**Transaktionen** Analog zu CVS mit checkout, update, (un)edit, commit, diff, patch, release etc. Also „lange Transaktionen“, die *nicht* die ACID-Eigenschaft haben (also Atomicity Consistency, Isolation, Durability).

**Hierarchisches Bauteilverzeichnis** Zum Erleichtern der Navigation im Modell könnte eine Hierarchie über die einzelnen Detaillierungsstufen angezeigt werden

## 1.3 Abgrenzungskriterien

Diese Kriterien sollen vom Prototypen bewusst nicht erreicht werden.

**Ablaufplanung** Auf eine Ablaufplanung für Projekte wurde in der Antragstellung bewusst verzichtet.

**Virtuelle Realität** Virtuelle Realität – auch in Ansätzen – mit Farben, Layern, Texturen etc. ist nicht erforderlich. (Verdeckte Linien werden nur bei einfacher Verwendbarkeit bereits bestehender Bibliotheken gerechnet).

**CAD** Es wird *keine* vollwertige CAD angestrebt. Es gibt *keine* Bemaßungsfunktion und *keine* parametrischen Objekte.

**Varietäten** Es werden explizit keine gekrümmten Objekte (Kurven, Flächen) vorgesehen.

## 1.4 Anwendungsbereiche

Dies ist eine Demo-Implementierung von Komplexen zu Forschungszwecken.

## 1.5 Zielgruppen

Zur Validierung des Konzepts ist von Planern/Architekten als Benutzern auszugehen.

## 1.6 Betriebsbedingungen

Die Umgebung ist eine Client-Server-Umgebung, wobei ein relationaler Datenbankserver betrieben wird, auf dem die Anwendungen als Clients zugreifen. Der Server läuft ständig, die Clients werden interaktiv bedient. Eventuell ist ein Teil des Komplex-Modells als Server zu implementieren.

# 2 Produktumgebung

Die Software soll unter folgenden Bedingungen eingesetzt werden.

## 2.1 Software

Verwendet werden Java 5 und MySQL. Auf dem Clientrechner muss also eine entsprechende virtuelle Maschine vorhanden sein. Auf dem Clientrechner sollte ebenfalls MySQL für eine Sandbox laufen. Das Projekt ist plattformunabhängig zu halten.

Auf dem Server muss ein JDBC-fähiger relationaler Datenbankserver laufen. Es wird zunächst nur MySQL verwendet. Eventuell werden auch Java-Dienste auf dem Server laufen.

## 2.2 Hardware

Die Software läuft auf je einem Standard-PC für Server und Client.

## 2.3 Orgware

Eine Netzwerkverbindung von Client zum Server ist erforderlich. Als Server kann allerdings auch der Client selbst (`localhost`) dienen.

## 2.4 Produktschnittstellen

Es werden (zunächst) keine Schnittstellen zu anderen Anwendungen gefordert.

## 3 Produktfunktion

Es folgt die genaue und detaillierte Beschreibung der einzelnen Produktfunktionen.

### 3.1 Login

Der Benutzer meldet sich unter einem Benutzernamen bei dem System an. Mit dem Benutzerkonto sind Rechte an der Bearbeitung/Einsicht der Projektdaten verknüpft.

### 3.2 Komplexauswahl

Der Benutzer erhält eine Auswahl derjenigen Projekte und Komplexe, zu deren Bearbeitung er autorisiert ist. Nach Auswahl eines Komplexes wird dieser in der vom Benutzer bevorzugten Weise grafisch dargestellt.

### 3.3 Darstellung

Der Komplex wird als 3D-Objekt am Bildschirm mit den gängigen Techniken aus der Computergraphik dargestellt. Der Benutzer muss diese Darstellung (Blickpunkt und Blickrichtung) steuern können. Es muss weiterhin möglich sein, Zellen mit dem Zeigegerät (Maus) individuell auszuwählen. Eine ausführliche Beschreibung wird im Dokument „Konzept der Geometrie des Prototyps The COMP“ gegeben.

### 3.4 Zoom/Pan

Der Benutzer ist in der Lage, die Ansicht des Komplexes zu bearbeiten. Diese Ansicht wird zusammen mit den Benutzerdaten auf dem Server gespeichert.

### 3.5 Arbeitsebene

Es wird eine (eventuell auch mehrere benannte) Arbeitsebene(n) als Hyperebene im  $\mathbb{R}^3$  angegeben. Dazu definiert der Benutzer ein orthogonales Koordinatensystem für diese Arbeitsebene. Die Spur des Komplexes in dieser Ebene wird besonders hervorgehoben. Der Benutzer wählt interaktiv Elemente des Komplexes durch Auswahl von von Spurelementen. Graphische Eingaben (Maus) werden auf diese Ebene projiziert.

### 3.6 Zeichnen

Es gibt grundsätzlich nur die elementaren Befehle des Anheftens und Teilens von Zellen. (Und die dazu inversen Befehle Löschen und Vereinigen). 0-Zellen werden an *Nichts* angeheftet (ebenfalls eine Anheftung an etwas, nämlich an eben dieses Nichts). Alle Zellen sind linear (eben, flächig). Weitere Zeichenbefehle sind nur zulässig, wenn sie als Makrobefehle nach diesem Konzept realisiert werden können.

### 3.7 Grundrisslayout

Ein eventuell vorhandener Volumenadjazenzgraph (VAG) kann im grafischen Dialog angezeigt werden. Zu diesen VAG kann ein Grundrisslayout gezeichnet werden (2D). Eine Extrusion zu einem Geschoss ist zu ermöglichen. Ebenfalls soll ein Geschoss in einen Gebäudekomplex eingefügt werden können.

### 3.8 Versionierung/Bearbeiten

Ein/e Versionsgeschichte/VersionsBaum nach dem Vorbild von CVS ist zu ermöglichen. Dazu sind die üblichen mit CVS (checkout, update, commit, diff, merge, patch) mit Komplexen zu implementieren.

### 3.9 Maßstäblichkeit

Durch Maßstabswahl wird der Detaillierungsgrad der Darstellung angepasst. Dazu gibt es verschiedene feste (statische) Maßstäbe. Z.B. 1:200 Skizze — 1:100 Entwurf — 1:50 Werkplan.

### 3.10 Schnitte

Durch Festlegung von Schnittebenen können (vereinfachte) Planzeichnungen generiert werden. Eine solche Planzeichnung ist Ausschnitt aus der Spur des Komplexes in der Schnittebene. Die Arbeitsebene kann zu einer derartigen Schnittebene deklariert werden. Eine Verallgemeinerung von Schnitten ist die Möglichkeit der gemeinsamen Verfeinerung von Komplexen (Überlagerung).

### 3.11 Konsistenz

Jede maßstäbliche Darstellung und jede Schnittzeichnung bildet einen eigenen Komplex. Diese Komplexe stehen durch Morphismen zueinander in Beziehung. Jede Änderung an einem der verbundenen Komplexe ist auch an den entsprechend anderen Komplexen entweder geeignet nachzuführen oder zu verbieten. Morphismen erfüllen also die Funktion von referentiellen Integritäten.

### 3.12 Detaillieren

Der Benutzer kann für einen Teil des Komplexes ein Detail festlegen. Dies geschieht entweder durch manuelles Detaillieren oder durch Auswahl eines Bibliotheksdetails. Dazu kann zunächst auf eine durch einen Morphismus verbundene Detailansicht gewechselt werden. Arbeitsebenen und Schnittzeichnungen gelten in den Detailansichten weiter.

### 3.13 Sachdaten

Den räumlichen Elementen können gewisse Sachdaten (Nutzung, Position im Raumprogramm etc.) zugeordnet werden. Dies sollte üblicherweise durch Referenzieren anderer Relationen geschehen. Weitere Daten sind die geometrischen Eigenschaften (Flächeninhalt). Reports erzeugen dann z.B. Raumbücher.

## 4 Produktdaten

Es gibt folgende aus Benutzersicht langfristig zu speichernde Daten.

### 4.1 Raumprogramm

Ein Verzeichnis aller Räume (Volumen), die im Entwurf vorkommen sollen mit Angabe von (meist bloß unteren) Schranken für die Raumgröße.

### 4.2 Raumbuch

In Analogie zum Raumprogramm ein Verzeichnis der tatsächlich vorkommenden Räume und raumbezogene Eigenschaften (Boden- und Wandbeläge, Möblierung etc.). Zudem ist eine Zuordnung der Räume zu Positionen im Raumprogramm notwendig.

### 4.3 Bauteilverzeichnis

Die Zuordnung Bauelement und Detail wurde im Antrag versprochen und ein entsprechendes Konzept ausgearbeitet. Idealerweise gibt es projektübergreifende Detailbibliotheken — zumindest ein projektbezogenes Detailverzeichnis ist anzustreben.

### 4.4 Katalog

Es ist ein Katalog von Komplexen und Morphismen anzulegen. Statische und dynamische Dimension sollen gleichermaßen gespeichert werden können.

### 4.5 Komplexe, Morphismen

Komplexe und Morphismen sind die zentralen Objekte und im *DKetKomp*-Konzept abzulegen.

### 4.6 Versionen

Für Komplexe ist eine Versionsgeschichte nach dem Vorbild von CVS abzulegen.

### 4.7 Geometrie

Die geometrische Realisierung  $|K|$  eines Komplex  $K$  (nur Polytope, Polyeder). Eventuell durch einen weiteren Datentyp „VertexLocation“ o.ä.

### 4.8 Projekte

Zuordnung von Komplexen zu Projekten, falls eine solche Zuordnung besteht.

### 4.9 Benutzer

Ein Verzeichnis der Benutzer und deren Rechte ist anzulegen. Es gibt Benutzerrechte bezüglich Komplexe und Rechte bezüglich Projekte.



## 4.10 Sitzungsdaten

Die Arbeitsumgebung der letzten Sitzung eines Benutzers. Dies betrifft insbesondere die gewählte Ansicht als Transformationsmatrix.

## 4.11 Sachdaten

Sachdaten sind sonstige Daten, die mit den Komplexobjekten assoziiert sind. Ein Raumprogramm ist ein Beispiel für Sachdaten.

# 5 Produktleistungen

Es werden keine besonderen Performanzanforderungen gestellt (Ausnahme: Höchstens polynomielle Laufzeit von Algorithmen). Der nachvollziehbare Bezug zur Theorie hat Vorrang.

# 6 Benutzerschnittstelle

Wichtigstes Darstellungsmedium ist das graphische Fenster zur Anzeige und Bearbeitung des Komplexes.

## 6.1 Sitzung

Eine Sitzung hat folgenden Ablauf:

1. Anmeldung bei einem Server mit Name, Passwort und Datenbank-URL. Es werden die zu bearbeitenden Projekte angezeigt.
2. Neu Anlegen oder Auswahl eines Projekts. Es wird der zuletzt bearbeitete Komplex ausgewählt.
3. Auswahl eines Komplex in dem Projekt. Die aktuelle Arbeitsumgebung wird geladen. Der Komplex wird graphisch dargestellt.
4. Update – Bearbeiten – Commit
5. Abwahl des Projekts. Die aktuelle Arbeitsumgebung wird gesichert.
6. Abmelden beim Server.

## 6.2 Befehle

Auswahl von Befehlen über das Menü. Interaktive Benutzereingabe mit Zeigegerät. Texteingabe evtl. über eine Konsole.

## 6.3 Arbeitsebene

In einer Arbeitsumgebung ist pro Komplex eine Arbeitsansicht und eine Arbeitsebene definiert. Die Arbeitsansicht legt fest, wie der Komplex dargestellt wird. Die Arbeitsebene legt grundsätzlich die Transformation der Benutzereingabe mit dem Zeigegerät fest. Die Spur des Komplexes in der Arbeitsebene wird besonders hervorgehoben dargestellt.

## 6.4 Darstellung

Affine Darstellung des 3D-Komplexes auf der 2D Oberfläche des Fensters. Eventuell Möglichkeit der perspektivischen Darstellung. Verdeckte Objekte werden nur gerechnet, wenn dazu bereits vorhandene Bibliotheksroutinen verwendet werden können (bekannte Probleme bei Java 3D). Ansonsten könnte eventuell auch eine Art Alpha-Nebel erzeugt werden. Das heißt, weiter entfernt liegende Objekte werden transparent (Alpha-Wert der Farbe) dargestellt. Diese Transparenz ist um so größer, je weiter das Objekt entfernt ist.

## 7 Qualitätsanforderungen

|                 | irrelevant | normal | gut | sehr gut |
|-----------------|------------|--------|-----|----------|
| Funktionsumfang |            | x      |     |          |
| Korrektheit     |            |        |     | X        |
| Zuverlässigkeit |            | X      |     |          |
| Benutzbarkeit   |            | X      |     |          |
| Effizienz       | X          |        |     |          |
| Übertragbarkeit |            |        | X   |          |

## 8 Nichtfunktionale Anforderungen

Implementierung in Java und JDBC.  
Plattformunabhängig.  
Lizensierung in GPL.

## 9 Testszenarioszenarien / Testfälle

Es sollte möglich sein, eine kleine Demoplanung auszuführen.

Ansonsten ist die Software testgetrieben zu entwickeln und hat die dabei entstehenden `junit`-Tests zu erfüllen.

## 10 Entwicklungsumgebung

Der Prototyp wird in Java mit Eclipse und CVS entwickelt.

### 10.1 Software

Verwendung von Java 5 (mit Generics) und eine Eclipse-Version, die an Generics angepasst ist. Getestet wird mit `junit`. Zugriff auf einen relationalen Datenbankserver (MySQL). Eventuell weitere Funktionalität auf den Server auslagern. Java3D sollte eingesetzt werden, solange dies die Plattformunabhängigkeit nicht verletzt.

### 10.2 Hardware

Standard PC.

## 10.3 Orgware

Versionsverwaltung mit CVS, Netzwerk.

## A Glossar

**Komplex** Verallgemeinerung von Graph: Eine topologischer Raum mit einer gewissen algebraischen Struktur, die ebenfalls als Komplex bezeichnet wird.

**Morphismus** Verallgemeinerung von Abbildung. Ein Morphismus ist eine gerichtete Beziehung zwischen zwei mathematischen Objekten, die auf die Struktur dieser Objekte Rücksicht nimmt. Beispiel Abbildung: Die Objekte sind Definitions- und Wertemenge, die Richtung geht von der Definitionsmenge zur Wertemenge. Die Berücksichtigung der Struktur ist: Wenn ein  $x$  Element der Definitionsmenge ist, dann ist  $f(x)$  Element der Wertemenge. Komplexmorphisme berücksichtigen die algebraische bzw. topologische Struktur der Komplexe.

**Raumbuch** Ein Raumbuch ist die Dokumentation der Raum- und Nutzungsverteilung in einem Projekt und die Ausstattung der einzelnen Räume. Das Raumbuch beschreibt im Gegensatz zum Raumprogramm die *Lösung* einer Planungsaufgabe.

**Raumprogramm** Ein Raumprogramm ist das „Pflichtenheft“ des Architekten. Es ist die detaillierte Auflistung der durch die Planung zu realisierenden Raumnutzungen. Das Raumprogramm beschreibt also im Gegensatz zum Raumbuch die *Aufgabenstellung*.

**Sandbox** Speicherort für die Arbeitsdaten auf Client-Seite. Der Benutzer lädt die aktuellen Projektdaten vom Server in die Sandbox (update). Er bearbeitet nur Daten in der Sandbox. Erst danach werden die modifizierten Daten von der Sandbox zum Server übertragen (commit) und dort abgespeichert.

**Spur** Eine Teilmenge  $A$  der Punkte  $X$  eines topologischen Raums  $(X, \mathcal{T})$  erzeugt einen Teilraum  $(A, \mathcal{T}|_A)$  dieses Raums  $(X, \mathcal{T})$ . Dieser Teilraum heißt dann auch Spur von  $A$  in  $(X, \mathcal{T})$ , seine Topologie ist die Spurtopologie. Dies ist auch auf Komplexe übertragbar.

**Volumenadjazenzgraph** Die Raumvolumen eines Gebäudes sind durch Raumübergänge (Türen, Durchgänge) untereinander verbunden. Die Volumen bilden somit die Knoten und die Übergänge die Kanten eines Graphen, der auch als Volumenadjazenzgraph bezeichnet wird.