

Dirk Donath, Frank Petzold, Jörg Braunes, Daniel Fehlhaber,
Helga Tauscher, Richard Junge, Roland Göttig

**IT-gestützte projekt- und zeitbezogene
Erfassung und Entscheidungsunter-
stützung in der frühen Phase der
Planung im Bestand (Initiierungsphase)
auf Grundlage eines IFC-basierten CMS**



F 2751

Bei dieser Veröffentlichung handelt es sich um die Kopie des Abschlußberichtes einer vom Bundesministerium für Verkehr, Bau und Stadtentwicklung -BMVBS- im Rahmen der Forschungsinitiative »Zukunft Bau« geförderten Forschungsarbeit. Die in dieser Forschungsarbeit enthaltenen Darstellungen und Empfehlungen geben die fachlichen Auffassungen der Verfasser wieder. Diese werden hier unverändert wiedergegeben, sie geben nicht unbedingt die Meinung des Zuwendungsgebers oder des Herausgebers wieder.

Dieser Forschungsbericht wurde mit modernsten Hochleistungskopierern auf Einzelanfrage hergestellt.

Die Originalmanuskripte wurden reprotechnisch, jedoch nicht inhaltlich überarbeitet. Die Druckqualität hängt von der reprotechnischen Eignung des Originalmanuskriptes ab, das uns vom Autor bzw. von der Forschungsstelle zur Verfügung gestellt wurde.

© by Fraunhofer IRB Verlag

2010

ISBN 978-3-8167-8341-1

Vervielfältigung, auch auszugsweise,
nur mit ausdrücklicher Zustimmung des Verlages.

Fraunhofer IRB Verlag

Fraunhofer-Informationszentrum Raum und Bau

Postfach 80 04 69

70504 Stuttgart

Nobelstraße 12

70569 Stuttgart

Telefon 07 11 9 70 - 25 00

Telefax 07 11 9 70 - 25 08

E-Mail irb@irb.fraunhofer.de

www.baufachinformation.de

www.irb.fraunhofer.de/bauforschung

Endbericht zum Forschungsprojekt

IT-gestützte projekt- und zeitbezogene Erfassung und Entscheidungsunterstützung in der frühen Phase der Planung im Bestand (Initiierungsphase) auf Grundlage eines IFC-basierten CMS

Der Forschungsbericht wurde mit Mitteln der Forschungsinitiative Zukunft Bau des Bundesamtes für Bauwesen und Raumordnung gefördert. Die Verantwortung für den Inhalt liegt beim Autor.

Projektzeit: 11/2007 – 12/2008



Bundesamt
für Bauwesen und
Raumordnung

Thema: IT-gestützte projekt- und zeitbezogene Erfassung und Entscheidungsunterstützung in der frühen Phase der Planung im Bestand (Initiierungsphase) auf Grundlage eines IFC-basierten CMS

Forschende Stelle

Name: Bauhaus-Universität Weimar, Professur Informatik in der Architektur
Professor: Prof. Dr.-Ing. Dirk Donath
Juniorprofessur Architekturinformatik
Prof. Dr.-Ing. Frank Petzold
Anschrift: Belvederer Allee 1
99423 Weimar
Tel.: 03643 - 58 42 01
Internet: <http://infar.architektur.uni-weimar.de>
Email: caad@archit.uni-weimar.de
wiss. Mitarbeiter: Dipl.-Ing. Jörg Braunes
Dipl.-Ing. Daniel Fehlhaber
Dipl.-Ing. Helga Tauscher

Name: TU München, Fachgebiet CAAD
Professor: Prof. Dipl.-Ing. Richard Junge
Anschrift: Arcisstr. 21
80333 München
Tel.: 089 - 289 22 171
Internet: www.caad.ar.tum.de
Email: junge@caad.arch.tu-muenchen.de
wiss. Mitarbeiter: Dipl.-Ing. Roland Göttig

Partner aus Industrie und Praxis:

Name: Kubit GmbH
Geschäftsführer: Dr.-Ing. Dipl.-Inform. Oliver Bringmann
Dipl.-Ing. Matthias Kokschi
Anschrift: Altplauen 19
01187 Dresden
Tel.: 0351 - 41 76 70
Email: info@kubit.de
Internet: www.kubit.de

Name: AWO Thüringen
Geschäftsführer: Dipl.- soz. Päd. Michael Hack
Anschrift: Pfeiffersgasse 12
99084 Erfurt
Tel.: 0361 - 21 03 10
Email: landesverband@awo-thueringen.de, michael.hack@awo-thueringen.de
Internet: www.awo-thueringen.de

Name: Ingenieurgesellschaft Ruf-Kraft-Schömig
Geschäftsführer: Dr.-Ing. Hermann Kraft
Anschrift: Scheidlerstr. 29
07745 Jena
Tel.: 03641 - 48 99 25
Email: rks-jena@rks.de
Internet: www.rks.de

Name: SYSLAB.COM GmbH
Ansprechpartner: Dipl.-Ing. Alexander Pilz
Anschrift: Landwehrstr.60-62
80336 München
Tel.: 089 - 3063589-0
Email: info@syslab.com
Internet: www.syslab.com

Inhalt

1. Einführung	7
1.1. Problembeschreibung	7
1.2. Zielstellung, angestrebtes Ergebnis	9
1.3. Angewandte Methoden	9
1.4. Projektaufbau	10
2. Systemkonzeption	12
3. Kosten-Risiko-Analyse	13
3.1. Stand der Forschung und Praxis	14
3.1.1. Kostenermittlung	14
3.1.2. Risikomanagement	15
3.1.3. Stochastik	16
3.2. Neue Ansätze der Kosten-Risiko-Analyse	16
3.3. Entwicklung von Algorithmen zur Bewertung von Kosten und Risiken	18
3.4. Fachwissen und Nutzerdaten als Berechnungsbasis	19
4. Datenmodellierung / Bauwerksmodell	21
4.1. Stand der Forschung und Technik	21
4.1.1. Bauwerksmodelle und Datenaustausch	21
4.1.2. Dynamische Bauwerksmodelle	21
4.2. Teilmodelle	22
4.2.1. Ink-Modell (IM)	23
4.2.2. Skizzeninterpretationsmodell (SIM)	23
4.2.3. Geometriemodell (GM)	24
4.2.4. Aufmaßmodell (AM)	24
4.2.5. Raum- und Bauteilmodell (RBM)	24
4.3. Grundlegende Konzepte des Raum- und Bauteilmodells	24
4.3.1. Basisklasse (BaseClass)	24
4.3.2. Gebäudeklasse (BuildingClass)	25
4.3.3. Raum- und Gebäudestruktur (SpatialStructure)	25
4.3.4. Bauteilstruktur (BuildingElementClass)	26
4.3.5. Mängel (Defect)	26
4.3.6. Maßnahmen (Task)	27
4.4. IFC-Kompatibilität	27
4.4.1. Raum- und Bauteilstruktur	27
4.4.2. Mängel	28
4.4.3. Geometrieabbildung	28
5. Skizzenbasierte Ersterfassung	30
5.1. Anforderungen an Benutzungsoberfläche und Interaktion beim Arbeiten vor Ort	30
5.2. Stand der Forschung: computergestützte Vor-Ort-Erfassung	30
5.2.1. Verarbeitung und Interpretation skizzenhaft eingegebener Informationen	31
5.2.2. Ergonomie stiftbasierter Interaktion	32

5.3.	Konzept eines skizzenbasierten Ersterfassungssystems	33
5.3.1.	Sketch Mode	34
5.3.2.	Capture Mode	34
5.3.3.	Survey Mode	35
5.3.4.	Raum- und Bauteilstruktur.....	35
6.	Content Management System	37
6.1.	Stand der Forschung und Praxis	37
6.2.	Anpassungen innerhalb des Projektes	38
7.	Prototypische Umsetzung	39
7.1.	Modul Ersterfassung (InitialDataCapture).....	39
7.1.1.	Funktionalität.....	40
7.1.2.	Implementation.....	41
7.1.3.	Potential Weiterentwicklung, Ausblick.....	41
7.2.	Modul IDC for hylasFM	41
7.2.1.	Funktionalität.....	42
7.2.2.	Implementation.....	42
7.2.3.	Potential Weiterentwicklung, Ausblick.....	42
7.3.	Modul Kosten-Risiko-Analyse	43
7.3.1.	Funktionalität.....	43
7.3.2.	Implementation.....	43
7.3.3.	Potential, Weiterentwicklung, Ausblick.....	44
7.4.	Modul CMS / IFC	44
7.4.1.	Funktionalität und Implementierung	45
7.4.2.	Potential Weiterentwicklung, Ausblick.....	46
8.	Evaluation	47
8.1.	Beispielobjekt Kindertagesstätte „Am Fuchsgrund“	47
8.1.1.	Bauform	47
8.1.2.	Schäden und Mängel	48
8.2.	Testszenarien	49
8.2.1.	Skizzenbasierte Ersterfassung.....	49
8.2.2.	Aufmaß	51
8.2.3.	Datenmodell und IFC Export.....	52
8.2.4.	Integration von Fachwissen und Kostenauswertung	53
9.	Zusammenfassung, Ausblick und offene Fragestellungen	55
9.1.	Datenmodellierung / Bauwerksmodell	55
9.2.	Skizzenbasierte Ersterfassung	56
9.3.	Kosten-Risiko-Analyse.....	56
10.	Abkürzungen und Synonyme.....	57
11.	Referenzen.....	58

Anhang	61
A Eigene Veröffentlichungen	61
B Funktionsumfang des Softwarekonzeptes / Capture-Mode.....	62
C Klassenstruktur und Aufbau des Raum- und Bauteilmodells	63
C 1. Einführung	63
C 2. Aufbau	63
C 2.1. Abstrakte Definitionen	63
C 2.2. Ressourcen.....	64
C 2.3. Gruppierungsmechanismus	65
C 2.4. Projektstruktur.....	65
C 2.5. Raumstruktur	66
C 2.6. Raumbegrenzung	69
C 2.7. Bauteilstruktur	70
C 2.8. Maßnahmenmodell	74
C 3. Klassendiagramm.....	77
D Technical Report - Stiftbasierte Benutzungsoberfläche für die Eingabe von Skizzen ..	78

1. Einführung

1.1. Problembeschreibung

Der Anteil von Baumaßnahmen im Bestand wächst unaufhaltsam. Bei der Planung und Ausführung von diesen Maßnahmen kommt es jedoch oftmals zu großen Problemen auf Grund von fehlenden Informationen, die zu verspäteten Entscheidungen und enormen Kostensteigerungen führen können. Das Bauen im Bestand ist daher bei vielen Auftraggebern in Verruf und dringend notwendige Sanierungsprojekte werden unter großen Ungewissheiten durchgeführt oder gar nicht erst begonnen.

Auftretende Probleme aufgrund von fehlerhaften Informationen, lassen sich in den meisten Fällen durch finanzielle Mittel in Verbindung mit verlängerten Bauzeiten lösen. Daher kommt dem rechtzeitigen Erkennen der anstehenden Aufgaben/Probleme und der daraus resultierenden Kostenermittlung in der Planung der Baumaßnahmen ein großer Wert am Erfolg eines Projektes zu. Bei Neubauprojekten wird diesem Umstand seit Jahrzehnten Rechnung getragen und es wurden umfangreiche Systeme und Datenbanken geschaffen, welche eine frühzeitige Kostenermittlung in Abhängigkeit der Projektziele mit relativ zuverlässigen Ergebnissen ermöglicht.

Auch im Altbau bzw. beim Bauen im Bestand gibt es mittlerweile Datenbanken, aus denen Werte für Kostenermittlungen entnommen werden können. Trotzdem gibt es noch immer große Schwierigkeiten durch kaum verwertbare Ergebnisse mit großen Unsicherheiten oder – für den Auftraggeber viel schlimmer – scheinbar präzise Ergebnisse, welche im Nachhinein nicht erreicht werden können.

„Obwohl Bauen im Bestand eine immer wichtigere Rolle spielt, sind viele Bauherren und ausführende Unternehmen zurückhaltend, wenn es um Erneuerungsmaßnahmen geht. Sanierungs- und Modernisierungsmaßnahmen im Wohnungsbau werden zum größten Teil (mehr als 90 %) von privaten Bauherren beauftragt. Bauen im Bestand ist für sie oft mit großen finanziellen Risiken behaftet, da eine Kostenschätzung zu Beginn eines Erneuerungsprojektes aufgrund fehlender Erfahrung häufig schwer fällt.“¹

„Das Volumen der Bauleistungen im Bestand wächst kontinuierlich: 2004 lag es bei 78 Mrd. €, während das Neubauvolumen etwa 52 Mrd. € betrug. Prognosen gehen davon aus, dass das Modernisierungsvolumen in Deutschland bis 2020 auf etwa 90 Mrd. € pro Jahr steigt. Auch über den Wohnungsbau hinaus gibt es im Hochbau ein erhebliches Marktpotenzial für Modernisierung. [...] Bauen im Bestand unterscheidet sich technisch und ökonomisch von der Neubautätigkeit: Bestandsmaßnahmen sind schwerer zu planen und bergen oft technische und finanzielle „Überraschungen“. Die durchschnittlichen Investitionen je Maßnahme sind deutlich niedriger als beim Neubau, besonders bei privaten Eigentümern.“²

Der große Unterschied zum Neubau ist der notwendige aber oftmals schwer vorhersehbare Umgang mit den bestehenden Bauteilen. Diese können bei einfacher Weiterverwendung ohne Änderungen zu erheblichen Kostenersparnissen gegenüber einem Neubau führen, jedoch bei Mängeln auch zu umfangreichen Maßnahmen. Die Ermittlung bzw. Planung, welche Maßnahmen an bestehenden Bauteilen durchzuführen sind, ist gleichermaßen

¹ Quelle RKW BAU-RATIONALISIERUNG – ibv · Juli 2005 Seite 11 ff.

² Quelle www.rkw.de/01_forschung/Bauwirtschaft/BauenBestand/index.html

komplex und wichtig für das Projektergebnis. Trotzdem wird diese zwingend notwendige Vorleistung für die erfolgreiche Abwicklung eines Projektes nicht ausreichend konsequent durchgeführt.

In den aktuellen Architektursystemen können nur Zustände eines Objektes, z.B. der Anfangszustand durch das Bauaufmaß oder der Endzustand durch die fertige Planung, abgebildet werden. Der Weg zwischen diesen Zuständen und damit die Maßnahmen, die zum Erreichen des Endzustandes erforderlich sind, werden in allen gängigen Systemen vernachlässigt.

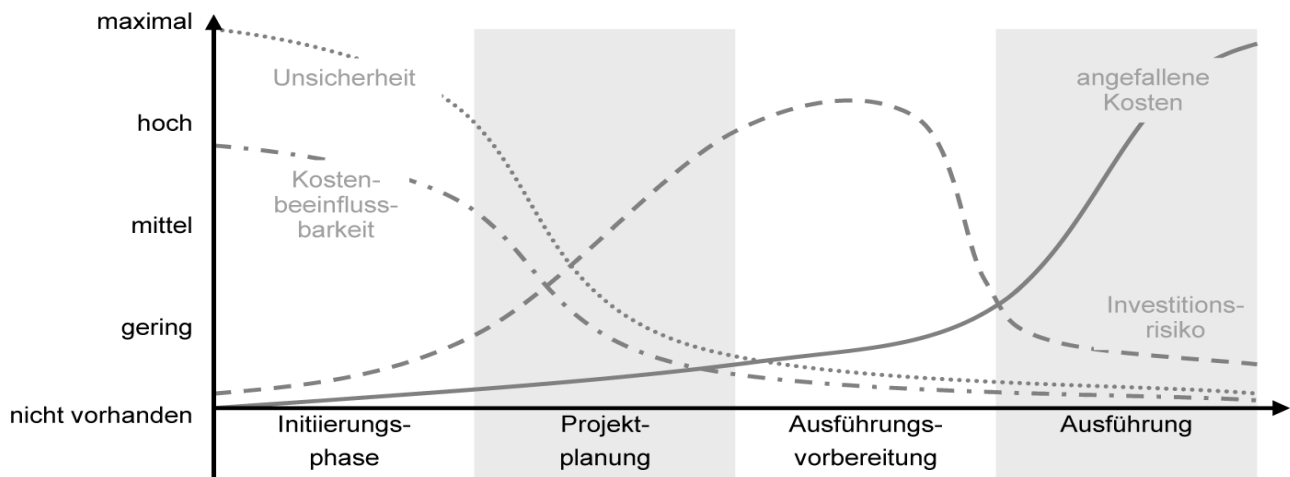


Abb. 1: Potentiale für Prozessoptimierung in den frühen Phasen der Planung

Beim Neubau reicht es für eine Kostenermittlung aus, den Endzustand zu kennen, da alle darin befindlichen Bauteile von einem „Nichts“ ausgehend erstellt werden müssen. Die einzelnen Bauteile können mit mehr oder weniger detaillierten Kostenkennwerten versehen werden und führen zu jeder Planungsphase zu einem relativ zuverlässigen Ergebnis, welches durch Anpassung des Endzustandes innerhalb der Planungsphasen auch auf ein gewünschtes Ziel ausgerichtet werden kann.

Beim Bauen im Bestand reicht diese Vorgehensweise nicht aus, da statt dem „Nichts“ der Status quo als Ausgangspunkt aller Maßnahmen angesetzt werden muss. Die Ermittlung von diesem Status (Bauweise, Materialien, Zustand) wird meistens erst parallel zur Planung durchgeführt, wodurch insbesondere in den frühen Planungsphasen sehr hohe Unsicherheiten gepaart mit einer noch höheren Unwissenheit die Ergebnisse von Kostenermittlungen und deren Deutung negativ beeinflussen. Selbst die Realisierbarkeit eines Sanierungsvorhabens (mit vertretbaren Aufwendungen) kann zu diesem frühen Zeitpunkt im Projekt schlecht vorhergesagt werden.

Die Unsicherheit und Unwissenheit führen zu einer mangelhaften Risikobewertung beim gesamten Vorhaben und daraus resultierend viel zu oft zu erheblichen Kostensteigerungen, Bauzeitverlängerungen und/oder Qualitätseinbußen. Neben den so genannten Sowieso-Kosten, welche auch bei rechtzeitigem Erkennen der Risiken unvermeidbar gewesen wären, kommt es durch Provisorien, Mehrfacharbeit und Baustillstände häufig auch zu enormen Zusatzkosten. Diese verursachen neben dem persönlichen wirtschaftlichen Nachteil des Auftraggebers einen erheblichen volkswirtschaftlichen Schaden. Mit jedem Prozent Baukosten, welches durch ein vorausschauendes System eingespart werden kann, stehen ca. 0,8 Mrd. € jährlich für zusätzliche Baumaßnahmen zur Verfügung. Darüber hinaus

können durch eine konsequente Datensammlung und -haltung sowohl der Planer als auch alle anderen am Bau beteiligten Mehrfachaufwendungen reduzieren. Die Nutzung aller zur Verfügung stehenden Daten ermöglicht zu jedem Zeitpunkt die häufig geforderte Transparenz und ein umfassendes Risikomanagement.

1.2. Zielstellung, angestrebtes Ergebnis

Ziel des Forschungsprojektes ist die Entwicklung eines Systems, in welchem Daten zum Bestandsobjekt schnell und dauerhaft erfasst werden können und die notwendigen Informationen für eine aussagekräftige Kostenermittlung und Risikobewertung schon in den frühen Leistungsphasen bereit gestellt werden. Wichtig ist natürlich, dass die erfassten Bestandsdaten für die nachfolgenden Planungsphasen und weitere Analyseverfahren anderer Fachdisziplinen zur Verfügung stehen.

Als Datengrundlage dieses Systems wird ein Gebäudemodell benötigt, welches den Status quo des Gebäudes abbildet, sowie die notwendigen Maßnahmen um den geplanten Endzustand zu erreichen. Da in den ersten Planungsphasen nicht alle notwendigen Entscheidungen eindeutig getroffen werden können, muss das System Unsicherheiten verwalten und auswerten können. Die Information, dass Daten fehlen oder Entscheidungen offen sind, muss im Ergebnis einer Kostenermittlung erkennbar sein, damit sowohl Auftraggeber, als auch Planer in der Lage sind, die möglichen Auswirkungen der Unsicherheiten zu erkennen, um darauf mit detaillierteren Untersuchungen bzw. anderen Planungsalternativen reagieren zu können.

Da das System nicht als eigenständiges durchgängiges Planungssystem über alle Phasen konzipiert wurde, musste ein Datenaustausch mit anderen Planungssystemen gewährleistet werden. Dabei dürfen jedoch zum einen keine Daten verloren gehen, und zum anderen sollte ein allgemein gültiges Datenmodell verwendet werden.

Während des gesamten Projektes ist auf ein einheitliches Datenmodell Wert zu legen, welches zentral und mit verteilten Nutzungsrechten den Projektbeteiligten zur Verfügung gestellt werden soll. Hierfür wird in dem Forschungsprojekt die Nutzung eines Content Management Systems in Kombination mit den Industry Foundation Classes (IFC) untersucht.

1.3. Angewandte Methoden

Im Rahmen des Forschungsprojektes kamen verschiedene methodische Herangehensweisen zur Anwendung. Für die Anforderungsanalyse an die notwendigen aufzunehmenden Bestandsdaten vor Ort konnte zunächst auf umfangreiche Untersuchungen vorangehender Forschungsprojekte zurückgegriffen werden ([SFB524_05a],[SFB524_05b]).

Die Detaillierung der Anforderungen an die aufzunehmenden Daten erfolgte im Rahmen der Bearbeitung des Arbeitspaketes „Kosten-Risiko-Analyse“ (Kap. 2). Hierbei wurden zunächst praxisübliche Verfahren untersucht und kritisch bewertet. Des Weiteren erfolgten die Betrachtung von Analyseverfahren anderer Fachdisziplinen und deren mögliche Anwendbarkeit innerhalb des Forschungsprojektes. Ergänzt wurden diese Analysen durch die praktischen Erfahrungen der am Projekt beteiligten Industriepartner

Zur Ausarbeitung des Systemkonzeptes wurden Methodiken und Notationen aus der Softwareentwicklung herangezogen.

Für die Erarbeitung der stiftbasierten Eingabe- und Interaktionstechniken erfolgte eine umfangreiche Recherche zu bestehenden Hard- und Softwarelösungen. Die Ergebnisse dieser Untersuchung sind in einem Technical Report zusammengefasst und dem Anhang zu entnehmen.

Als Nachweis der Umsetzbarkeit einzelner Aspekte des Softwarekonzeptes wurden verschiedene Prototypen entwickelt (proof of concept) und zunächst unter Laborbedingungen getestet. Zur Implementierung dieser Prototypen kamen unterschiedliche Entwicklungsumgebungen und Softwarewerkzeuge zum Einsatz. Die Auswahl dieser Werkzeuge erfolgte jeweils unter dem Aspekt einer schnellen und einfachen Umsetzbarkeit. Die Funktionalitäten und zugrundeliegenden Softwarewerkzeuge der Prototypen sind im Kap. 7 dargelegt.

Zur Gesamtbewertung des konzipierten Systems erfolgte eine Evaluation der Prototypen an einem Beispielobjekt. Die Auswahl des Gebäudes erfolgte in enger Zusammenarbeit mit dem Praxispartner AWO Thüringen nach eingehender Betrachtung verschiedener in Frage kommender Objekte.

1.4. Projektaufbau

Kern des Projektes war es, einen möglichst praxisorientierten Ansatz der frühzeitigen Kosten- und Risikoanalyse zu entwickeln und in einem vor Ort nutzbaren Softwaresystem zur Ersterfassung und Bewertung konzeptionell umzusetzen. Um dies zu erreichen erfolgte die Bearbeitung des Projektes in enger Zusammenarbeit mit den Partnern aus der Industrie.

Die Rolle der Praxispartner lässt sich in zwei wesentliche Aufgaben unterteilen:

- Die Partner RKS und AWO übernahmen beratende Tätigkeiten bei der Anforderungsanalyse der notwendigen aufzunehmenden Bestandsdaten. Die AWO stellte ferner eine Auswahl geeigneter Objekte für die Evaluierung der Prototypen zur Verfügung, klärte Zugänglichkeiten und stellte Objektinformationen und Unterlagen zusammen.
- Die Partner Syslab und Kubit unterstützten die softwaretechnische Umsetzung der Prototypen. Syslab stellte die Content Management Plattform auf Basis des freien Systems Plone zur Verfügung und richtete in Zusammenarbeit mit der TU München den Projektraum ein. Kubit stellte Softwarewerkzeuge zur Datenmodellierung (hylaFM) bereit und implementierte Teile der Prototypen, insbesondere Schnittstellen zwischen verschiedenen Modulen.

Die Projektbearbeitung erfolgte in verschiedenen Arbeitspaketen, welche jeweils Teilaspekte der Gesamtkonzeption beinhalteten. Die Erarbeitung erfolgte zum Teil parallel, da sich Inhalte überschneiden bzw. sich gegenseitig beeinflussten, und durch verschiedene Projektbeteiligte bearbeitet wurden. Die Hauptarbeitspakete lassen sich wie folgt unterteilen:

- Kosten-Risiko-Analyse
- Datenmodellierung / Bauwerksmodell
- Skizzenbasierte Ersterfassung
- Content Management System

Eine genaue Darlegung der Inhalte und erarbeiteten Ergebnisse folgt in den weiteren Kapiteln.

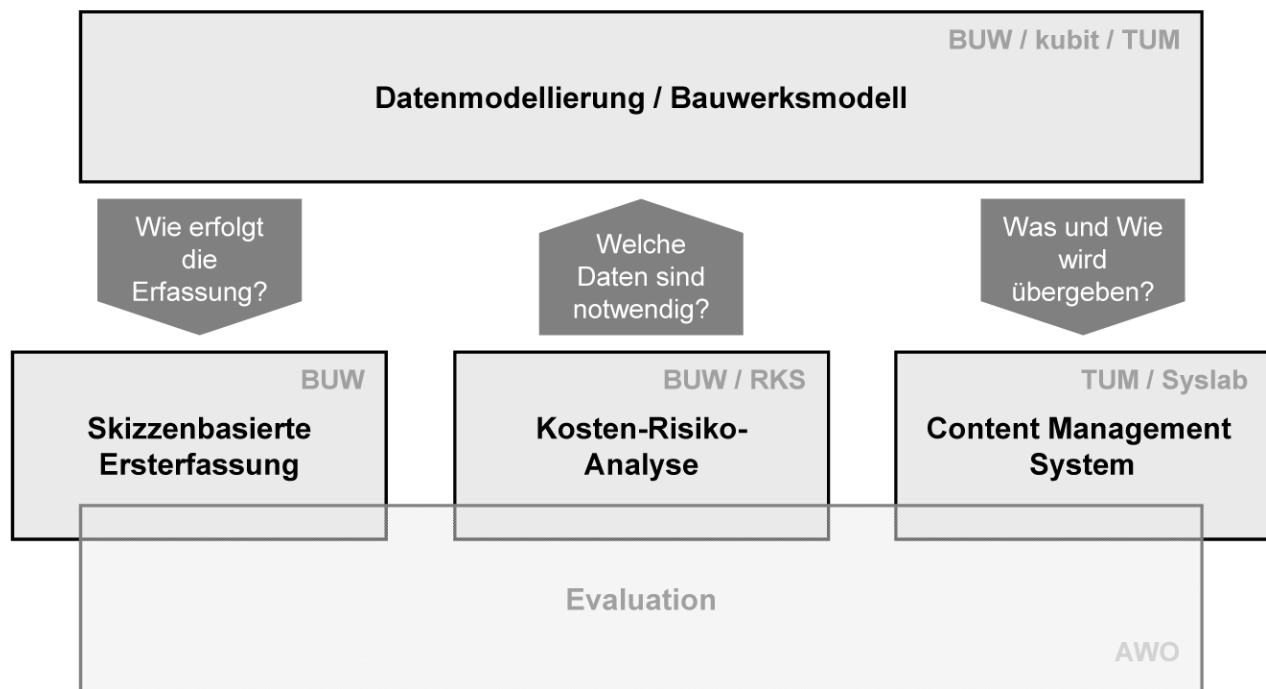


Abb. 2: Arbeitspakete und deren bearbeitende Partner sowie gegenseitige Anforderungen

Innerhalb dieser Arbeitspakete erfolgten jeweils Testimplementationen in Form verschiedener Softwareprototypen. Funktionalität, Umsetzung und das Potential möglicher Weiterentwicklungen dieser Prototypen sind separat im Kapitel 7 erläutert. Abb. 2 zeigt die Arbeitspakete, inklusive der bearbeitenden Partner, und deren Zusammenhänge.

2. Systemkonzeption

Bevor auf die einzelnen bearbeiteten Teilaspekte näher eingegangen wird, erfolgt zunächst die Betrachtung der Gesamtkonzeption.

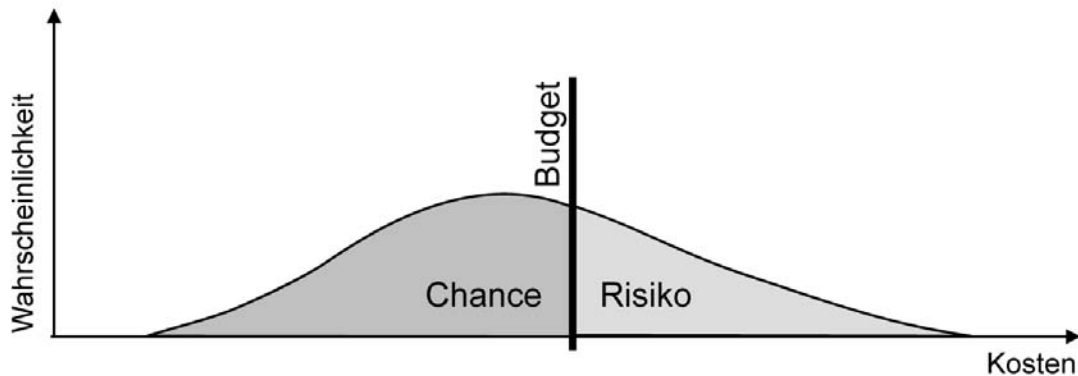


Abb. 3: Wahrscheinlichkeiten der zu erwartenden Kosten nach Berücksichtigung aller möglichen Maßnahmenvarianten bei einem Revitalisierungsvorhaben

Mit Hilfe des konzipierten Systems wird der aktuelle Gebäudezustand erfasst, bewertet, und die Kosten der notwendigen Maßnahmen – insbesondere Instandsetzungsmaßnahmen – ermittelt. Da in der Initiierungsphase eines Revitalisierungsvorhabens der aktuelle und der angestrebte Gebäudezustand nicht vollständig bekannt sein können, sollen alle möglichen Maßnahmenvarianten Berücksichtigung finden. Hierzu wird das stochastische Verfahren der Monte-Carlo-Simulation verwendet, um die Wahrscheinlichkeiten der zu erwartenden Kosten zu berechnen (Abb. 3).

Ausgehend von dem festgesetzten Budget für das Revitalisierungsprojekt lassen sich anhand der Wahrscheinlichkeitsverteilung der zu erwartenden Kosten die Chancen und Risiken der Projektfinanzierung ablesen, welche als transparente und nachvollziehbare Entscheidungsunterstützung für den Auftraggeber dienen.

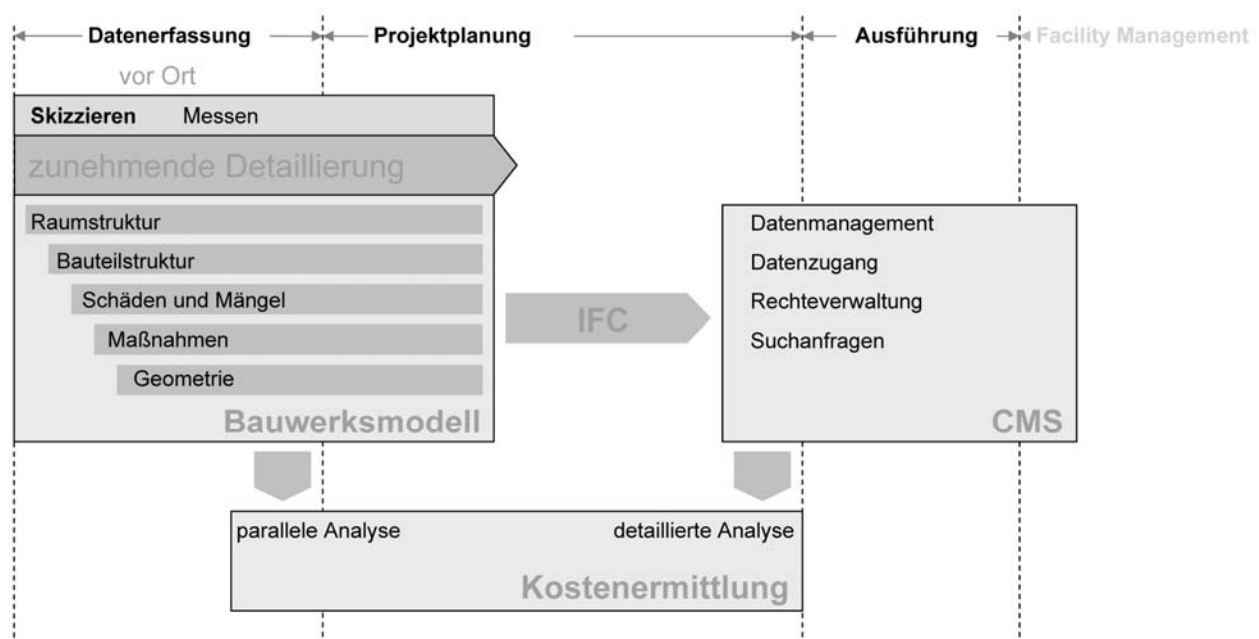


Abb. 4: Systemkonzept und dessen Teilmodule

Die für diese Auswertung notwendigen Daten werden vor Ort computergestützt erfasst und in einem Bauwerksmodell abgelegt. Diese Ersterfassung soll dabei schnell und effektiv unter Verwendung einer skizzenhaften Eingabe und darauf aufbauender Interpretation der Daten erfolgen. Zu Erfassen sind insbesondere die Raum- und Bauteilstruktur des Gebäudes sowie Bauteilzustände und sichtbare Mängel. Anhand dieser Daten können durch den Nutzer Revitalisierungsmaßnahmen zugewiesen werden, welche letztendlich zu Kosten führen.

Nach Abschluss der Ersterfassung und ersten Kostenermittlung sollen die aufgenommenen Daten für die folgenden Planungsprozesse weiterhin zur Verfügung stehen. Hierzu wird das Bauwerksmodell IFC-kompatibel aufgebaut und in entsprechende IFC-Dateien exportiert. Diese IFC-Daten werden anschließend in ein web-basiertes Content Management System (CMS) eingelesen und somit allen an der Planung Beteiligten zur Verfügung gestellt. Das CMS erlaubt gezielte Suchanfragen auf Teile des Bauwerksmodells unter bestimmten architektonische oder technischen Kriterien in Kombination mit Benutzerzugriffsrechten. Auch Projekt übergreifende Recherchen in mehreren Dateien können über das CMS realisiert werden. Abb. 4 verdeutlicht schematisch das Systemkonzept und dessen Teilmodule.

3. Kosten-Risiko-Analyse

3.1. Stand der Forschung und Praxis

In dem speziellen fachübergreifenden Kontext dieses Forschungsvorhabens gibt es kaum relevante Veröffentlichungen, die als Basis des zu konzipierenden Systems herangezogen werden konnten. Selbstverständlich gibt es jedoch für einzelne Teilaspekte Lösungsansätze, die nachfolgend genauer beleuchtet werden sollen. Die drei wichtigsten angeschnittenen Fachgebiete bei der Kosten-Risiko-Analyse sind Kostenermittlungen im Hochbau, Risikomanagement und Stochastik als Teilgebiet der Mathematik.

3.1.1. Kostenermittlung

Grundsätzlich wurden für die Kostenermittlungen die Regelungen der [HOAI08] sowie der [DIN276] und alle weiteren Normen und Verordnungen in ihrer jeweils gültigen Fassung zu Grunde gelegt. Auch andere Standardwerke zur Kostenermittlung wie die umfangreichen Baupreisdatenbanken von [BKI08], [Schmitz08], [Sirados08] wurden direkt im System angewandt und bilden die Grundlage der erforderlichen Beispieldaten.

Das systematische Vorgehen sowie die Analyse von vorhandenen Schwierigkeiten bei den Kostenermittlungen werden u. a. bei [Schach01] und [Kraft02] beschrieben. Für die Analyse und Beschreibung von Projektabläufen und erforderlichen Genauigkeiten / Informationen wurde [Schulte02] herangezogen.

Aus den ausgewerteten Unterlagen sowie aus der Erfahrung der beteiligten Partner aus der Praxis ergab sich, dass sowohl die systematischen Ansätze als auch die Datensammlungen zu einem Großteil auf den Neubau von Objekten ausgerichtet sind. Dabei wird für jedes geplante Bauteil nach dem aktuellen Stand der Detaillierung ein Vergleichswert der kompletten Herstellungskosten zugeordnet. Schwankungen ergeben sich dabei vor allem aus den Qualitäten bzw. aus den unterschiedlichen Konstruktionsarten, wobei für das gesamte Objekt weitestgehend homogene Strukturen zu untersuchen sind.

Bei einer Sanierung ist dagegen der Umgang mit der meist deutlich komplexeren vorhandenen Bebauung notwendig, die Planung muss sich an bestehenden Elementen ausrichten. Kosten entstehen daher nicht nur durch funktionale und gestalterische Absichten, sondern auch durch die Behebung von Mängeln und Schäden sowie durch die Anpassung älterer Bauteile und Konstruktionen an die heutigen statischen, technischen, funktionalen und ästhetischen Ansprüche. Vor allem zu Beginn des Projektes macht sich ein geringer Kenntnisstand über die bestehende Struktur und den Zustand des Objektes bemerkbar. Dies erhöht durch fehlende Informationen oder Annahmen die Unsicherheit bei Entscheidungen und führt häufig zu falschen Prognosen und fehlgeschlagenen Projekten [Fehlhaber05].

Im Gegensatz zu den Neubauprojekten, bei denen heutzutage durch konsequente bauteilorientierte Planung bereits detaillierte Kostenermittlungen aus dem Planungssystem (z.B. Nemetschek Allplan BCM) herausgeschrieben werden können, müssen diese Aufgaben beim Bauen im Bestand zu einem großen Teil händisch durchgeführt werden. Es können zwar bei einer 3D-Planung ebenfalls die notwendigen Mengen aus dem Planungssystem entnommen werden, aber der Anteil der Projekte mit einer 3D-Planung ist im Vergleich zum Neubau äußerst gering. Die Zuordnung von Informationen bei bestimmten Bauteilen und den

daraus resultierenden erforderlichen Maßnahmen (mit ihren zugehörigen Kosten) müssen noch immer per Hand durchgeführt werden. Daraus resultiert vor allem in frühen Planungsphasen entweder ein sehr hoher Arbeitsaufwand oder als Folge dessen eine nicht praxisgerechte Vereinheitlichung von heterogenen Strukturen. Mögliche Abweichungen zwischen den Kostenermittlungen und den letztendlich tatsächlich anfallenden Kosten sind daher deutlich höher als die beim Neubau üblichen bzw. juristisch vertretbaren Werte (s. Tab. 1)

Leistungsphase HOAI	Stufen der Kostenermittlung	Schwankung
1. Grundlagenermittlung	Kostenrahmen	± 30%
2. Vorplanung	Kostenschätzung	± 20 %
3. Entwurfsplanung	Kostenberechnung	± 10 %
4. ...		

Tab. 1: Vertretbare Schwankungen bei Kostenermittlungen im Neubau

3.1.2. Risikomanagement

Obwohl das Bauen an sich und vor allem das Bauen im Bestand zu den risikoreichsten Investitionen überhaupt gezählt werden, wird ein richtiges Risikomanagement nur selten durchgeführt. Trotz der hohen Investitionskosten und der vielen Probleme, die sich in einem Projekt ergeben können, werden bei Bauprojekten weiterhin viele Entscheidungen auf Grundlage von vagen Informationen oder aus dem Bauch heraus getroffen.

Der bewusste und strategische Umgang mit Risiken in der Gesellschaft und vor allem der Wirtschaft setzt sich immer mehr durch, eine Einführung von ausgewählten Risikomanagementverfahren beschränkt sich jedoch bei Bauprojekten hauptsächlich auf den Bereich der Projektentwicklungen. Auch in der Literatur sind zu dem Themenbereich Risiko und Bauprojekte kaum relevante Veröffentlichungen zu finden [Fehlhaber05]. Die beste Ausarbeitung aus betriebswirtschaftlicher Sicht bietet [Gondring07], welcher jedoch hauptsächlich auf die Verteilung von Risiken bei verschiedenen Objekten und nicht auf die Steuerung bei einzelnen Projekten eingeht.

In diesem Forschungsprojekt sind daher die Grundlagen des Risikomanagements auf die Prozesse eines Bauprojektes zu übertragen. In vergleichbaren Fachbereichen wurde das bereits mehrfach durchgeführt, wodurch hier auf zumindest ansatzweise übertragbare Lösungen zurückgegriffen werden kann. Insbesondere sind hier zu nennen die Diplomarbeit von [Schmuck02] sowie der Artikel von [Gleißner99] zum Bereich Versicherungswirtschaft.

Der grundlegende Begriff des Fachbereiches ist das „Risiko“, welches [Biffel04] als rein abstrakte Vorstellung eines möglichen künftigen Ereignisses bezeichnet wird und zu unerwünschten Folgen führen kann. Aus dieser Definition ergibt sich, dass ein Risiko erst zu einem Problem wird, wenn es sich materialisiert. Der Ursprung des Wortes „Risiko“ kommt aus dem lateinischen „risicare“ und bedeutet „Klippen umschiffen“, woran man sehr gut verdeutlichen kann, dass nur bekannte Risiken sinnvoll umgangen werden können [Fehlhaber05].

Risikomanagement ist nicht dazu da, Risiken auszuschließen, sondern alle Risiken zu erkennen, zu bewerten und tragbare Risiken einzugehen, wodurch sich zwangsläufig eine

Chance ergibt, welche als das direkte Gegenteil zum Risiko betrachtet wird. Risikomanagement beschäftigt sich demnach mit den Ursachen von Risiken und mit deren strategischen Umgang und zwar Projekt begleitend bis diese entweder eintreten oder entfallen. Dieser Vorgehensweise steht in der gängigen Praxis das Krisenmanagement diametral gegenüber, welches sich erst nach dem Eintritt eines Risikos mit dessen Auswirkung, dem entstehenden Schaden, befasst.

3.1.3. Stochastik

Da auf Grund der am Projektanfang häufig fehlenden und ungenauen Daten viele Unsicherheiten im System berücksichtigt werden müssen, wurden viele Ansätze der Wahrscheinlichkeitsrechnung und der Statistik untersucht. Hierbei konnten sowohl die im vorangegangenen Kapitel erwähnten Arbeiten zur angewandten Stochastik bei der Aggregation von Risiken dienen als auch die Skripte von C. J. Luchsinger ([Luchsinger02a], [Luchsinger02b]) und [Werner03]. Speziell zur Anwendung von Monte-Carlo-Simulationen gibt es eine ganze Reihe weiterer Literatur, die sich jedoch hauptsächlich mit speziellen Varianten der Simulation oder mit der Einführung in anderen Fachbereichen beschäftigt.

Einen guten Überblick hierzu bietet u. a. [Frey01]. Die theoretischen Grundlagen dieser deterministischen Analysemethoden sind demnach sehr umfassend beschrieben und die Anwendung im Bereich des Risikomanagements hat sich in Verbindung mit den steigenden Rechenleistungen durchgesetzt. Speziell im Hinblick auf die in der Praxis der Kostenermittlung bereits analog durchgeführten Variantenuntersuchungen ergibt sich hier ein großes Potential für Arbeitserleichterungen auf der einen Seite und eine größere Flexibilität mit höherer Genauigkeit auf der anderen Seite.

Neben der Monte-Carlo-Simulation wurde auch auf andere stochastische Methoden und Berechnungen zurückgegriffen, die jedoch allgemein gebräuchlich sind und deswegen an dieser Stelle nicht weiter beschrieben werden.

3.2. Neue Ansätze der Kosten-Risiko-Analyse

Bereits seit den 20ern des letzten Jahrhunderts gibt es ein starkes Bestreben zur Vereinheitlichung bzw. Normung der Bauteile und der Prozesse am Bau. In den 30ern entstanden daraufhin die Normen DIN 276 und DIN 277 und wurden seitdem mehrfach überarbeitet. Sie bieten in der aktuellen Version eine äußerst praxistaugliche Struktur zur funktionalen Gruppierung und Bewertung von Bauteilen bzw. Flächen und Rauminhalten. In der DIN 276 von 11-2006 wurden neben den bekannten Methoden zur Kostenermittlung (LP 2 – Kostenschätzung, LP 3 – Kostenberechnung, LP 5-7 – Kostenanschlag und LP 8 – Kostenfeststellung) zusätzliche Anforderungen an den Planer aufgenommen. So muss dieser sich neben einem Kostenrahmen als Kostenermittlung in der Leistungsphase 1 jetzt auch mit Kostenvorgaben des Auftraggebers und Kostenrisiken auseinandersetzen. An diesen Normen orientiert sich der grundlegende Aufbau des Gebäudemodells für das konzipierte System, u. a. da nur so die bereits vorhandenen Vergleichsdaten genutzt werden können.

Die Vergleichsdaten beruhen bei Neubauprojekten auf der kompletten Errichtung eines bestimmten Bauteils oder einer gesamten Bauteilgruppe. Bei Sanierungsvorhaben sind aber die verschiedenen Fälle zu berücksichtigen, dass an einem bestehenden Bauteil etwas entfernt, ergänzt, angepasst oder ausgetauscht werden muss. Demnach kann beispielsweise das Vorhandensein einer definierten Wand im Planungs-SOLL aber sowohl bedeuten, dass

gar keine Maßnahmen an ihr auszuführen sind (weil sie in genau dem geplanten Zustand schon existiert), dass kleinere oder größere Änderungen an ihr vorzunehmen sind (weil sie dem geplanten Zustand nicht exakt entspricht) oder dass sie komplett neu hergestellt werden muss (weil sie gar nicht existiert oder nicht wieder verwendbar ist). Bis diese unterschiedlichen Varianten jedoch nicht zweifelsfrei geklärt werden können, ist die Zuordnung nur eines Vergleichswertes zu dem Bauteil nicht möglich bzw. nicht richtig. Die Festlegung auf eine bestimmte Ausführungsvariante kann nicht ohne detaillierte Kenntnis des IST-Zustandes eines Bauteils erfolgen.

Welche Variante mit welchem Vergleichswert am Ende die richtige ist, hängt von vielen objektiven (z.B. Ort, Zeit, Funktion, Zustand, Normen) und auch einigen subjektiven (z.B. Nutzerwünsche, Ästhetik, Qualität, Auftragslage) Faktoren ab, welche soweit möglich in den Daten des Bauteils bzw. des Projektes mit abgebildet werden müssen. Dabei sollte aus Gründen der Transparenz immer der Grad der Genauigkeit und der Gewissheit der Daten sowie der bearbeitenden Personen mit abgebildet werden.

Die Aufteilung der Investitionskosten ist mit den Kostengruppen der DIN 276 vorgegeben und die drei großen Aufgaben des Forschungsprojektes waren demnach:

1. Wie können Bauteile so strukturiert aufgenommen werden, dass auch bei unterschiedlichen Detaillierungsstufen die Struktur der DIN 276 im System erhalten bleibt.
2. Wie können unterschiedliche Vergleichswerte an unterschiedlich detaillierte Daten geknüpft werden.
3. Wie können die unterschiedlichen Informationsstände und die unterschiedlichen Detaillierungsstufen in einer einheitlichen Berechnung/Simulation zu einem transparenten und Fehler unanfälligen Ergebnis führen.

Grundlegend wurde eine Reihe von abstrakten Bauteilgruppen eingeführt, die viele verschiedene Daten aufnehmen können aber nicht müssen. So ist zum Beispiel für viele Bauteilgruppen nicht zwingend erforderlich, dass sie eine geometrische Repräsentation besitzen. In den gängigen CAAD-Systemen gibt es im Gegensatz dazu häufig genug geometrische Repräsentationen ohne jegliche für die Kostenermittlung relevante Information.

Des Weiteren werden in Maßnahmengruppen Varianten von unterschiedlichen möglichen Vorgehensweisen beim Übergang vom IST- zum SOLL-Zustand eines Bauteils zusammengefasst und über eine jeweilige vorgegebene oder nutzerabhängige Eintrittswahrscheinlichkeit im Bezug zum Zustand bzw. zu den Planungsabsichten bewertet.

Die Simulation bei der Kostenermittlung berücksichtigt anschließend alle vorhandenen aber auch die nicht vorhandenen Daten und berechnet unter Beachtung der vorhandenen Unsicherheiten viele Projektverläufe, um die Ergebnisse dann am Ende in einer detaillierten Wahrscheinlichkeitsverteilung auszuwerten. Darüber hinaus können weitere Erkenntnisse zu den Bauteilen mit den größten Unsicherheiten und andere Variantenuntersuchungen über eine ähnliche Simulation gewonnen werden.

3.3. Entwicklung von Algorithmen zur Bewertung von Kosten und Risiken

Für die Systemkonzeption ist die Erfassung der Bauteilstruktur als Elemente der Kostengruppen nach DIN 276 vorgesehen. Der Nutzer bewertet bei der Aufnahme den Zustand des Gebäudes (mangelfrei, gut, schlecht, desaströs) und erfasst sichtbare Mängel. Diese Erfassung und Bewertung kann dabei in zunehmender Detaillierung erfolgen – angefangen von der Bewertung des Gesamtgebäudes, über die Bewertung von Bauteilgruppen (z.B. alle Außenwände) bis hin zur detaillierten Beurteilung einzelner Bauteile. Diese zunehmende Detaillierung lässt sich für die Kostenermittlung in 3 Stufen untergliedern (s. Tab. 2).

Stufe 1	Bewertung des gesamten Objektes und Kostenermittlung u.a. durch Vergleichswerte entsprechend der:
	<ul style="list-style-type: none">▪ Aufnahme der Außenmaße und Anzahl der Geschosse des Gebäudes▪ Daten zu Nutzung, Bautyp und Zustand▪ Daten zu Region, Bauzeit und Baunebenkosten
Stufe 2	Ergänzung der vorhandenen Werte durch z.B.:
	<ul style="list-style-type: none">▪ Genaue Angaben zu Grundfläche und Nutzfläche▪ Genaue Angaben zu Typ, Zustand und Größe einzelner Bauteilgruppen▪ Zusätzliche/ersetzende Maßnahmen▪ Ergänzung von einzelnen Mängeln
Stufe 3	Ergänzung der vorhandenen Werte durch z.B.:
	<ul style="list-style-type: none">▪ Genaue Angaben zu einzelnen Bauteilen▪ Genaue Angaben zu Mengen und Maßen▪ Festlegung einzelner Maßnahmen▪ Ergänzung weiterer Mängel

Tab. 2: Stufen der Kostenermittlung bei zunehmender Detaillierung der Gebäudedaten

Die Detaillierung der Daten erfolgt dabei fließend und muss nicht zwangsläufig für das gesamte Gebäude gleichmäßig erfolgen. So kann bspw. ein Gebäudeteil detailliert bis zu den einzelnen Bauteilen erfasst werden (Stufe 3), während andere lediglich in ihrer Außenkubatur und allgemeinem Zustand aufgenommen wurden (Stufe 1).

Anhand der Beurteilung der Zustände und der konkreten Erfassung von Mängeln (z.B. Wasserschaden an Wand mit Angabe der Fläche) kann der Nutzer mögliche Maßnahmen (und damit verbunden die Kosten dieser Maßnahmen) zur Beseitigung des Mangels bzw. Behebung eines schlechten Zustandes zuweisen. Da zum Zeitpunkt der Erfassung das Ausmaß des Mangels bzw. Zustandes nicht präzise aufgenommen werden kann, werden im System die relevanten möglichen Maßnahmen und deren Eintrittswahrscheinlichkeiten (EW) in Maßnahmengruppen abgebildet. Durch die Abbildung verschiedener Maßnahmen ergibt sich eine Vielzahl an möglichen Revitalisierungsvarianten, welche wiederum unterschiedliche Kosten verursachen. Zur Auswertung dieser Kosten und zur Ermittlung der Wahrscheinlichkeit ihres Eintretens, wird auf das stochastische Verfahren der Monte-Carlo-Simulation zurückgegriffen.

Die Funktionsweise der Monte-Carlo-Simulation ist in etwa vergleichbar mit einfachen Variantenuntersuchungen, wobei diese nicht nur auf eine sondern auf viele notwendige Entscheidungen gleichzeitig angewandt wird. Dabei wird ein Zufallsexperiment so oft wiederholt, dass über das Gesetz der großen Zahlen systembedingte Mess- bzw. Rechenfehler auf einen vernachlässigbaren Wert reduziert werden.

Bei nicht eindeutigen Festlegungen (wenn mindestens 2 unterschiedliche Maßnahmenvarianten bei einem Bauteil in Frage kommen) wird in jedem Durchlauf eine der möglichen Varianten über das Verhältnis einer Zufallszahl zur Eintrittswahrscheinlichkeit als richtig angenommen und der zugehörige Wert in der Kostenermittlung berücksichtigt. In jedem Durchlauf werden alle Unsicherheiten berücksichtigt und so über eine große Zahl von Durchläufen alle möglichen Projektverläufe bzw. alle möglichen Ergebnisse für die Kostenermittlungen erreicht. Die Häufigkeit bestimmter Ergebnisbereiche bietet anschließend Aufschluss über die wahrscheinlichsten, möglichen oder unmöglichen Projektergebnisse.

Da das Rechensystem mit einer ausreichend hohen Anzahl an Durchläufen eine minimale Fehlerquote hat (in Tests bei 1.000.000 Durchläufen kleiner 0,1 %), ist es für den Einsatz bei Kostenermittlungen unter Unsicherheiten gut geeignet. Wichtig ist bei den Berechnungen, dass die zu Grunde liegenden Daten richtig sind. Da alle Ungenauigkeiten/Unsicherheiten (auch Messtoleranzen) berücksichtigt werden, führen lediglich falsche Annahmen oder unpräzise Eingaben zu mangelhaften Ergebnissen. Diese sollen daher mit Unterstützung des Systems weitestgehend verhindert werden.

3.4. Fachwissen und Nutzerdaten als Berechnungsbasis

Unabhängig davon, wie komplex oder einmalig ein Bauprojekt ist, der allergrößte Teil aller notwendigen Arbeiten wurde in identischer oder zumindest vergleichbarer Weise schon einmal bei einem anderen Projekt durchgeführt. Die Nutzung dieses detaillierten Wissens und die richtige Übernahme der vorhandenen Daten auf die aktuelle Aufgabe spielt eine Schlüsselrolle in der erfolgreichen Bearbeitung eines Projektes.

Die Existenz möglicher Vergleichsdaten kann vom System geprüft und eventuelle Treffer als Vorschlag dem Nutzer angezeigt werden. Dabei kann bei typischen Fällen sogar ein vordefinierter Satz von Maßnahmenvarianten mit dem Zustand entsprechenden Eintrittswahrscheinlichkeiten angeboten werden. Letztendlich muss jedoch immer der Nutzer die erforderlichen Festlegungen auf eine spezielle Variante treffen, andernfalls würde in der Simulation die Unsicherheit einer Systemvorgabe mit bewertet werden. So bleibt die Verwaltung der Datenflut dem System vorbehalten, die Entscheidungsgewalt und damit der Einfluss auf Detaillierung und Genauigkeit der Ergebnisse verbleiben beim Nutzer.

Durch gezielte Hinweise auf bekannte Probleme bei bestimmten Konstruktionen (z.B. häufige Mängel, Konflikte mit aktuellen Normen, etc.) könnte der Nutzer bei der Aufnahme eines Objektes bei der Risikoidentifikation unterstützt werden. Diese Berücksichtigung von typischen Risiken bietet sich auch bei der Kostenermittlung an, wenn beispielsweise bei Umnutzungen die Notwendigkeit von Brandschutzertüchtigungen hinzukommt, welche in den ersten Projektphasen selten detailliert untersucht werden. Das Auftreten von Abhängigkeiten sich gegenseitig bedingender Maßnahmen wurde bei der Systemkonzeption, insbesondere beim Entwurf des Bauwerksmodells, diskutiert. Eine detaillierte Bearbeitung im Rahmen des Forschungsprojektes war im vorgegebenen Zeitrahmen allerdings nicht möglich. Diese speziellen Anforderungen und Möglichkeiten sollten jedoch in weiteren Forschungsprojekten genauer untersucht werden.

Innerhalb dieses Projektes wurden die erforderlichen Daten aus den nachfolgenden Standardwerken der Kostenermittlung entnommen:

BKI – aus den Büchern BKI Objektdaten A1 (Stand 2001) sowie BKI Objektdaten A5 (Stand 2007) des Baukosteninformationszentrum Deutscher Architektenkammern wurden Maßnahmen und Vergleichsdaten für die Kostengruppen der 1. Gliederungsebene und 2. Gliederungsebene übernommen.

Schmitz/Krings/Dahlhaus/Meisel – aus den Büchern Baukosten 2004 und 2008 (jeweils der Band für Sanierung) wurden einzelne Maßnahmen und Vergleichsdaten für die Kostengruppen der 3. Gliederungsebene sowie für die Bewertung der gesamten Kostengruppe 700 Baunebenkosten übernommen.

Aufgrund der offenen Struktur möglicher Maßnahmen (Bezug auf spezielle Bauteile oder abstrakte Bauteilgruppen) kann das System beliebig mit neuen Daten gefüllt werden. Dabei ist nicht nur der Import aus anderen Datenbanken (DBD, sirAdos, etc.) möglich, sondern vor allem auch die Übernahme von Daten aus eigenen abgeschlossenen Projekten, solange die Werte auf einen gemeinsamen Indexwert (z.B. Durchschnitt Deutschland 2007) umgerechnet werden können.

4. Datenmodellierung / Bauwerksmodell

4.1. Stand der Forschung und Technik

4.1.1. Bauwerksmodelle und Datenaustausch

Bauwerksmodelle dienen der rechnerinternen Abbildung von Gebäudedaten und bilden eine Untermenge von Produktmodellen. Entsprechend ISO 10303 stellen Produktmodelle eine abstrakte Definition der physischen und funktionalen Aspekte eines Produktes dar. Die Modellierung erfolgt in der Regel nach dem objektorientierten Ansatz in Form von Klassendefinition, (Objekt-)Instanzen und Relationen.

Aktuelle CAAD-Systeme verwenden applikationsspezifische Bauwerksmodelle zur Abbildung der im System erstellten bzw. verarbeiteten Gebäudedaten. In der Regel arbeiten die Systeme Konstruktionsorientiert, das heißt die Strukturierung erfolgt in einzelnen Bauteile und Baugruppen repräsentierenden Objekten. Über Relationen erfolgt eine Zuordnung der Bauteile zu räumlichen Struktur des Gebäudes: Räume, Raumzonen, Geschosse, Gebäude etc.

Ein Austausch von Gebäudemolldaten zwischen verschiedenen Applikationen ist aufgrund unterschiedlicher Bauwerksmodelle in der Regel nicht bzw. nur eingeschränkt möglich. Um dennoch die Daten zwischen den Applikationen austauschen zu können, ist die Verwendung eines neutralen objektorientierten Datenaustauschformates für Produktmolldaten erforderlich. Ein solches Datenaustauschformat für Bauwerksmodelle stellen die Industry Foundation Classes (IFC) dar.

Die IFC werden seit 1994 von der International Alliance of Interoperability (IAI), einem Zusammenschluss aus mittlerweile über 600 Firmen und Forschungseinrichtungen der Bauindustrie, als offenes Austauschformat für das Bauwesen entwickelt. Die erste Fassung der IFC (Release 1) erschien 1997, die aktuelle Version IFC2x3 ist seit 2007 verfügbar. Die IFC2x4 befindet sich als alpha-Version derzeit im Test. [IAI09_1], [IAI09_2]

Die IFC sind als internationaler Standard (ISO16739) registriert und werden zunehmend durch zahlreiche CAAD-Applikationen unterstützt.

4.1.2. Dynamische Bauwerksmodelle

Ein Problem bei der Modellierung von Bauwerksdaten stellt der meist unikate Charakter von Gebäuden dar. Es ist kaum möglich ein allgemeingültiges Bauwerksmodell zu definieren, um alle möglichen Aspekte, Elemente und Methoden vorzudenken. Gerade im Altbaubereich ist dies aufgrund verschiedener Gebäudetypen und Bauzeiten sowie unterschiedlicher Ausprägung von Bauteilen und Baugruppen problematisch.

Zur Lösung dieser Problematik wurden im Rahmen der Forschung dynamische Ansätze entwickelt, welche es erlauben, Bauwerksmodelle zur Laufzeit anzupassen und zu erweitern. Entsprechende Ansätze sind u.a. in [Hauschild03], [Willenbacher02], [Willenbacher04], [SFB524_05b] und [SFB524_05c] zu finden.

Im Rahmen der Konzeptentwicklung wie auch der Implementierung der Prototypen wurden diese Ansätze insofern berücksichtigt, dass sich die Systemkonzeption als Teilmodell (bzw. als Verbund von Teilmodellen, siehe Kapitel 4.2) in ein solches dynamisches

Bauwerksmodell integriert, welches zur Zeit im Rahmen weiterer Forschung an der Professur „Informatik in der Architektur“ der Bauhaus-Universität Weimar weiterentwickelt wird [Thurow08].

4.2. Teilmodelle

Wie bereits erwähnt, integriert sich das entwickelte Systemkonzept in den Ansatz eines dynamischen Bauwerksmodells, welcher im Rahmen weiterer Forschung entwickelt wurde.

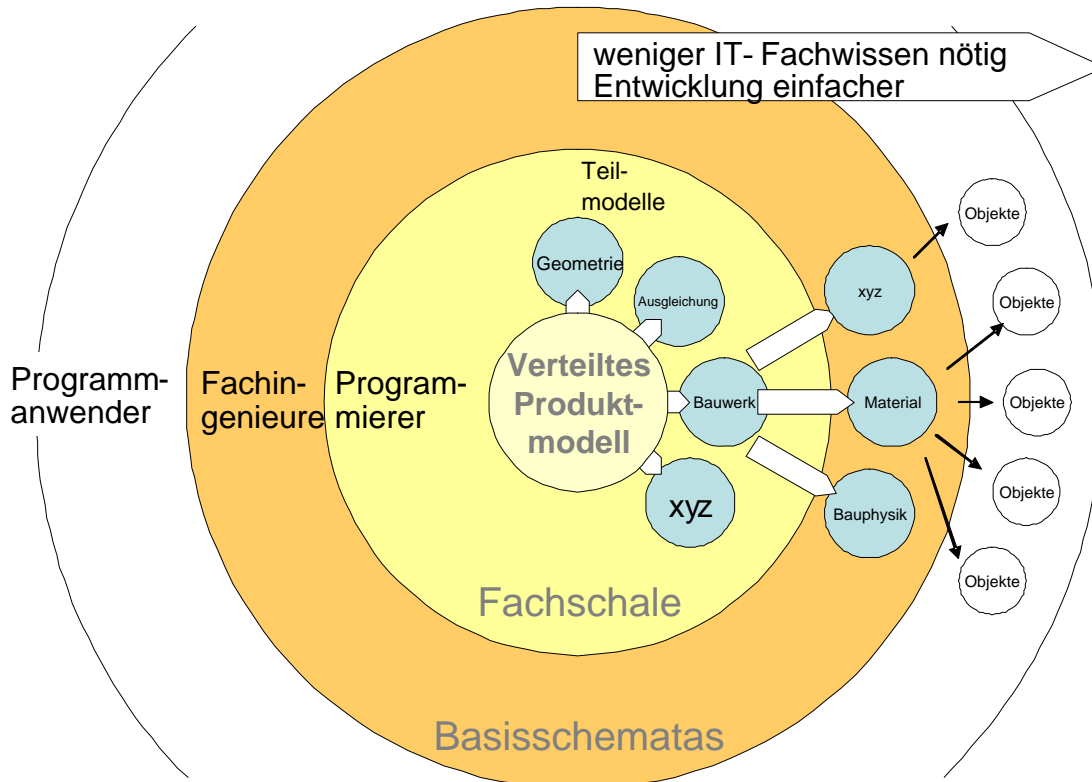


Abb. 5: Aufbau des dynamischen Bauwerksmodells

Das applikationsspezifische Bauwerksmodell ist dabei aus mehreren Teilmodellen aufgebaut, welche miteinander verzahnt sind, d.h. die Objekte unterschiedlicher Modelle können gegenseitig referenziert werden. Das dynamische Konzept erlaubt dabei sowohl die Integration neuer Teilmodelle, als auch die Erweiterung und Anpassung bestehender auf unterschiedlichen Manipulationsebenen zur Laufzeit. zeigt den prinzipiellen Aufbau dieses Konzeptes. Detaillierte Informationen finden sich in [Thurow08]³

Für das Systemkonzept sind nachfolgend erläuterte Teilmodelle vorgesehen (Kap. 4.2.1 bis Kap. 4.2.5), welche gemeinsam das applikationsspezifische Bauwerksmodell bilden.

³ Die Ausarbeitung und Implementierung dieses dynamischen Ansatzes befinden sich derzeit in Arbeit. Für die Umsetzung der Prototypen wurde daher zum Teil auf statische Strukturen zurückgegriffen. Konzeptionell lässt sich aber das entwickelte System in den dynamischen Ansatz integrieren.

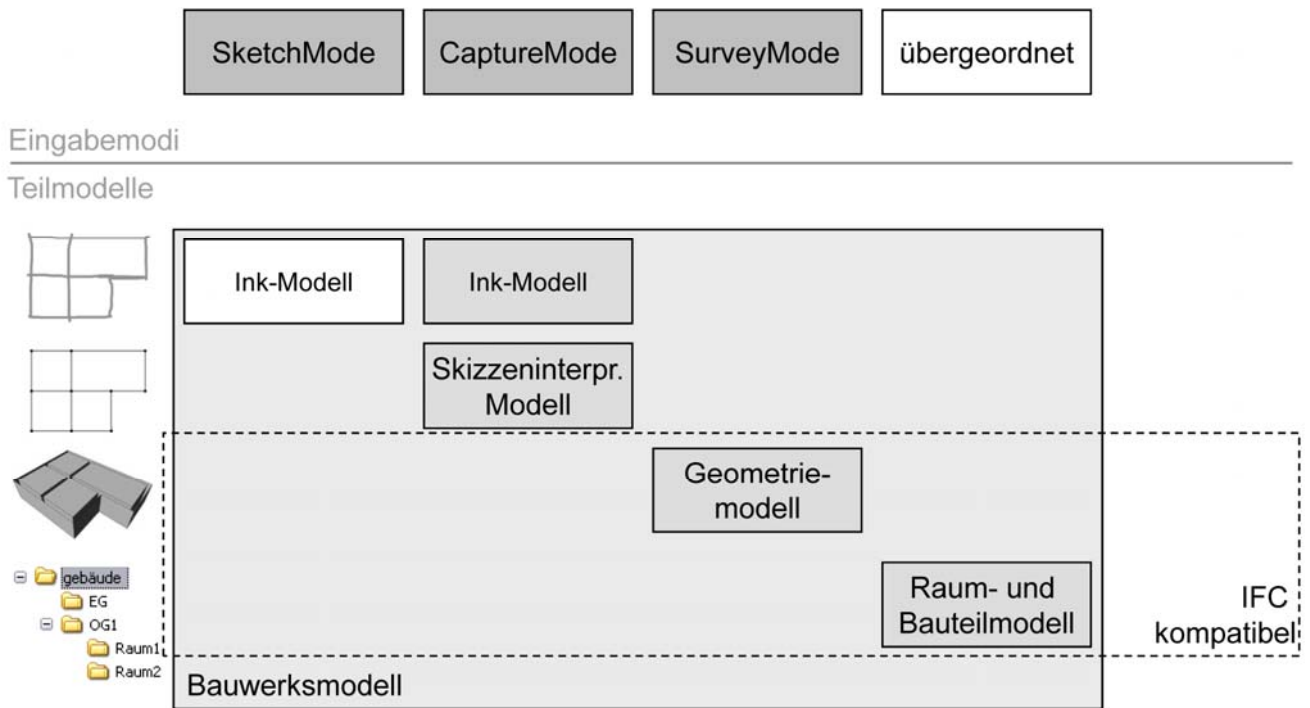
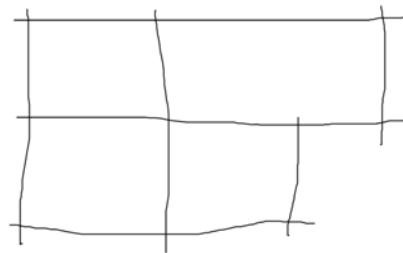


Abb. 6: Das Bauwerksmodell als Verbund von Teilmodellen und deren Verwendung in den unterschiedlichen Eingabemodi (siehe Kap. 5.3)

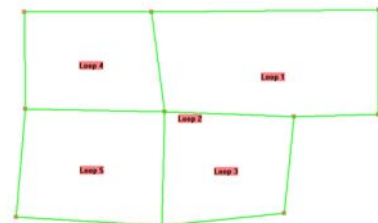
4.2.1. Ink-Modell (IM)

Das Ink-Modell dient der Eingabe und Darstellung sowie Manipulation interpretierbarer Skizzen (Sketch-Mode) unter Verwendung digitaler Tinte. Die einzelnen Strokes werden miteinander verschnitten. Es entstehen separate Teilstriche mit gemeinsamem Schnittpunkt.



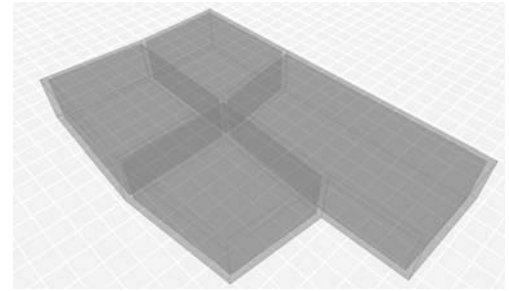
4.2.2. Skizzeninterpretationsmodell (SIM)

Das Skizzeninterpretationsmodell entsteht als interpretiertes Ergebnis während der Stifteingabe (Ink-Modell). Es dient der vereinfachten Darstellung von Elementen des Oberflächen- sowie Raum- und Bauteilmodells als Menge von Knoten und Kanten im 2D-Raum. Eine Kante verbindet zwei Knoten, ist einem Bauteil (*BuildingElementClass* im Raum- und Bauteilmodell) zugeordnet und besitzt zwei Seiten. Jede Seite verweist auf einen Raum (*Space* im Raum- und Bauteilmodell) und auf eine Oberfläche (*Polygon* im Oberflächenmodell). Öffnungen sowie Mängel (*Door*, *Window* und *Defect* im Raum- und Bauteilmodell) verweisen ebenso auf die Kante mit Angabe der Position relativ zum Anfangspunkt.



4.2.3. Geometriemodell (GM)

Das Geometriemodell dient der Darstellung und Manipulation der resultierenden 3D-Geometrie aus der Interpretation der digitalen Tinte, sowie der Repräsentation von Bauteilen und Räumen. Das Modell bildet die sichtbaren Oberflächen des Gebäudes ab. Als Grundgeometrien sind Punkte, Linien und Polygone vorgesehen, wobei Linien und Polygone jeweils auf Punkte verweisen und somit von diesen abhängig sind.



4.2.4. Aufmaßmodell (AM)

Das Aufmaßmodell dient der Abbildung unterschiedlichster Messungen und geometrischen Abhängigkeiten bzw. Restriktionen (z.B. Rechtwinkligkeit, Parallelität, Symmetrien), mit welchen die Maßlichkeit des Geometriemodells schrittweise der Realität angepasst wird. Kern des Modells ist ein geodätischer Ausgleichungsansatz. Die Konzipierung dieses Modells war nicht Bestandteil des Forschungsvorhabens, vielmehr wurde auf bereits abgeschlossene Arbeiten zurückgegriffen ([SFB524_05b], [Thurow04]).

4.2.5. Raum- und Bauteilmodell (RBM)

Das Raum- und Bauteilmodell dient der Abbildung aller für die Auswertung relevanter Daten zu Bauteilen, zur räumlichen Struktur, zu Mängeln und Maßnahmen. Der Aufbau des Modells orientiert sich zum einen an den Vorgaben der Kosten- und Risikoanalyse und zum anderen an der IFC-Spezifikation, um einen möglichst einfachen Datenexport zu realisieren. Die geometrische Repräsentation einzelner Objekte (z.B. Bauteile) erfolgt über Verweise zu Objekten im Geometriemodell (3D Repräsentation über Oberflächen) und im Skizzeninterpretationsmodell bzw. Ink-Modell (skizzenhafte Repräsentation bei der Eingabe/Erfassung).

4.3. Grundlegende Konzepte des Raum- und Bauteilmodells

Kern der Datenhaltung bildet das Raum- und Bauteilmodell. Der Aufbau der Klassenstruktur des Modells ist im Wesentlichen durch die Anforderungen aus der Kosten-Risiko-Analyse geprägt, sowie der IFC-Spezifikation, um einen möglichst einfachen Datenaustausch zu gewährleisten.

Im Folgenden sind grundlegende Konzepte des Modells beschrieben. Einen vollständigen Überblick über die vorgesehenen Klassen findet sich im Anhang C.

4.3.1. Basisklasse (BaseClass)

Das Konzept der Basisklasse entspricht weitestgehend dem der Klasse *IfcRoot* innerhalb der IFC-Spezifikation. Die Basisklasse bildet den Ausgangspunkt für weitere Untersetzungen in die Bauteilstruktur, die Raumstruktur, Mängel und Maßnahmen etc. in der Vererbungshierarchie. Die Basisklasse sieht die Relation zu einem *OwnerHistory* Objekt vor, welches Informationen zum Erstellungszeitpunkt, zum Zeitpunkt der letzten Änderung, zum Bearbeiter bei Erstellung und zum Bearbeiter der letzten Änderung vorsieht. Dieses Konzept wurde aus den IFC übernommen. Im Rahmen der Kostenauswertung bietet es den Vorteil,

dass für jedes neu erstellte und geänderte Objekt der Bearbeiter (*Person*) nachvollzogen werden kann. Die Klasse *Person* wiederum verfügt über das Attribut *competence* (Enumeration: Laie, Student, Planer, Sachverständiger, nutzerdefiniert). Auf diese Weise wird die Nachvollziehbarkeit und Qualität der eingegebenen Daten sichergestellt, welche in die Kosten-Risiko-Analyse mit einfließen.

Innerhalb des Systemkonzepts erfolgt beim Starten eines Projektes eine Anmeldeprozedur für den aktuellen Nutzer, um die Zuordnung sicherzustellen.

4.3.2. Gebäudeklasse (*BuildingClass*)

Die Gebäudeklasse bildet die abstrakte Superklasse für alle Elemente eines Gebäudes⁴. Die Klasse wird von *GeometryClass* abgeleitet, welche die Verknüpfung zum Geometriemodell herstellt, d.h. sämtliche Instanzen aller abgeleiteten Klassen können geometrisch repräsentiert werden. Über das Attribut *condition* (Enumeration: mangelfrei, gut, schlecht, desaströs) wird der aktuelle Zustand eines Elementes abgebildet. Das Attribut *conditionInherit* legt fest, ob der Zustand für das Objekt selbst gesetzt wurde, oder aus einem übergeordneten Objekt übernommen ist. Welches Objekt als übergeordnet betrachtet wird, ist auf Ebene der abgeleiteten Klassen definiert, zum Beispiel:

Für ein Gebäude wurde global der Zustand mit gut bewertet. Beim Anlegen neuer Geschosse wird dieser Zustand zunächst übernommen (vererbt, *conditionInherit* = *true*). Der Nutzer kann anschließend den Zustand des Geschosses manuell ändern, falls notwendig (*conditionInherit*=*false*).

Über die Methode *getQuantity(type :string)* kann die Menge eines Elementes abgefragt werden. Der Übergabeparameter *type* (Art der abzufragenden Menge) bestimmt dabei, welche Menge tatsächlich relevant ist. Die Definition über Art und Berechnung der Menge erfolgt auf Ebene der abgeleiteten Klassen, zum Beispiel:

Für eine Wand können u.a. die Typen „Außenwandfläche“, „Innenwandfläche“, „Konstruktionsgrundfläche“ etc. für die Kostenberechnung abgefragt werden. Für ein Gebäude z.B. „Bruttogrundfläche“, „Bruttorauminhalt“ etc.

Die Menge wird dabei entweder aus den Attributen des Objektes (bzw. aus den Attributen verknüpfter Objekte) berechnet, oder direkt aus dem Geometriemodell abgeleitet (sofern eine geometrische Repräsentation vorliegt). Der Nutzer hat immer die Möglichkeit bestimmte Größen manuell einzugeben, oder eine Messung im Geometriemodell durchzuführen. Innerhalb der Definition der abgeleiteten Klassen wird für jedes numerische Attribut, das eine Größe darstellt (z.B. *buildingHeight* in Klasse *Building*), ein Attribut vorgesehen, welches die Herkunft des Attributwertes angibt (z.B. *buildingHeightOrigin* als Enumeration: nutzerdefiniert, errechnet, aus Modell).

4.3.3. Raum- und Gebäudestruktur (*SpatialStructure*)

Die Abbildung der Raum- und Gebäudestruktur erfolgt über Ableitungen der abstrakten Klasse *SpatialStructureClass*. Als Untersetzungen folgen Grundstück (*Site*), Gebäude (*Building*), Geschoss (*BuildingStorey*) und Raum (*Space*). Die räumliche Gliederung erfolgt über Relationen zwischen den einzelnen Klassen, Ausgangspunkt bildet die Klasse *Project*:

⁴ Als Element eines Gebäudes sind sowohl physische Objekte, wie Bauteile, als auch Objekte zur räumlichen Gliederung, wie Geschosse und Räume, gemeint.

Project → hasSites → **Site** → hasBuildings → **Building** → hasStoreys → **BuildingStorey** → hasSpaces → **Space**

Die Verknüpfung zur Bauteilstruktur wird über die Relation *hasBuildingElements* (definiert in *SpatialStructureClass*) und der umgekehrten Relation *containedInSpatialStructure* (definiert in *BuildingElementClass*) realisiert:

SpatialStructureClass → hasBuildingElements → **BuildingElementClass**

BuildingElementClass → containedInSpatialStructure → **SpatialStructureClass**

4.3.4. Bauteilstruktur (BuildingElementClass)

Die Abbildung einzelner Bauteile sowie zusammengesetzter Bauteilgruppen erfolgt mittels abgeleiteter Klassen von *BuildingElementClass*. Über die Methode *getCostCategory()* wird die Kostengruppe des Objektes entsprechend DIN276 zurückgegeben. Um eine zunehmende Detaillierung der Bauteile zu ermöglichen, wurde auf das Konzept der Composition und Decomposition zurückgegriffen, wie sie in ähnlicher Form in der IFC-Spezifikation definiert ist. Über das Attribut *isComplex* (Enumeration: complex, element, partial) ist definiert, ob es sich bei dem Objekt um ein einzelnes Bauteil handelt (=element), eine Bauteilgruppe aus mehreren Bauteilen bildet (=complex) oder Teil einer Bauteilgruppe ist (=partial). Handelt es sich um eine Bauteilgruppe oder den Teil einer Gruppe, erfolgt die Relation zu den entsprechenden Objekten mittels *isDecomposedBy* bzw. *Decomposes*. Beispiel:

Die bei der Interpretation des Ink-Modells erzeugten Kanten im Skizzeninterpretationsmodell werden mit Objekten der Klasse Wand (*Wall*) verknüpft, ohne dass eine detaillierte Information über Wandkonstruktion oder Wandbekleidung vorliegt. Handelt es sich um eine Innenwand, wird die Kostengruppe 340 (*getCostCategory()* = 340, *isComplex* = element) zurückgegeben. Mit zunehmender Detaillierung kann die Konstruktion präzisiert werden, d.h. die Wand wird nun als Baugruppe bestehend aus Wandkonstruktion (*Wall*) und Bekleidungen (*Covering*) betrachtet (*isComplex*=complex) und mit den entsprechenden Teilen in Relation gesetzt (Abb. 7).

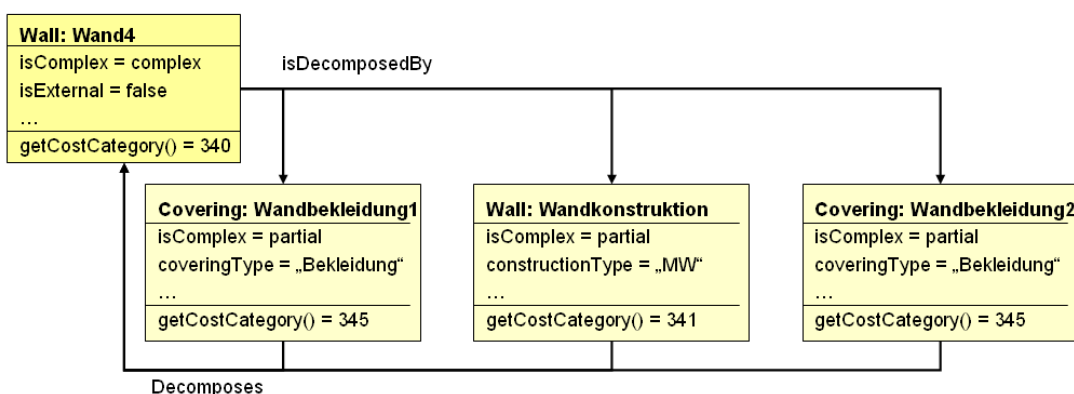


Abb. 7: Prinzip der Decomposition von Bauteilen (Baugruppen) in einzelne Bestandteile

4.3.5. Mängel (Defect)

Mängel werden als eigene Objekte im Modell abgebildet, welche in Relation zu Elementen der Bauteilstruktur und der Raum- und Gebäudestruktur stehen. Über das Attribut *type* wird unterschieden, um welche Art von Mangel es sich handelt (Enumeration: funktional,

konstruktiv oder ästhetisch). Als abgeleitete Klasse von *GeometryClass*, können Mängel geometrisch repräsentiert werden. Hierzu sind Spezialisierungen der Klasse als punktförmiger, linienförmiger und flächenförmiger Mangel vorgesehen, entsprechend der Definition eines Bauschadens nach DIN1356-6 [DIN1356]. Zur Kostenermittlung von Maßnahmen zur Behebung eines spezifischen Mangels, kann dieser über die Methode *getQuantity((type :string))* nach seiner Menge abgefragt werden.

4.3.6. Maßnahmen (Task)

Die Klasse *Task* dient der Abbildung sowohl von geplanten Maßnahmen (*plannedTask*), als auch von abgeschlossenen Maßnahmen (*finishedTask*). Über die Klasse *TaskGroup* können mögliche geplante Maßnahmen mit einem Bauteil oder einem Mangel in Relation gesetzt werden. Auf diese Weise werden die verschiedenen Maßnahmenvarianten abgebildet, welche über die Monte-Carlo-Simulation auszuwerten sind. Hierzu sieht die Klasse *plannedTask* das Attribut *occurrenceProbability* (Eintrittswahrscheinlichkeit) vor. Mögliche anwendbare Maßnahmen werden aus einer externen Datenbank abgefragt (siehe Kapitel 3.4). Die Definition der Eintrittswahrscheinlichkeit einer Maßnahme erfolgt entweder vom Nutzer, oder kann ebenfalls in der Datenbank hinterlegt werden (bspw. kann nach Abschluss mehrerer Projekte die Häufigkeit der tatsächlich ausgeführten Maßnahmen ermittelt werden, um entsprechende Eintrittswahrscheinlichkeiten bereits vorzugeben). Zur Unterscheidung der Herkunft der Daten, wird dies im Attribut *occurrenceProbabilityOrigin* (Enumeration: Datenbank, errechnet, nutzerdefiniert) hinterlegt.

Über die Klasse *finishedTask* können die tatsächlich ausgeführten und abgeschlossenen Maßnahmen abgebildet werden. Diese dient zum einen der Dokumentation von in der Vergangenheit durchgeführten Arbeiten bei der Datenerfassung, und zum anderen der Dokumentation der tatsächlich ausgeführten Maßnahmen nach Projektabschluss.

4.4. IFC-Kompatibilität

Eine wesentliche Anforderung des Projektes stellt die Weiterverwendbarkeit der Daten in folgenden Planungsprozessen und anderen Planungswerkzeugen dar. Um dies zu erreichen ist eine Übertragung der aufgenommenen Daten in das IFC-Datenformat vorgesehen (siehe Abb. 6). Bei der Entwicklung des Raum- und Bauteilmodells wurde daher auf eine weitestgehende Kompatibilität mit der IFC-Spezifikation geachtet:

4.4.1. Raum- und Bauteilstruktur

In den IFC wird die Raumstruktur anhand abgeleiteter Klassen von *IfcSpatialStructureElement* abgebildet und die Bauteilstruktur durch die abstrakte Klasse *IfcBuildingElement* und entsprechende Spezialisierungen. Die Relation zwischen Bauteilstruktur und Raumstruktur wird mittels *IfcRelContainInSpatialStructure* und *IfcRelSpaceBoundary* beschrieben (Abb. 8).

Innerhalb der Klassendefinition des Raum- und Bauteilmodells sind spezifische Attribute zur Beschreibung von Objekteigenschaften vorgesehen (z.B. *condition* zur Abbildung des allgemeinen Zustandes eines Elementes). Zur Abbildung von Eigenschaften stellen die IFC die Funktionalität der Eigenschaftssätze (*IfcPropertySetDefinition*) zur Verfügung. Innerhalb dieser können zusätzliche, nicht durch die IFC vordefinierte Eigenschaften für alle Klassen die von *IfcObject* abgeleitet sind, definiert werden. Auf diese Weise ist es möglich für die

spezifischen Attribute des Raum- und Bauteilmodells, welche in die IFC übertragen werden sollen, eigene objektspezifische Eigenschaftssätze zu definieren.

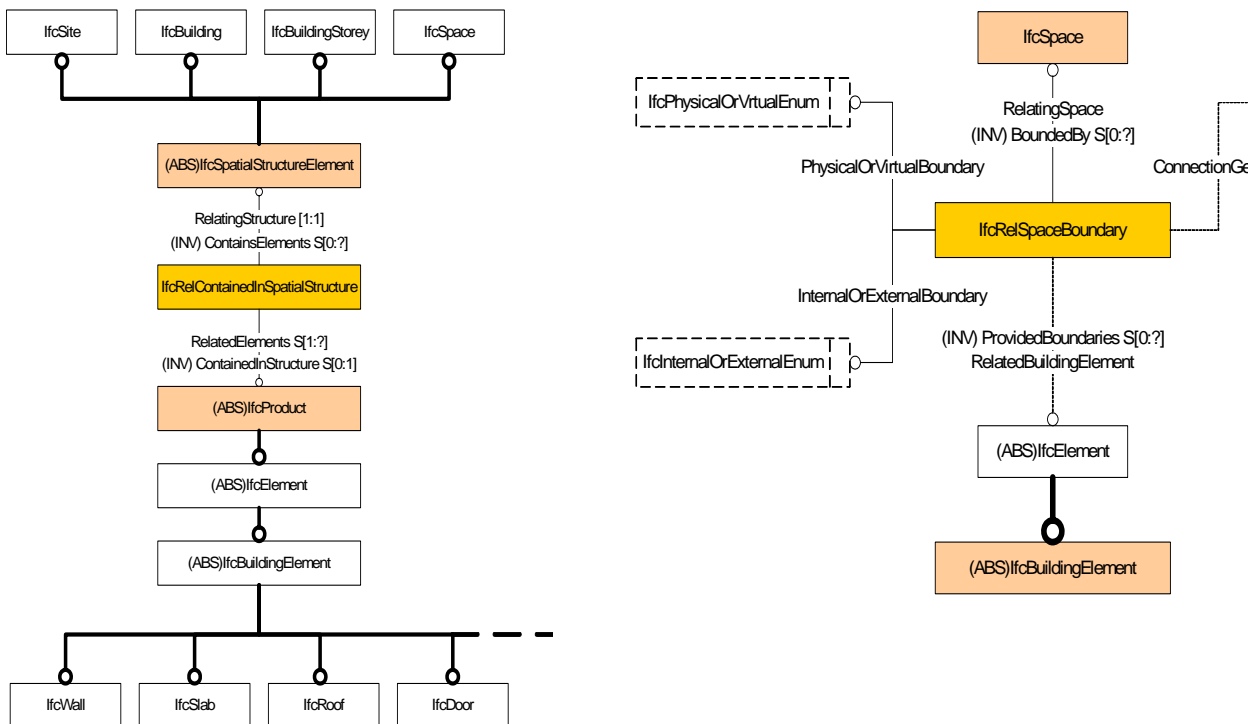


Abb. 8: Über die Relation *IfcRelContainedInSpatialStructure* erfolgt die Zuordnung einzelner Bauteile (*IfcBuildingElement*) zur räumlichen Struktur (links). Gleichzeitig ist die Relation zwischen einem Raum und seinen raumbegrenzenden Bauteilen mittels *IfcRelSpaceBoundary* gegeben (rechts).

4.4.2. Mängel

Die Abbildung von Mängeln in Form von Objekten ist derzeit in den IFC nicht vorgesehen. Allerdings existiert die Möglichkeit, nicht definierte Objekte mit Hilfe von *IfcProxy* abzubilden. Proxy-Objekte dienen dem Datenaustausch von nicht durch die IFC spezifizierten Objekten zwischen verschiedenen Applikationen. Über Verknüpfungen mit Eigenschaftssätzen können diese näher spezifiziert werden, so bspw. zur Abbildung von Mängeln und Schäden. Proxy-Objekte können dabei als Platzhalter für jegliche Ableitungen von *IfcObject* stehen (definiert durch das Attribut *ProxyType*). Steht das Proxy-Objekt für ein physisches Objekt (*ProxyType* = *.PRODUCT*, entspricht *IfcProduct*), kann dieses auch geometrisch repräsentiert werden. Über die Relationen *IfcRelAssignsToProduct* und *IfcContainInSpatialStructure* kann das Proxy-Objekt sowohl mit der Bauteil-, als auch mit der Raumstruktur verknüpft werden.

4.4.3. Geometrieabbildung

Das Geometriemodell verwendet zur geometrischen Abbildung von Bestandsgebäuden ein Oberflächenmodell (siehe Kapitel 4.2.3).

Innerhalb den IFC werden verschiedene Formen der geometrischen Modellierung sowohl für 2D-, als auch für 3D-Modelle unterstützt:

Type	
Curve2D	2 dimensional curves

GeometricSet	points, curves, surfaces (2 or 3 dimensional)
GeometricCurveSet	points, curves (2 or 3 dimensional)
SurfaceModel	face based and shell based surface model
SolidModel	including swept solid, Boolean results and Brep bodies more specific types are:
SweptSolid	swept area solids, by extrusion and revolution
Brep	faceted Brep's with and without voids
CSG	Boolean results of operations between solid models, half spaces and Boolean results
Clipping	Boolean differences between swept area solids, half spaces and Boolean results
AdvancedSweptSolid	swept area solids created by sweeping a profile along a directrix
<i>additional types</i>	some additional representation types are given:
BoundingBox	simplistic 3D representation by a bounding box
SectionedSpine	cross section based representation of a spine curve and planar cross sections. It can represent a surface or a solid and the interpolations of the between the cross sections is not defined
MappedRepresentation	representation based on mapped item(s), referring to a representation map. Note: it can be seen as an inserted block reference. The shape representation of the mapped item has a representation type declaring the type of its representation items.

Tab. 3: Unterstützte Geometriemodellierung in den IFC [IFC09]

Für die Bestandsplanung ist insbesondere die dreidimensionale, geometrische Abbildung von Bauteilen (*IfcBuildingElement*) und Räumen (*IfcSpace*) bzw. raumbegrenzenden Oberflächen relevant.

Für die Bauteile erlauben die IFC u.a. die Verwendung von *SurfaceModel* und Abbildung mittels *IfcFaceBasedSurfaceModel*. Dies erlaubt auch die geometrische Repräsentation durch eine nicht vollständig geschlossene Flächenmodellierung. Für Räume sieht die IFC u.a. die Verwendung von *BRep* mittels *IfcFacetedBrep* vor, nicht aber *SurfaceModel*. Dies bedeutet, dass Räume stets als geschlossene Volumen abzubilden sind.

IfcFacetedBrep und *IfcFaceBasedSurfaceModel* bilden letztendlich die dreidimensionale Geometrie durch Verwendung eines Sets von *IfcPolyLoops*. Diese beschreiben einzelne planare Flächen anhand der flächenbegrenzenden Punkte.

Die 3D-Geometrie kann somit aus einzelnen *IfcPolyLoops* aufgebaut sein, welche sowohl durch das *SurfaceModel* als auch durch das *BRep* referenziert werden. Die Übertragung der Oberflächen (Polygone) des Geometriemodells in die IFC ist somit möglich.

5. Skizzenbasierte Ersterfassung

5.1. Anforderungen an Benutzungsoberfläche und Interaktion beim Arbeiten vor Ort

Die Kosten-Risiko-Analyse im frühen Projektstadium (Initiierungsphase) basiert auf Informationen über das Gebäude, die mit einer schnellen und effizienten Ersterfassung oder der Ergänzung und Validierung vorhandener Informationen vor Ort gewonnen werden.

Die Arbeitssituation vor Ort ist geprägt durch schlecht optimierbare Licht- und thermische Verhältnisse, eingeschränkte Begehrbarkeit sowie durch Zeitdruck, der neben den bereits genannten Witterungs-Bedingungen auch dem frühen Projektstadium und den technischen Rahmenbedingungen geschuldet ist. Da die Erfassung von Anfang an digital erfolgen soll, werden als Äquivalent zu den unter diesen Bedingungen gebräuchlichen Klemmbrettern tragbare Ultra Mobile PCs eingesetzt, die einhändig mit einem Stift bedient werden.



Abb. 9: Beispielsituation vor Ort (links) und Ultramobile PC (UMPC) (rechts)

Aus der Charakteristik vor Ort ergeben sich Konsequenzen für die Konzeption und Gestaltung der Benutzungsoberfläche:

- Die Benutzungsoberfläche muss mit einem Stift bedienbar sein, weil die Geräte mindestens im mobilen Modus auf Maus und Tastatur verzichten.
- Die Oberfläche sollte möglichst übersichtlich und mit relativ großen und kontrastreichen Interaktionselementen ausgestattet sein, um negative Umgebungsbedingungen zu kompensieren.
- Wegen der relativ kleinen Displays sollte die Fläche des Eingabebereichs permanent oder temporär maximiert werden.

5.2. Stand der Forschung: computergestützte Vor-Ort-Erfassung

Zeichnen und Skizzieren als traditionelle Arbeitsmethoden der Architekten haben sich nicht nur für das Festhalten von Entwurfsideen, sondern auch für die Notation bestehender baulicher Objekte über Jahrhunderte bewährt. Mit dem zunehmenden Bedürfnis möglichst frühzeitig digitale Werkzeuge in den Arbeitsablauf zu integrieren, haben sich zwei Möglichkeiten der Computerunterstützung herausgebildet: das nachträgliche Einpflegen

analog auf Papier erfasster Informationen in ein Computersystem und die Nutzung von Standard-CAD-Software auf mobilen Geräten vor Ort. Während die erste Variante mit redundanter Erfassung, Erhöhung des zeitlichen Aufwands sowie der Gefahr einer fehlerhaften Übertragung einhergeht, ist die zweite Variante defizitär hinsichtlich des effizienten Arbeitens vor Ort, da hier die für den Einsatz am Schreibtisch entwickelten Programme zwangsläufig versagen.

Stiftbasierte Mensch-Computer-Schnittstellen adaptieren die einfache Form der Interaktion mit einem Stift auf einem Blatt Papier für den Computer und versuchen dabei, den Informationsgehalt der gezeichneten Liniengrafik für die elektronische Informationsverarbeitung nutzbar zu machen. Mit der Entwicklung und Verbreitung entsprechender stiftbasierter Hardware - ultramobile Computer (UMPCs) – rückt die Vision eines solchen Skizzierwerkzeugs näher. Es erfordert allerdings Software, deren Interface für das Arbeiten mit dem Stift angepasst ist (im Gegensatz zu den derzeitigen für maus- und tastaturbasierte Interaktion entwickelten Schnittstellen) und die über zusätzliche Funktionalität verfügt, um den Informationsgehalt der Skizzen zu erschließen.

Die Suche nach etablierten Möglichkeiten die Intuitivität und Effizienz der skizzenhaften Erfassung auf Papier in ein Computersystem zu übertragen, wurde entsprechend dieser Erfordernisse an zwei Schwerpunkten ausgerichtet: der Ergonomie stiftbasierter Benutzungsoberflächen einerseits und Funktionalitäten zur Verarbeitung und Interpretation skizzenhaft eingegebener Informationen durch Computersysteme andererseits.

In den beiden folgenden Unterabschnitten sind die Ergebnisse der Recherche zusammengefasst. Eine ausführliche Betrachtung inklusive Software-Review bereits vorhandener kommerzieller und experimenteller Systeme ist dem Technical Report im Anhang D zu entnehmen, welcher im Rahmen des Forschungsprojektes erarbeitet wurde.

5.2.1. Verarbeitung und Interpretation skizzenhaft eingegebener Informationen

Die freihändige Eingabe mittels Stift erzeugt eine Liniengrafik, welche in der Regel als Vektorgrafik durch die Aufnahme von Kontrollpunkten entlang der Linie rechnerintern abgebildet wird. Die einzelnen Elemente der Grafik werden *Strokes* (Striche) genannt, die gesamte Menge der Elemente wird als *Ink* (Tinte) bezeichnet⁵. Den Elementen dieser Vektorgrafik werden meist verschiedene visuelle Attribute wie Breite, Farbe in Analogie zu den verschiedenen traditionellen Stiftspitzen zugewiesen. Diese Attribute dienen der visuellen Unterscheidung der Elemente durch den Nutzer, können aber auch mit einer Bedeutung verknüpft und damit durch den Rechner semantisch interpretiert werden.

Die Verarbeitung als Vektorgrafik ermöglicht zudem die Auswahl, Transformation und Manipulation der einzelnen Strokes bzw. der einzelnen Kontrollpunkte. Analog zum Anklicken mit der Maus hat sich hierfür das Antippen mit dem Stift in den meisten Systemen als gebräuchlich durchgesetzt. Für die Mehrfachauswahl von Strokes ergeben sich – im Gegensatz zur Mausbedienung – durch die Stiftbedienung andere Interaktionstechniken, wie Umkreisen oder Durchstreichen der Elemente, welche den Einsatz der Tastatur (bspw. zum Wählen der Umschalttaste) unnötig machen. Ebenso ist die Verwendung von Stiftgesten zum Aufrufen von Befehlen möglich, wobei eine Unterscheidung zwischen dem Ausführen der Geste oder dem Zeichnen von Elementen vorgenommen werden muss.

⁵ Diese Begriffe haben sich ausgehend von der Terminologie des [Microsoft08]Tablet PC SDK durchgesetzt.

Skizzierte oder handschriftliche Informationen können sehr zügig notiert werden, so dass sich schnell eine größere Informationsmenge ansammelt. Damit steigt die Notwendigkeit geeigneter Ordnungsmechanismen zur Verwaltung der Informationen. Häufig wird auf die Analogie des Notizbuches zurückgegriffen, bei der einzelne „Skizzenblätter“ mit Namen und Datums- bzw. Zeitinformation abgelegt werden. Als Erweiterung dieses Prinzips können Skizzen auch semantisch Elementen zugeordnet werden; bspw. die (Grundriss-)Skizze einem Geschoss.

Auch das Prinzip der Überlagerung von Ebenen (auch Layer genannt) – wie es in CAD-Systemen üblich ist – findet sich in vielen Anwendungen. Dies stellt wiederum eine Analogie zum Übereinanderlegen von analogen Skizzen auf Transparentpapier dar.

Um die in der zeichnerischen Eingabe enthaltenden Informationen extrahieren zu können, bedarf es der Interpretation. Hierzu wird die Zeichnung in topologisch strukturierte Geometrie überführt, welche mit Methoden der Mustererkennung durchsucht werden kann. Bei der geometrischen Interpretation werden einfache Grundformen wie Liniensegmente, Kreise, Quadrate sowie Orthogonalität oder Parallelität erkannt. Von einer semantischen Interpretation spricht man, wenn bestimmte geometrische Formen mit einer Bedeutung versehen werden.

5.2.2. Ergonomie stiftbasierter Interaktion

Allgemeine Anforderungen an Benutzungsoberflächen gelten nicht nur für Maus bzw. Tastatur bedienbare Interaktionen, sondern auch für stiftbasierte. Grafische Benutzeroberflächen nach dem WIMP-Paradigma (Benutzeroberflächen mit den Elementen Window, Icon, Menü, Pointing Device) haben sich dabei weitestgehend durchgesetzt. Die Anforderungen an eine Benutzeroberfläche ergeben sich aus Ergonomie und die Nutzung weit verbreiteter und etablierter Standards⁶.

Während bei der Maus die punktförmige Interaktion (Zeigen und Klicken) als Standardform betrachtet werden kann und die linienförmige Interaktion (Klicken, Gedrückt-Halten und Ziehen) eher umständlich und seltener ist, verhält es sich für den Stift genau umgekehrt. Das Tippen mit dem Stift ist eher unergonomisch, während die linienförmige Bewegung dem Eingabegerät entspricht. Dennoch lassen sich die meisten für die Bedienung mit der Maus entwickelten Oberflächen relativ einfach auf Stifteingabe umstellen, indem Klicken durch Tippen ersetzt wird und das Ziehen mit gedrückter Maustaste durch linienförmige Bewegung direkt auf dem Bildschirm.

Problematisch erweist sich der Moduswechsel. Während bei der Maus der Doppelklick oder die rechte Maustaste zur Verfügung steht, ist eine zusätzlich Taste am Stift eher unergonomisch. Verschiedene Systeme arbeiten beim Moduswechsel mit Verzögerung (Tippen und anschließende Verzögerung), welche aber den Arbeitsfluss mit dem Stift verzögert. Wenn mehrere Funktionen mit der gleichen Stiftbewegung aufgerufen werden sollen, muss für mindestens eine ein expliziter Moduswechsel erfolgen. Aus den klassischen WIMP-GUIs ist der explizite Moduswechsel durch die Werkzeugmetapher bekannt. Durch Auswahl eines Werkzeugs werden die folgenden Interaktionen in einer bestimmten Weise interpretiert, die Form des Mauszeigers gibt ein Feedback, welcher Modus gerade aktiv ist. Meist sind die Werkzeuge in Werkzeugkästen angeordnet, die verschiebbar und anpassbar sind.

⁶ Anforderungen an eine grafische Benutzungsoberfläche sind in Teil 110 der EN ISO 9241 beschrieben

Für das Skizzieren ist eine möglichst große Eingabefläche wünschenswert, die frei von Interaktionselementen ist. Für mobile Geräte mit einem kleineren Bildschirm muss die grafische Oberfläche so angepasst werden, dass möglichst viel Platz für die Skizzeneingabe bleibt. Eine Möglichkeit auf klassische Interaktionselemente wie Werkzeug- und Menüleiste verzichten zu können, ist die Verwendung von Kreismenüs. Das Kreismenü integriert über die Verbindung einer hierarchischen Menüstruktur mit Icons hinaus Eigenschaften, die vom Kontextmenü her bekannt sind: die Menüstruktur entfaltet sich ausgehend von einem einzelnen Punkt. Hinzu kommt der Vorteil der Kombinierbarkeit mit Stiftgesten, die sich aus Linien in Richtung der Menüpunkte zusammensetzen. Kreismenüs können dabei den Lernprozess dieser Gesten unterstützen, wenn die Menüpunkte immer gleich angeordnet sind.

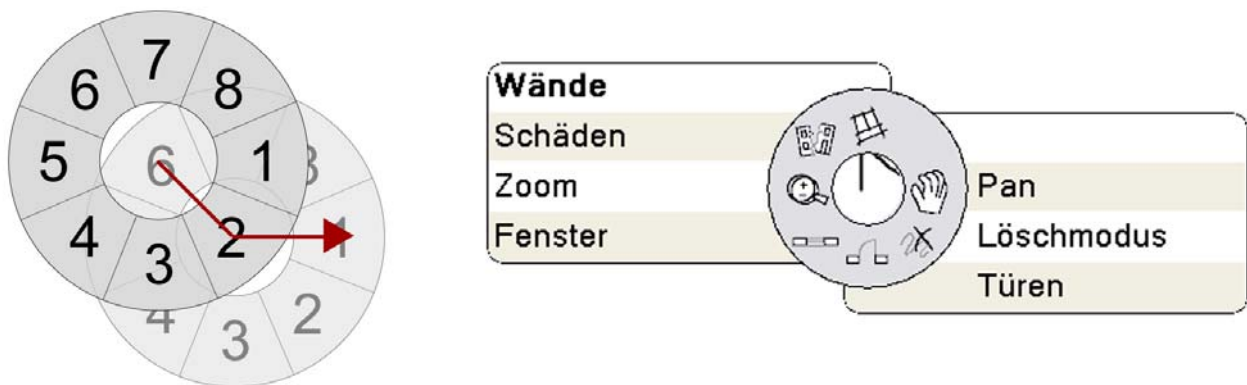


Abb. 10: Prinzip der Stiftgesten bei Kreismenüs: Auswahl des Befehls an Position 2.1 (links), implementiertes Kreismenü im Prototyp InitialDataCaptur (rechts)

5.3. Konzept eines skizzenbasierten Ersterfassungssystems

Im Folgenden soll das entwickelte Benutzungs- und Interaktionskonzept des Ersterfassungswerkzeuges zur schnellen skizzenhaften Erfassung planungsrelevanter Bestandsdaten vor Ort näher betrachtet werden. Der Fokus des Systems liegt in einer schnellen und strukturierten Erfassung der Daten in unterschiedlichen Detaillierungsstufen. Hierzu ist ein Werkzeug notwendig, welches es erlaubt, zunächst die Raum- und Bauteilstruktur des Gebäudes in ihrem aktuellen Zustand sowie vorhandene Mängel zu erfassen. Darüber hinaus sind auch informale Daten (z.B. Anmerkungen, Fotos, vorhandene Planunterlagen) einzubeziehen. Mit zunehmender Detaillierung der Daten, sind Messwerte einzupflegen, um von einem grob geometrischen Modell zu einem maßlich korrekten Abbild des Gebäudes zu gelangen.

Für die Eingabe dieser Daten sind im System drei verschiedene Modi vorgesehen, welche sich hinsichtlich der Eingabe- und Interaktionstechnik sowie Interpretation und Repräsentation der einzugebenden Daten unterscheiden:

- Sketch Mode
- Capture Mode
- Survey Mode.

5.3.1. Sketch Mode

Dieser Modus dient der ergänzenden Erfassung informaler Daten in Form von freien Handskizzen. Diese werden vom System nicht interpretiert und können daher ohne Einschränkungen gezeichnet werden (im Gegensatz zum Capture Mode, dessen Algorithmen zur Skizzeninterpretation vorgegebene Eingabeabläufe erzwingen). Die freien Skizzen können durch den Import von Pixelgrafiken ergänzt werden.

Als Organisationsprinzip wird die Analogie des Skizzenbuches verwendet. Die informellen Daten in Form der Skizze werden als separates Objekt (vergleichbar mit einer Ebene bzw. einem Skizzenblatt) mit Namen und Zeitstempel abgelegt. Diese Objekte können den Elementen des Raum- und Bauteilmodells als zusätzliche Repräsentation in beliebiger Anzahl zugeordnet werden. Da neben der freihändigen Eingabe von Grafik auch das Einbinden von Pixelgrafiken möglich ist, können bspw. gescannte Planunterlagen als Repräsentation eines Gebäudegeschosses dienen und als Grundlage für die Skizzenerstellung im Capture-Mode genutzt werden.

5.3.2. Capture Mode

Dieser Modus stellt die Standard-Eingabefunktionalität für die Erfassung der Raum- und Bauteilstruktur sowie die Eingabe von Mängeln dar. Dem Nutzer stehen Werkzeuge zur Erstellung von Wandlinien und Raumgrenzen, Öffnungen und Mängeln zu Verfügung. Der Geschossgrundriss wird dabei mit dem Eingabestift freihändig skizziert. Das System interpretiert die Skizzengrafik (Ink-Modell) und überführt diese nach Verschneidung aller geraden Linienzüge in das Skizzeninterpretationsmodell. Das System greift dabei auf Funktionalitäten des Software Development Kit (SDK) der Microsoft Windows XP Tablet PC Edition zurück [Microsoft08].

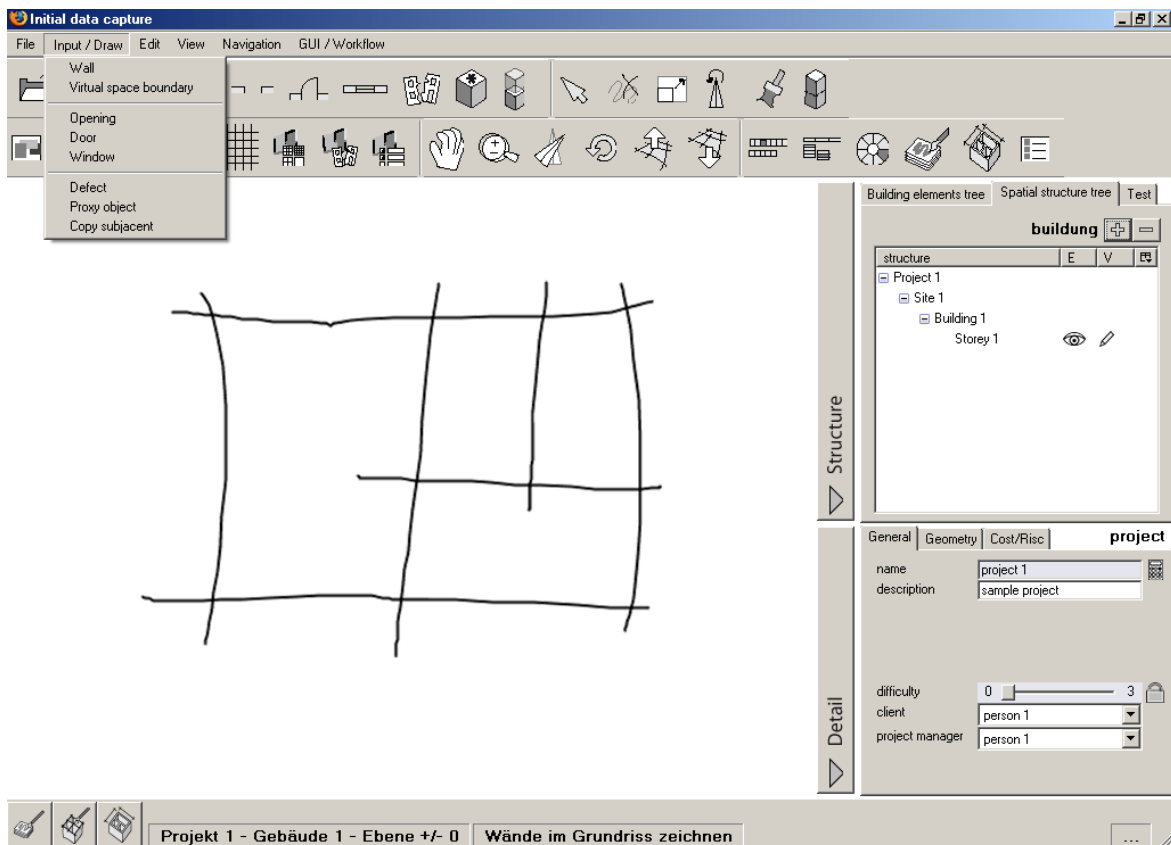


Abb. 11: Konzept der Benutzeroberfläche – Capture-Mode

Für den Nutzer bleibt die Darstellung der Skizze erhalten, so dass die ursprüngliche Skizze die Ungenauigkeit der Eingabe veranschaulicht und kein exakt gemessenes Modell suggeriert.

Neben der Interpretation der Skizze wird gleichzeitig die Raum- und Bauteilstruktur angelegt. Linienzüge werden als Wandsegmente oder virtuelle Raumgrenzen interpretiert; bilden sie einen geschlossenen Bereich, wird ein Raum angelegt. Öffnungen und Mängel können als Symbole in die Skizzengrafik eingefügt und gleichzeitig als Elemente des Bauteilmodells angelegt werden. Der konzipierte Funktionsumfang des Capture Mode ist detailliert im Anhang B aufgeführt.

5.3.3. Survey Mode

Das Skizzeninterpretationsmodell wird in ein 3D-Oberflächenmodell überführt und als solches dem Nutzer sichtbar gemacht. Aus den einzelnen Linienzügen werden jeweils zwei planare Flächen erzeugt, welche die Bauteiloberflächen der Wandsegmente repräsentieren. Innerhalb dieser Sicht können geometrische Filter (Parallelität, Rechtwinkligkeit, Symmetrie etc.) gesetzt und Messwerte eingegeben werden. Mittels einer Ausgleichsrechnung wird das 3D-Modell schrittweise an die gemessenen Werte angepasst. Diese Funktionalität greift auf bereits getätigte Forschungsarbeiten zurück und ist ausführlich in [Thurow04], [SFB524_05a] und [SFB524_05b] beschrieben.

5.3.4. Raum- und Bauteilstruktur

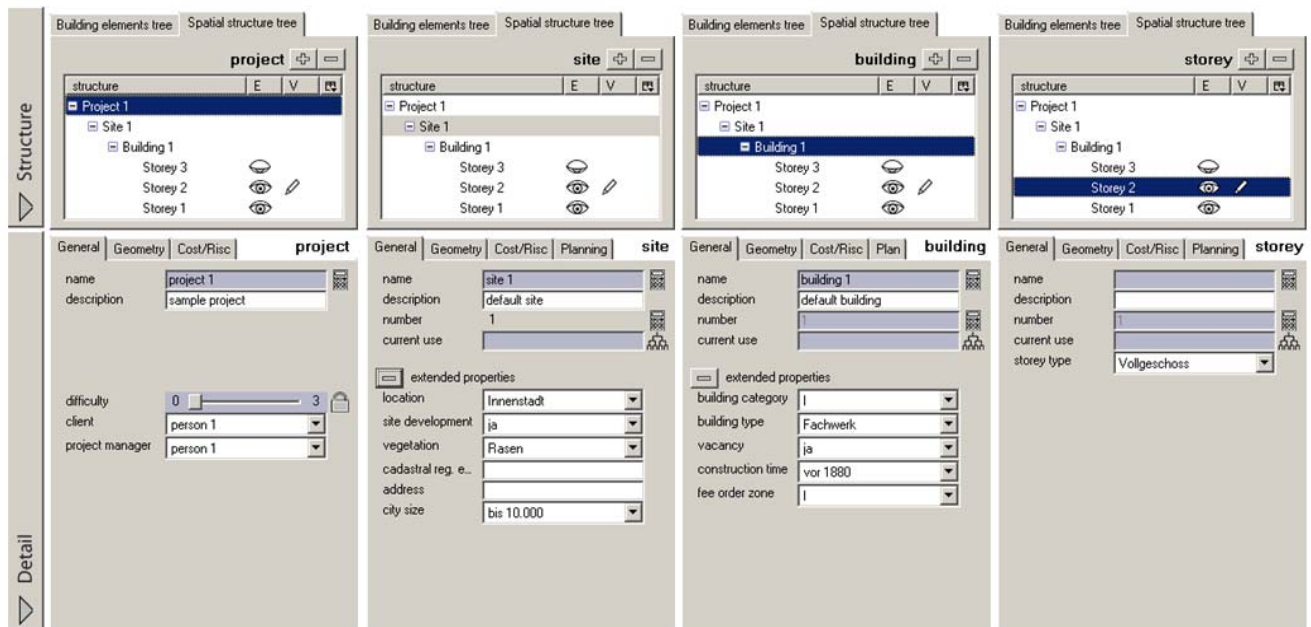


Abb. 12: Abbildung der Gebäudestruktur und Eingabedialoge für Projekt, Grundstück, Gebäude und Geschoss (von links nach rechts)

Innerhalb der drei genannten Eingabemodi lässt sich das Raum- und Bauteilmodell jeweils in seiner räumlichen Gliederung (Projekt, Gebäude, Geschoss, Raum) und Bauteilgliederung als Baumstruktur (treeview) darstellen. Die Bauteilgliederung orientiert sich an den Kostengruppenelementen der DIN 276, welche die Grundlage für die Kostenermittlung

darstellt. Des Weiteren ist die Eingabe und Manipulation der Objekteigenschaften möglich. Auch hier ist sichergestellt, dass die Herkunft der Werte ablesbar bleibt. Dies geschieht zum einen durch farbliche Kodierung als Unterscheidung zwischen manuell gesetzten Werten und Systemvorgaben, und zum anderen durch Darstellung der Datenherkunft mittels Icons (default, berechnet, geerbt, aus Skizze).

6. Content Management System

6.1. Stand der Forschung und Praxis

Zur Speicherung von Gebäudemodellen bieten sich datenbankbasierte Systeme an, mit denen Daten bzw. Dateien in logischen Strukturen nach selbst definierten Regeln abgelegt werden können. Dies lässt sich insbesondere mittels so genannter "Content Management Systeme" (CMS) realisieren, da diese Systeme individuell anpassbar sind [McCay05]. CMS kann ins Deutsche mit dem Begriff "Inhaltsverwaltungssystem" übersetzt werden. CMS zeichnen sich dadurch aus, dass beliebiger Inhalt (engl. Content) z.B. in Form von Text-, Bild- oder CAD-Dokumenten innerhalb eines Systems organisiert wird. Hierbei kommen Schnittstellen für den Import und Export zum Einsatz und die Inhalte lassen sich nach individuellen Regeln verwalten. Es ist beispielsweise möglich, zu importierende Daten, denen eine bestimmte Datenstruktur zugrunde liegt, über selbst geschriebene Importschnittstellen zu zerlegen und umzuwandeln (= Parser). Die Daten werden dann in einer eigenen Tabelle abgelegt mit Zugriffsmöglichkeit auf Einzelteile der zuvor importierten Daten.

Ein Vorteil mancher Content Management Systeme ist die Tatsache, dass in den professionell eingesetzten Systemen bereits Verfahren zum Anlegen von Rechten und Rollen vorhanden sind. Damit kann man definieren, wer zu welchem Zeitpunkt auf festgelegte Daten zugreifen und gegebenenfalls neue Daten anlegen darf. Manchen Benutzern können so beispielsweise nur Leserechte auf einige Inhalte zugewiesen werden. Dies ist bei dem hier eingesetzten CMS der Fall.

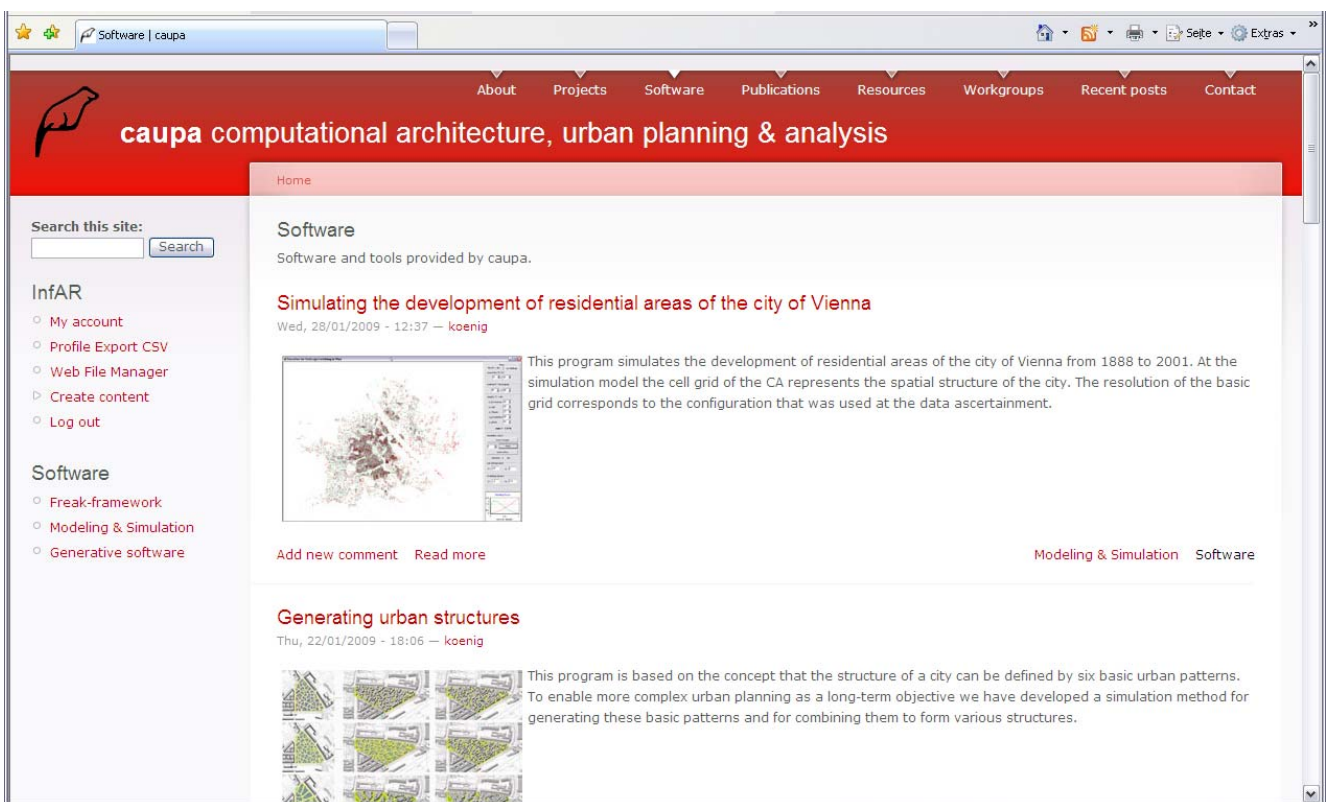


Abb. 13: Typische Oberfläche eines CMS (<http://caupa.prodieas.de/>)

Das Einstellen von Inhalten geschieht meist über eine internetbasierte Benutzeroberfläche (Abb. 13). Ein Nutzer kann ein solches System auch ohne Programmier- oder HTML-Kenntnisse bedienen, allerdings sind Programmierkenntnisse erforderlich, wenn individuelle Funktionen eingebaut werden sollen. Für ein solches, angepasstes System, sind die Autoren (Lieferanten der Inhalte des CMS) speziell einzuweisen. Alle Bedienungsschritte werden über eine Browseroberfläche abgewickelt, so dass für die Benutzer der Eindruck einer üblichen Internet-Seite entsteht, von der Informationen und Dokumente heruntergeladen, sowie Informationen hochgeladen werden können.

6.2. Anpassungen innerhalb des Projektes

Im Rahmen dieses Forschungsprojektes war es notwendig, Gebäudemodelldaten in Form von IFC-Dateien zu verwalten. Diese Dateien liegen als ASCII-Text-Dateien vor und können mit einem CMS auch als solche hochgeladen werden. Problematisch erscheint jedoch die Suche nach bestimmten Elementen bzw. Bauteilen innerhalb der IFC-Dateien, die ja ganze Gebäude mit den zu den Bauteilen gehörenden Mängeln repräsentieren. Da Mängel derzeit durch die IFC nicht unterstützt werden, wurden diese zunächst mittels *IfcProxy* abgebildet (siehe Kapitel 4.4.2). Falls gezielt nach den Bauteilen bzw. den mangelbehafteten Bauteilen gesucht werden soll, ist es bei den zu erwartenden großen Datenmengen unsinnig, reine Textsuchen auf einige sehr große Dateien anzuwenden. Dieses Verfahren würde zu viel Rechenzeit in Anspruch nehmen. Aus diesem Grund wurde ein Parser entwickelt, der die Bauteilbezeichnungen sowie angefügte Mängellisten und Sanierungsmaßnahmen ausgibt (siehe Kapitel 7.4.2.).

Zudem stellte sich die Frage, wie man die zu erwartenden Daten insgesamt organisiert. Am einfachsten erschien eine Lösung, die dem üblichen architektonischen Planungsprozess möglichst nahe kommt, nicht zuletzt deshalb, weil damit die Hemmschwelle, ein komplett neues System anzuwenden, herabgesetzt werden sollte. Innerhalb des CMS soll also zunächst ein Projekt angelegt werden können. Häufig ist mit dem Auftrag, eine Bauaufnahme durchzuführen, verbunden, im Anschluss auch eine Planung zur Sanierung oder Umnutzung durchzuführen. Beim Anlegen eines Projektes müssen also entsprechende Ordner und Ordnerstrukturen für die Leistungsphasen der [HOAI08] erstellt werden.

Darauf aufbauend benötigt man eine Lösung für die im Planungsprozess übliche Form der Indizierung. Diese dient dazu, wesentliche Änderungen innerhalb einer Planungsphase zu strukturieren. Beispiel:

Im Rahmen der Werkplanung wird zunächst ein Abbruch mehrerer nichttragender Wände zur Verbindung von kleineren Zimmern zu größeren Einheiten geplant. Der Bauherr entscheidet sich jedoch (relativ spät) dazu, den Abbruch der Wände in zwei Fällen nicht durchführen zu wollen, sondern dort nur zwei Durchgänge vorzusehen.

Damit keine Verwechslung mit den vorangegangenen Planungen stattfindet, wird ein neuer Index (a, b, c, usw.) vom Projektleiter vergeben. Das Gebäudemodell und sämtliche Pläne müssen auf den neuen Stand gebracht und den Projektbeteiligten (Statiker, Haustechniker, ausführende Firmen) zugänglich gemacht werden. Diese Form der Indizierung war bisher in CMS nicht vorhanden, konnte aber als Funktion hinzugefügt werden.

7. Prototypische Umsetzung

Anhand der Systemkonzeption erfolgten prototypische Implementationen zum Nachweis der Umsetzbarkeit verschiedener Teilaspekte. Die Implementation erfolgte dabei in verschiedenen Modulen unter Verwendung unterschiedlicher Entwicklungsumgebungen und zum Teil auf Basis bestehender Softwaresysteme. Die einzelnen Module sind über Schnittstellen miteinander verzahnt, um den Arbeitsfluss der Systemkonzeption von der Datenaufnahme und -detaillierung über die Kosten-/Risikoanalyse bis zum IFC-Export und die Übergabe an das Content Management System abzubilden. Die einzelnen Protypen sind frei über die Webplattform *caupa*⁷ verfügbar.

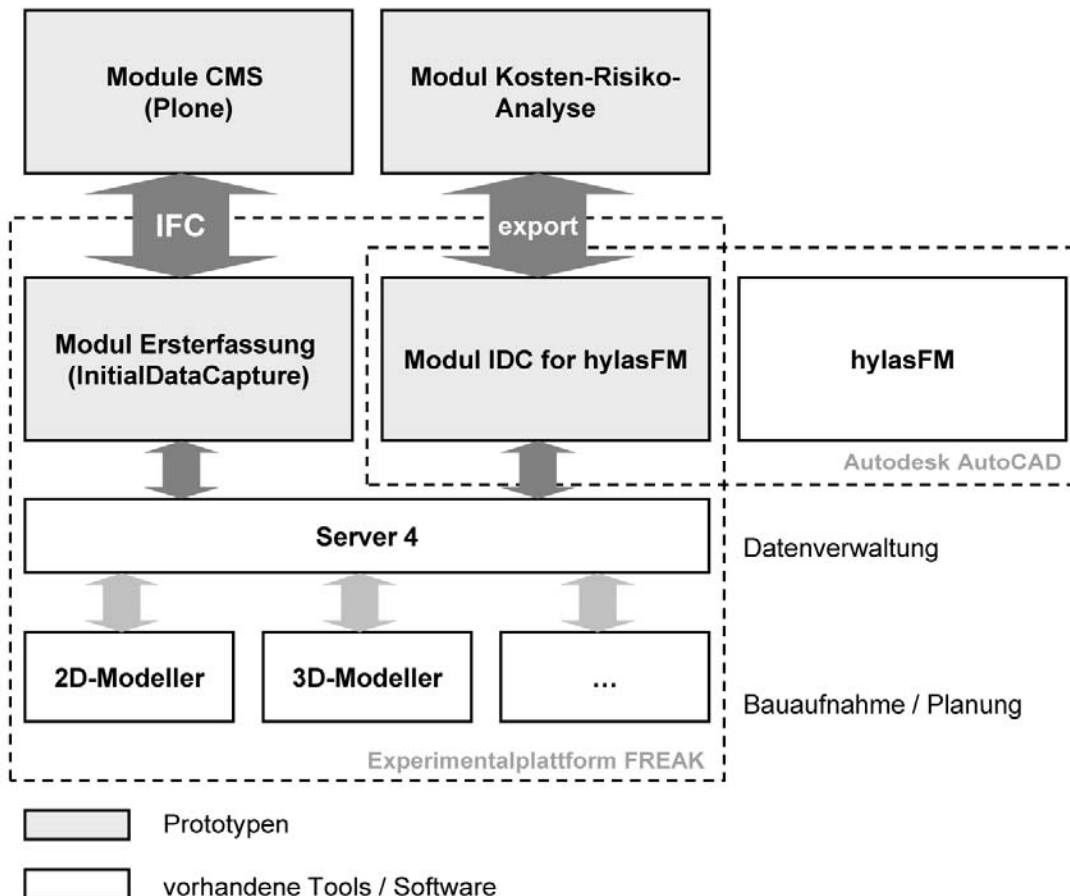


Abb. 14: Teilmodule der prototypischen Arbeitsumgebung und deren Verknüpfung

7.1. Modul Ersterfassung (InitialDataCapture)

Das Modul entspricht in seiner Funktionalität weitestgehend dem Capture-Mode des konzipierten Ersterfassungssystems in reduziertem Umfang. Der Prototyp zeigt die skizzenbasierte Erfassung eines Gebäudes in seiner räumlichen Struktur und Bauteilstruktur sowie die Aufnahme von Zuständen und Mängeln.

⁷ <http://www.caupa.de/projects/zukunftBau>

7.1.1. Funktionalität

Der Prototyp erlaubt das Anlegen von Geschossgrundrissen mittels Freihandlinien auf einem TabletPC oder UMPC. Die Freihandlinien (Ink-Modell) werden vom System als Wandsegmente bzw. virtuelle Raumgrenzen interpretiert (Skizzeninterpretationsmodell). Entsprechend der identifizierten Elemente werden Objekte der Klassen Raum (*Space*) und Wand (*Wall*) bzw. Raumgrenze (*VirtuellSpaceBoundary*) im Raum- und Bauteilmodell angelegt. Weitere Funktionen erlauben das Einfügen von Öffnungen sowie Mängeln sowie das Hinterlegen von Grafiken (z.B. gescannte Planvorlage). Die Interpretation der Freihandlinien erfolgt während der Eingabe in Echtzeit. Eine Umwandlung des Skizzeninterpretationsmodells in ein 3d-Geometriemodell kann anschließend erfolgen. Ein Export der resultierenden 3D-Geometrie sowie der Elemente des Raum- und Bauteilmodells als IFC-Datei ist möglich.

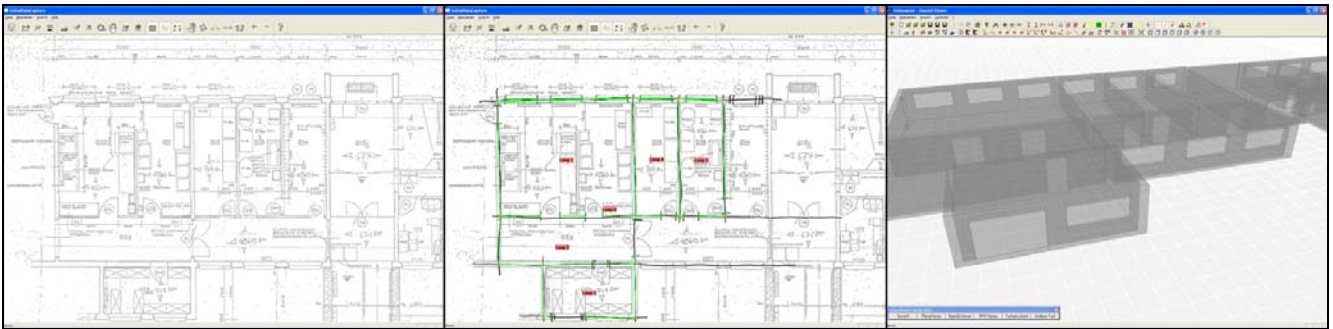


Abb. 15: Modul Ersterfassung – Hinterlegen von Grafiken (Scan), Erstellen von Freihandlinien und Interpretation, Ableiten der 3D-Geometrie (von links nach rechts)

Entsprechend der Recherche stiftgerechter Eingabe- und Interaktionstechniken wurden das GUI (Graphical User Interface) und die Benutzungsführung optimiert. Hierzu entstand zunächst ein Mock-up (funktionsfreier Prototyp) der Benutzeroberfläche, welcher Menüleisten und Schaltflächen für nahezu alle vorgesehenen Funktionen enthält (Abb. 16).

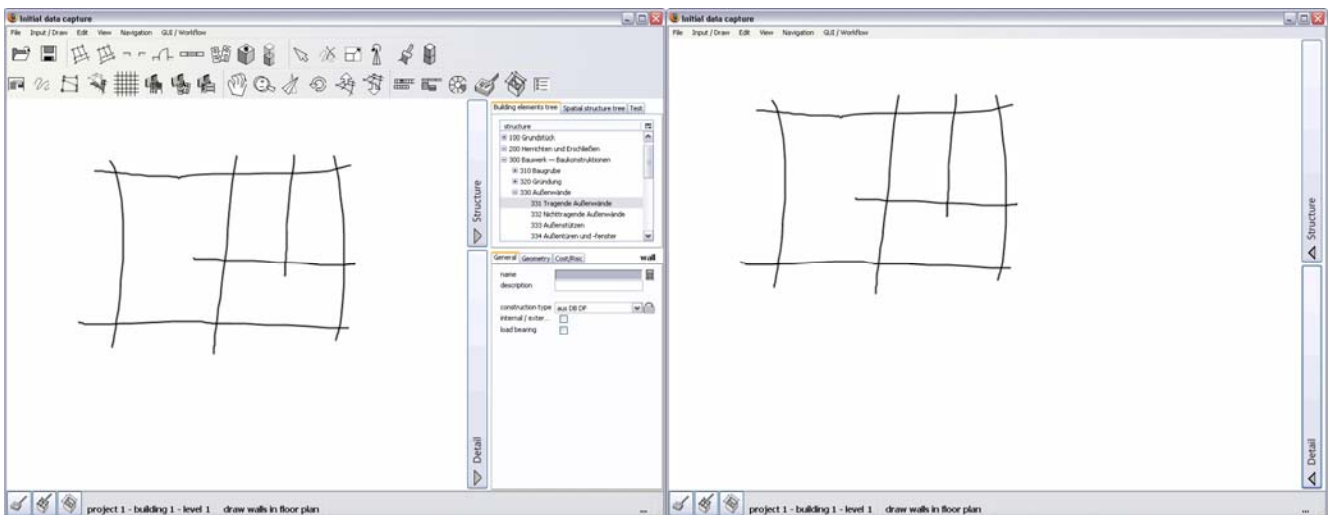


Abb. 16: Mock-up der Benutzeroberfläche mit eingeblendeten Menüs (links) und maximierter Eingabefläche (rechts)

Wichtig beim Design der Benutzeroberfläche erschien die maximale Ausnutzung des Bildschirms zur Erstellung der Freihandlinien. Hierzu lassen sich die Werkzeugleisten ausblenden. Der Zugriff auf die Bauwerksstruktur und die Eigenschaftsdialoge erfolgt über

temporär ausklappbare Reiter am Bildschirmrand. Bei der Eingabe von alphanumerischen Werten wird temporär die Handschrifterkennung aktiviert.

Als zweites Eingabekonzept, neben üblichen Werkzeugleisten, erfolgte die Implementierung eines Kreismenüs, welches zum einen kurze Wege und zum anderen die Nutzung von Gesten bei der Werkzeugauswahl erlaubt (siehe Kapitel 5.2.2).

7.1.2. Implementation

Der Prototyp *InitialDataCapture* wurde als eigenständiger Client für die an der Professur „Informatik in der Architektur“ der Bauhaus-Universität Weimar entwickelten *Experimentalplattform Freak* umgesetzt. Dabei handelt es sich um eine Client-Server Plattform, die es erlaubt, verschiedene experimentelle Softwareapplikationen über einen gemeinsamen Server miteinander zu verbinden. Die Datenverwaltung erfolgt innerhalb des Servers, die Clients greifen auf die jeweils benötigten Strukturen zu. Innerhalb des Servers können verschiedene miteinander verknüpfte Teilmodelle (Kernel) verwaltet werden. Für die Entwicklung des Prototyps konnte dabei auf bereits vorhandene Modelle (Geometrie, Ausgleichung) zurückgegriffen werden. Neue Modelle (Ink-Modell, Skizzeninterpretationsmodell) wurden ergänzt und in die Plattform integriert. Die Implementation des Raum- und Bauteilmodells erfolgte auf Basis des bereits vorhandenen Modellverwaltungssystems (MVS), welches die Definition und Manipulation von Klassen und deren Relationen nach dem Paradigma der objektorientierten Programmierung zur Laufzeit erlaubt ([SFB524_05b], [SFB524_05c]).

7.1.3. Potential Weiterentwicklung, Ausblick

Bis zum Projektabschluss konnte nur ein Teil der konzipierten Funktionalitäten umgesetzt werden. Der Fokus des Prototyps lag vor allem in der Verwendbarkeit von Freihandskizzen als Eingabemittel für Gebäudemodelle. Während der Evaluation konnte dies positiv beurteilt werden.

Die Implementierung des Raum- und Bauteilmodells erfolgte nur in Teilen. Eine komplette Abbildung der konzipierten Modellstruktur auf Basis des in Freak integrierten MVS erwies sich als ineffektiv. Hierfür wurde auf das System hylasFM des Partners Kubit zurückgegriffen (siehe Kapitel 7.2). Beide Systeme wurden allerdings über eine Schnittstelle miteinander verknüpft, so dass der Arbeitsfluss der Datenaufnahme und Auswertung dennoch realisiert werden konnte.

Trotz Abschluss des Projektes ist eine Weiterentwicklung des Prototyps im Rahmen weiterer Forschung geplant. Die Erkenntnisse aus der Evaluierung werden darin einfließen. Insbesondere das Konzept der skizzenhaften Aufnahme von Gebäudegrundrissen und deren Interpretation als semantisches Modell erweist sich auch für andere Anwendungsfälle als vielversprechend.

7.2. Modul IDC for hylasFM

HylasFM erlaubt die Modellierung von Bauteilen und deren Beziehungen untereinander und ist als Zusatzapplikation in das CAD-System Autodesk AutoCAD eingebunden. Auf Basis von HylasFM konnte das Raum- und Bauteilmodell schnell und flexibel umgesetzt werden, um

die konzipierten Strukturen zu testen und gegebenenfalls anzupassen. HylasFM fungierte dabei als Modellverwaltungssystem.

7.2.1. Funktionalität

Der Prototyp erlaubt die Darstellung und Manipulation der Raum- und Bauteilstruktur in Form einer Baumansicht, wie sie in der Systemkonzeption vorgesehen ist. Darüber hinaus demonstriert der Prototyp die Anbindung einer externen Datenbank zum Abrufen praxisüblicher Maßnahmen (siehe Kapitel 3.4). Mittels Export stehen die Daten anschließend für die Auswertung im Modul Kosten-Risiko-Analyse (Kap. 7.2.3) zur Verfügung.

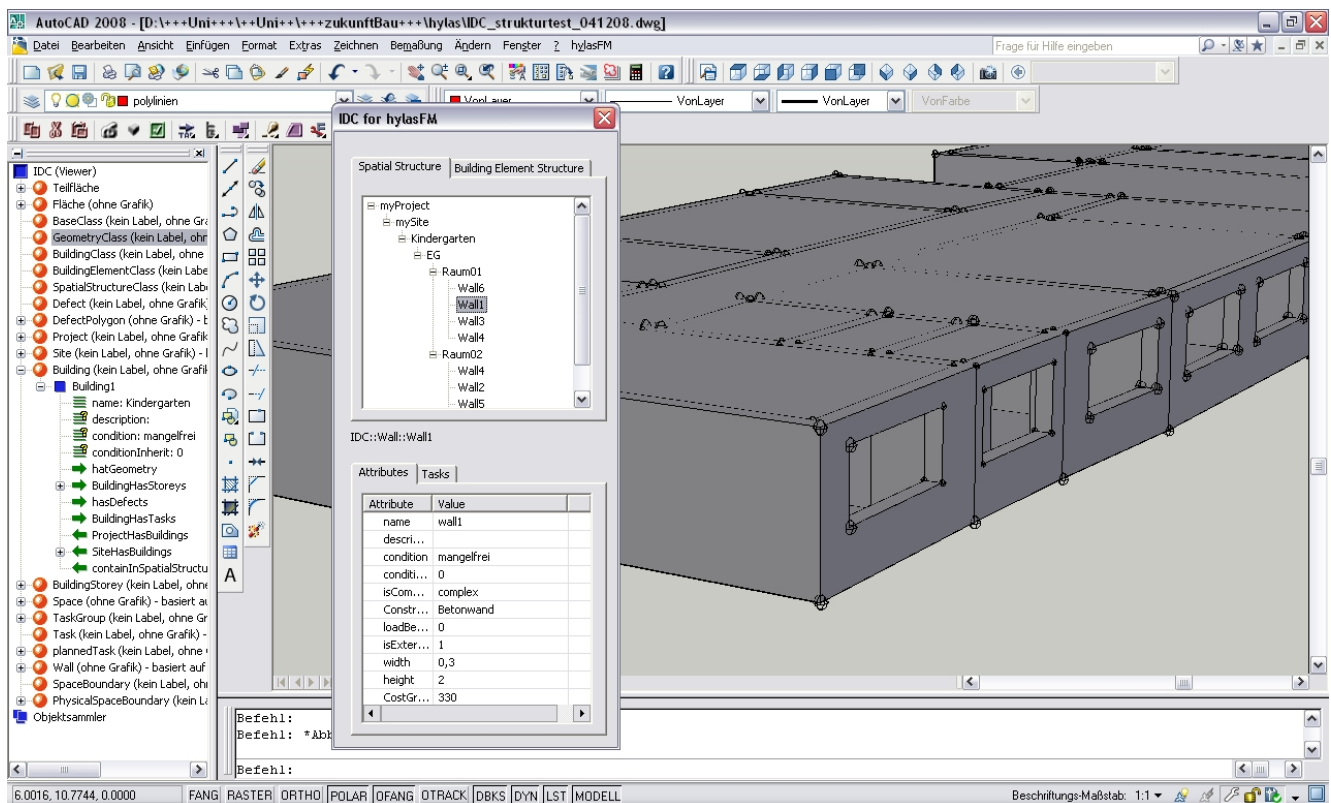


Abb. 17: Prototyp IDC for hylasFM

7.2.2. Implementation

Die Implementation erfolgte über die VBA-Programmierschnittstelle von AutoCad / hylasFM. Die Einbindung von „Fachwissen“ erfolgte mittels SQL-Abfrage in einer externen Access-Datenbank. In dieser wurde ein Datensatz von ca. 250 Maßnahmen mit entsprechender Zuordnung zu Kostengruppennummer, Menge und Kosten hinterlegt.

Die Verknüpfung der HylasFM-Datenstruktur zu dem Modul Ersterfassung erfolgte durch Einbindung von AutoCad / HylasFM als Client in die Experimentalplattform Freak. Die 3D-Geometrie wird dabei innerhalb der AutoCad Zeichnungsumgebung dargestellt und mit den Objekten der HylasFM-Datenstruktur verknüpft. Eine Verknüpfung der Objekte des Freak-MVS mit der HylasFM-Datenstruktur wurde ebenfalls realisiert.

7.2.3. Potential Weiterentwicklung, Ausblick

HylasFM bietet eine flexible Softwareumgebung zum Testen konzeptioneller Datenstrukturen, welche gegenüber dem Modellverwaltungssystem der

Experimentalplattform derzeit einige Vorteile aufweist, u.a. Typsicherheit bei Attributen und Berechnungsattribute. Die im Rahmen des Projektes geschaffene Schnittstelle zwischen Freak und AutoCad / HylasFM wird daher auch im Rahmen weiterer Forschung von Nutzen sein.

7.3. Modul Kosten-Risiko-Analyse

Dieses kleine Modul wurde als Test der konzipierten Auswertung programmiert und sollte sowohl die Verwendung der aufgenommenen Bauteildaten als auch das Rechensystem verifizieren. So ging es zum einen darum, die mögliche Geschwindigkeit der Berechnung zu ermitteln als auch die Fehleranfälligkeit der Berechnung.

7.3.1. Funktionalität

Notwendig war für das Modul das Einlesen und Verbinden von verschiedenen Datenbanken (Maßnahmen und Bauteile), die Anzeige und Veränderung der eingelesenen Datensätze sowie die Simulation und deren Auswertung an sich. Auf besondere grafische Auswertungen oder die Erstellung von Berichten etc. wurde im Rahmen des Forschungsprojektes verzichtet, da die reine Berechnung und statistische Auswertung der Simulation der entscheidende Punkt für das Gesamtsystem waren. Die Ergebnisse von Beispielberechnungen konnten jedoch zur einfachen Visualisierung in Exceltabellen übernommen und so grafisch ausgewertet werden.

7.3.2. Implementation

Das Modul wurde in einem gesonderten VB.net-Programm erstellt und auf Grund der Kürze des Forschungsprojektes noch nicht in die übrige Freak-Umgebung integriert. Das Einlesen von Access-Datenbanken (*.mdb) wurde dabei auch als Ersatz für die Datenübernahme vom Server verwendet. Die erforderliche Funktionalität zur Datenaufnahme und Verwaltung konnte somit gewährleistet werden und stellt für ein endgültiges Modul kein Problem dar.

Maßnahmengruppe		Maßnahme	Kosten mittel in €	EW in %
ID	Kostengruppe			
2	331	M1 keine Änderung	0,00	10
Einheit		M2 Mauerwerk, Fugen freistimmen	23,00	20
m² AWF		M3 Neuverfugen von Mauerwerk	20,50	50
Menge mittel		M4 Sanierputz auf Mauerwerk	57,00	20
9,19		M5	0,00	0
Mengenabw. in %		M6	0,00	0
Modell				
Bezeichnung				
Mauerwerk				

Berechnungen: 1000000

Simulation starten Auswert. zeigen Löschen

Abb. 18: Modul Kosten-Risiko-Analyse: Eingabedialog Maßnahmengruppen

Der Kernpunkt des Testmoduls ist die Simulation und Auswertung des Projektes. Bei einem kleinen Beispielprojekt konnten bereits mit einem einfachen Notebookprozessor (2,0 GHz) hohe Geschwindigkeiten (1.000.000 Durchläufe in ca. 6 s) erreicht werden.

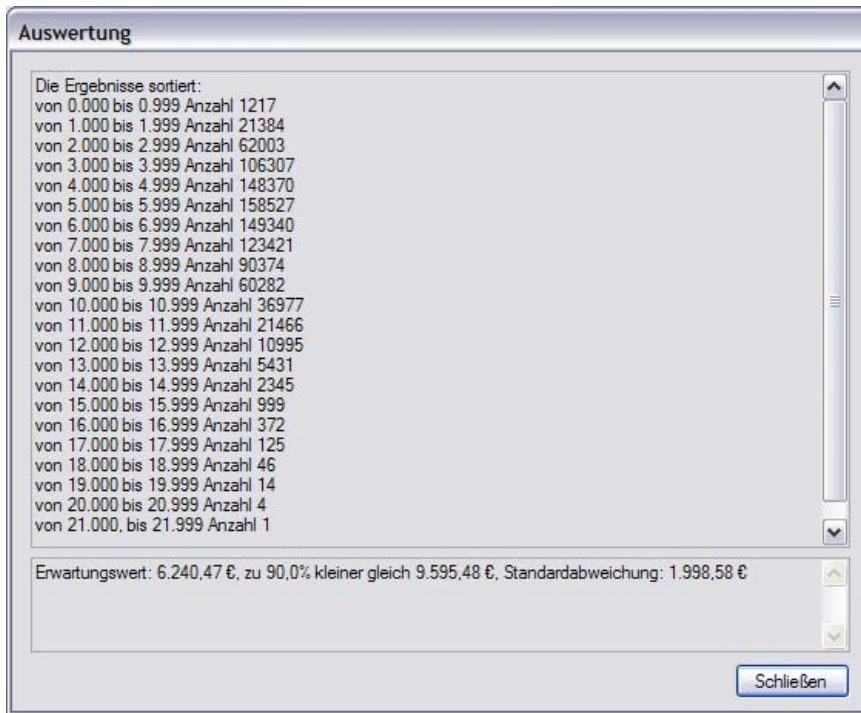


Abb. 19: Ergebnis einer Simulation

7.3.3. Potential, Weiterentwicklung, Ausblick

Das Modul hat gezeigt, dass selbst bei mittleren bis größeren Projekten mit ca. 200-500 aufgenommenen Bauteilen die Berechnungszeiten zwischen 1 und 3 Minuten zu erwarten sind. Bei Programmierung mittels hochperformanter Sprachen (bspw. C++) und Verwendung leistungsfähiger Hardware, können diese weiter reduziert werden. Mit dem Modul konnte gezeigt werden, dass die als Grundlage der stochastischen Auswertung verwendete Monte-Carlo-Simulation für die Kostenermittlung ausreichend präzise ist, und in angemessener Zeit durchgeführt werden kann.

Es erscheint darüber hinaus möglich, eine geringfügig eingeschränkte Auswertung selbst auf einem Kleinstrechner (UMPC) während der Bauaufnahme parallel mitlaufen zu lassen und so zusätzliche Erkenntnisse für den Nutzer zu gewinnen. So könnte sofort auf Bauteile mit hohen Unsicherheiten und der Notwendigkeit zur Informationsverdichtung hingewiesen werden.

Innerhalb eines Gesamtsystems zur Ersterfassung ist die stochastische Auswertung direkt zu integrieren und eine grafische Auswertung parallel zur Erfassung der Bauteile zu gewährleisten.

7.4. Modul CMS / IFC

Die Umsetzung dieses Moduls erfolgte auf Basis des Content Management Systems Plone. Das System wurde entsprechend den Projektanforderungen angepasst. Um IFC-Daten verwalten zu können, wurde ein IFC-Parser entwickelt und in Teilen prototypisch umgesetzt.

7.4.1. Funktionalität und Implementierung

Der Parser für IFC-Dateien gibt die Bauteilbezeichnungen sowie angefügte Mängellisten und Sanierungsmaßnahmen aus, wenn die Methode *ifc_view* ausgeführt wird. Innerhalb von Plone wurde für IFC-Dateien ein eigener Mime-Type (Multipurpose Internet Mail Extension; Spezifikation für Daten, die über das Internet versandt werden können) definiert. (Abb. 20)

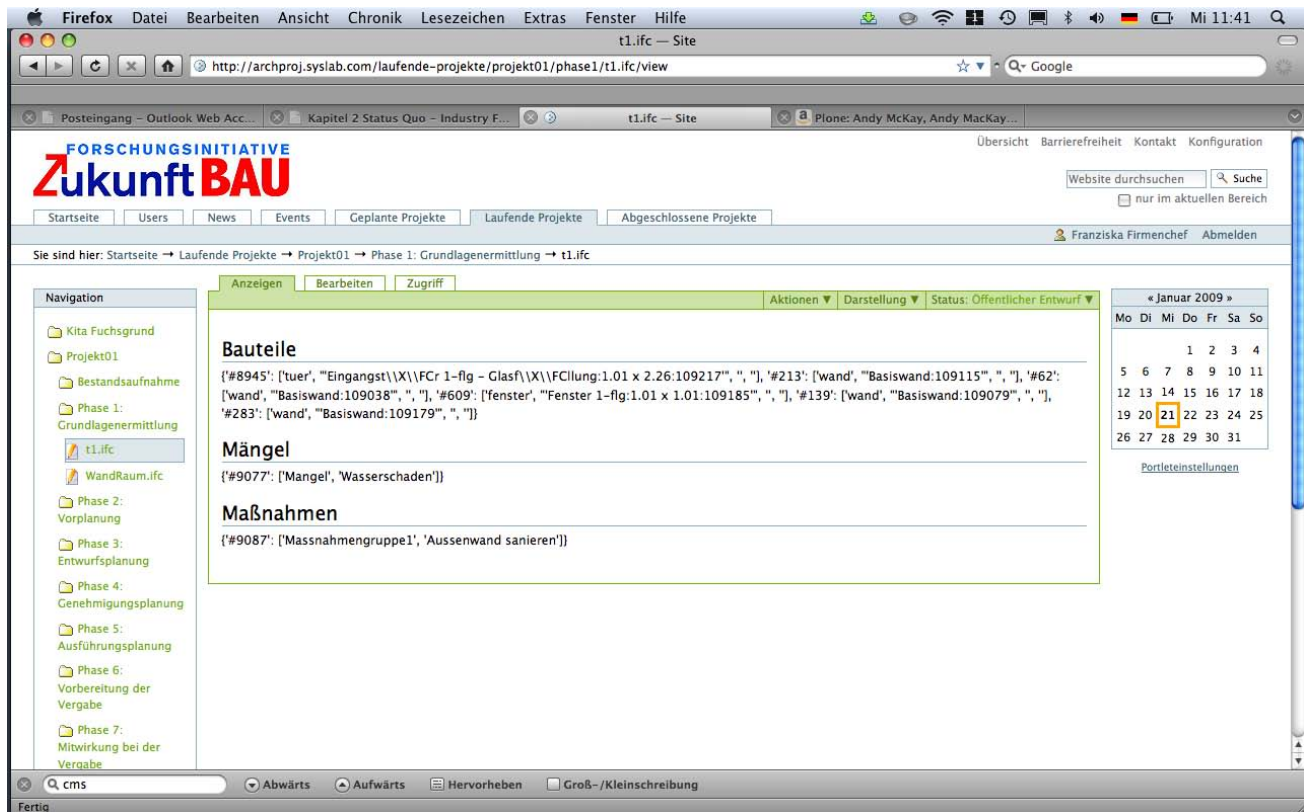


Abb. 20: IFC-Parser innerhalb des CMS: Bauteilbezeichnungen, angefügte Mängellisten und Sanierungsmaßnahmen werden beim Hochladen einer IFC-Datei ausgelesen und in einer Tabelle abgelegt. Dies ermöglicht einen schnelleren Zugriff auf die Daten

Beim Anlegen eines Projektes wird ein Ordner „Ressourcen“ angelegt, der zur Aufnahme beliebiger vorhandener Daten dient. Das können z.B. Bilder von der Baustelle, Textdokumente, CAD-Dateien oder Plandokumente (PDF) sein. Innerhalb dieses Ordners können Unterordner angelegt werden. Weiterhin wird ein Ordner „Bestandsaufnahme“ angelegt, der zur Speicherung der IFC-Dateien mit den Mängellisten dient. Auch dort können beliebige weitere Dateien abgelegt werden. Die ebenfalls erzeugten Phasen 1 bis 9 gemäß der Strukturierung der HOAI dienen zur bequemen Ordnung der Planungen nach der Bauaufnahme. Wenn sie nicht benötigt werden, können sie durch einen Klick auf „Inhalt“, Auswahl der 9 Phasen und „Löschen“ entfernt werden. Das System erscheint wie eine gewohnte Ordnerstruktur und lässt sich auch nahezu genauso einfach bedienen.

Innerhalb des Ordners „Bauaufnahme“ und innerhalb der neun Planungsphasen kann die Methode *index_listing* ausgeführt werden. Dadurch kann ein Projektleiter einen neuen Index vergeben. Alle zuvor vorhandenen Zustände können wiederhergestellt werden, indem man in einem Pulldown-Menü auf den entsprechenden Buchstaben des Index klickt (Abb. 20). Das ist besonders dann von Vorteil, wenn bereits viele Dateien in einer Phase vorhanden sind. Es

werden jeweils nur die Dateien eingeblendet, die innerhalb des ausgewählten Index aktuell sind.

Die Zuweisung bestimmter Rechte konnte mit Hilfe der in Plone existierenden Rechteverwaltung gelöst werden. Es wurden 4 Kategorien oder Rollen angelegt: Firmenchef, Projektleiter, Architekt und Handwerker. Wobei der Firmenchef Administrationsrechte für das gesamte System besitzt, der Projektleiter darf innerhalb eines Projektes alle Manipulationen an Daten sowie die Indizierung vornehmen. Der Architekt darf lesen und schreiben in allen Ordnern und Phasen, der Handwerker darf nur die Dokumente lesen, die für ihn vorgesehen wurden, sowie Angebotsdateien hochladen.

7.4.2. Potential Weiterentwicklung, Ausblick

Das System konnte bis zum Projektabschluss nur mit Testdatensätzen erprobt werden. Ein Test unter realen Bedingungen war aufgrund der begrenzten Projektlaufzeit und dem späten Vorliegen geeigneter Daten aus den anderen Modulen nicht möglich. Ein umfassender Praxistest um Schwachstellen erkennen zu können, wäre daher wünschenswert.

Die Bedienung der Methoden *ifc_view* und *index_listing* kann noch benutzerfreundlicher gestaltet werden, sie ist momentan nur über einen Zusatz in der URL des jeweiligen Ordners möglich. Das ganze System kann so erweitert werden, dass Informationen an die jeweils notwendigen Projektbeteiligten á Block verschickt werden. Mittels der Kalenderfunktion könnten zudem Terminvorgaben gesteuert werden.

8. Evaluation

Die entwickelten Softwareprototypen sollten in einer Evaluationsphase auf Ihre Anwendbarkeit vor Ort getestet werden. Hierzu wurde aus einer Auswahl verschiedener in Frage kommender Objekte, welche durch den Projektpartner AWO Thüringen verwaltet werden, eine Kindertagesstätte in Erfurt ausgewählt.

8.1. Beispielobjekt Kindertagesstätte „Am Fuchsgrund“

Als Evaluationsobjekt stand die Kindertagesstätte „Am Fuchsgrund 32“ in der Thüringer Landeshauptstadt Erfurt zu Verfügung. Die Einrichtung wird seit dem 1.04.2003 durch die Arbeiterwohlfahrt Thüringen e.V. (AWO) als Träger verwaltet. Grundstückseigentümer und Hausbesitzer ist die Stadt Erfurt.

Die Kindertagesstätte wurde am 7.10.1968 als kombinierte Einrichtung aus Kindergruppe für ca. 100 Kinder und Kindergarten für 160 Kinder gegründet. Derzeit besuchen 138 Kinder im Alter von 2 bis 6 Jahren die Einrichtung.



Abb. 21: Luftbildaufnahme der Kindertagesstätte (Quelle: www.bing.com/maps)

8.1.1. Bauform

Bei dem Gebäude handelt es sich um einen für die damalige Zeit typischen Plattenbau. In der Stadt Erfurt existierten zwei weitere Kindertagesstätten gleicher Bauart, die aber bereits wegen Baufälligkeit abgebrochen wurden. Die äußere Bauform besteht aus zwei länglichen, parallel verlaufende Gebäudeteile, welche durch zwei verglaste Riegel verbunden sind. Der Straßen seitige Gebäudeteil (Ost) erstreckt sich über 2 Etagen, nicht unterkellert, während der zweite Gebäudeteil (West) über eine Etage und Keller verfügt. Die beiden Gebäudeteile sind jeweils in ihrer Höhe um ein halbes Geschoss (ca. 1,65 m) versetzt. Gegenläufige Rampen innerhalb der verbindenden Riegel ermöglichen den Zugang zu den Etagen.

Im Obergeschoss des Gebäudeteiles Ost befinden sich die Gruppenräume des ehemaligen Krippenbereiches, während im Erdgeschoss Versorgungs- und Personalräume sowie eine nicht mehr genutzte Küche untergebracht sind. Der Gebäudeteil West schließt mit einer über die gesamte Westseite verlaufenden Terrasse an den Garten an. Im Erdgeschoss sind die Gruppenräume des ehemaligen Kindergartens zu finden.



Abb. 22: Eingangsbereich Gebäudeteil Ost, Innenhof mit Verbindungsriegel, Terrasse Gebäudeteil West (von links nach rechts)

Die beiden Gebäudeteile folgen einem strengen Konstruktionsraster (acht hintereinandergereihte Module, Achsmaß 7,2 m, unterbrochen durch zwei 3,6 m breite Halbmodule in Höhe der Verbinder) und sind aus Stahlbetonplatten gefertigt. Die flachgeneigten Satteldächer sind als Bitumenkiesdach ausgebildet.

8.1.2. Schäden und Mängel

Trotz aktiver Nutzung weist das Gebäude eine größere Zahl baulicher Schäden und Mängel auf. Seit dem Bestehen der Einrichtung wurden nur die nötigsten Instandhaltungsmaßnahmen durchgeführt. Mit der Übernahme der Einrichtung durch die AWO Thüringen im Jahr 2003, erfolgten die Instandsetzung der Fenster, sowie die Modernisierung der Heizungsanlage. In den Jahren 2004-2006 wurde auf Initiative der Kita-Leitung sowie Eltern eine Umgestaltung der Außenanlagen in Eigenarbeit durchgeführt.

Die Gebäudeteile weisen im Fassadenbereich eine große Anzahl an Rissen und Putzabplatzungen auf. Risse sind insbesondere entlang der Stoßfugen der Stahlbetonplatten wahrzunehmen. Die Terrasse entlang des Gebäudeteils West weist große bauliche Mängel in Form von Rissen, Abplatzungen und großflächigen Bodenunebenheiten auf. Diese sind zu einem großen Teil auf das Hochwasser der Gera aus dem Jahr 1994 zurückzuführen, welches bis an die Westseite des Gebäudes führte.

Die beiden Verbinder weisen Undichtigkeiten im Deckenbereich und den großflächigen Verglasungen aus. Die Glaseinfassungen sind z.T. soweit geschädigt, dass Glasflächen fast lose in den Rahmen liegen. Hinzu kommt der mangelhafte Wärmeschutz der aktuellen Verglasung. Die beiden Rampen im Innern der Verbinder weisen eine zu große Steigung von ca. 14% auf und stellen damit einen funktionalen Mangel dar, welcher im Zuge einer Komplettsanierung zu beheben ist.

Die Fenster und Türen der Gebäudeteile Ost und West sind größtenteils in einem schlechten Zustand. Im Zusammenhang mit einer Verbesserung des Wärmeschutzes, ist bei einer Sanierung ein Komplettaustausch vorzusehen. Ebenso ist die Elektrik komplett zu erneuern – hier ist es bereits zu Schmorbränden gekommen – sowie die sanitären Einrichtungen zu modernisieren.

Die Innenräume präsentieren sich in einem vergleichsweise guten Zustand. Dies ist insbesondere auf Eigenleistungen der Mitarbeiter der Einrichtung sowie der Eltern der Kinder zurückzuführen. Dennoch bedarf es der Erneuerung der Fußböden.



Abb. 23: Schadensbilder im Innen- und Außenbereich der Kindertagesstätte (Auswahl)

Im Obergeschoss des Gebäudeteils Ost erstreckt sich im Auflagerbereich der Decke an die Außenwand fast durchgängig ein ca. 5 cm breiter Spalt. Der Ursprung dieses Spaltes ist im Zuge der Sanierungsplanung zu klären. Bei der Begehung vor Ort wurde diese aber zunächst als statisch nicht bedenklich bewertet.

8.2. Testszenarien

8.2.1. Skizzenbasierte Ersterfassung

Getestet wurde die Anwendbarkeit des entwickelten Prototypen *InitialDataCapture* (Kap. 7.1) in realer Umgebung der beschriebenen Kindertagesstätte. Im Fokus des Testes stand insbesondere die Nutzbarkeit der stiftbasierten Eingabe auf einem TabletPC bezogen auf die vorhandene Hardware, sowie die Effektivität der entwickelten GUI und Interaktionsmöglichkeiten (Kap. 5.1 und Kap. 5.3).

Aufgrund der Einschränkungen des entwickelten Prototypen, konnte nur ein Teilbereich des Gebäudes erfasst werden: der südliche Abschnitt der beiden Gebäudeteile (Ost und West) inklusive Verbinder in der Erdgeschossesebene (Abb. 24).

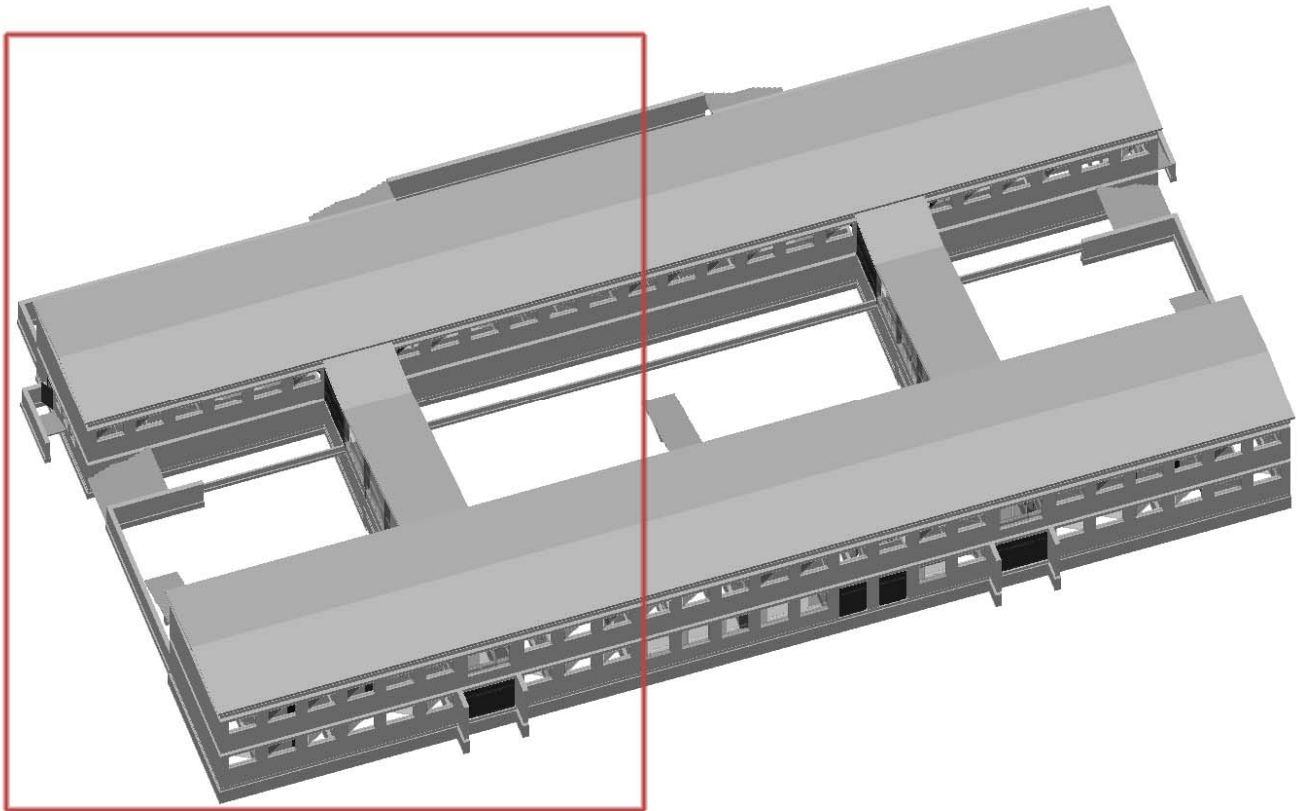


Abb. 24: Bearbeiteter Ausschnitt des Gebäudes⁸

Ein Eingabegerät stand ein TabletPC mit 12,1 Zoll Bildschirmdiagonale und einer Auflösung von 1400 x 1050 Pixeln zur Verfügung. Der große Eingabebereich sowie die großen Schaltflächen des Softwareprototyps erwiesen sich als gut nutzbar bei der Stifteingabe. Ebenso die Texterkennung bei Eingabe von alphanumerischen Daten. Die Arbeit mit dem System ist durchaus auch bei einem noch kleineren Display durchführbar.

Zur Eingabe der Grundrissgeometrie wurde sowohl mit einem leeren Eingabefeld gearbeitet, als auch mit einem hinterlegten unmaßstäblichen Planscan (Funktionalität des Sketch Mode, Kap. 5.3.1). Bei der Eingabe ohne Planscan erwies sich das raumweise Skizzieren als praktikabler (*Bottom up* Prinzip; man nimmt zunächst einen Raum komplett auf und wechselt anschließend zum nächsten), während ein hinterlegter Planscan die Eingabe von zunächst der äußeren Kubatur und anschließender Detaillierung durch Einfügen der Raumabtrennungen begünstigt (*Top down* Prinzip; die Eingabe erfolgt vom Großem zum Kleinen). Im direkten Vergleich erweist sich das *Top Down* Prinzip als weniger fehleranfällig bei der Interpretation der Skizze. Bei der *Bottom up* Eingabe muss der Nutzer wesentlich mehr darauf achten, die Skizze an bereits vorhandenen Knotenpunkten (Schnittpunkte zweier Wandsegmente; vergl. Kap. 4.2.2) fortzusetzen ohne eine fehlerhafte Eingabe durchzuführen. Bei weiterführender Entwicklung der Skizzeninterpretation gilt es an dieser Stelle die Handhabung für den Nutzer zu vereinfachen und Korrekturmöglichkeiten anzubieten. Beide Eingabeprozesse – *Bottom up* als auch *Top down* – sind aber notwendig um eine schnelle effektive Aufnahme zu ermöglichen.

Neben der Eingabefunktionalität wurde des Weiteren die Ableitung des Raum- und Bauteilmodells getestet. Vorausgesetzt der korrekten Skizze erfolgte die Ableitung der

⁸ Das dargestellte Modell entstand im Rahmen der Diplomarbeit von Ronny Möhler zum Thema „Untersuchung zweier Planungsmethoden in der Architektur“, welche parallel zum Projekt bearbeitet wurde.

Räume und raumbegrenzenden Bauteile sowie die Erstellung des 3D-Modells problemlos (Abb. 25).

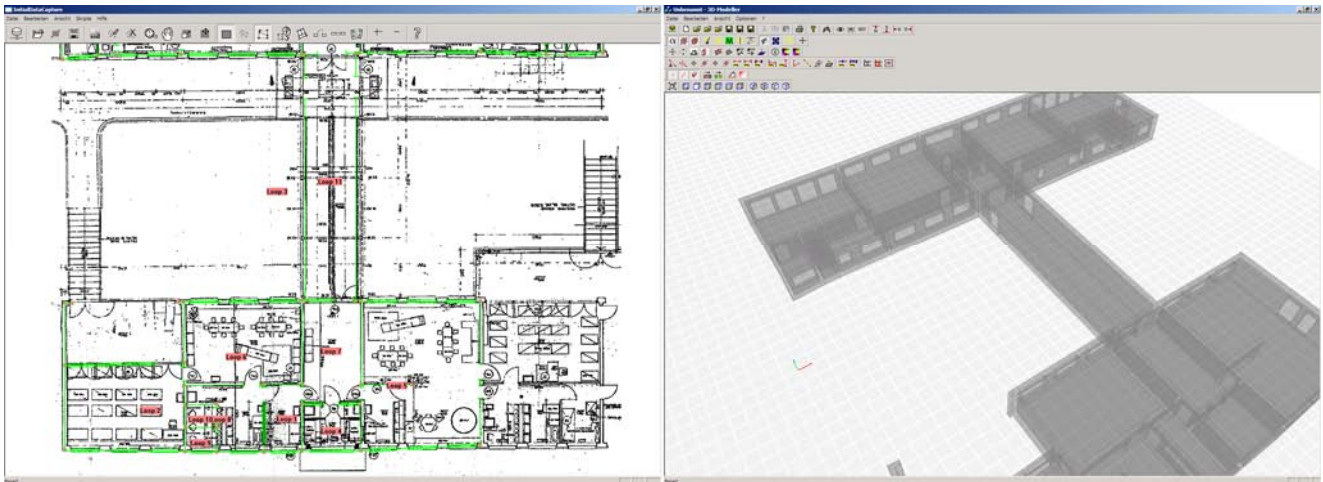


Abb. 25: Angefertigte Skizze über hinterlegten Planscan (links) und resultierendes 3D-Modell nach der Interpretation (rechts).

8.2.2. Aufmaß

Das aus der Skizze abgeleitete 3D-Modell ist Ausgangspunkt für das Gebäudeaufmaß. Wie in der Konzeption erläutert, ist das Aufmaß nicht obligatorisch (vergl. Kap. 5.3). Eine Auswertung nach Kosten und Risiken kann in der Initiierungsphase des Projektes auch anhand von geschätzten bzw. unpräzisen Maßen vorgenommen werden.

Die vorgeschlagene Systemkonzeption gliedert sich allerdings nahtlos in ein bereits bestehendes experimentelles Softwareframework ein, welches auch Applikationen zur geometrischen Bauaufnahme umfasst (vergl. Kap. 7).

Im Rahmen der Evaluierung wurde auf ein komplettes Aufmaß verzichtet. Diese Funktionalität war nicht Bestandteil des Forschungsprojektes, sondern wurde aus vorangegangenen Projekten übernommen. Dennoch erfolgte eine Ausgleichung der abgeleiteten 3D-Geometrie mittels geometrischer Filter nach Parallelität, Orthogonalität und vorhandenen Ebenen, sowie die Anpassung an aufgenommene Maße der Außenkubatur. Diese Ausgleichung erfolgte in den Softwareprototypen *2D-Modeller* bzw. *3D-Modeller*. Wie auch der Prototyp *InitialDataCapture* sind diese Applikationen Bestandteil der *Experimentalplattform Freak* und greifen daher auf das gleiche Datenmodell zurück (vergl. Kap. 7.1.2).

Abbildung Abb. 26 (links) zeigt die aus der Skizze resultierende Gebäudegeometrie als 2D-Ansicht. Diese weist erhebliche Unregelmäßigkeiten auf, welche aus der freihändigen Eingabe der Skizze resultieren. Nach Filterung der Geometrie nach geometrischen Abhängigkeiten und Ausgleichung ergibt sich ein bereinigtes Bild. Anschließend erfolgten händische Messungen (mit Hilfe eines Laserdistanzmessers) an den Außenkanten des Gebäudes. Das Ergebnis des ausgeglichenen 3D-Modells nach Anwendung der Filter und Messungen ist in der Abbildung Abb. 26 (rechts) zu sehen.

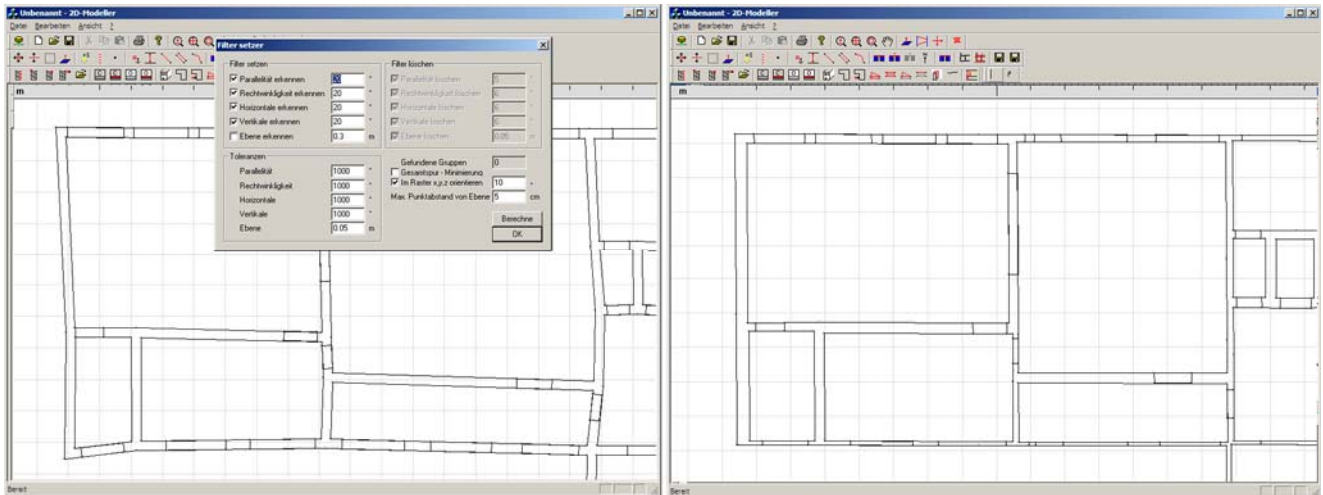


Abb. 26: Resultierendes Modell nach Interpretation der Skizze (links) und Ergebnis nach Ausgleichung (rechts)

8.2.3. Datenmodell und IFC Export

Die Weiterverarbeitung der Daten erfolgte mit Hilfe des Prototypen *IDC for hylasFM* (Kap. 7.2). Die Software hylasFM diente dabei als Modellverwaltungssystem und erlaubte die flexible Modellierung und schnelle Anpassung des konzipierten Raum- und Bauteilmodells. Durch die Umsetzung des Prototyps als Client der Experimentalplattform Freak, war eine direkte Weiterverarbeitung der Daten aus der skizzenbasierten Erfassung und dem Aufmaß möglich. Aus der Interpretation der Skizzeneingabe des Prototypen *InitialDataCapture* resultierte das Raum- und Bauteilmodell, welches direkt mit der Geometrie verknüpft ist (Abb. 27).

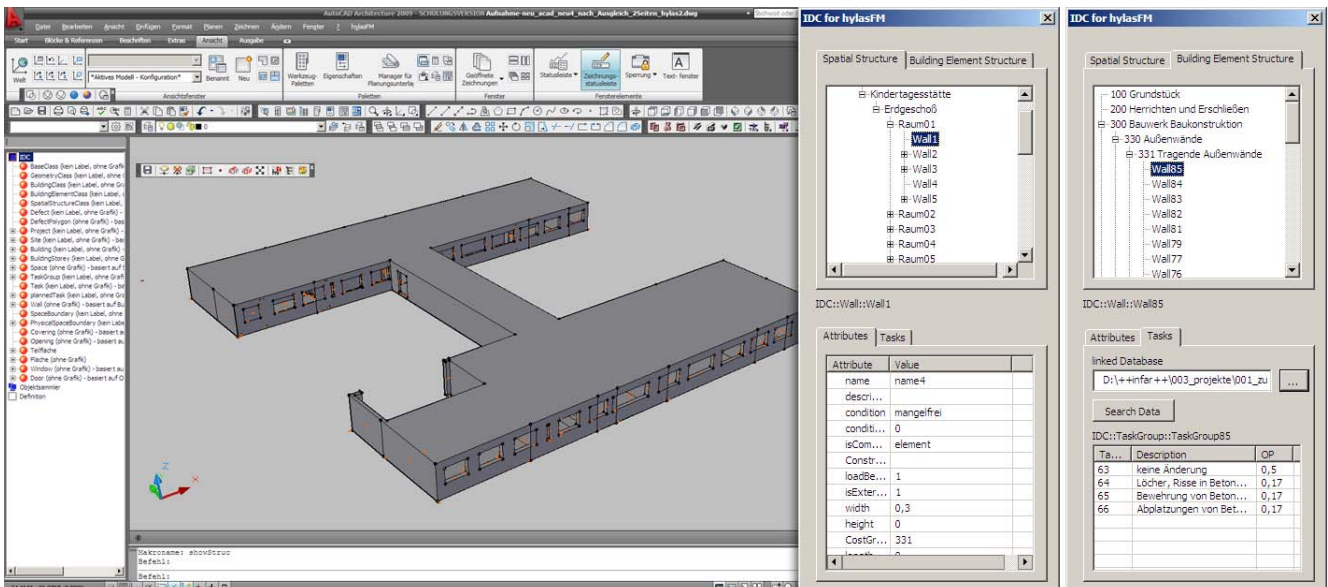


Abb. 27: 3D-Darstellung des aufgenommenen Modells im Prototyp hylasFM (links) sowie Darstellung der räumlichen Struktur (mitte) und Bauteilstruktur nach DIN 276 (rechts)

Wie in Kapitel 4.3 beschrieben, orientiert sich der Aufbau des Raum- und Bauteilmodells in grundlegenden Strukturen an den Industry Foundation Classes, wodurch ein Export in das IFC-Format realisiert werden konnten. Abb. 28 zeigt das exportierte Ergebnis des Aufmaßes

in einem IFC-Viewer⁹. Ebenfalls zu erkennen ist die exportierte räumliche Struktur und die Beziehung dieser zu den Bauteilen.

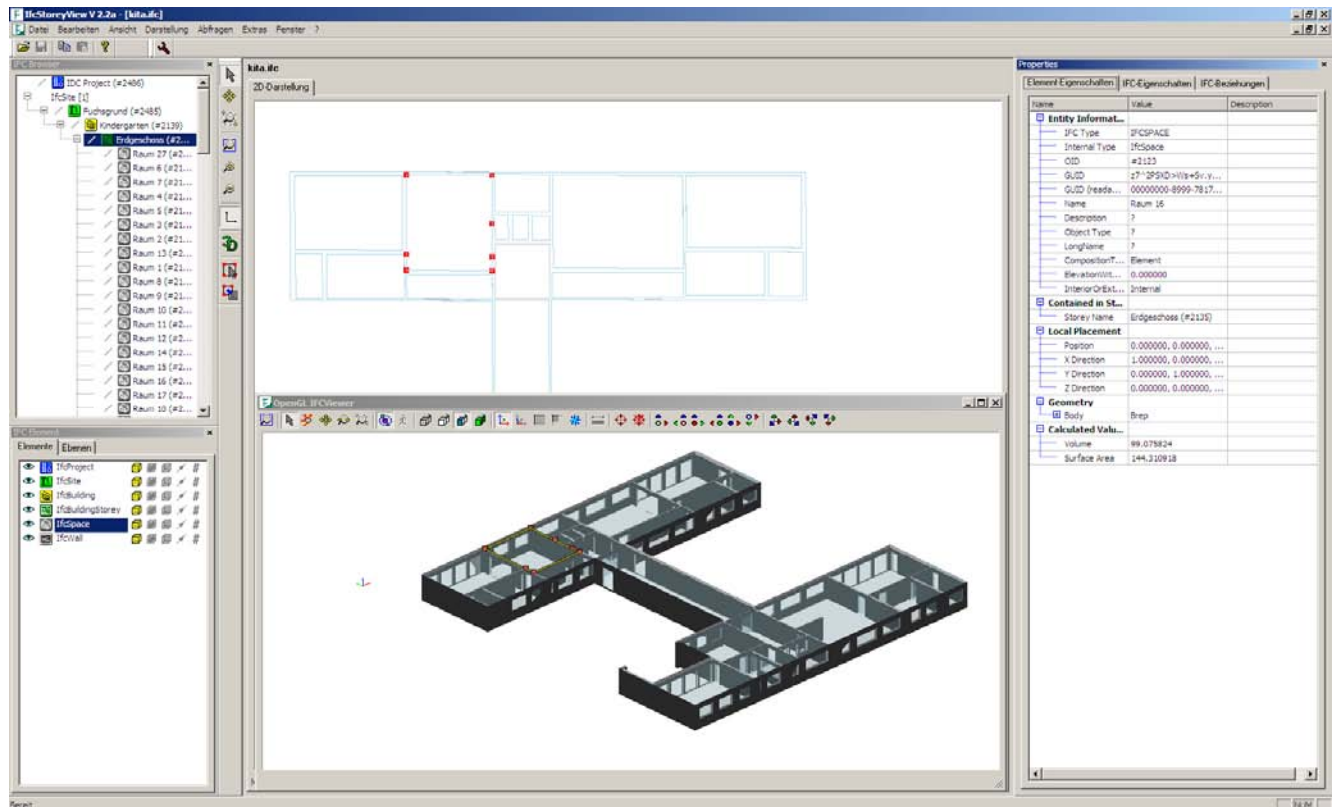


Abb. 28: Exportierte IFC-Datei des aufgenommenen Kindergartens in einem IFC-Viewer

8.2.4. Integration von Fachwissen und Kostenauswertung

Über den Prototypen erfolgte ebenso die Anbindung an eine externe Datenbank zur Auswahl praxisüblicher Maßnahmen. Maßgeblich für die Datenbanksuche sind die Kostengruppennummer sowie die Konstruktionsart des gewählten bzw. der gewählten Elemente des Bauteilmodells. Entsprechend dieser Attribute erfolgt eine Filterung des Datensatzes, so dass nur die relevanten Maßnahmen dem Nutzer angezeigt werden. Im Testszenario erfolgte sowohl die Zuordnung von Maßnahmen zu einem Bauteil (Bsp.: Innenwand + tragend = KGr 341; Konstruktionsart: Mauerwerk), als auch zu einer Kostengruppenkategorie (Bsp: alle tragenden Außenwände = KGr 331; Konstruktionsart: Betonwand). Die Mehrfachauswahl über die Kostengruppenkategorie erwies sich als sehr praktikabel, da schnell für alle gleichartigen Bauteile entsprechende mögliche Maßnahmen zugeordnet werden können. Die Einzelauswahl zeigt sich dann als zweckmäßig, wenn bei der Erstbegehung spezifische Mängel vorgefunden werden, auf die gesondert reagiert werden muss (mit spezifischen Maßnahmen).

⁹ Verwendet wurde der freie IFC-Viewer *IfcStoreyView* des Forschungszentrums Karlsruhe, www.iai.fzk.de

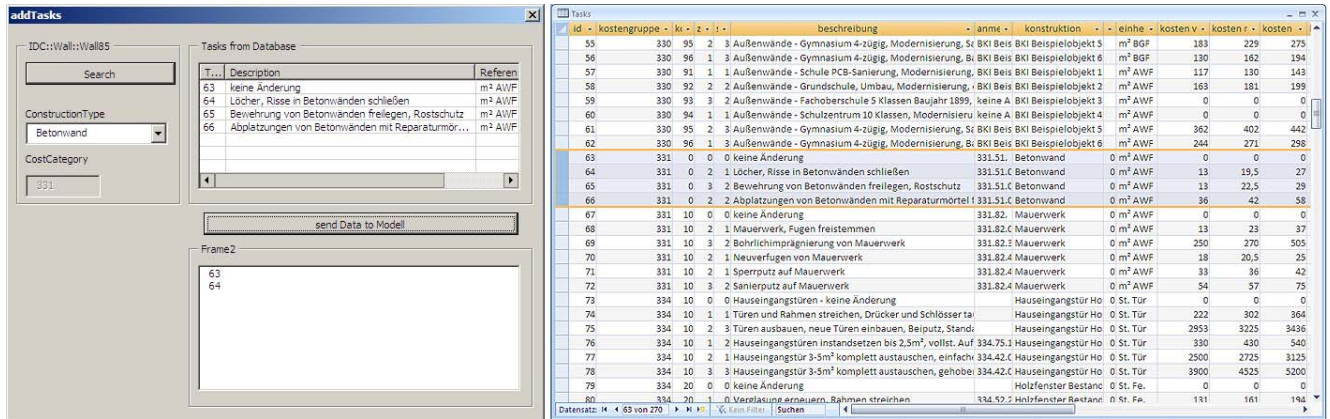


Abb. 29: Auswahldialog Maßnahmen (links) und Darstellung der abgerufenen Datensätze aus der Maßnahmendatenbank (rechts)

Abschließend erfolgte die Übergabe aller Maßnahmengruppen an das Modul Kosten-Risiko-Analyse und eine entsprechende Berechnung der zu erwartenden Kosten. Da nur ein begrenzter Gebäudeausschnitt erfasst wurde, erfolgte keine komplette Bewertung der zu erwartenden Kosten für eine Sanierung. Der Test zeigt aber anschaulich die Möglichkeiten der Auswertestrategie.

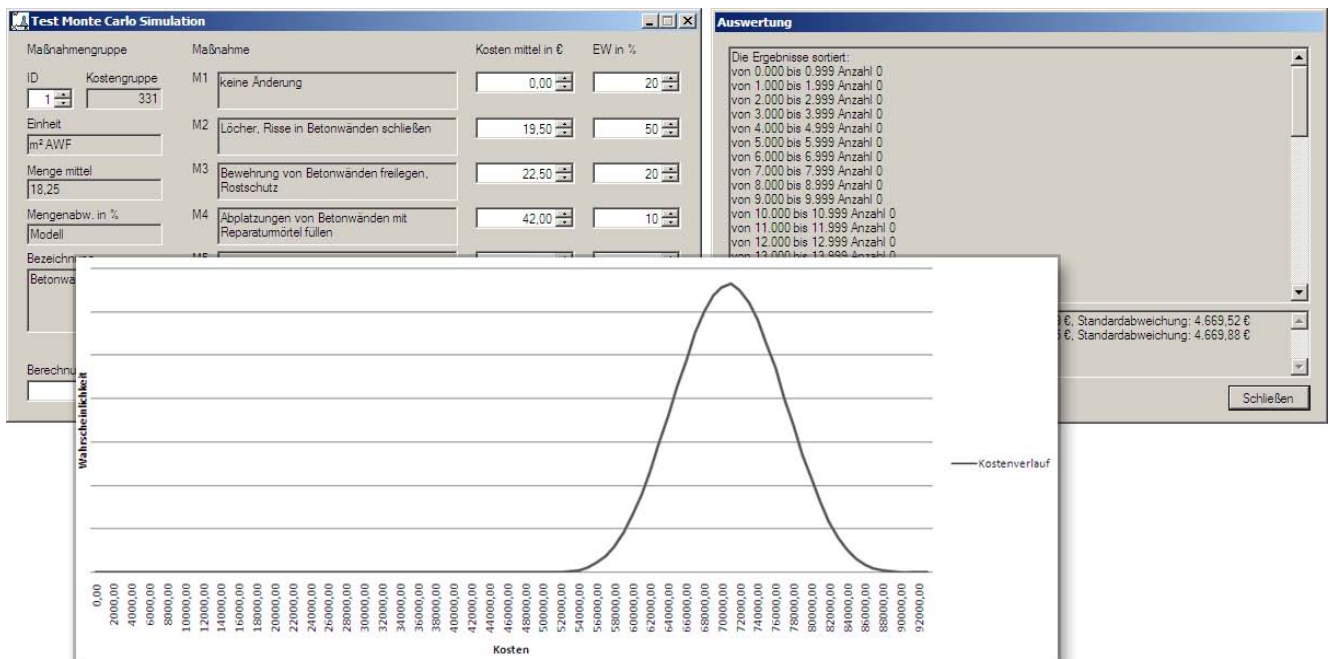


Abb. 30: Ergebnisse der Kostenbewertung

9. Zusammenfassung, Ausblick und offene Fragestellungen

Das Projekt zielte auf die Entwicklung eines Computersystems zur schnellen Ersterfassung von für die Planung im Bestand relevanter Gebäudedaten. Die vorgestellte Vorgehensweise sollte dabei möglichst bestehende Arbeitsweisen adaptieren, jedoch eine digitale Unterstützung erfahren. Wesentliches Merkmal dieser Unterstützung ist die Erfassung verschiedener planungsrelevanter Daten und Speicherung in einem digitalen Gebäudemodell. Dieses ermöglicht die Fortschreibung der Daten ohne Informationsverlust im weiteren Verlauf des Planungsprozesses und darüber hinaus im gesamten Lebenszyklus eines Gebäudes. Aktuelle Trends und Entwicklungen bei Software für Architekten und Ingenieure bestätigen die Notwendigkeit solcher Gebäudemodelle.

9.1. Datenmodellierung / Bauwerksmodell

Das Erstellen von Gebäudemodellen im Zuge einer Bauaufnahme ist mit aktuellen Systemen derzeit nur eingeschränkt oder auch gar nicht möglich. Das Forschungsvorhaben steigt in dieser Frage direkt zu Beginn der Projektarbeit einer Sanierungsplanung ein. Bereits bei der Erstbegehung eines Bestandsgebäudes – noch vor der Entscheidung für oder gegen eine Sanierung – werden bereits Informationen über das Gebäude in einem digitalen Gebäudemodell abgelegt und für erste Kosten- und Risikoanalysen bereitgestellt. Die hierfür konzipierte Klassenstruktur des Raum- und Bauteilmodells, stellt zwar ein auf den Anwendungsfall angepasstes Modell dar, dennoch wurde eine Weiterverwendung im Planungsprozess sowie die Kompatibilität zur IFC-Spezifikation berücksichtigt.

Im Kapitel 4.1.2 wurde kurz die Notwendigkeit der Verwendung von dynamischen Gebäudemodellen diskutiert. Diese bilden die Struktur eines Gebäudemodells nicht wie bisher statisch ab, sondern sind offen für spätere notwendige Erweiterung aufgrund von neuen oder geänderten Projektanforderungen. Im Rahmen des BMBF geförderten Projektes nubau¹⁰, wird derzeit an der Bauhaus-Universität Weimar ein solches dynamisches Modell in Form eines verteilten Produktmodells entwickelt, welches als Grundlage für die Zusammenarbeit verschiedener am Bau beteiligter Fachdisziplinen dient. Das im Rahmen des hier vorgestellten Forschungsvorhaben entwickelte Bauwerksmodell erfährt seine Umsetzung als Teilmodell dieses verteilten Produktmodells, und dient damit als Grundlage für weitere Applikationen für Bauaufnahme und Planung im Bestand.

Als zertifizierter Standard sind die IFC bei der Weitergabe von Gebäudemolldaten mittlerweile unumgänglich. Die besonderen Anforderungen, die sich bei der Abbildung von Bestandsdaten in einer IFC-Datei ergeben, wurden bisher jedoch nicht näher betrachtet. Im Projekt wurden hierfür erste Vorschläge erarbeitet und auf Difizite aufmerksam gemacht (bspw. die derzeit nicht vorhandene Möglichkeit der Abbildung von Mängeln). Hier bedarf es weiterer Forschung. Wünschenswert wäre die Erarbeitung einer *IFC Model View Definition* (MDV) für die Abbildung von Gebäuden aus einer Bestandserfassung heraus. Die IFC-MDVs beschreiben die Basisanforderungen an einen Austausch von Gebäudemolldaten im IFC Format für einen spezifischen Anwendungsfall in der Bauindustrie¹¹. Als Beispiel sei hier der

¹⁰ Methoden und Werkstoffe für die nutzerorientierte Bausanierung; www.nubau.de

¹¹ Vergl. <http://www.blis-project.org/IAI-MVD/>

IFC Basic FM HandOver view genannt, welcher derzeit als Grundlage für den Datenaustausch im Facility Management durch die BuildingSmart definiert wird¹².

9.2. Skizzenbasierte Ersterfassung

Die Erstbegehung als Vorbereitung einer Sanierungsplanung stellt einen wesentlichen Arbeitsschritt dar, da hierbei erste Entscheidungen über den weiteren Verlauf der Projektbearbeitung getroffen werden. Gleichwohl erfolgt die Erstbegehung meist unter Zeit- und Kostendruck. Ein Ziel des Forschungsvorhabens war es, diesen Arbeitsschritt durch eine digitale Datenerfassung zu unterstützen, und dennoch eine schnelle und effektive Arbeitsweise anzubieten. Die stiftbasierte Interaktion auf einem UMPC versucht dabei traditionelle Arbeitsweisen zu adaptieren, um die Nutzung digitaler Werkzeuge vor Ort nicht als Behinderung wahrzunehmen. Eine Interpretation der Skizze, wie sie im Capture Mode konzeptionell vorgesehen und im Prototypen InitialDataCapture umgesetzt wurde, erwies sich hierfür als sehr vielversprechend, wie die Testszenarien vor Ort gezeigt haben.

Die Vorgehensweise des intuitiven Skizzierens von Grundrissen bei gleichzeitigem Anlegen einer Raum- und Bauteilstruktur ist dabei nicht nur auf den Anwendungsfall der Erstbegehung begrenzt. Die Methodik ist auch auf den Entwurfsprozess bei der Neubauplanung übertragbar. Die große Resonanz bei Präsentationen des Prototyps auf diversen Konferenzen und im Gespräch mit Fachkollegen hat dies bestätigt.

9.3. Kosten-Risiko-Analyse

Die Testszenarien haben gezeigt, dass die vorgeschlagene Kostenbewertung mit Hilfe der Monte-Carlo-Simulation schnell bewertbare Ergebnisse erzielen kann. Die Qualität der Ergebnisse hängt allerdings stark von der Zuverlässigkeit der eingegebenen Werte für die Maßnahmenvarianten ab. Wie bereits dargelegt, existiert mittlerweile eine große Anzahl von Quellen, in welchen Maßnahmen für typische Problemstellungen bei einer Sanierung mit Kosten hinterlegt sind.

Im Rahmen einer kommerziellen Nutzung sollte es von Interesse sein, ein möglichst breites Spektrum vorhandener Quellen einzubeziehen. Über die beschriebene Datenbankabfrage mittels Filterung der erfassten Bauteile (anhand der Kostengruppenzuordnung sowie Konstruktionsart und Zustand der Bauteile), sollte es kein Problem sein auch verschiedene Datenquellen in Kombination einzusetzen.

Diese Integration von Fachwissen kann aber nur als ein Schritt betrachtet werden. Der Fokus innerhalb eines solchen Systems, muss die Unterstützung des Aufnehmenden/ Planenden bei der Vermeidung von Unwissenheit sein. Dies kann zum Beispiel über Hinweise zu bekannten Problemen bei bestimmten Konstruktionen und über Vorschläge von Maßnahmen erfolgen, die eventuell notwendig werden könnten aber vom Nutzer nicht erkannt wurden. Dies bedarf aber der Abbildung von Abhängigkeiten zwischen bestimmten Maßnahmen untereinander. Hinzu kommt die Integration von Regeln, welche sich u.a. aus Normen und Gesetzen ergeben. Die Abbildung solcher Abhängigkeiten und Integration in ein solches System sollte Schwerpunkt weiterer Forschung sein.

¹² Vergl. http://www.iai-tech.org/products/ifc_specification/ifc-view-definition/basic-fm-handover-view/basic-fm-handover-view-summary

10. Abkürzungen und Synonyme

BIM	Building Information Model bzw. Building Information Modelling (Gebäude-Informationen-Modell bzw. -modellierung)
CA(A)D	Computer Aided (Architectural) Design
CMS	Content Management System
GUI	Graphical User Interface (Grafische Benutzeroberfläche)
HOAI	Honorarordnung für Architekten und Ingenieure
IFC	Industry Foundation Classes
Mock-up	Synonym für einen funktionsfreien Prototypen
MVS	Modellverwaltungssystem
SDK	Software Development Kit
UMPC	Ultra Mobile Personal Computer
VBA	Visual Basis for Applications (weit verbreitete Skript- Programmiersprache)
WIMP	Window, Icon, Menue, Pointing Device Synonym für einen allgemein gängigen Standard beim der Gestaltung von grafischen Benutzeroberflächen

11. Referenzen

- [Biffl04] Biffl, S., und M. Heindl, C. Kozarits, K. Schmidt: Skriptum zur Lehrveranstaltung Risikomanagement VU, Wien 2004, URL: qse.ifs.tuwien.ac.at/courses/RM/RMVU_SkriptumSS2004_MH_040226.pdf. Stand 11.2004
- [BKI08] Baukosteninformationszentrum Deutscher Architektenkammern, <http://www.bki.de>, Stand 12.2008
- [Blecken07] Blecken, U. und Hasselmann, W.: Kosten im Hochbau – Praxishandbuch und Kommentar zur DIN 276; Verlagsgesellschaft Rudolf Müller, Köln 2007
- [Depenbrock02] Depenbrock, F. H. und Vogler O.: Honorarordnung für Architekten und Ingenieure HOAI, Bundesanzeiger Verlag, 2. Auflage; Köln 2002
- [DIN1356] DIN Deutsches Institut für Normung e.V.: DIN 1356-6 - Technische Produktdokumentation Bauzeichnungen, Teil 6 – Bauaufnahmezeichnungen, Entwurf. Beuth Verlag GmbH, Berlin, 2003
- [DIN276] Deutsches Institut für Normung e.V.: DIN 276 – Kosten im Hochbau, Beuth Verlag, Berlin, 2006
- [Fehlhaber05] Fehlhaber, D.: Kosten-/Risiko-Analyse-System für Sanierungsprojekte; Diplomarbeit, nicht veröffentlicht, Weimar 2005
- [Frey01] Frey, H. C. und Nießen, G.: Monte Carlo Simulation, Gerling Akademie Verlag, München, 2001
- [Gleißner99] Gleißner, W. und Meier, G.: Risikoaggregation mittels Monte-Carlo-Simulation, in: Versicherungswirtschaft, Heft 13/1999; Verlag Versicherungswirtschaft; Karlsruhe 1999
- [Gondring07] Gondring, H.: Risiko Immobilie – Methoden und Techniken der Risikomessung bei Immobilieninvestitionen; Oldenbourg Wissenschaftsverlag, 2007
- [Hauschild03] Hauschild, T.: Computer Supported Cooperative Work-Applikationen in der Bauwerksplanung auf Basis einer integrierten Bauwerksmodellverwaltung, Weimar, Bauhaus-Universität, Fakultät Bauingenieurwesen Dissertation, 2003,
- [HOAI08] Neuenfeld, et. al.: Honorarordnung für Architekten und Ingenieure, Kohlhammer, Stuttgart, 2008
- [IAI09_1] International Alliance for Interoperability. URL: <http://www.iai-international.org>. Stand 01.2009
- [IAI09_2] Industrieallianz für Interoperabilität e.V. – IAI. URL: <http://www.buildingsmart.de>. Stand 01.2009
- [IFC09] IAI; Liebich, T. (Hrsg.): IFC2x Edition 3. Technical Corrigendum 1. URL: <http://www.iai-tech.org/ifc/IFC2x3/TC1/html/index.htm>, Stand 01.2009

- [Kraft02] Kraft, H.: Baupreisermittlung für Bauschadenssachverständige, Lehrbrief für Bausachverständige; ASD – Arbeitskreis im BWA zur Aus- und Weiterbildung von Sachverständigen in Deutschland; 2002
- [Luchsinger02a] Luchsinger, C. J.: Statistische Methoden – Grundlagen der Statistik, 2002, www.luchsinger-mathematics.ch/sm3.pdf, Stand Dezember 2004
- [Luchsinger02b] Luchsinger, C. J.: Statistische Methoden – Schätztheorie und Konfidenzintervalle, 2002, www.luchsinger-mathematics.ch/sm5.pdf, Stand 12.2004
- [McCay05] McCay, A.: Plone, Addison-Wesley, München, 2005
- [Microsoft08] Microsoft: Tablet PC SDK. URL <<http://www.microsoft.com>>, 2005. zuletzt abgerufen 05/2008
- [Neddermann07] Neddermann, R.: Kostenermittlung im Altbau – Aktuelle Baupreise, rechtliche Grundlagen, technische Beurteilung, Kostenermittlungsmethoden; Werner Verlag, Köln 2007
- [Petzold01] Petzold, F.: Computergestützte Bauaufnahme als Grundlage für die Planung im Bestand - Untersuchungen zur digitalen Erfassung und Modellbildung, Weimar, Bauhaus-Universität, Fakultät Architektur, Dissertation, 2001
- [Schach01] Schach, R.: Baukosten: Kostensteuerung in Planung und Ausführung; Springer-Verlag Berlin; Berlin 2001
- [Schmitz04] Schmitz, H.: Baukosten 2004. Verlag für Wirtschaft und Verwaltung Hubert Wingen Essen, 2004
- [Schmitz08] Schmitz, H.: Baukosten 2008. Verlag für Wirtschaft und Verwaltung Hubert Wingen Essen, 2008
- [Schmuck02] Schmuck, N.: Diplomarbeit - Konzeption einer projektorientierten Risikoanalyse zur Auftragsselektion in der Bauindustrie unter Verwendung der „Monte Carlo Simulation“; Diplomarbeit, nicht veröffentlicht; Weimar 2002
- [Schulte02] Schulte, K.-W., und Bone-Winkel, S., Fischer, C.: Handbuch Immobilien-Projektentwicklung; Immobilien Informationsverlag Rudolf Müller; 2. Auflage; Köln 2002
- [SFB524_05a] SFB 524, D2 (Donath/Petzold): Bauplanungsrelevantes digitales Gebäudeaufnahme- und Informationssystem, Sonderforschungsbereich SFB 524 - Werkstoffe und Konstruktionen für die Revitalisierung von Bauwerken, Arbeits- und Ergebnisbericht für die 1. Förderperiode 07'99-06'02, Weimar, Februar 2002
- [SFB524_05b] SFB 524, D2 (Donath/Petzold): Bestandsorientierte Unterstützung der Planung, Sonderforschungsbereich SFB 524 - Werkstoffe und Konstruktionen für die Revitalisierung von Bauwerken, Bericht der 2. Förderperiode 07'02-06'05 und Antrag auf Finanzierung der 3. Förderperiode 07'05-06'09, Band 2/2, Weimar, Januar 2005

- [SFB524_05c] SFB 524, D3 (Hübler): Digitales Bauwerksmodell als Grundlage der Prozessintegration, Sonderforschungsbereich SFB 524 - Werkstoffe und Konstruktionen für die Revitalisierung von Bauwerken, Bericht der 2. Förderperiode 07'02-06'05 und Antrag auf Finanzierung der 3. Förderperiode 07'05-06'09, Band 2/2, Weimar, Januar 2005
- [Sirados08] Sirados Baudaten, www.sirados.de, Stand 12.2008
- [Smith99] Smith, N. J.: Managing Risk in Construction Projects; Blackwell Science Ltd., Oxford 1999
- [Thurow04] Thurow, T.: Digitaler Architekturbestand - Untersuchungen zur computergestützten, schrittweisen Erfassung und Abbildung der Geometrie von Gebäuden im Kontext der planungsrelevanten Bauaufnahme. Weimar, Bauhaus-Universität, Fakultät Architektur, Dissertation, 2004
- [Thurow08] Thurow, T.: Kombination verschiedener Fachaspekte auf Basis eines dynamisch erweiterbaren Bauwerksmodells. Beitrag zum 1.nuBau - Workshop, Weimar, Bauhaus-Universität, 07.11.2008
- [Werner03] Werner, B.: Einführung in elementare Stochastik – Wahrscheinlichkeitsrechnung und Statistik, Hamburg 2003, www.math.uni-hamburg.de/home/werner/Stochastik.pdf, Stand Dezember 2004
- [Willenbacher02] Willenbacher, H.: Interaktive verknüpfungsbasierte Bauwerksmodellierung als Integrationsplattform für den Bauwerkslebenszyklus, Weimar, Bauhaus-Universität, Fakultät Bauingenieurwesen, Dissertation, 2002,
- [Willenbacher04] Willenbacher, H. & Hübler, R.: Intelligent Link-Management for the Support of Integration in Building Life Cycle International Conference on Computing in Civil and Building Engineering, ICCCBE, Weimar 2004

Anhang

A Eigene Veröffentlichungen

Donath, D., Braunes, J., Fehlhaber, D., Tauscher, H.:

Ein IFC Gebäudemodell für die CAAD integrierte Planung im Bestand. In: R. Windisch, G. Faschingbauer, P. Katranuschkov (Hrsg.), Forum Bauinformatik 2008 - Junge Wissenschaftler forschen, Institut für Bauinformatik, Fakultät Bauingenieurwesen, TU Dresden, 2008, pp. 19-26

Donath, D., Braunes, J., Fehlhaber, D., Tauscher, H.:

Sketch-based building information capture for cost and risk assessment in the early stages of planning in existing built contexts. In: M. Muylle (eds.), Architecture 'in computro' - eCAADe conference, Antwerp, Belgium, 2008, page 109-116

Goettig, R., Braunes, J.:

Building Survey in Combination with Building Information Modelling for the Architectural Planning Process. In: G. Çağdaş and B. Çolakoğlu (eds.), Computation - The new realm of architectural design, eCAADe conference, Istanbul, Turkey, 2009, pp. 69-74

Projektdokumentation, Prototypen und aktuelle Weiterentwicklungen sind auf der Webplattform *caupa* (<http://www.caupa.de/projects/zukunftBau>) zusammengefasst.

B Funktionsumfang des Softwarekonzeptes / Capture-Mode

Funktionsgruppe	Funktionalität	Funktionalität englisch		Befehlsaufruf						Eingabe Parameter		Ergebnis	
			Modusänderung?	Befehl	Befehl englisch	Icon	Pos. (Kreis)menu	Geste	anderer Aufruf	direkte Interaktion in der Repräsentation des Modells	anderer Interaktionsbereich	Feedback in der Benutzungs- Oberfläche	Modelländerung
Eingabe	neues Projekt anlegen / öffnen	create / open new project	-	Projekt	project	+	-		+ Symbol in SF				
	neues Gebäude anlegen	create new building	-	Gebäude	building	+	-		+ Symbol in SF				
	neues Geschoss anlegen	create new level	-	Geschoss	level	+	-		+ Symbol in SF				
	Wände zeichnen	draw walls	x	Wand	wall		2.1.			Linie zeichnen			
	virtuelle Raumbegrenzungen zeichnen	draw virtual room boundaries	x	Grenze	boundary		2.2.			Linie zeichnen			
	Durchbruch zeichnen	draw opening	x	Durchbruch	opening		2.3.						
	Türen zeichnen	draw doors	x	Tür	door		2.4.			Punkt auf Wand antippen			
	Fenster zeichnen	draw windows	x	Fenster	window		2.5.			Punkt auf Wand antippen			
	darunterliegende Bauteile kopieren	copy subjacent building elements	x	Kopieren	copy		2.8.			überzeichnen der darunterliegenden Objekte		Ink von unten übernehmen? oder nur interpretierte Geometrie	
	Vertikalschließung Var. A (nach unten)	vertical circulation Var. A	x	Vertikalraum	vertical room					Raum auswählen (s.u.)		Markierung Luftraum in der Grundriss-Skizze	Raumattribut Anzahl Geschosse
	Vertikalschließung Var. B (nach oben)	vertical circulation Var. B	x	Vertikalraum	vertical room		3.6.			Raum auswählen (s.u.)	Popup Anzahl Geschosse	Anzeige Raum in darüberliegenden Geschossen	Abgleich darüberliegende Geschosse, Raumattribut Anz Geschosse
	allgemeines Objekt (Proxy-Objekt)	enter proxy object	x	Proxyobjekt	proxy object		2.7.						
	Mangel eingeben	enter deficiencies / defects	x	Mangel	deficiency / defect		2.6.			Position, zugehöriges Bauteil antippen	Popup Art des Mangels (erzwungene Attributeingabe)	Marker Mangel	
	Zustand eingeben (Proj., Geb., Gesch., Raum)	enter condition of building or room	-	Zustand	condition	-	-	-	EF oder Werkzeug Eigensch. übertragen	vorherige Auswahl (s.u.)	Slider in EF	Repräsentation Ink: farbliche Kodierung, ob Bewertung erfolgt	
	geschätzte Maße eingeben (z.B. Ausdehnung)	enter estimated dimensions	-	Text	text	-	-	-	EF	maßstäbliches Skizzieren der Strecke o. Fläche in der aktuelle Ansicht	Text handschriftlich in EF	Texterkennung -> Anzeige erkannte Zeichen / Zahlen	
Auswahl	weitere Attribute eingeben (Proj., ... s.o.)	enter other attributes	-	Ausdehnung	dimension	-	-	-	EF	vorherige Auswahl (s.u.)	maßstäbliches Skizzieren der Strecke o. Fläche in Zahlenwert bzw. Texterkennung -> Anzeige erkannte Zeichen / Zahlen		
	erzwungene Attributeingabe ???		-			-	-	-	EF		maßstäbliches Skizzieren der Strecke o. Fläche in Zahlenwert bzw. Texterkennung -> Anzeige erkannte Zeichen / Zahlen		
	Attributwerte auf Default / Berechnung / Ver. A	reset values to default / calculate	-	Wert zurücksetzen	reset value				auto		Popup im Eingabebereich (letzte Stiftposition oder Position zugehöriges Bauteil oder Raum)		
Auswahl	Attributwerte übernehmen	inherit attribute values	x	Wertübernahme	inherit value		3.5.						
	Bauteile auswählen	select building elements							SF	Objekt von dem übernommen wird, vorher auswählen; auf die übertragen wird danach	Auswahl in SF	Aktualisierung EF, Hervorhebung gewähltes Bauteil, Bearbeitungshandler	
	Räume auswählen	select rooms							SF	antippen Bauteilrepräsentation (Ink)	Auswahl in SF	Aktualisierung EF, Hervorhebung gewählter Raum, Bearbeitungshandler	
	Mängel auswählen	select deficiencies / defects	x	Ausw./Bearb.	select / edit		3.1.		SF	antippen Raumbezeichnung	Auswahl in SF	Aktualisierung EF, Hervorhebung gewählter Mangel, Bearbeitungshandler	
	Ink auswählen	select ink	-				s.o.		SF	antippen Mangelmarker	Auswahl in SF		
Bearbeitung	Geschoss auswählen	select level	-				-		SF		Auswahl in SF	EF, Anzeige Grundriss-Skizze ausgewähltes Geschoss, Handler gesamte Skizze an fixer Position, Warnung wenn keine Zustands-Bewertung	
	Gebäude auswählen	select building	-				-		SF		Auswahl in SF	EF, Anzeige aller Grundriss-Skizzen überlagert oder unterstes (Default-) Geschoss?	
	Geometrie auswählen (Punkte)	select geometry	x	Ausw./Bearb.	select / edit		s.o.		SF		Auswahl in SF	Hervorhebung ausgewählte Punkte, Stiftspitze Verschieben-Symbol	
	Auswahl aufheben	deselect all	-	Ausw./Bearb.	deselect		-			antippen / einkreisen		alle Markierungen entfernen	
	Ink-Objekte verschieben	move ink objects	-	Ink schieben	move ink		-		Handler	Translation zeichnen nach Auswahl			
Navigation (2,5D)	Ink-Objekte rotieren	rotate ink objects	-	Ink drehen	rotate ink		-		Handler	Bewegen des Rotieren-Handlers nach Auswahl			
	Ink-Objekte skalieren	scale ink objects	-	Ink skalieren	scale ink		-		Handler	Bewegen des Skalieren-Handlers nach Auswahl			
	Ink-Objekte löschen	delete ink objects	-	Ink löschen	delete ink		3.2.	Kritzeln	Handler?	Antippen des Löschen-Handlers nach Auswahl			
	Punktbearbeitung (verschieben)	point editing	-	Punkt schieben	move point		s.o.		Handler	Bewegen des Verschieben-Handlers nach Auswahl			
	Grundriss verschieben	-	-										
Navigation (2,5D)	Grundriss rotieren (Norden)	rotate floor plan	-	Norden	enter north		3.4.						
	Grundriss skalieren (Maßstab)	scale floor plan	-	Maßstab	enter scale		3.3.						
	Wechseln in Navigationsmodus	switch to navigation mode	x						Umschalttaste			Navigationshandler sichtbar	
	Pan	pan	-				5.1.			Translation zeichnen nach Umschalttaste			
	Drehen (Ansicht genordet / nicht genordet)	rotate (orientation to north)	-				5.3.		Nord-Symbol	tippen auf Nord-Symbol			
Ansichtssteuerung	Ansicht frei drehen	rotate view	-				5.4.						
	Zoom (Ansichtsmaßstab)	zoom (view scale)	-				5.2.		Maßstabsleiste	Interaktion mit Maßstabsleiste			
	Geschoss-Ebene wechseln (abwärts)	change levels (down)	-	Ebene runter	level down		5.6.						
	Geschoss-Ebene wechseln (aufwärts)	change levels (up)	-	Ebene hoch			5.5.				s.o. Ebene auswählen in SF		
	Raster ein/ausblenden	show/hide grid	-	Raster an/aus	grid on/off		4.5.					Raster anzeigen	
Ansichtssteuerung	Skizzenblatt anzeigen	show/hide sketch sheet	-	Skizzenblatt an/aus	sketch on/off		4.1.		SF		Sichtbarkeitssteuerung im SF RaumFS sichtbar/unsichtbar		
	darunterliegende Geschoss-Skizze einblend	show/hide subjacent floor level	-	unteres Geschoss	subjacent floor		4.4.					Geschossskizze sichtbar/unsichtbar grau	
	beliebige Geschossskizzen einblenden	show/hide floors	-				-		SF - Augesymbol				
	interpretierte Skizze anzeigen	show interpretation	-	OS an/aus	interpretation on/off		4.3.		Symbol in Statusleiste?			Interpretierte Linien sichtbar (Standard) / unsichtbar	
	Ink anzeigen	show sketched ink	-	IS an/aus	ink on/off		4.2.		Symbol in Statusleiste?				
Arbeits Schritte	Informationsdichte visualisieren	visualize information density	-	Informationsdichte	show data density		4.6.					farbliche Kodierung der Bauteile und Raummarken, je nachdem wieviele / welche Attribute schon eingegeben wurden	
	Kosten visualisieren	visualize costs	-	Kostenvisualis.	show costs		nötig ??? x					farbliche Kodierung der Bauteile und Raummarken nach Kostenberechnung	
	Zustand visualisieren	visualize condition	-	Zustandsvisualis.	show condition		4.7.						
	Informationsbedarf visualisieren	visualize information needs	-	Informationsbedarf	show info needs		4.8.					farbliche Kodierung der Bauteile und Raummarken, die schlechten Zustand und niedrige Informationsdichte vereinen	
	Grundriss-Skizzen zuordnen (Geschoss)	attach sketch sheet to floor level	-										
GUI-Anpassung	Datei speichern	save file		speichern	save		1.2.						
	Datei öffnen	open file		öffnen	open		1.1.						
	mit Server4 verbinden	connect to Server Four		verbinden	connect		-		automatisch beim Start des Clients				
	Menüleiste ein/ausschalten	show/hide menu bar		Menüleiste	menu bar		6.2.						
	Kreismenu ein/ausschalten	show/hide radial menu		Kreismenu	radial menu		6.3.						
Arbeits Schritte	Werkzeugkästen ein/ausschalten	show/hide toolboxes		Werkzeugkästen	toolbox		6.1.						
	Optionen / Vorgaben / Defaults anpassen	change options / defaults		Optionen	options		6.6.						
	zum Skizzenblock wechseln	jump to sketchpad		Skizzenblock	sketchpad		6.4.						
	zu Aufmaßwerkzeugen wechseln	jump to surveying tools		Aufmaß	survey		6.5.						

C Klassenstruktur und Aufbau des Raum- und Bauteilmodells

C 1. Einführung

Definition eines Raum- und Bauteilmodells für Initial Data Capture und die daran angeschlossenen Kosten-/Risikoauswertung.

Ziel war der Aufbau der Klassendefinition und deren Relationen untereinander. Der Aufbau erfolgt vorrangig nach den Anforderungen der Kosten-/Risikoanalyse, berücksichtigt aber gleichzeitig die Übertragbarkeit in die IFC.

C 2. Aufbau

Legende:

<i>Superklasse</i>	- eine Vererbungshierarchie höher
<i>Klassenname</i>	- (Sub-)Klasse
<i>(abs) Klassenname</i>	- abstrakte Klasse
Attributname : Datentyp	- Attribute
Methodenname(Übergabeparameter) : Rückgabedatentyp	- Methoden
Relationsname : (Kardinalität) Klassenname	- Relationen zu anderen Objekten

Anmerkung:

Bei enum-Datentypen ist generell der Eintrag *nutzerdefiniert* vorzusehen und ein zusätzliches Attribut vom Typ *String*, welches die Beschreibung (Wert) dieses nutzerdefinierten Eintrages enthält. Dies ist in der Beschreibung der Klassen nicht überall so eingetragen!

C 2.1. Abstrakte Definitionen

Superklasse für alle Klassen des Raum- und Bauteilmodells ist:

Klassenname	Attribute / Methoden / Relationen	Beschreibung
<i>(abs) Baseclass</i>	name : String	optional
	description : String	optional
	<i>getClassType() : string</i>	gibt Namen der Klasse zurück
	<i>OwnerHistory : (1...1) OwnerHistory</i>	
<i>(abs) BaseClass</i> <i>(abs) GeometryClass</i>		abstrakte Superklasse für alle Elemente, die geometrisch repräsentiert werden können!
	<i>Representation: (0...*) Body</i>	Relation zur Geometrie (anderes Teilmodell) die Repräsentation ist optional bzw. kann im Laufe der Lebensdauer eines Objektes auch ergänzt / geändert werden

... (abs) GeometryClass (abs) BuildingClass		abstrakte Superklasse für alle Elemente der Raum- und Bauteilstruktur
	condition : enum (mangelfrei, gut, schlecht, desaströs)	Zustand der jeweiligen Instanz
	conditionInherit : bool	true bedeutet, dass der Zustand aus dem Objekt in der Hierarchiestufe höher übernommen wurde. Hierarchien sind: - Decomposes (Teil und Ganzes) - ContainedInSpatialStructure (beinhaltet in Raumstruktur)
	getQuantityOrigin : enum (Modell, nutzerdefiniert)	gibt an, wie die Methode <i>getQuantity</i> die Menge ermittelt: aus der tatsächlichen Modellgeometrie oder als berechneter Wert einer Nutzereingabe. Die Art der Mengenermittlung hat einen Einfluß auf die Monte-Carlo-Simulation und wird dort als Unsicherheitsfaktor berücksichtigt!
	getQuantity(type : String) : double	Menge ... über den String wird die Art (type) der abzufragenden Menge übergeben z.B. Außenwandfläche (Definitionen der Mengen-Arten lt. DIN 277-3). Die abgeleiteten Klassen „wissen“ selbst, welche Arten von Mengen sie zurückgeben können. Ist die Art nicht bekannt, wird NULL zurückgegeben!
	Defects : (0...*) Defect	beschreibt die Relation zwischen Objekt und ihm anhaftenden Mangel. Ein Mangel kann auch von mehreren Objekten gleichzeitig referenziert sein!
	PlannedTasks : (1 ... 1) TaskGroup	zugeordnete Maßnahmengruppe
	FinishedTasks : (0 ... *) finishedTask	bereits abgeschlossene Maßnahmen

C 2.2. Ressourcen

Klassenname	Attribute / Methoden / Relationen	Beschreibung
OwnerHistoryClass		Informationen über Ersteller und Erstellungs- bzw. Änderungszeit eines Objektes ... entspricht weitestgehend IFC
	OwningApplication : String	Name des Clients der das Objekt erstellt / geändert hat! (im Rahmen des Projektes ist dies derzeit immer <i>InitialDataCapture</i> . Bei Erweiterung des gesamten Client-Server Ansatzes kann dies immer ein anderer Client sein)
	getCreationDate() : int	Erstellungsdatum/-zeit (in Sekunden seit Beginn 1970; entsprechend Spezifikation IFC)
	getLastModifyingDate() : int	Änderungsdatum/-zeit
	OwningUser : (1...1) PersonAndOrganization	
	LastModifyingUser : (1...1) PersonAndOrganization	
Address		Adresse
	street : string	
	city : string	
	postcode : string	Postleitzahl
	federalState : String	Bundesland

	country : string	
(abs)PersonAndOrganization		Superklasse für Person / Unternehmen
	telephonNumber : string	
	faxNumber : string	
	emailAddress : string	
	webSite : string	URL
	role : enum (Auftraggeber, Bearbeiter, Eigentümer, nutzerdefiniert)	
	roleUserDefined : string	
	Address : (1...*) Address	
(abs)PersonAndOrganization Person		Person
	competence : enum (Laie, Student, Planer, Sachverständiger, nutzerdefiniert)	Die Kompetenz des Bearbeiters geht als Unsicherheitsfaktor in die Monte-Carlo-Simulation mit ein.
	competenceUserDefined : string	
(abs)PersonAndOrganization Organization		Büro / Unternehmen
	ContactPerson : (1 ... 1) Person	
	Manager : (1 ... 1) Person	

C 2.3. Gruppierungsmechanismus

Klassenname	Attribute / Methoden / Relationen	Beschreibung
(abs) BaseClass Group		allgemeine Gruppe ... die referenzierten Elemente können auch Bestandteil mehrerer Gruppen sein
	GroupedObjects : (1...*) BaseClass	Relation zu den gruppierten Elementen

C 2.4. Projektstruktur

Über eine Instanz der Klasse Projekt, können alle Elemente im Bauwerksmodell abgefragt werden. Das Projekt bildet die oberste Hierarchiestufe.

Klassenname	Attribute / Methoden / Relationen	Beschreibung
(abs) BaseClass Project		Projekt
	difficulty : enum (sehr gering, gering, durchschnittlich, überdurchschnittlich, anspruchsvoll)	Schwierigkeitsgrad des Projektes
	getCost(costCategory: int) : double	KGr 700 siehe SpatialStructureClass
	getEditors() : setOf(1...*) Person	alle Bearbeiter, die bei der Erfassung beteiligt sind
	getCurrentEditor() : Person	aktueller Bearbeiter
	Client : (1...1) PersonAndOrganization	Auftraggeber
	ProjectManager : (1...1) Person	Projektverantwortlicher

	hasBuildings : (1...*) Building	zum Projekt gehörende Gebäude
	hasSites : (1...*) Site	zum Projekt gehörende Grundstücke

C 2.5. Raumstruktur

Alle Elemente, die die räumliche Gliederung eines Projektes (Gebäudes) beschreiben:

Klassenname	Attribute / Methoden / Relationen	Beschreibung
... (abs) BuildingClass (abs) SpatialStructureClass		abstrakte Superklasse für alle Elemente der Raumstruktur
	number : int	optional: Nummer zur identifikation
	currentUse : String	aktuelle Nutzung
	plannedUse : String	geplante Nutzung
	plannedStandard : enum (einfach, mittel, hoch)	geplanter Standard
	currentUseInherit : bool plannedUseInherit : bool plannedStandardInherit : bool	= true bedeutet, dass der Wert aus dem Objekt in der Hierarchiestufe höher übernommen wurde, d.h. wenn auf true gesetzt, muss der Wert aus dem Objekt „darüber“ gelesen und entsprechend gesetzt werden. Wird der Wert überschrieben, muss *Inherit auf false gesetzt werden. Hierarchie ist in dem Fall die Raumstruktur : site->building->buildingStorey->space
	getCost(costCategory : int) : double	gibt die Kosten zurück, die durch die übergebene Kostengruppe erzeugt wird. Betrifft 1 Stufe der Kostengruppen (100 ... 700), welche KGr tatsächlich einen Rückgabewert haben, wird durch die abgeleiteten Klassen bestimmt!
	hasBuildingElements : (0 ... *) BuildingElementClass	Hierüber können alle Elemente referenziert werden, die dem Objekt der Raumstruktur zugeordnet sein können, d.h. hierüber lassen sich bspw. alle Wände eines Geschosses abfragen!
... (abs) SpatialStructureClass Site		Grundstück
	quality : enum (eben, geneigt, starke Hanglage, nutzerdefiniert)	Beschaffenheit
	qualityUserDefined : string	
	vegetation : enum (rasen, straeucher, kleine Baeume, große Baeume, nutzerdefiniert)	
	vegetationUserDefined : string	
	cadastralRegisterEntry : String	Eintragungen im Grundbuch
	location : enum (Innenstadt, Stadtrand, Laendlich)	Region
	citySize : enum(bis 10.000, bis 25.000, bis 50.000, bis 100.000, bis 250.000, bis 500.000, bis 1.000.000, über 1.000.000)	
	siteDevelopment : enum (ja, nein, teilweise, nutzdefiniert)	Erschließung

	siteDevelopmentUserDefined : string	
	siteArea : double	Grundstücksfläche als Nutzereingabe
	getQuantityOrigin : enum (Modell, nutzerdefiniert)	siehe <i>BuildingClass</i>
	getCost(costCategory : int) : double	KGr 100,200,500
	getQuantity(type : String) : double	zulässige Types: „Grundstücksfläche“ (aus siteArea) „Bebaute Fläche“ (summe Bruttogrundflächen aller Gebäude) „Unbebaute Fläche“ (Grundstücksfläche – bebaute Fläche) Die Methode unterscheidet, welcher Wert zurückgegeben wird. Entweder aus Nutzereingabe, oder aus dem Modell ermittelt (über Attr. getQuantityOrigin).
	Address : (1 ... 1) Address	
	hasBuildings : (1...*) Building	zum Grundstück gehörende Gebäude
... (abs) <i>SpatialStructureClass</i> <i>Building</i>		Gebäude
	vecancy : enum (ja, nein, teilweise)	Leerstand
	buildingCategory : enum (GK1, GK2, GK3, GK4, GK5)	Gebäudeklasse nach ThürBauO
	buildingTypeSchmitz : enum (Fachwerk, Gründerzeit, Bauten 1920-1939, Bauten 1950-1959, Umnutzungen (Fabriken/Kloster /Krankenhäuser)	Gebäudetyp nach [Schmitz08]
	buildingTypeBWZK : enum	Gebäudetyp nach Bauwerkszuordnungskatalog
	constructionTime : enum (vor 1880, 1880-1919, 1920-1945, 1946-1959, ab 1960)int	
	planningAim : enum (Modernisierung, Instandhaltung, Instandsetzung, Sanierung, Umnutzung, Umbau, Erweiterung, Rückbau)	Planungsziel(nach HOAI) : relevant bei der Auswahl von Maßnahmen!
	planningFocus : enum (energetisch/haustechnisch, schallschutztechnisch, brandschutztechnisch, gefahrenstofftechnisch	
	feeOrderZone : enum (I,II, III, IV, V)	Zone nach HOAI ...
	buildingLengthX : double	Werte für die Gebäudegröße können vom Nutzer unabhängig von der erstellten Geometrie für detailliertere Kostenbewertung eingetragen werden wenn Nutzer buildingHeight setzt, und buildingHeightOrigin = errechnet bzw. aus Modell ist, muss eine Warnung/Abfrage erscheinen, welche Werte zu verwenden sind, bzw. angepasst werden sollen.
	buildingLengthY : double	
	buildingHeight : double	
	buildingHeightOrigin : enum (nutzerdefiniert, errechnet, aus Modell)	errechnet: wenn das attribut buildingHeight sich aus den Geschosshöhen + Höhe über Grund erstes Vollgeschoss errechnet nutzerdefiniert: wenn durch Nutzereingabe aus Modell: wenn aus der (Modell)Geometrie ermittelt

	<code>setBuildingHeightOrigin(type : string)</code>	<p>wenn Nutzer <code>buildingHeight</code> setzt, wird <code>buildingHeightOrigin</code> auf „nutzerdefiniert“ gesetzt</p> <p>wenn Gebäude über Geschosse verfügt, wird <code>buildingHeightOrigin</code> auf „errechnet“ gesetzt. <code>buildingHeight</code> wird dann aus Summe der Geschosshöhen ermittelt</p> <p>wenn <code>buildingHeightOrigin</code> auf Modell gesetzt wird, wird <code>buildingHeight</code> aus der Modellhöhe (Abstand zwischen Unterkante erstes Vollgeschoss bis Oberkante letztes Voll- bzw. Dachgeschoss) ermittelt</p>
	<code>getCost(Kostengruppe : int) : double</code>	KGr 300, 400
	<code>getQuantity(type : String) : double</code>	<p>zulässige Types:</p> <p>„Bruttogrundfläche“ (aus Skizze/Modell, oder aus <code>buildingLenghtX</code> * <code>buildingLenghtY</code> wenn diese gesetzt sind)</p> <p>„Bruttorauminhalt“ (aus Skizze/Modell, oder aus <code>buildingLenghtX</code> * <code>buildingLenghtY</code> * <code>buildingHeight</code> wenn diese gesetzt sind)</p> <p>„Gebäudehöhe“ (<code>buildingHeight</code>)</p>
	<code>hasStoreys : (1...*) BuildingStoreys</code>	zum Gebäude gehörende Geschosse
<p>...</p> <p>(abs)</p> <p><i>SpatialStructureClass</i></p> <p><i>BuildingStorey</i></p>		Geschoss
	<code>type : enum (Kellergeschoss, Vollgeschoss, Dachgeschoss)</code>	
	<code>buildingStoreyHeight : double</code>	Nutzereingabe für Geschosshöhe ... Default-Wert für Höhe beim Anlegen eines neuen Geschosses oder ermittelt aus <code>buildingHeight</code> (wenn dies gesetzt ist) / Anzahl der Geschosse
	<code>buildingStoreyHeightOrigin : enum (default, nutzerdefiniert, errechnet, aus Modell)</code>	<p>errechnet: wenn das Attribut <code>buildingHeight</code> sich aus den Geschosshöhen errechnet</p> <p>nutzerdefiniert: wenn durch Nutzereingabe</p> <p>aus Modell: wenn aus der (Modell)Geometrie ermittelt</p>
	<code>buildingStoreyHeightOverGround : double</code>	Höhe des Geschossniveaus über Grund (Abstand Unterkante Geschoss bis Boden)
	<code>buildingStoreyHeightOverGroundOrigin : enum (default, nutzerdefiniert, errechnet, aus Modell)</code>	
	<code>setBuildingStoreyHeightOrigin(type : string)</code>	<p>wenn Nutzer <code>buildingStoreyHeight</code> setzt, wird <code>buildingStoreyHeightOrigin</code> auf „nutzerdefiniert“ gesetzt</p> <p>Wenn Gebäudehöhe nutzerdefiniert (<code>buildingHeightOrigin</code> = nutzerdefiniert), dann werden Geschosshöhen errechnet, die auf default stehen! Nutzerdefinierte werden nicht überschrieben (hier müsste eine Kontrolle auf Plausibilität erfolgen, z.B. wenn Geschosshöhen nutzerdefiniert sind, dürfen diese in der Summe nicht größer sein, als die Gebäudehöhe, bzw. Errechnete Höhen dürfen nicht zu klein werden!)</p> <p>Wenn <code>buildingStoreyHeightOrigin</code> auf Modell gesetzt ist, wird <code>buildingStoreyHeight</code> aus der Modellhöhe (Abstand zwischen Unterkante Geschoss bis Oberkante nächstes Geschoss) ermittelt.</p>

	setBuildingStoreyHeightOverGroundOrigin(type : string)	
	getCost(costCategory : int) : double	KGr 300, 400
	getQuantity(type : String) : double	zulässige types: „Bruttogrundfläche“ (aus Skizze/Modell) „Nettogrundfläche“ (aus Skizze/Modell, summe aller Raumgrundflächen) „Konstruktionsgrundfläche“ (Bruttogrundfläche - Nettogrundfläche) „Geschosshöhe“ (buildingStoreyHeight) „Höhe über Grund“ (buildingStoreyHeightOverGround) „Nutzfläche“ (Summe Raumgrundfläche aller Räume mit Nutzungsart NF) „Technische Funktionsfläche“ (Summe Raumgrundfläche aller Räume mit Nutzungsart TF) „Verkehrsfläche“ (Summe Raumgrundfläche aller Räume mit Nutzungsart VF)
	hasSpaces : (1...*) Space	zum Geschoss gehörende Räume
... <i>(abs)</i> SpatialStructureClass Space		Raum
	use : String	z.B. Wohnen
	usingGroup : enum (NF, TF, VF)	Nutzungsgruppe nach DIN 277
	spaceHeight : double	Raumhöhe
	spaceHeightOrigin : enum (default, nutzerdefiniert, aus Modell)	default-Wert sollte mit der Geschosshöhe gekoppelt sein,
	setSpaceHeightOrigin (type : string)	
	getQuantity(type : String) : double	zulässige types: „Raumgrundfläche“ (aus Modell) „Raumhöhe“ (aus Skizze/Modell oder spaceHeight)
	containedInStorey : (1...*) BuildingStorey	umgekehrte Relation zu BuildingStorey::hasSpaces Beschreibt zur welchen Geschossen der Raum gehört

C 2.6. Raumbegrenzung

Die Raumbegrenzung ist eine logische Verknüpfung zwischen Raum und begrenzendem Bauteil. Sie wird als Klasse abgebildet, da sie die Eigenschaft virtuell oder physisch haben kann.

Ist die Raumgrenze virtuell, hat sie die Relation zu zwei Räumen und kann über eine Fläche im Geometriemodell repräsentiert werden.

Ist die Raumgrenze physisch, hat sie die Relation zu einem Raum und einem Bauteil und kann durch eine Fläche im Geometriemodell, welche auch durch das Bauteil referenziert wird, repräsentiert werden.

Klassenname	Attribute / Methoden / Relationen	Beschreibung
... (abs) GeometryClass (abs) SpaceBoundary		Raumbegrenzung
	RelatingSpace : (1...1) Space	
... (abs) SpaceBoundary VirtualSpaceBoundary		virtuelle Raumgrenze
	OtherRelatingSpace : (1...1) Space	
... (abs) SpaceBoundary PhysicalSpaceBoundary		physische Raumgrenze
	RelatingElement : (1...1) BuildingElementClass	

C 2.7. Bauteilstruktur

Alle Elemente, die physisch vorhanden sind und das Gebäude bilden.

Bauteile können aus mehreren Bauteilen zusammengesetzt sein (über Basisklasse realisiert).

Klassenname	Attribute / Methoden / Relationen	Beschreibung
... (abs) BuildingClass (abs) BuildingElementClass		abstrakte Superklasse für alle Bauteile
	isComplex : enum (complex, element, partial)	complex = das Bauteil ist aus mehreren Bauteilen zusammengesetzt (Relation isDecomposedBy) element = das Bauteil ist weder zusammengesetzt, noch Teil eines Anderen partial = das Bauteil ist Teil eines Anderen (Relation Decomposes)
	constructionType : string	Konstruktionsart Über die Konstruktionsart kann eine detaillierte Abfrage nach geeigneten Maßnahmen in der externen Datenbank erfolgen
	getCostCategory() : int	gibt die Kostengruppe nach DIN 276 des Bauteils zurück
	isDecomposedBy : (0 ... *) BuildingElementClass	ist zusammengesetzt aus ...
	Decomposes : (0 ... 1) BuildingElementClass	setzt zusammen (ist Teil von) ...
... (abs) BuildingElementClass Wall		Wand(-segment)
	loadBearing : bool	tragend / nichttragend
	isExternal : bool	Innen / Außen
	width : double	Wandstärke (Abstand zwischen beiden Wandoberflächen)
	widthOrigin : enum (default, nutzerdefiniert, aus Modell)	

	height : double	Wand höhe
	heightOrigin : enum (errechnet, nutzerdefiniert, aus Modell)	
	setWidthOrigin (type : string)	
	setHeightOrigin (type : string)	errechnet = aus der Raumhöhe bei nutzerdefiniert kann die Wand eine andere Höhe haben, als der Raum
	getCostCategory() : int	330 – außen + complex (beinhaltet konstruktion + fenster + bekleidung ...) 340 – innen + complex (inkl. opening, covering) 331 – tragend + außen + partial/element 332 – nichttragend + außen + partial/element 341 – tragend + innen + partial/element 342 – nichttragend + innen + partial/element
	getQuantity(type : String) : double	zulässige types: „Dicke“ „Höhe“ „Konstruktionsgrundfläche“ „Fläche“ „Anteilige Fläche“ (ohne Öffnungen)
	hasOpenings : (0...*) Opening	Öffnungen
... (abs) BuildingElementClass Column		Stütze
	loadBearing : bool	tragend / nichttragend
	isExternal : bool	Innen / Außen
	widthX : double	Dicke (Abstand zwischen beiden Oberflächen)
	widthY : double	
	diameterWidth : double	Durchmesser, bei runder Stütze
	widthOrigin : enum (default, nutzerdefiniert, aus Modell)	
	height : double	
	heightOrigin : enum (errechnet, nutzerdefiniert, aus Modell)	
	setWidthOrigin (type : string)	
	setHeightOrigin (type : string)	errechnet = aus der Raumhöhe bei nutzerdefiniert kann die Stütze eine andere Höhe haben, als der Raum
	getCostCategory() : int	333 – außen 343 – innen
	getQuantity(type : String) : double	zulässige types: „DickeX“ „DickeY“ „Konstruktionsgrundfläche“ „Höhe“

<p>...</p> <p>(abs) <i>BuildingElementClass</i></p> <p><i>Slab</i></p>		Decke
	loadBearing : bool	tragend / nichttragend
	isExternal : bool	Innen / Außen
	width : double	Dicke (Abstand zwischen beiden Oberflächen)
	widthOrigin : enum (default, nutzerdefiniert, aus Modell)	
	setWidthOrigin (type : string)	
	getCostCategory() : int	350 – complex (inkl. covering) 351 – element/partial
	getQuantity(type : String) : double	zulässige types: „Dicke“ „Fläche“ „Anteilige Fläche“ (ohne Öffnungen)
	hasOpenings : (0...*) Opening	Öffnungen
<p>...</p> <p>(abs) <i>BuildingElementClass</i></p> <p><i>Roof</i></p>		Dach
	width : double	Dicke (Abstand zwischen beiden Oberflächen)
	widthOrigin : enum (default, nutzerdefiniert, aus Modell)	
	slope : double	Dachneigung in Grad Eine Dachfläche! Bei Sattel-, Walmdächer etc. setzt sich das Dach aus mehreren Elementen zusammen z.B.: Roof.complex ➔ isDecomposed : Roof.partial ➔ isDecomposed : Roof.partial
	slopeOrigin : enum (default, nutzerdefiniert, aus Modell)	
	setWidthOrigin (type : string)	
	setSlopeOrigin (type : string)	
	getCostCategory() : int	360 – complex (inkl. openings, covering) 361 – element/partial
	getQuantity(type : String) : double	zulässige types: „Dicke“ „Neigung“ „Fläche“ „Anteilige Fläche“ (ohne Öffnungen)
	hasOpenings : (0...*) Opening	Öffnungen
<p>...</p> <p>(abs) <i>BuildingElementClass</i></p> <p><i>Covering</i></p>		Belag/Bekleidung
	isExternal : bool	Innen / Außen

	coveringType : enum (Wandbekleidung, Deckenbekleidung, Deckenbelag, Dachbekleidung, Dachbelag, nutzerdefiniert)	Wandbekleidung = Decomposes zeigt auf Wall Deckenbekleidung/-belag = Decomposes zeigt auf Slab Dachbekleidung/-belag = Decomposes zeigt auf Roof
	coveringTypeUserDefined : string	wenn coveringType = nutzerdefiniert
	width : double	Dicke (Abstand zwischen beiden Oberflächen)
	widthOrigin : enum (default, nutzerdefiniert, aus Modell)	
	setWidthOrigin (type : string)	
	getCostCategory() : int	335 - wenn coveringType = Wandbekleidung, wall.isExternal=true und isExternal = true 336 - wenn coveringType = Wandbekleidung, wall.isExternal=true und isExternal = false 345 - wenn coveringType = Wandbekleidung, wall.isExternal = false und isExternal = false 352 – wenn coveringType = Deckenbelag 353 – wenn coveringType = Deckenbekleidung 363 – wenn coveringType = Dachbelag 364 – wenn coveringType = Dachbekleidung
	getQuantity(type : String) : double	zulässige types: „Dicke“ „Fläche“ „Anteilige Fläche“ (ohne Öffnungen)
	hasOpenings : (0...*) Opening	Öffnungen
... <i>(abs) BuildingElementClass</i> <i>Opening</i>		Öffnung allgemein
	isExternal : bool	Innen / Außen
	width : double	Breite (Abstand zwischen beiden Oberflächen)
	length: double	Länge bzw. Höhe
	widthLengthOrigin : enum (default, nutzerdefiniert, aus Modell)	
	setWidthHeightOrigin (type : string)	
	getQuantity(type : String) : double	„Breite“ „Höhe“ „Fläche“
	isOpeningIn : (1 ... *) BuildingElementClass	Wo ist die Öffnung drinn ...
... <i>Opening</i> <i>Window</i>		Fenster
	parapetWallHeight : double	Brüstungshöhe
	parapetWallHeightOrigin : enum (default, nutzerdefiniert, aus Modell)	
	setParapetWallHeightOrigin (type : string)	

	<code>getCostCategory() : int</code>	334 – wenn isExternal = true, und isOpeningIn auf wall zeigt 344 – wenn isExternal = false, und isOpeningIn auf wall zeigt 362 – wenn isOpeningIn auf roof zeigt
	<code>getQuantity(type : String) : double</code>	„Brüstungshöhe“
... Opening Door		Tür
	<code>doorsill : double</code>	Höhe Türschwelle
	<code>doorSillOrigin : enum (default, nutzerdefiniert, aus Modell)</code>	
	<code>setDoorSillOrigin (type : string)</code>	
	<code>getCostCategory() : int</code>	334 – wenn isExternal = true, und isOpeningIn auf wall zeigt 344 – wenn isExternal = false, und isOpeningIn auf wall zeigt
	<code>getQuantity(type : String) : double</code>	„Türschwelle“
... (abs) BuildingElementClass BuildingElementProxy		BauteilProxy ... Platzhalter für nicht näher definierte Bauteile
	<code>costCategory : int</code>	Kostengruppennummer für die das Bauteil steht
	<code>getCostCategory() : int</code>	
	<code>getQuantity(type : String) : double</code>	„Fläche“ Gibt die Fläche zurück, falls Objekt mit Geometrie verknüpft ist ... hier kann allerdings nur die Summe aller verknüpften Flächen zurückgegeben werden, da für ein nicht näher definiertes Bauteil keine andere Berechnungsvorschrift vorgesehen werden kann!
	<code>hasOpenings : (0...*) Opening</code>	Öffnungen

C 2.8. Maßnahmenmodell

Wenn ein Mangel auf einen bestimmten Bereich begrenzt ist, dann ist dieser in der Regel flächenförmig! d.h. er gibt als Methode (getQuantity) seine Fläche zurück.

In der DIN 1356-6 werden auch punktförmige und linienförmige Schäden vorgesehen. Diese werden als eigenständige Mangelklasse (Defect) berücksichtigt!

Klassenname	Attribute / Methoden / Relationen	Beschreibung
... (abs) GeometryClass (abs) Defect		Mangel
	<code>type : enum (funktional, konstruktiv, ästhetisch)</code>	
	<code>defectID : string</code>	Schadenschlüsselnummer nach DIN 1356-6

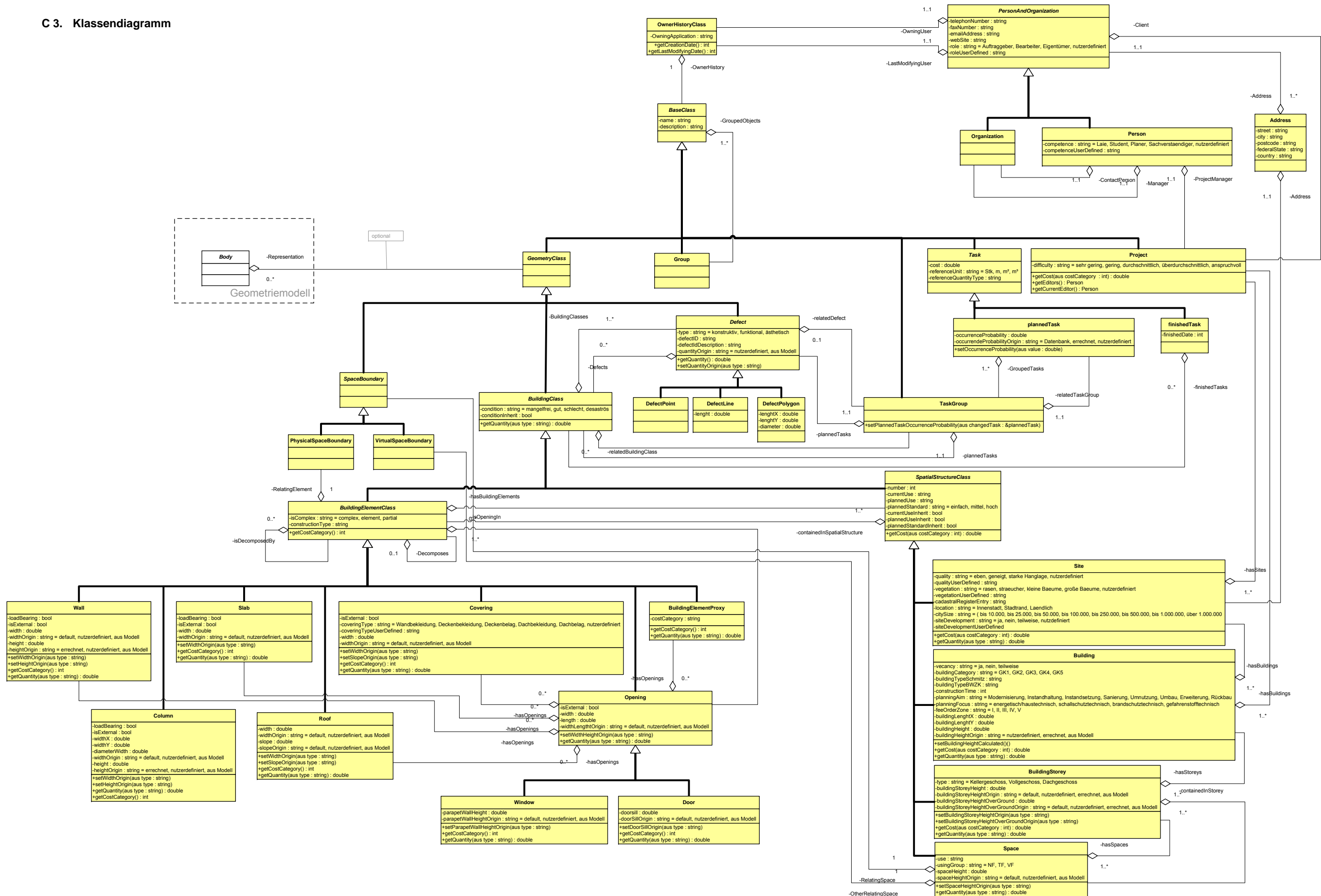
	defectIdDescription : string	+ Beschreibung
	quantityOrigin : enum (nutzerdefiniert, aus Modell)	
	getQuantity() : double	gibt Menge des Mangels zurück, sofern dieser mit Geometrie verknüpft ist!
	setQuantityOrigin(type : string)	wenn Mangel Geometrie hat, kann die Menge aus dem Modell ermittelt werden!
	PlannedTasks : (1 ... 1) TaskGroup	zugeordnete Maßnahmengruppe
	BuildingClasses : (1 ... *) BuildingClass	Elemente auf die sich der Mangel bezieht Ein Mangel kann auch mehreren Elementen zugeordnet sein
... (abs) Defect DefectPoint		punktförmiger Mangel
... (abs) Defect DefectLine		linienförmiger Mangel
	length : double	Nutzereingabe für Linienlänge
... (abs) Defect DefectPolygon		flächenförmiger Mangel
	lengthX : double	Nutzereingabe über Länge x Breite oder durchmesser ... es darf immer nur eines von beiden eingegeben werden!
	lengthY : double	
	diameter : double	

Maßnahmen können sowohl bereits abgeschlossen (z.B. wird bei der Ersterfassung festgestellt, dass die Fenster vor 5 Jahren erneuert wurden), als auch geplant sein. Die geplanten Maßnahmen sind maßgebend für die Kostenermittlung. Die Attribute description, cost und referenceQuantity werden beim Anlegen von Maßnahme aus der externen Datenbank der relevanten Maßnahmen (Fachwissen) entsprechend des verknüpften Bauteils bzw. Mangels ausgelesen!

Klassenname	Attribute / Methoden / Relationen	Beschreibung
(abs) BaseClass (abs) Task		Abstrakte Klasse für Maßnahme
	description : string	Beschreibung ->Datenbank aus BaseClass
	cost : double	pro Bezugseinheit ->Datenbank optional für finishedTask
	referenceQuantityType : String	auf welchen Mengentyp bezieht sich die Maßnahme? ->Datenbank z.B.: Außenwandfläche, Innenwandfläche ...
	referenceUnit : enum (Stk, m, m², m³)	optional für finishedTask
... (abs) Task finishedTask		Abgeschlossene Maßnahme
	finishedDate : int	Jahr der Ausführung

<p>...</p> <p>(abs) Task</p> <p>plannedTask</p>		Geplante Maßnahme
	occurrenceProbability : double	Eintrittswahrscheinlichkeit Wert zwischen 0 und 1
	occurrenceProbabilityOrigin : enum (Datenbank, errechnet, nutzerdefiniert)	die Eintrittswahrscheinlichkeit kann entweder aus der Datenbank gelesen werden (default), oder vom Nutzer selbst gesetzt werden. Wenn vom Nutzer selbst gesetzt, dürfen innerhalb einer Maßnahmengruppe die Eintrittswahrscheinlichkeiten insgesamt nur 100 ergeben, d.h. bei Nutzeranpassung einer Wahrscheinlichkeit werden die anderen errechnet.
	setOccurenceProbability(value : double)	setzt den Wert für occurrenceProbability ruft die Methode setPlannedTaskOccurrenceProbability() der verknüpften TaskGroup auf
	relatedTaskGroup : (1...1) TaskGroup	zugehörige Maßnahmengruppe
<p>BaseClass</p> <p>TaskGroup</p>		Maßnahmengruppe: gruppiert alle möglichen Maßnahmenvarianten (plannedTask)
	setPlannedTaskOccurrenceProbability(changedTask : &plannedTask)	wenn ein Nutzer die Eintrittswahrscheinlichkeit einer Maßnahme ändert, wird die Methode aufgerufen und eine pointer auf die geänderte Maßnahmengruppe übergeben. Die Methode passt die Eintrittswahrscheinlichkeiten aller anderen Maßnahmen in der Gruppe an, so dass die Summe immer 1 ergibt. occurrenceProbabilityOrigin dieser Maßnahmen wird auf errechnet gesetzt.
	GroupedTasks : (1...*) plannedTask	gruppiert nur Maßnahmen
	relatedDefect : (0 ... 1) Defect	zugehöriger Mangel
	relatedBuildingClass : (0 ... *) BuildingClass	zugehöriges Bauteil / Raumelement Maßnahmengruppe muß mit mindestens einem von beiden verknüpft sein!! Maßnahmengruppe kann auch mit mehreren Bauteilen verknüpft sein, z.B. allen Außenwänden ... wenn dann eine Maßnahme manipuliert wird, muss eine Abfrage erscheinen, ob dies für alle verknüpften Bauteile zutrifft, oder nur für das aktive. Wenn nur für das aktive, muss eine Kopie der Maßnahmengruppe nur für das aktive Bauteil erstellt werden!

C 3. Klassendiagramm



D Technical Report - Stiftbasierte Benutzungsoberfläche für die Eingabe von Skizzen

Technical Report

Stiftbasierte Benutzungsoberfläche für die Eingabe von Skizzen

Bauhaus Universität Weimar
Professur Informatik in der Architektur
Bearbeiter: Helga Tauscher

26. Juli 2008

Inhaltsverzeichnis

1	Einführung	5
1.1	Rahmen	5
1.2	Problemstellung	5
1.3	Begriffe	5
1.4	Gliederung	6
1.5	Methode des Software-Reviews	6
2	Funktionen der Skizzeneingabe	8
2.1	Erstellen und Bearbeiten von Skizzen	8
2.1.1	Zeichnen (Skizzeneingabe)	8
2.1.2	Bearbeiten der Zeichnungselemente	9
2.1.3	Auswahl der Zeichnungselemente	9
2.1.4	Zeichnungsassistent / „Beautification“	10
2.2	Verwalten von Skizzen	10
2.2.1	Verwalten mehrerer Skizzen	10
2.2.2	Strukturierung innerhalb der Skizze und Verknüpfung nach außen	11
2.2.3	Überlagerung (Ebenen)	11
2.2.4	Level of Detail	12
2.3	Datenaustausch	12
2.3.1	Einbinden bestehender Daten (Ref.)	12
2.3.2	Import	13
2.3.3	Export	13
2.4	Interpretation, semantische Anreicherung	13
2.4.1	Geometrische Erkennung	13
2.4.2	Symbolerkennung / Handschrifterkennung	14
2.4.3	Geometrische Rekonstruktion	14
2.5	Modellieren	15
3	Ergonomie stiftbasierter Interaktion	16
3.1	allgemeine Gesetze des Designs von WIMP-GUIs	16
3.2	Hardware-Varianten	17
3.2.1	externes Tablett	17
3.2.2	externe Tracker für klassisches Stift-Papier-“Interface“	18
3.2.3	stiftbedienter Eingabe-Bildschirm	18
3.3	Stiftgemäße grafische Benutzungsoberfläche	19
3.3.1	Stift als pointing device - Adaption der Maus-Interaktion	19

3.3.2	Modeswitching	19
3.3.3	maximierter Eingabebereich	20
3.3.4	angepasste Interaktionselemente: Kreismenüs	20
3.3.5	angepasste Interaktionselemente: Stiftgesten	21
3.4	Grafische Benutzungsoberfläche zur Eingabe skizzenförmiger grafischer Inhalte	22
3.4.1	Auflösung der Befehlssyntax	22
3.4.2	Navigation	22
3.4.3	Feedback	23
3.4.4	Ungenauigkeit und Ambivalenz	24
4	Software-Review	25
4.1	Kategorien	25
4.1.1	nach Domain	25
4.1.2	nach Art der bearbeiteten Informationen	26
4.1.3	nach Umfang der Interpretation	27
4.2	allgemeine Beschreibung	27
4.3	Beschreibung Funktionsumfang	36
5	Fazit: GUI-Entwicklung im Rahmen des Forschungsprojekts	39
5.1	Funktionalität des idealen prototypischen Systems	39
5.2	Oberfläche und Interaktion im prototypischen System	40
5.3	GUI prototyping Methoden und GUI Notationen	41
	Literaturverzeichnis	41
6	Quellen	42
6.1	Literatur	42
6.2	Software	44

1 Einführung

1.1 Rahmen

Im Projekt „Digitaler Gebäudebestand als Grundlage für Projektentscheidungen im Altbau“ werden die Potentiale der Nutzung digitaler Technologien für die Kosten-Risiko-Analyse im frühen Projektstadium (Initiierungsphase) von Bauprojekten im Altbau untersucht. Diese Analyse basiert auf Informationen über das Gebäude, die mit einer schnellen und effizienten Ersterfassung oder der Ergänzung und Validierung vorhandener Informationen vor Ort gewonnen werden. Die Werkzeuge und Methoden der Erfassung müssen dabei die Besonderheiten des Arbeitens vor Ort berücksichtigen und entsprechend des Projektstadiums mit minimalem Aufwand zu den notwendigen Ergebnissen führen. Sie stützen sich deshalb auf den Einsatz ultramobiler Computer, die einhändige Bedienung mit einem Stift und intuitiv anwendbare, dem fachlich geschulten Anwender vertraute EingabeprozEDUREN. In diesem Bericht werden die Recherchen zu den stiftbasierten Erfassungsmethoden zusammengefasst.

1.2 Problemstellung

Zeichnen und Skizzieren sind traditionelle Arbeitsmethoden der Architekten, die sich sowohl für das Festhalten von Entwurfsideen oder -ansätzen, als auch für die Notation bestehender baulicher Objekte im Rahmen der Bauaufnahme über Jahrhunderte entwickelt und bewährt haben. Stiftbasierte Mensch-Computer-Schnittstellen adaptieren die einfache Form der Interaktion mit einem Stift auf einem Blatt Papier für den Computer und versuchen dabei, den Informationsgehalt der gezeichneten Liniengrafik für die elektronische Informationsverarbeitung nutzbar zu machen. Sowohl die Gestaltung der Interaktion als auch die Aufbereitung und Verarbeitung der eingegebenen Informationen weichen von den herkömmlichen auf Tastatur und Maus beruhenden Schnittstellen ab.

1.3 Begriffe

Für Benutzungsschnittstellen, die auf der Eingabe mit dem Stift basieren, werden in der Literatur verschiedene Begriffe verwendet, die auf unterschiedliche Anwendungskontexte verweisen:

Stiftbasierte Schnittstellen (pen-based interfaces) sind nach dem Eingabegerät bezeichnet, ohne die Art der Eingabe zu spezifizieren.

Kalligraphische Schnittstellen (calligraphic interfaces) verdanken ihre Bezeichnung der Tatsache, dass die intuitivste Art, Schriftzeichen mit dem Stift einzugeben, die Handschrift ist.

Skizzenbasierte Schnittstellen (sketch-based interfaces) orientieren auf eine bildhafte Eingabe.

Weitere Bezeichnungen sind pen-based computing (emphasiert die Übertragung in die digitale Welt) und sketch-based modeling (weist darauf hin, dass die Skizze nicht das Ziel der Eingabe ist, sondern nur die Methode, um ein 3D-Modell zu erstellen)

1.4 Gliederung

Der vorliegende Bericht umfasst die Recherche des Forschungsstandes zu skizzenbasierten Benutzungsoberflächen sowie einen Software-Review für das Gebiet. Der theoretische Teil ist in zwei Kapitel zur Funktionalität und zur Ergonomie unterteilt. Das abschließende Kapitel fasst den Report im Hinblick auf die Entwicklung einer eigenen prototypischen skizzenbasierten Schnittstelle zusammen.

1.5 Methode des Software-Reviews

Der durchgeführte Software-Review wurde mit der Zielstellung durchgeführt, relevante Funktionen für die Skizzeneingabe zu extrahieren und bereits etablierte Möglichkeiten für die Umsetzung dieser Funktionalität in einer stiftbasierten Benutzungsschnittstelle zu recherchieren. Ziel war also weder eine Bewertung der Software-Funktionalität hinsichtlich einer Nutzungs- (oder Kauf-)empfehlung für bestimmte Aufgaben noch eine Bewertung der Benutzbarkeit (Usability) hinsichtlich einer Weiterentwicklung der Software.

Die Durchführung des Software-Reviews erfolgte deshalb als Durchsicht und nicht nach vorgegebenen Evaluationsschemata. Für eine stringente Evaluation mit konsistentem Test-Szenario wies das vorliegende Spektrum an Software eine zu breite Streuung der Anwendungsgebiete, Zielgruppen und Anspruchsniveaus auf. Gleichwohl wurde angestrebt, jeweils an einem Test-Beispiel - ggf. anhand eines Tutorials oder der Dokumentation - den vollständigen Funktionsumfang der Software zu explorieren.

Die vorgefundenen Funktionalitäten und Elemente der Benutzungsschnittstelle wurden in einem ersten Durchlauf stichpunktartig gesammelt. Nach der Zusammenfassung jeweils gleichartiger Funktionen und Interaktionsmechanismen in einer Tabelle (s. Abschnitt 4.3) konnte die Zusammenstellung in einem zweiten Durchlauf durch Abarbeiten der nun vorliegenden Liste an Funktionalitäten vervollständigt werden. Neben einem Profil der jeweiligen Software (spaltenweise) lag mit der Tabelle nach Abschluss der Software-Durchsicht für jede Funktionalität ein Überblick über verschiedene Möglichkeiten der Umsetzung und den Umfang der Umsetzung in den einzelnen

Software-Tools vor (zeilenweise). Diese Erkenntnisse konnten in die theoretischen Kapitel 2 und 3 einfließen.

2 Funktionen der Skizzeneingabe

2.1 Erstellen und Bearbeiten von Skizzen

2.1.1 Zeichnen (Skizzeneingabe)

Die zeichnerische Interaktion und die Nutzung des Stifts als Eingabegerät sind durch die lange Tradition der Grafik miteinander verbunden. Die freihändige Eingabe von Zeichnungselementen mit dem Stift ergibt eine Liniengrafik, da durch die Bewegung der Stiftspitze eine Linie erzeugt wird. Diese wird i.d.R. als Vektorgrafik durch die Aufnahme von Kontrollpunkten erfasst. Die einzelnen Elemente der Grafik werden Striche genannt (strokes), die gesamte Menge der Linien heißt Tinte (ink). Diese Begriffe haben sich ausgehend von der Terminologie von Microsofts Tablet PC SDK [54] durchgesetzt.

Durch die Auswahl verschiedener Werkzeuge in Analogie zu verschiedenen Stiftspitzen der traditionellen Zeichengeräte, sowie durch die Festlegung von Strichstärken und Farben können den Elementen der Vektorgrafik Eigenschaften zugewiesen werden. Während die Eingabe von grafischen Elementen an sich essentiell für die meisten untersuchten Software-Tools ist, schwankt der Umfang der zuweisbaren Attribute von der Beschränkung auf die reinen geometrischen Informationen der Grafik bis hin zu umfangreichen, selbstdefinierbaren Paletten aus Stiftspitzen, Linienarten, Linienstärken und Farben. Gelegentlich werden visuelle Attribute der Zeichnungselemente mit Bedeutungen verknüpft, d.h. verschiedenfarbig oder in verschiedenen Linienarten gezeichnete Zeichnungselemente werden semantisch interpretiert (z.B. Esquise [51]) oder verschiedene mit der Liniengrafik assoziierte semantische Objekte bewirken eine visuelle Unterscheidung der Zeichnungselemente.

Neben der Geometrie können auch Druck und Anstellwinkel des Zeichengeräts ausgewertet werden, die untersuchten Programme berücksichtigen diese Werte jedoch durchgängig nicht.

Eine alternative Eingabemöglichkeit durch Anzeigen der Position einzelner Kontrollpunkte mit dem Stift ist oft vorzufinden. Einerseits ermöglicht sie eine gewisse „Rückwärtskompatibilität“, d.h. dass die Programme auch mit der Maus bedienbar sind, andererseits ist das Antippen der Eckpunkte die effektivste Möglichkeit, geradlinige Segmente einzugeben. Solche Segmente, von denen bekannt ist, dass sie gerade sein sollen, trotzdem mit einer Freihandlinie zu erfassen, kann aber für die Darstellung von Ungenauigkeiten sinnvoll sein.

2.1.2 Bearbeiten der Zeichnungselemente

Die Möglichkeit, Eingaben zu korrigieren bzw. mindestens rückgängig zu machen oder zu löschen ist wesentliche Voraussetzung für die Bedienbarkeit einer Benutzeroberfläche (siehe 3.1). Die Bearbeitungsmöglichkeiten umfassen das Verschieben, Skalieren, Rotieren und Löschen von Zeichnungsteilen, sowie die Vertexbearbeitung (nur Verschieben und Löschen) und die Änderung von grafischen Attributen.

Die Methode der Änderung durch Überzeichnen ist dem stiftbasierten Workflow besonders adäquat - vorausgesetzt, das System kann die Absicht des Nutzers richtig interpretieren. Das gleiche gilt für Änderungsoperationen durch Stiftgesten (siehe 3.3.4).

Bei Systemen, bei denen nicht nur eine grafische Darstellung erzeugt, sondern die Zeichnung durch das Software-System interpretiert und in andere Modelle überführt werden soll (s. 2.4), müssen zudem für die Bearbeitungsoperationen Methoden vorgesehen sein, die die Änderungen auch in das interpretierte Modell übernehmen.

2.1.3 Auswahl der Zeichnungselemente

Zur differenzierten Bearbeitung müssen die Zeichnungsteile, auf die die Bearbeitungsoperationen angewendet werden, vorher ausgewählt werden. Analog zum Anklicken mit der Maus hat sich das Tippen mit dem Stift als gebräuchlichster Auswahlmechanismus durchgesetzt.

Fortgesetztes Tippen auf verschiedene Elemente kann Mehrfachauswahl bedeuten (z.B. Freak Sketch Client [58]) oder den Wechsel des ausgewählten Elementes. Bei Maus-basierten Oberflächen hat sich die wechselnde Auswahl eines Einzelelementes etabliert, Mehrfachauswahl wird durch Verwendung einer Umschalttaste realisiert oder durch Aufziehen eines Fensters. Beide Methoden erweisen sich in stift- und skizzenbasierten Umgebungen als schwierig - die Umschalttaste, weil meist keine Tastatur vorhanden ist, und das Fenster, weil der diagonale Strich, der das Fenster definiert, von einem Strich unterschieden werden muss, der als grafische Eingabe gilt. In stiftbasierten Oberflächen findet sich deshalb häufig das Einkreisen von Elementen für die Mehrfachauswahl (z.B. Slidewriter [34], InkSeine [40]), wobei jedoch ebenfalls ein Moduswechsels (s. 3.3.2) erfolgen muss, da der Kreis als Geste interpretiert werden muss und nicht als Zeichnungselement.

Die Auswahl der Objekte kann vor oder nach der Angabe der Bearbeitungsoperation erfolgen. Für die geometrischen Änderungsoperationen Verschieben, Skalieren, Drehen hat sich die vorherige Auswahl durchgesetzt. Die Operation wird dann mit Hilfe von Interaktionselementen ausgewählt und durchgeführt, die nach der Auswahl um das Objekt herum erscheinen („Griffe“). Andere Operationen, wie etwa das Löschen, werden häufig im Sinne der Werkzeug-Metapher zuerst ausgewählt und dann durch Auswahl der Objekte auf diese angewendet.

2.1.4 Zeichnungsassistent / „Beautification“

Die eingegebenen Freihandlinien führen in den allermeisten Fällen wie beim Zeichnen auf Papier zu einem sofortigen visuellen Ergebnis auf dem Bildschirm (Ausnahme z.B. optionale Darstellung bei der Palm Graffiti Eingabe). Die Darstellung auf dem Bildschirm ist aber kein physikalischer Automatismus wie beim Zeichnen auf Papier, sondern muss durch die Software erfolgen. Beim Rendern der erfassten Stiftbewegungen können die grafischen Eigenschaften der eingegebenen Linien verändert werden, z.B. wird häufig die Geometrie geglättet. Desweiteren können annähernd gerade, vertikale und horizontale Linien erkannt und in ihrer Repräsentation angepasst werden, sowie einfache geometrische Erkennung durchgeführt werden.

Die optional anschaltbare „Zeichnungsassistent“ im Programm Slidewriter [34] besteht beispielsweise in der automatischen Erkennung und geometrischen Anpassung von Linien im orthogonalen Raster sowie Kreisen. Andere Beispiele finden sich in etablierten Grafikdesign-Software-Paketen, wie etwa die Erkennung annähernd geschlossener Linienzüge mit Lücken in einstellbaren Toleranzbereichen (Flash) oder Fang-Mechanismen für markante Punkte der bereits gezeichneten Geometrie (Adobe Illustrator, Corel Draw).

Bei Pegasus [42] fließt die Erkennung von Orthogonalität, Symmetrie, gleichen Längen in ein Vorschlagssystem ein und bietet so ein Äquivalent zu Raster-, Hilfslinien- und Punktfangsystemen, das ohne explizite Befehle auskommt und sich besser in den Skizziervorgang integrieren lässt. Da das automatische Erkennen und Verändern eine Einschränkung der Nutzerautonomie bei der Eingabe darstellt und Fehlinterpretationen sehr störend wirken, werden Vorschlagssysteme (z.B. auch bei Chateau [43]) genutzt, um zwischen automatischer Erkennung und expliziter Nutzerentscheidung zu vermitteln, ohne den Skizzierfluss zu unterbrechen.

2.2 Verwalten von Skizzen

Skizzierte oder handschriftliche Informationen können sehr zügig notiert werden, so dass sich schnell eine größere Informationsmenge ansammelt. Die Nutzung digitaler Technologien verspricht, diese Informationen besser verwalten, strukturieren und gezielt abrufen zu können. Dafür bedarf es geeigneter Ordnungsprinzipien, die Sätze von kompletten Skizzen oder Teile von Skizzen betreffen.

2.2.1 Verwalten mehrerer Skizzen

In Programmen, die die Funktionen des klassischen Notizbuchs adaptieren, wird häufig auf die gewohnte chronologische Ordnung zurückgegriffen. Die Speicherung und Dateibenennung mit einem Namensschema, das Datums- und Uhrzeitinformaten enthält, erfolgt dann automatisch (z.B. beim one-click-save der Penoffice-Notizzettel [57]). Das Anhängen einer automatischen Nummerierung an einen voreingestellten Dateinamen ist eine weitere Möglichkeit (z.B. Sketchbook [30]).

Bei Systemen, die Modelle mit taxonomischer Struktur aufbauen, können die Skizzen im Ganzen Elementen dieser Strukturen zugeordnet werden. Im Architektur-Bereich werden am häufigsten Grundrisszeichnungen den Gebäudeebenen zugeordnet (z.B. Esquise [51] oder SAM [44]), es ist aber auch die Zuordnung von Ansichten, Fenster-teilung, Fliesenspiegeln usw. zu den entsprechenden Bauteilen denkbar. Zeichnungen, die sich nicht eindeutig Elementen zuordnen lassen (wie etwa Schnitte), weil sie mehrere Elemente oder mehrere Ebenen der Taxonomie umfassen, können mit allen enthaltenen Elementen der taxonomischen Struktur verlinkt werden. Für eine eindeutige Zuordnung bietet sich dann allenfalls die Lokalisierung im dreidimensionalen Geometrie-Modell an.

Einige Programme (z.B. Slidewriter [34]) bieten die Benennung von Einzelskizzen mit einer ikonografischen Bezeichnung an. Sie besteht aus einem mit dem Stift gezeichneten Symbol, das die Skizze für Navigation (siehe 3.4.2) und Anzeigesteuerung repräsentiert. Esquise [51] und Sketchbook [30] nutzen dieses Prinzip für übereinanderliegende Zeichnungen in einem Ebenenstapel (s. 2.2.3).

2.2.2 Strukturierung innerhalb der Skizze und Verknüpfung nach außen

Die Strukturierung innerhalb der Skizze ist ein erster Schritt auf dem Weg zur Anreicherung der grafischen Information mit Bedeutung. Dafür werden Zeichnungselemente in Gruppen zusammengefasst und zueinander in Beziehung gesetzt. Dieser Prozess kann manuell oder automatisch erfolgen. Bei Denim [50] werden beispielsweise übereinanderliegende Zeichnungsstriche automatisch zusammengefasst, die Gruppen können manuell getrennt und um weitere Elemente ergänzt werden. Durch diese Zusammenfassung kann ein logischer Zoom realisiert werden (siehe 2.2.4).

Wie die komplette Skizze in einem Satz von Zeichnungen können auch einzelne oder gruppierte Teile der Skizze benennbar sein, um sie identifizieren und auffinden zu können. Ebenso kann die Zuordnung zu Elementen der taxonomischen Struktur in Systemen, die eine solche Struktur aufbauen, auch auf dem Level der einzelnen Zeichnungselemente oder Gruppen aus Zeichnungselementen erfolgen. In Grundriss-orientierten Architekturskizzen ist dies beispielsweise für Räume, Fenster und Türen sinnfälliger (vgl. SAM [44], Esquise [51]). Die Verknüpfung kann auch über Marker ohne weitere Zeichnungsgeometrie erfolgen (Freak Draft View [61]). Ggf. kann wie dort eine automatische Nummerierung der benannten Zeichnungsteile erfolgen.

2.2.3 Überlagerung (Ebenen)

Viele der durchgesehenen Programme nutzen die aus Grafik- und CAD-Programme etablierte Ebenenmetapher in unterschiedlicher Ausprägung. In der Minimalvariante können verschiedene Hintergrundgrafiken eingeblendet werden (z.B. die Imitation linierten und karierten Papiers bei NoteLab [48]). Slidewriter bietet lediglich eine zweite ein- und ausblendbare Ebene „private ink“, auf der Zeichnungen und Notizen angelegt werden können, die nicht für die Präsentation, sondern nur als Anmerkungen gedacht sind. Bei Sketchbook [30] dagegen steht der komplette bekannte Funktionsumfang

von Ebenen inklusive Sperren und Verändern der Reihenfolge zur Verfügung. Neben verbaler Benennung und Thumbnails kann auch eine ikonografische Bezeichnung zur Identifikation verwendet werden.

CAD-Programme im Architekturbereich nutzen die Ebenenmetapher zur Abbildung von übereinanderliegenden Geschossebenen und das Ineinanderblenden, um übereinanderliegende Bauteile identifizieren zu können. Dieser Verwendung des Ebenenprinzips lässt sich gut mit dem Ansatz einer Grundrisskizze pro Gebäudeebene verbinden und wird z.B. in Esquise [51] eingesetzt.

2.2.4 Level of Detail

Werden Skizzen in verschiedenen Maßstäben angelegt, werden gleichen Zeichnungsteile entsprechend des Maßstabs in verschiedenen Detail-Stufen gezeichnet. Abgebildete Objekte erhalten mehrere Repräsentationen verschiedener Abstraktion. Dieses Prinzip ist aus der Computergrafik als Level of Detail (LoD) bekannt, um beim Rendern eine der Entfernung entsprechend aufgelöste Version von 3D-Objekten zu verwenden und so Rechner-Ressourcen zu sparen. Maßstabsabhängige Abstraktionsstufen der Darstellung im zweidimensionalen Bereich sind aus der Kartographie bekannt und unter der Bezeichnung logischer Zoom in die Bildschirmdarstellung eingegangen.

Logischer Zoom macht eine Überblicksdarstellung von Information, die in kleinteilerem Maßstab erfasst wurde erst möglich. Beispielhaft implementiert ist ein logischer Zoom in der Denim [50], einer Software für skizzenbasiertes Webdesign. In der größten Zoom stufe werden die einzelnen Seiten einer Web-Site nur als Symbole mit ihren Verknüpfungen angezeigt, mit steigender Detaillierung kommen die Titel der Seiten und die grobe Flächenaufteilung hinzu. In Zoomstufen der Überblickssicht für einige Seiten werden nur die Boundingboxen gruppierter Elemente angezeigt, in Detailsichten jeder einzelne gezeichnete Strich.

2.3 Datenaustausch

2.3.1 Einbinden bestehender Daten (Ref.)

Skizzenhafte und handschriftliche Anmerkungen werden in kreativen Prozessen der Ideenfindung verwendet, um vorhandenes Material zu ordnen, Wesentliches herauszuarbeiten, zu markieren und zu annotieren, sowie durch die Bearbeitung zu erschließen. Programme zum Informationsmanagement bieten deshalb Möglichkeiten, vorhandenes Datenmaterial als Grundlage für den Skizziervorgang zu verwenden. Sowohl PenOffice [57] als auch InkSeine [40] erlauben das Zeichnen auf beliebige Bildschirmbereiche und das Speichern des zu Grunde liegenden Screenshots mit der darübergezeichneten Tintenzeichnung. Auch Sketchbook [30] enthält einen einfachen Zugang zu einem Bildschirmfoto. Es stellt außerdem sämtliche Dateien, die in einem Rasterbild-Format in dem aktuellen Ordner liegen, gemeinsam mit den in den selben Ordner gespeicherten Skizzen zur Navigation (Blättern) zur Verfügung.

Darüberhinaus integrieren zunehmend Programme, die nicht schwerpunktmäßig für die Stiftbearbeitung gedacht sind, Funktionen oder Module zur freihändigen Annotation ihrer nativen Elemente wie etwa Powerpoint oder Design Review [29]. Die Annotationen werden dabei aber in der Regel nicht semantisch weiterverarbeitet.

Für den Bereich der Architektur ist die Einbindung von Bestandsplänen oder nicht interpretierbaren Skizzen als Grundlage für die Erstellung strukturierter Skizzen von besonderem Interesse.

2.3.2 Import

Der Import, im Gegensatz zum Referenzieren mit einer Überführung in die Datenstruktur der erfassten Tinte verbunden, kommt ohne eine komplexe Umwandlung nur für Vektorformate in Frage. Die meisten TabletPC-Programme unterstützen jedoch nur das Öffnen des Ink-Formates, das mit Hilfe von Microsofts Ink-Bibliothek [54] einfach geschrieben und gelesen werden kann. Es existieren auch Bestrebungen, eine entsprechende Open-Source-Bibliothek aufzubauen (siehe Open Ink [27]).

2.3.3 Export

Da das Skizzieren vorrangig für konzeptionelle Aufgaben eingesetzt wird, an die sich oft eine vertiefte Bearbeitung mit anderen Methoden anschließt, ist die Möglichkeit der Weiternutzung der Daten von großem Interesse. Meistens ist zusätzlich zum Ink-Format jedoch nur der Export von gerasterten Grafik-Daten möglich (jpg, bmp, gif), selten der Export von Vektor-Daten (svg bei NoteLab [48]), bei denen zumindest die geometrische Information in größerem Umfang erhalten werden könnte. Um auch semantische Information mitzutransportieren kommen, ohnehin nur den Inhalten entsprechende spezielle Datenaustauschformate in Betracht wie IFC oder domainspezifische XML-Derivaten.

2.4 Interpretation, semantische Anreicherung

Um die in der zeichnerischen Eingabe enthaltenen Informationen extrahieren zu können, bedarf es der Interpretation. Dazu wird die Zeichnung in eine topologisch strukturierte Geometrie überführt, die mit Methoden der Mustererkennung durchsucht werden kann. Werden bestimmte Konstellationen gefundener geometrischer Grundformen mit Bedeutung versehen, handelt es sich um eine semantische Interpretation. Bei der Interpretation kann die Reihenfolge der Eingabe berücksichtigt werden.

2.4.1 Geometrische Erkennung

Bei der geometrischen Erkennung werden einfache Grundformen wie gerade Linien-segmente, Kreise, Quadrate, Dreiecke (u.a. Tablet PC SDK [54] und PenOffice [57]) sowie Grundkonstellationen wie Rechtwinkligkeit und Parallelität erkannt. Diese die-

nen zunächst zur „Beautification“ der Zeichnung, d.h. zur gefälligeren Darstellung, sind aber dabei schon eine Interpretation der Intention des skizzierenden Nutzers.

Eine weitergehende topologische Strukturierung erfolgt durch Verschneiden (z.B. Lea [32]), ggf. das Entfernen von Überständen oder Schließen von Lücken in definierten Toleranzbereichen und das Extrahieren geschlossener Polygonzüge (z.B. Sketchboard [45]). Esquise [51] geht noch darüber hinaus, indem durch Agenten mehrfach skizzierte Linien, unterbrochene (gestrichelte und punktierte) Linien und Schraffuren erkannt werden.

2.4.2 Symbolerkennung / Handschrifterkennung

Von Symbolerkennung wird gesprochen, wenn erkannte geometrische Grundformen oder vordefinierte Konstellationen dieser Grundformen mit Bedeutung versehen werden. Etablierte Symbolsysteme existieren für stark formalisierte grafische Notationen, wie für diagrammatische Darstellungen verschiedener Domains oder architektonische Ausführungspläne. Die Symbolsysteme sind oft sehr ausgereift und werden auch in weniger formalisierten skizzenhaften Zeichnungen eingesetzt.

Das Programm Lea [32] verwendet beispielsweise an die Notation in Werkplänen angelehnte leicht abgewandelte Symbole für Türen und Fenster. Aufbauend auf die geometrische Erkennung des Programms Esquise [51] könnten erkannten Schraffuren Materialien zugeordnet werden. Darüber hinaus können erkannte Symbole als Befehle interpretiert werden, man spricht dann von Gestenerkennung (siehe 3.3.4).

Bei der Schrift handelt es sich um die Spezial-Form eines Symbolsystems. Die Buchstaben des Alphabets werden als geometrische Symbole detektiert und mit dem entsprechenden Zeichensatz abgeglichen. Einige Skizzenprogramme unterstützen die Handschrifterkennung in der kompletten Skizze (InkSeine [40]), andere nur in entsprechenden Eingabesituationen (SmartSketches [33] für die Bemaßung), das Tablet PC SDK [54] erlaubt die Anbindung verschiedener Recognition-Engines.

2.4.3 Geometrische Rekonstruktion

Bei der geometrischen Rekonstruktion („geometrical reconstruction“, „line drawing reconstruction“, „reconstruction“) wird aus einer zweidimensionalen Zeichnung durch Interpretation die Geometrie und Topologie eines dreidimensionalen Objektes wiedergewonnen. Es ist der entgegengesetzte Prozess zur Beschreibung von dreidimensionalen Objekten durch zweidimensionale Zeichnungen, sinngemäß das „Lesen“ der Zeichnung als 3D-Modell. In Abgrenzung geht es bei der „restitution“ oder „recognition“ (Symbolerkennung) um das Erkennen von Objekten in der zweidimensionalen Ausgangszeichnung.

Da die skizzenhafte Eingabe im zweidimensionalen Raum stattfindet, wurde die geometrische Rekonstruktion als adäquates Mittel für das stiftbasierte Modellieren dreidimensionaler Objekte untersucht. Die entwickelten Prototypen unterstützen dabei meist die Rekonstruktion bestimmter Arten geometrischer Körper (z.B. Digital Clay [59] Polyeder mit orthogonalem Kanten, 3DJournal [52] Polyeder mit vorwiegend

orthogonalen Kanten und eingeschränkt gekrümmte Flächen, SmoothSketch [47] Körper mit beliebig gekrümmten Flächen). Einige Programme erfordern, die verdeckten Linien mitzuzeichnen, andere interpretieren alle Linien als sichtbare Kanten.

Problematisch ist bei diesem Ansatz für die Modellierung, dass die Eingabe ein hohes Maß an räumlichem Vorstellungsvermögen erfordert, das auch die meisten trainierten Menschen mit suchenden Zeichenbewegungen ergänzen müssen. Diese widersprechen jedoch der gleichzeitig für die Interpretation erforderlichen eindeutigen Linienführung.

2.5 Modellieren

Company u.a. [6] grenzen für das skizzenbasierte Modellieren Gestensysteme von den beschriebenen Rekonstruktionssystemen ab. Bei den Gestensystemen werden bewährte Funktionen des dreidimensionalen Modellierens verwendet und diese nur in stiftgerechter Form aufgerufen und ausgeführt. Systeme, die beide Formen kombinieren (Rekonstruktion für Grundkörper, Gesten z.B. für Transformation und Volumenkombination), werden dort als hybride Systeme bezeichnet. Wenn Karpenko [17] von hybriden Systemen spricht, meint sie dagegen eine Kombination aus Rekonstruktion und Objekterkennung durch Vergleich mit einer Datenbank dreidimensionaler Objekte.

Ein weiterer Ansatz des Modellierens mit Skizzen ist die Festlegung von wechselnden Ebenen im Raum, auf die die zweidimensionale Zeichnung projiziert wird. Diese Zeichenebenen können virtuelle Konstruktionsebenen sein wie bei Chateau [43] oder durch Flächen von bestehenden Objekten definiert werden. Die auf die Fläche projizierte Kontur dient dann als Ausgangspunkt für weitere Operationen wie das Einschneiden von Löchern (3D Journal [52]), das Extrudieren (SketchUp [28]) oder das „Aufblasen“ (Teddy [41]).

3 Ergonomie stiftbasierter Interaktion

Die allgemeinen Anforderungen an grafische Benutzungsoberflächen gelten nicht nur für klassische Oberflächen im WIMP-Stil, sondern auch für die stiftbasierte Eingabe, und werden daher zunächst zusammengefasst (3.1). Zusätzlich ergeben sich aus dem Eingabewerkzeug (Hardwarevarianten 3.2) Besonderheiten, die für die Oberflächen-Ergonomie stiftbasierter Anwendungen Beachtung finden müssen (3.3). Zuletzt schlagen sich auch die Besonderheiten der Skizze als im kreativen Prozess erzeugter grafischer Inhalt in der Benutzungsoberfläche nieder (3.4).

3.1 allgemeine Gesetze des Designs von WIMP-GUIs

Grafische Benutzungsoberflächen, die auf dem nach den Elementen Window, Icon, Menü, Pointing Device benannten WIMP-Paradigma beruhen, sind inzwischen weit entwickelt und ausgereift. Die Anforderungen an diese Benutzungsoberflächen sind einerseits ergonomisch durch physiologische Eigenschaften des menschlichen Nutzers begründet, andererseits resultieren sie aus durch die weite Verbreitung etablierten Standards. Eine Oberfläche, die mit zu vielen Gewohnheiten auf einmal bricht, wird ebenfalls unergonomisch.

Die Anforderungen an eine grafische Benutzungsoberfläche sind in Teil 110 der EN ISO 9241 beschrieben, die Normen für die Schnittstelle zwischen Mensch und Computer beschreibt. Sie definiert die folgenden Anforderungen:

- Aufgabenangemessenheit
- Selbstbeschreibungsfähigkeit
- Steuerbarkeit
- Erwartungskonformität
- Fehlertoleranz
- Individualisierbarkeit
- Lernförderlichkeit

Blazic [4] konkretisiert in Anlehnung an Shneiderman [26] die grundlegenden Anforderungen so:

- Korrekte Funktionalität

- Zuordnung von Software und der realen Welt
- Datenanzeige in Beziehung zur Aufgabe
- Datenausgabe kompatibel zur Dateneingabe
- Leichte Umkehr von Aktionen
- Fehlervermeidung

Zusätzlich fordert er für Computer Aided Conceptual Design (CACD)

- Unterstützung der kreativen Tätigkeit
- Natürliche Interaktionsmethoden
- Schnelles Feedback und Evaluation

Während die allgemein formulierten Anforderungen an grafische Benutzungsoberflächen auch für stiftbasierte Oberflächen gelten, müssen für Details der Umsetzung die grundlegenden Unterschiede zwischen der Ergonomie der Maus und der Ergonomie des Stifts berücksichtigt werden. Dies gilt etwa für Fitts's Law [9], mit dem sich die benötigte Zeit, um ein Ziel auf dem Bildschirm zu erreichen, in Abhängigkeit von der Größe des Ziels sowie der Entfernung vom Startpunkt der Bewegung bestimmen lässt.

Fitts's Law gilt zwar allgemein für die Aktion des Zeigens sowohl auf Bildschirmobjekte als auch auf Objekte der realen Welt, es berücksichtigt jedoch nicht die spezifischen Umstände der Zeige-Aktion, z.B. das Verdecken mit der Hand oder die Bildschirmränder, es gilt außerdem natürlich nicht für andere Interaktionsformen als das Antippen, die der ergonomischen Handhabung des Stifts besser entsprechen.

3.2 Hardware-Varianten

3.2.1 externes Tablett

Stiftbedienbare Tabletts als externe Eingabegeräte werden genutzt, um den Weg von der physisch mit Stift auf Papier festgehaltenen grafischen Information hin zur Repräsentation im Computer abzukürzen, die zuvor meist durch gerastertes Erfassen der fertiggestellten grafischen Vorlage erfolgte.

Das größte Manko der externen Grafiktabletts ist, dass der Ort der Eingabe und das visuelle Feedback auf dem Bildschirm räumlich auseinanderfallen. Bei der etablierten Eingabe mit der Maus ist dies zwar ebenfalls der Fall, genauso wie bei der Eingabe per Tastatur, jedoch scheint hier eine Adaption der Gewohnheiten erfolgt zu sein (ggf. durch explizites Lernen antrainiert wie beim „blinden“ Maschine-Schreiben mit dem Zehn-Finger-System), das bei der Stifteingabe durch die lange kulturelle Tradition des Schreibens und Zeichnens erschwert wird bzw. durch den besonders engen Zusammenhang von zeichnerischer oder schreibender Aktion und visuell sichtbarem Ergebnis unmöglich ist.

3.2.2 externe Tracker für klassisches Stift-Papier-“Interface“

Alternativ zu den externen Tablettts wurden Lösungen entwickelt, bei denen der physische Vorgang des Auftragens von Tinte auf Papier und damit das gewohnte Feedback bei der Eingabe beibehalten wurde. Zusätzlich zum traditionellen Schreib- oder Zeichenvorgang werden dabei die Bewegungen des Stiftes aufgezeichnet und gespeichert. Das Tracking kann auf verschiedene Weise erfolgen: durch mit einem speziellen Raster präpariertes Papier (z.B. Paperium [22], Destiny IO digital pen [24], vormals Logitech-Produkt), durch eine untergelegte Fläche, die den räumlichen Bezug herstellt (z.B. IBM Thinkpad) oder durch einen an den normalen Skizzenblock anklemmbaren Tracker, der die Stiftposition aus den Signallaufzeiten errechnet und per USB direkt an einen angeschlossenen Computer weiterleitet oder im Tracking-Gerät für die spätere Übertragung speichert (z.B. GeneralKeys Digitaler Notizblock [21]).

Während die Intuitivität der Eingabeprozedur so erhalten werden kann, ist es nicht möglich, auf diese Weise über die zeichnerische und handschriftliche Eingabe hinaus mit dem Rechner zu interagieren. Im Sinne einer ergonomisch gestalteten Benutzungsschnittstelle wird aber angestrebt, die Eingabegeräte innerhalb einer Anwendung möglichst selten oder nicht zu wechseln, so dass sich Aufgabenpakete komplett mit der Tastatur oder mit der Maus oder eben mit dem Stift erledigen lassen.

3.2.3 stiftbedienter Eingabe-Bildschirm

Versuche einer verbesserten Verbindung der Eingabe per Stift und der grafischen Benutzungsoberfläche wurden bereits durch die Auflage spezieller Folien mit aufgedruckten Elementen der grafischen Benutzungsoberfläche auf externe Tablettts (z.B. für CAD-Programme) unternommen, so dass Befehlsschaltflächen direkt im Eingabebereich sichtbar waren. Visuelles Feedback am Ort der Eingabe konnte diese Konstellation natürlich nicht leisten.

Ein qualitativer Sprung wurde erst durch die Zusammenlegung von Eingabe- und Ausgabefläche im Bildschirm erreicht. Diese stiftbedienbaren Bildschirme erlauben die komplette Umstellung der Schnittstelle und der grafischen Oberfläche auf stiftbasiertes Arbeiten. Damit konnte erstmals komplett auf zusätzliche Eingabegeräte wie Maus und Tastatur verzichtet werden, gleichzeitig wurde aber die Entwicklung von Software (vom spezialisierten Programm bis zum Betriebssystem) erforderlich, bei der alle erforderlichen Interaktionen mit dem Stift vorgenommen werden können.

Einen ersten ernsthaften Entwicklungsschub bekamen stiftbasierte Computer Anfang der 90er, als Microsoft eine spezielle Betriebssystemversion für Windows 3.1 vorlegte, die jedoch die Leistungsfähigkeit der marktüblichen Hardware überschritt und sich daher nicht durchsetzte. Erst 2002 realisierte Microsoft in einem neuen Anlauf ein erfolgreiches Software-Produkt, die Windows XP Tablet PC Edition, das zeitgleich mit verschiedenen stiftbedienbaren Geräten (Tablet PCs) auf den Markt kam. Inzwischen sind die Funktionalitäten für stiftbasiertes Arbeiten in Microsofts reguläre Betriebssystem-Edition (Vista) eingeflossen.

3.3 Stiftgemäße grafische Benutzungsoberfläche

3.3.1 Stift als pointing device - Adaption der Maus-Interaktion

Während bei der Maus die punktförmige Interaktion (Zeigen und Klicken) als Standardform betrachtet werden kann und die linienförmige Interaktion (Klicken, Gedrückt-Halten und Ziehen) eher umständlich und seltener ist, verhält es sich für den Stift genau umgekehrt. Das Tippen mit dem Stift ist eher unergonomisch, während die linienförmige Bewegung dem Eingabegerät entspricht. Dennoch lassen sich die meisten für die Bedienung mit der Maus entwickelten Oberflächen relativ einfach auf Stiftteingabe umstellen, indem Klicken durch Tippen ersetzt wird, Ziehen bei gedrückter Maustaste durch linienförmige Bewegung direkt auf dem Bildschirm (mit Druck auf den Bildschirm), sowie die Mausbewegung ohne gedrückte Taste durch linienförmige Bewegung mit leichtem Abstand vom Bildschirm.

Einschränkungen der Ergonomie ergeben sich bei der Übertragung des Doppelklicks und der rechten Maustaste in stiftbasierte Schnittstellen. Ersteres wird meist durch unergonomisches Doppeltippen gelöst, letzteres durch eine zusätzliche Taste am Stift, bei deren Benutzung der Stift jedoch seine intuitive Handhabbarkeit einbüßt.

Schwierig sind außerdem Fälle, in denen Anwendungen durch Beibehalten und Adaption der vorhandenen Mausinteraktionen umgestellt werden und zusätzlich einen dem Stift adäquaten Interaktionsmodus anbieten wollen, etwa einen Zeichen-, Handschrift- oder Gestenmodus. Dafür muss dann entweder auf Funktionen, die zuvor mit der Maus erreichbar waren, verzichtet werden (z.B. Kontextmenü auf der rechten Maustaste) oder es müssen weitere Möglichkeiten des Moduswechsels eingeführt werden. Eine Variante ist z.B. eine Verzögerung am Beginn oder Ende der Stiftbewegung auf dem Bildschirm, die bei PenOffice „start delay“ und „end delay“ genannt wird [23]. Die Webseiten-Prototyping-Software Denim [50] nutzt die Methode des Tippens mit darauffolgender Verzögerung („tap and hold“) zur Auswahl von Elementen und eine verzögert begonnene Bewegung zum Verschieben derselben. Diese verzögerten Interaktionen stören jedoch das flüssige Arbeiten mit dem Stift erheblich.

Eine andere Möglichkeit des Moduswechsels ist das Umdrehen des Stifts (z.B. zum Radieren). Dies muss jedoch vom Eingabegerät unterstützt werden.

3.3.2 Modeswitching

Kara u.a. [16] beschreiben Moduswechsel als zentrales Problem, das bei der Evaluierung ihrer Skizziersoftware für industrielles Produktdesign aufgetreten ist. Implizite oder automatisierte Moduswechsel sind am bequemsten für den Nutzer, da er nicht durch einen expliziten Befehl den Zeichenfluss unterbrechen muss.

Ein Beispiel für einen impliziten Moduswechsel ist die Auswahl durch Tippen: Sobald das erste Tippen erfolgt, ist der Auswahlmodus aktiv, wenn wieder eine Bewegung auf dem Bildschirm ausgeführt wird, wird in den Zeichenmodus zurückgeschaltet. Diese Lösung funktioniert nicht mehr, wenn die Interaktion des Tippens auch für andere Modi verwendet wird, z.B. für das Zeichnen geradliniger Segmente, oder wenn eine Verschieben-Funktion durch Tippen und sofort anschließendes Ziehen ausgeführt

werden soll. Wenn mehrere Funktionen mit der gleichen Stiftbewegung aufgerufen werden, muss für mindestens eine ein expliziter Moduswechsel erfolgen.

Aus den klassischen WIMP-GUIs ist der explizite Moduswechsel durch die Werkzeugmetapher bekannt. Durch Auswahl eines Werkzeugs werden die folgenden Interaktionen in einer bestimmten Weise interpretiert, die Form des Mauszeigers gibt ein Feedback, welcher Modus gerade aktiv ist. Meist sind die Werkzeuge in Werkzeugkästen angeordnet, die verschiebbar und anpassbar sind. Bei experimenteller Software wird auch versucht, die Werkzeuge an beliebigen Stellen auf dem Bildschirm ablegen zu können (Denim [50]), um sie schnell griffbereit zu haben. So soll die Benutzer-Anpassung der grafischen Oberfläche intuitiver und flexibler gestaltet werden können.

In den gewohnten grafischen Oberflächen werden außerdem Umschalttasten für Moduswechsel verwendet, so etwa die Shift-Taste um zwischen verschiedenen Objektauswahl-Varianten umzuschalten oder in 3D-Programmen das Umschalten in den Navigationsmodus. Für tastaturlose Umgebungen werden alternativ Moduswechsel durch Stiftgesten propagiert. Die Gesten müssen dann jedoch sehr einfach gestaltet sein, um nicht komplizierter als ein expliziter Befehl zu sein. Schon die verzögerte Stiftbewegung wie in Denim [50] zum Auswählen und Verschieben von Elementen unterbricht den Arbeitsfluss merklich. Eine wenig beachtete Möglichkeit neben der Taste am Stift ist die alternate-Taste am Tablet-PC, die jedoch alle Varianten der Umschalt-Tasten auf der Tastatur zusammen ersetzen muss (z.B. K-Sketch [35]).

3.3.3 maximierter Eingabebereich

Für das Skizzieren ist eine möglichst große Eingabefläche wünschenswert, die frei von Interaktionselementen ist. In Umgebungen mit mehreren Bildschirmen ist die Analogie zum leeren Skizzierpapier einfach umzusetzen, für mobile Geräte mit nur einem zudem meist kleineren Bildschirm muss die grafische Oberfläche so angepasst werden, dass möglichst viel Platz für die Skizzeneingabe bleibt.

Klassische WIMP-Software in den professionellen Bereichen des Grafikdesign, des CAD, des 3D-Modellierens und der Computeranimation erlauben bereits heute das Ausblenden sämtlicher Interaktionselemente, Dialoge und zusätzlicher Fenster mit einem einfachen Tastendruck, so dass der Nutzer mit voller Konzentration auf den Zeichen- oder Modellbereich arbeiten kann. Auch die stiftbasierten Benutzungsoberflächen versuchen die eigentliche Arbeitsfläche zu maximieren (z.B. Sketchbook [30] und Chateau [43]) und zusätzliche Fenster, Menüs, Werkzeugleisten höchstens bei Bedarf einzublenden.

3.3.4 angepasste Interaktionselemente: Kreismenüs

Die Interaktionselemente der WIMP-GUIs sind das Menü und der Werkzeugkasten mit Icons. Beide sind meist um den Arbeitsbereich herum angeordnet und verkleinern ihn nicht nur, sondern lenken auch von der eigentlichen Eingabe ab. Außerdem unterbricht die Bewegung an den Bildschirmrand den Zeichenfluss.

Eine weitere Alternative ist, eine Kombination aus beiden in einer minimierten Form

am Bildschirmrand oder frei verschiebbar auf dem Bildschirm anzuordnen. Die Kombination aus Icons und Beschriftung - also die Zusammenführung von Werkzeugleiste und Menü - ist ein Trend, der z.B. bei der letzten Generation von Microsofts Office-Produkten beobachtet werden kann. Damit wird das Auffinden der Befehle vereinfacht.

Das Kreismenü integriert über die Verbindung einer hierarchischen Menüstruktur mit Icons hinaus Eigenschaften, die vom Kontextmenü her bekannt sind: die Menüstruktur entfaltet sich ausgehend von einem einzelnen Punkt. Bereits die 3d-Modellierungssoftware Maya nutzte hierarchische Kontextmenüs (mit Textbefehlen). Kreismenüs finden sich in InkSeine [40] und Sketchbook [30].

Kreismenüs haben auch den Vorteil, dass sie sich mit Stiftgesten kombinieren lassen, die sich aus Linien in Richtung der Menüpunkte zusammensetzen. Kreismenüs können den Lernprozess dieser Gesten unterstützen, wenn die Menüpunkte immer gleich angeordnet sind. Mit diesem Anspruch laufen sie jedoch dem Prinzip der Kontextmenüs entgegen, die nicht nur am Punkt des Cursors erscheinen, sondern auch eine der Situation (ausgewählte Objekte, gewählte Ansicht ...) entsprechende und damit wechselnde Auswahl an sinnvollen Befehlen anpassen.

3.3.5 angepasste Interaktionselemente: Stiftgesten

Eine weitere Möglichkeit, Befehle aufzurufen, ohne Werkzeugleisten oder Menüs zur Verfügung zu haben, war bisher das Ausführen von Tastaturkürzeln. Gerade im professionellen Bereich erlaubten Tastaturkürzel, den Bildschirm von allen unnötigen Interaktionselementen zu befreien. Bei stiftbasierten mobilen Geräten wird zugunsten des Gewichts meist auf die Tastatur verzichtet, so dass statt dessen die Gestenerkennung bemüht wird.

Universelle Gesten, die sich wie die Tastatur-Kürzel Strg-X/C/V bereits zu etablieren scheinen, sind die Kritzel-Geste für das Löschen von Objekten, das Auswählen durch Einkreisen sowie eine kleine rotierende Bewegung zum Scrollen (InkSeine [40], Slidewriter [34]). Bei Digital Clay [59] werden Gesten „doodles“ (Kritzeleien) genannt. Pen Commander [56] benutzt handschriftliche Textkommandos, die durch Einkreisen als Befehle interpretiert und ausgeführt werden.

Das Problem der Unterscheidung zwischen Geste und gezeichnetem Inhalt wird bei der Evaluationen von Nutzungsoberflächen als Problem identifiziert (u.a. [16]). Bei Systemen, die die Skizze als gezeichnete Eingabe interaktiv inhaltlich interpretieren, verkompliziert sich die Problematik, da hier theoretisch mehrere Interpretations-Engines nebeneinander laufen müssten und sich die Symbolsysteme von Stiftgesten und Skizzenvokabular nicht überschneiden dürfen. Einfach Lösungen sind hierbei nur durch verzögerte Interpretation oder expliziten Moduswechsel denkbar.

3.4 Grafische Benutzungsoberfläche zur Eingabe skizzenförmiger grafischer Inhalte

3.4.1 Auflösung der Befehlssyntax

Company u.a. [6] sehen das Hauptproblem der Kommunikation zwischen CAD-System und Designer darin, dass CAD-Systeme eine nicht-sequentielle (grafische) Ausgabe haben, während sie nur sequentielle (verbale) Eingabe akzeptieren. Dies liege in der sequentiellen Natur algorithmischer Sprachen. Das konzeptionelle Modell und das Interaktions-Frontend müssten „grafisch“ im Sinne von nicht-sequentiell sein, um diese Kluft zu überbrücken.

Eine ähnliche Argumentation findet sich auch bei Shneidermans These vom „Verschwinden der Syntax“ [26] bei grafischen Benutzungsoberflächen - die Befehle müssen nicht mehr in einer bestimmten Reihenfolge aufgerufen werden. Die These erscheint allerdings fraglich, da auch in grafischen Benutzungsoberflächen eine bestimmte Reihenfolge der Eingabe von Befehl und Argumenten vorgegeben ist. Oft erfolgt zunächst die Werkzeugauswahl, dann die Eingabe der Parameter (die auch grafisch erfolgen kann) meist auch in einer bestimmten Reihenfolge. Wenn die Reihenfolge umgekehrt implementiert wird, handelt es sich dabei ebenso um eine syntaktische Variante wie bei verschiedenen Formen der Argumentangabe in Kommandozeilen-Umgebungen.

Vorschlagssysteme versuchen die Befehlssyntax aufzuheben, indem sie jedwede Nutzereingabe einer Analyse unterziehen, um aus den ausgewählten Objekten und eingegebenen Parametern Vorschläge zu generieren. Befehle müssen hier tatsächlich nicht mehr explizit eingegeben werden. Beispiele dafür sind Chateau [43] und SmartSketches [33].

3.4.2 Navigation

Wenn die Navigation innerhalb einer Skizze oder eines Modells ebenfalls mit dem Stift erfolgen soll, ist dafür die Unterscheidung von zeichnender und navigierender Stiftbewegung erforderlich. Dies kann über eine Modusumschaltung geschehen oder durch die Bereitstellung von Navigationsschaltflächen, auf denen die navigierende Bewegung ausgeführt wird (z.B. zoomslider). Zusätzliche Schaltflächen widersprechen aber der Forderung nach einem möglichst großen und freien Eingabebereich.

Neben der Umschaltung in den Navigationsmodus selber muss zwischen den einzelnen Navigationsarten Zoom, Pan und Rotate unterschieden werden. Chateau [43] implementiert dafür ein Gestenset. Dafür wird der Anfang der Navigationsbewegung analysiert: Beginnt die Bewegung horizontal, dann erfolgt eine Pan-Navigation, beginnt die Bewegung vertikal, erfolgt eine Zoom-Navigation und erfolgt der Beginn der Bewegung vom Bildschirmrand Richtung Bildschirmmitte, dann erfolgt eine Orbit-Navigation. Die Umschaltung in den Navigations- und gleichzeitig Gestenerkennungsmodus erfolgt durch Drücken der rechten Maustaste oder der Zusatz Taste am Stift.

Sketchbook [30] nutzt ein „Zoom&Pan-Tool“ für die Navigation, es handelt sich dabei um ein Kreismenü, bei dem die Befehle nicht in Kreissegmenten sondern in

konzentrischen Kreisen angeordnet sind (Zwiebelschalen-Menü). Das gleich Interaktionselement kommt für die Bearbeitung (Skaliern, Drehen, Verschieben) der gesamten Skizze oder von Teilen davon zum Einsatz, wodurch die Handhabung sehr konsistent wird.

Neben der Navigation innerhalb der Skizze ist für Programme, die die Verwaltung mehrerer Skizzen unterstützen, die Navigation in diesen Sätzen von Skizzen erforderlich. Dafür kann eine scrollbare Thumbnail-Leiste genutzt werden (z.B. Slidewriter [34]), Stiftgesten zum Blättern (z.B. Sketchbook [30]) oder die Navigation in einklappbaren Baumstrukturen, wenn Programme die Skizzen einer hierarchischen Taxonomie zuordnen (z.B. SAM [44]).

3.4.3 Feedback

Unter Feedback wird hier verstanden, dass der Nutzer über den Erfolg seiner Eingabe informiert wird und seine Eingaben kontrollieren kann. Im einfachsten Fall erfolgt dies durch das Rendern der gezeichneten Linie auf den Bildschirm. In interpretierenden Systemen ist wichtiger noch das Feedback zur erfolgten Interpretation der Eingabe. Meist werden dabei die eingegebenen Daten verändert und intern in ein anderes Modell überführt, Feedback kann dann durch Umschalten zwischen den Repräsentationen der verschiedenen Modelle oder durch Ineinanderblenden erfolgen.

Eine kritischer Punkt beim Feedback entsteht durch die Verzögerung der Interpretation. Company u.a. [6] erörtern die Vor- und Nachteile der interaktiven und stapelweisen (verzögerten oder sogar explizit ausgelösten) Geometrieerkennung am Beispiel der „Beautification“: Während die interaktive Variante besser für das sofortige Feedback und die Integration in eine grafische Benutzungsoberfläche ist, eignet sich die Stapelverarbeitung besser für Analysen über die gesamte Skizze (z.B. Symmetrierkennung). Bereits bei der Handschriftinterpretation kann die verfrühte Interpretation zu Fehlern führen, wenn Schriftzeichen möglicherweise noch nicht vollständig eingegeben sind. Pen Office [23] führt das Feedback der Handschriftinterpretation mit noch größerer Verzögerung („deferred recognition“) aus - es wartet auf die Eingabe ganzer Wörter und die eventuelle Markierung als Stiftkommando. Auch Programme die grafische diagrammatische Symbole erkennen, beruhen auf verzögerter Erkennung, Electronic Cocktail Napkin [39] beispielsweise startet erst mit der Interpretation, wenn der Benutzer eine 5 Sekunden lange Zeichenpause eingelegt hat.

Auch im für das Projekt zu entwickelnden Modul zur Skizzenerfassung wird sich die Frage stellen, in welchem Ausmaß Interpretation sofort erfolgen kann und muss, bzw. in welchem Ausmaß eine verzögerte, explizit ausgelöste Interpretation von Vorteil ist.

Eine andere Form des Feedbacks betrifft die Modusänderung: Der Nutzer muss immer informiert werden, in welchem Eingabemodus das Programm sich befindet. Etabliert hat sich dafür die Veränderung des Mauszeigers wie bei der Werkzeugmetapher bereits beschrieben (s. 3.3.2).

3.4.4 Ungenauigkeit und Ambivalenz

Die skizzenhafte Eingabe von Informationen in frühen konzeptionellen Phasen des Design-Prozesses ist in großem Umfang mit Ungenauigkeiten und Ambivalenz behaftet. Darauf müssen Software-Tools, die diese Phase unterstützen, in zweierlei Hinsicht reagieren: Einerseits müssen diese Eigenschaften der eingegebenen Information bei der Interpretation berücksichtigt werden, andererseits müssen sie in die Repräsentation der eingegebenen und der interpretierten Ergebnisse Eingang finden, um nicht die trügerische Vorstellung einer Genauigkeit und Eindeutigkeit zu erwecken, die nicht vorhanden ist.

In die Repräsentation von Modellen, die skizzenhafte Eingabe interpretieren, kann die Darstellung der eingegebenen Zeichnung integriert werden und so die Ausgangszeichnung als Visualisierung der Eingabegenauigkeit erhalten bleiben (siehe u.a. 3D Journal [52] oder Lea [32]). Dann stellt sich jedoch die Frage, ob und wie Änderung im interpretierten Modell in die Eingabeskizze zurück überführt werden, sofern das System die Möglichkeiten dieser Bearbeitung bietet. Auch wenn die Eingabe-Skizze nicht beibehalten wird, kann die Visualisierung des interpretierten Modells die Eingabegenauigkeit reflektieren. Für zweidimensionale Geometrien werden beispielsweise Linienstärke und Linienendüberstand als Indikatoren für die Genauigkeit eingesetzt (SAM [44]), für dreidimensionale Geometrien Non-Photorealistic Rendering (NPR), wie Cartoon-Rendering (z.B. SmartSketches [33] oder SketchUp [28]).

Eine Möglichkeit, Ungenauigkeiten bei der Interpretation der eingegebenen Geometrie zu berücksichtigen, haben beispielsweise Lank u.a. mit dem „sloppy selection“ Mechanismus entwickelt [18]. Dabei wird die Mehrfachauswahl von Objekten durch Einkreisen mit dem Stift in Abhängigkeit von der Zeichengeschwindigkeit mit unterschiedlichen Toleranzbereichen durchgeführt.

Ambivalente Eingaben müssen weitestgehend ausgeschlossen werden, um die Interpretation der Eingabe zu ermöglichen. Dazu müssen entweder Restriktionen der Eingabe vorgenommen werden oder ein eindringliches Feedback über die erfolgte oder unmögliche Interpretation bereitgestellt werden.

Uneindeutigkeiten der Eingabe können auch mit Vorschlagssystemen behandelt werden, die dann mehrere Möglichkeiten der Interpretation anbieten und dem Benutzer die Konkretisierung seiner Intention überlassen (siehe z.B. [43]).

4 Software-Review

In diesem Teil werden die untersuchten und in den vorhergehenden Teilen 2 und 3 bereits referenzierten Software-Tools beschrieben.

4.1 Kategorien

Die Zusammenfassung der untersuchten Software in Gruppen ähnlicher Tools kann anhand der Kategorien Anwendungsbereich, Art der behandelten Information und Grad der Interpretation dieser Informationen erfolgen.

4.1.1 nach Domain

Skizzenbasierte Anwendung sind für viele Fachbereiche vorstellbar, da die handschriftliche Notiz ergänzt um zeichnerische Darstellungen ein probates Mittel zur Notation, Präsentation und Kommunikation ist. Entsprechend existieren Software-Tools mit Schwerpunkten in allen drei Bereichen, die in unterschiedlichem Umfang um Funktionen aus der fachlichen Domain ergänzt sind. Aus dem fachlichen Anwendungsbereich ergibt sich oft auch die Art der Information (siehe 4.1.2), die vorrangig behandelt wird.

Die Benutzung von Notizbüchern mit handschriftlichen Vermerken und Skizzen ist eine Form des Persönlichen Informationsmanagement (PIM), die in vielen Fachbereichen angewendet wird, nicht nur in kreativen Bereichen. Dienel [8] untersucht beispielsweise die Notizbücher von Ingenieuren. Anwendungen, die das Persönliche Informationsmanagement und das Notieren unterstützen, haben ihren funktionalen Schwerpunkt meist im Bereich der Verwaltung von Skizzen und der Einbindung externer Informationen.

In anderen Anwendungen kommen skizzenhafte Zeichnungen als reiner Eingabemodus für die anderweitige Weiterverwendung zur Anwendung z.B. für mathematische Formeln [19], Text, Schaltpläne bis hin zu 3D-Modellen. Der Schwerpunkt der Anwendungen liegt dann meist auf der Symbol- und Handschrifterkennung oder der geometrischen Rekonstruktion. Mit LADDER [11] wurde eine Formbeschreibungssprache entwickelt, mit der symbolische Darstellungen fachspezifischer Objekte beschrieben werden können, mit dem Ziel, letztendlich stiftbasierte Interfaces herzustellen. Bei den Symbolen handelt es sich um Elemente von zweidimensionalen Diagrammen.

Das Skizzieren im Bereich des Grafikdesign kann auf eine längere Tradition zurückgreifen. Skizzenanwendungen, deren Zweck die Erstellung von Grafiken und visuellen Präsentationen ist, weisen besonders ausgeprägte Funktionalität im Bereich der Zeichenfunktionen auf und bemühen sich um besonders intuitive Integration des Stifts

als Eingabewerkzeug, da professionelle Benutzer in diesem Bereich meist die physische Benutzung verschiedener Stifte und Pinsel gewohnt sind.

Das Skizzieren und Annotieren bestehender Materialien erfolgt nicht nur beim Persönlichen Informationsmanagement, sondern auch zur Kommunikation, zum Informationsaustausch und im Bereich des Lehrens und Lernens verschiedener Fachgebiete. Als Beispiel sei eine Anwendung aus dem medizinischen Bereich angeführt, die die interaktive Interpretation und Kommunikation der Ergebnisse bildgebender diagnostischer Verfahren erlaubt: MIBlob [1]. Ein anderes Beispiel sind Programme zum Anfügen von Annotationen an 3D-Modellen (Autodesk Design Review [29])

4.1.2 nach Art der bearbeiteten Informationen

Die Art der vorrangig behandelten Information hängt eng mit dem fachlichen Anwendungsbereich zusammen, wobei jedoch der 2D-Linien-Grafik als natives Ergebnis der Bewegung des Stifts (linienförmig) auf dem Bildschirm (zweidimensional) besonders allgemeine Bedeutung zukommt. Die als zweidimensionale Grafik eingegebene Information muss dann durch Mechanismen der Interpretation in andere Informationsarten umgewandelt werden. Diese reichen von Text über verschiedene andere Symbolsysteme bis hin zu dreidimensionalen Modellen.

Text: Zur Interpretation der stiftbasierten Eingabe von Texten und Annotationen wird die Handschrifterkennung eingesetzt, sowie die automatische Extraktion von Bereichen mit handschriftlichen Texten in Grafiken. Die Eingabe mathematischer Formeln ist zwischen Text und 2D-Grafik angesiedelt.

2D-Grafik: Die zweite Dimension ist die native Dimension der Stiftbasierten Eingabe. Formalisierte diagrammatische Darstellungen sind durch Symbol- und Texterkennung interpretierbar.

2,5D : Das 2,5-dimensionale Modell bildet den idealen Vermittler zwischen zweidimensionaler Skizze und dreidimensionalem Modell, und wird oft in architekturnahen Skizzenanwendungen verwendet, wie z.B. Esquise [51] und Lea [32], da viele architektonische Bauteile 2,5-dimensional beschreibbar sind.

3D : Die dreidimensionale Modellierung mit Hilfe von Skizzen unterstützt verschiedene Geometrien, von polygonalen (die für die meisten architektonischen Geometrien ausreichen) bis zu Freiform-Flächen. Die Interpretation kann mit Hilfe der geometrischen Rekonstruktion erfolgen.

2D + Zeit : Animationsprogramme fügen den zwei räumlichen Dimensionen die zeitliche hinzu, auch hierzu existieren Umsetzungen in stiftbasierte Anwendungen, z.B. K-Sketch [35].

3D + Zeit : Programme zum Animieren dreidimensionaler Modelle scheinen bisher zu komplex, um mittels skizzenhafter Eingabe bewältigt zu werden. Entsprechende Programme liegen nicht vor, die Verbindung von Funktionen aus den Bereichen 2D + Zeit und 3D ist jedoch prinzipiell vorstellbar.

4.1.3 nach Umfang der Interpretation

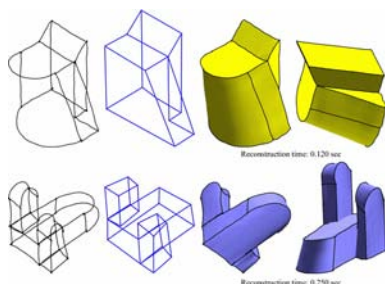
Da das Ergebnis der Skizzeneingabe eine zweidimensionale Grafik ist, können Programme, die zweidimensionale grafische Information zum Gegenstand haben, auch ohne Interpretation damit weiterarbeiten. Auch die zweidimensionale Grafik kann aber einer Interpretation unterzogen werden.

Alle Anwendungen, die mit anderen Informationsarten arbeiten, müssen eine Interpretation in größerem oder kleinerem Umfang vornehmen, wenn sie auf der freihändig eingegebenen Zeichnung beruhen. Wenn sie hingegen keine skizzenhafte Eingabe unterstützen, sondern nur die etablierten Methoden der Bearbeitung der entsprechenden Informationen für stiftbasierte Schnittstellen adaptieren, werden Interpretationsmechanismen meist ebenfalls überflüssig.

4.2 allgemeine Beschreibung

Im Folgenden werden die untersuchten Softwaretools kurz vorgestellt. Es wurde jeweils ein charakteristischer Screenshot dokumentiert, das Jahr der letzten veröffentlichten Version und ggf. die Downloadquelle, bzw. - sofern eine solche nicht zu benennen ist - eine Veröffentlichung, in der die Funktionalität beschrieben ist. Desweiteren wurden für evtl. weitere Untersuchungen (als weiterführender Usability-Test oder als Review auf Quelltext-Ebene) festgehalten, ob es sich um eine kommerzielle oder kostenfreie Lizenz handelt, ob der Quelltext offengelegt ist und ggf. welche Programmier-Sprache verwendet wurde. Weiterführend ist für weitere Untersuchungen von Interesse, ob bestimmte Installationsvoraussetzungen (z.B. Laufzeitumgebungen) erforderlich sind und welche Toolkits und Bibliotheken verwendet wurden.

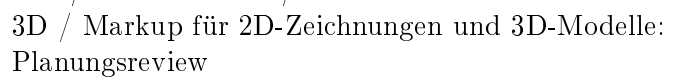
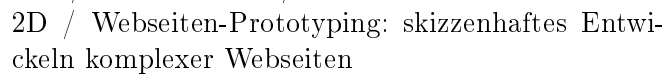
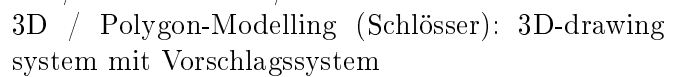
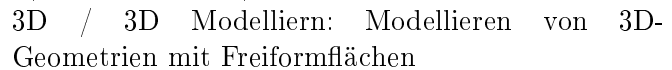
Der besseren Auffindbarkeit halber sind die Software-Pakete alphabetisch sortiert.

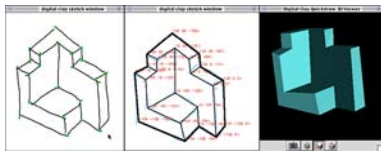


3D Journal 2005 [52] [20]

frei / Closed Source / -

3D / Modellieren: geometrische Rekonstruktion von
3D-Modellen aus 2D-Skizzen





Digital Clay 1998 [59] [25]

- / Closed Source / -

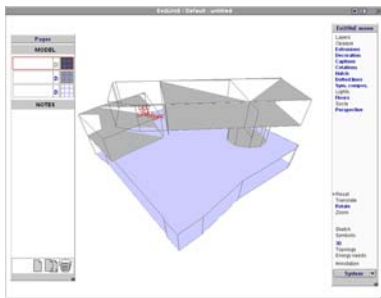
3D / Modellieren: Geometrische Rekonstruktion



Electronic Cocktail Napkin 1996 [39] [10]

- / Closed Source / Common Lisp

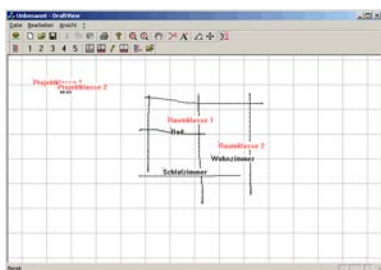
2D / Design-Skizzen: Notizbuch mit Mustererkennung



Esquise 2007 [51] [15]

auf Anfrage / Closed Source / -

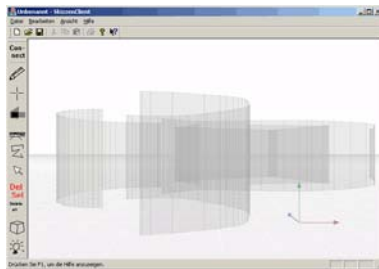
2,5D / Architektur-Skizze: architektonisches 3D-Modell durch Skizzen mit Hilfe Mustererkennung durch Agenten



Freak Draft View 2007 [61]

intern / Closed Source / C++

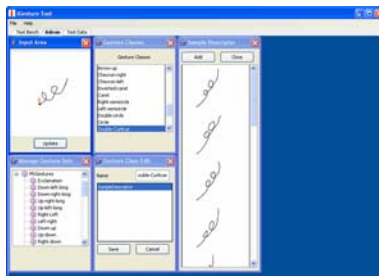
2D / Bauwerksmodellierung: Verknüpfen von geometriefreier Bauwerkstruktur mit Markern auf Skizzen



Freak Sketch Client 2007 [58]

intern / Closed Source / C++

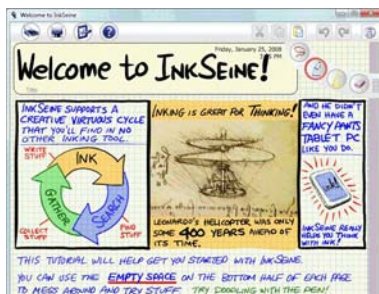
2,5D / Modellieren 3D: Skizzieren in der Freak-Umgebung



iGesture 2007 [31]

Frei (Apache) / Closed Source / Java

2D / Software-Entwicklung: Gesten-Erkennungs-Framework zur Entwicklung und zum Testen von Gesten und gestenbasierten Anwendungen



InkSeine 2008 [40] [12]

frei / Closed Source / -

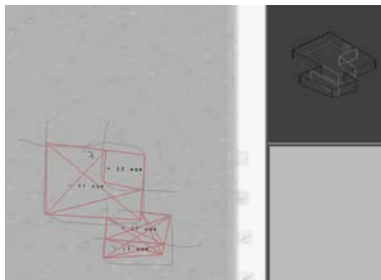
2D / Notizen, Informationsmanagement:



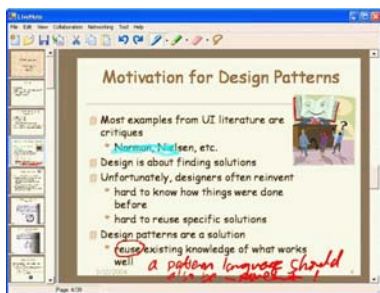
K-Sketch 2008 [35]

frei / Open source / C#

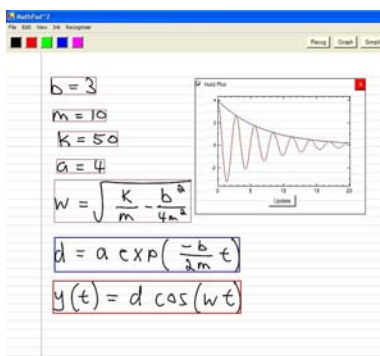
2D+T / Animation: einfaches Erstellen informaler Animationen aus Skizzen, für Laien und zum Prototyping



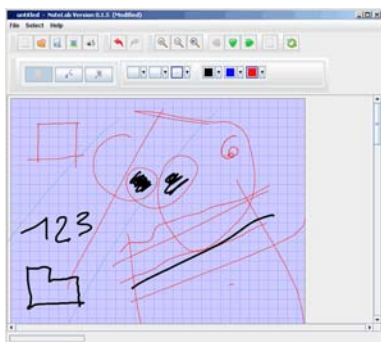
Lea 2003 [32] [2]
intern / Closed Source / -
2,5D / Architekturskizze: architektonisches 3D-Modell erstellen durch Grundrisskizzen



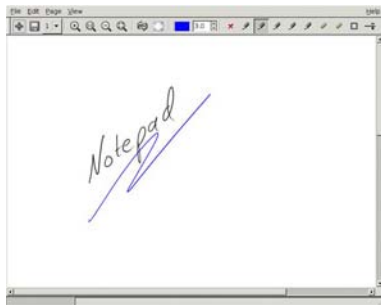
Livenotes 2005 [62]
frei (BSD) / Open source / C#
2D / interaktives Lernen: Unterstützung des Anlegens und Austauschs von Notizen in kleinen Gruppen, Importieren von ppts, Annotationen und instant messaging



MathPad 2007 [49] [19]
- / Closed Source / -
2D / Formeleingabe: Mathematische Formeln



NoteLab 2007 [48]
frei (GPL) / Open source / Java
2D / Notizen: „digital notebook“ für Tablet-PCs

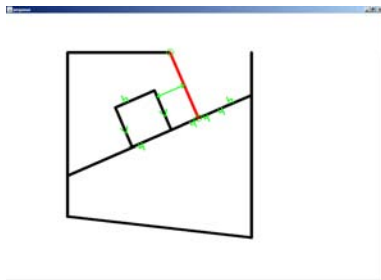


Notepad 2006 [37]

frei (GPL) / Open source / Python
2D / Notizen: stiftbasierte Notizen in Python

Open Ink [27]

- / Closed Source / -
/ Entwicklung: Alternative zu MS Tablet-PC Ink
SDK



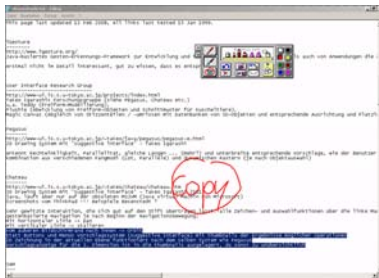
Pegasus 1998 [42] [14]

frei / Closed Source / Java
2D / Vektorgrafik: 2D-Zeichen-System mit suggesti-
ve Interface



Pencil 2008 [55]

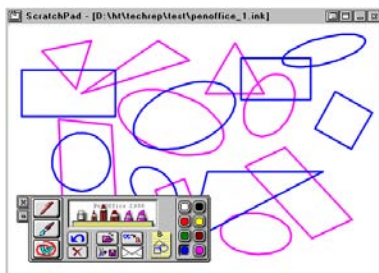
frei / Open source / C++
2D+T / Animation: einfaches Programm zum Er-
stellen von klassischen 2D-Animationen



PenCommander 2005 [56] [23]

Trial 30 Tage / Closed Source / -

2D / Betriebssystem: Ausführen von Programmen und Befehlen über selbstdefinierte textbasierte Gesten



PenOffice 2005 [57] [23]

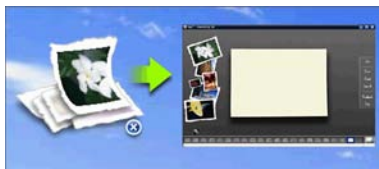
Trial 30 Tage / Closed Source / -

2D / Notizen: Handschrifterkennung + versch. Skizzentools, wie Klebezettel (Scratchpad), Geometriererkennung in Notizen

Piccolo.NET User Interface Library 2005 [38]

Frei (BSD) / Open source / Java, C#

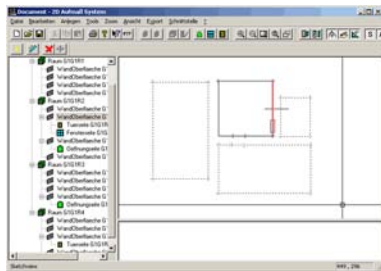
/ Entwicklung: Toolkit für ZUI (zoomable interfaces) für hierarchische Daten in Java, C#



Ript 2008 [53]

frei / Closed Source / -

2D / Hobby-Kreativität: „Ausreißen“, Sammeln, Collagieren und Beschriften von Informationen aus Internetseiten

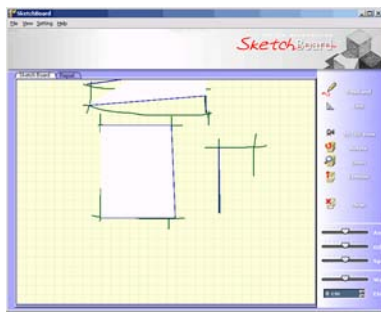


SAM 1998 [44]

intern / Closed Source / -
2,5D / Bauaufnahme: Bauaufnahme tool

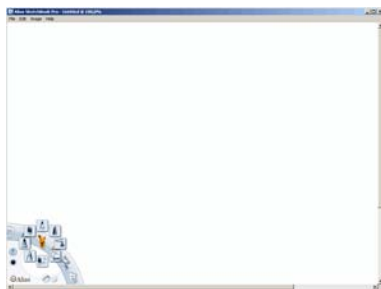
SiGeR 2005 [60]

frei / Open source / C#, VB .net
/ Software-Entwicklung: „Simple Gesture Recognition“ - Bibliothek für einfache Gestenerkennung



Sketchboard 2004 [45]

Frei / Closed Source / C++
3D / CAD-Modellierung 3D: Experimenteller CAD-Modeller



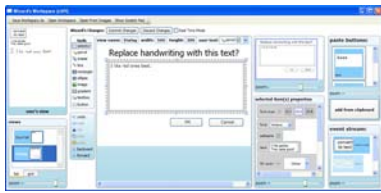
Sketchbook Pro 2008 [30]

Trial / Closed Source / -
2D / Entwurfsskizzen: Skizzenbuch für Designer



SketchUp 2007 [28]

Frei / Closed Source / -
3D / 3D-Modellieren:



SketchWizard 2007 [36] [7]

frei / Open source / C#

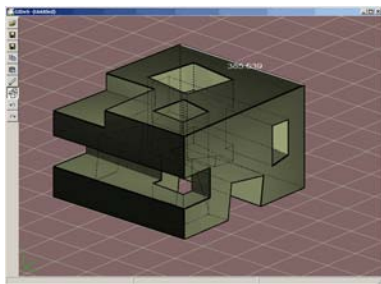
2D+T / GUI-Prototyping: GUI-Prototyping-
Applikation, die Wizard-of-Oz-Testing ermöglicht



Slidewriter 2006 [34]

frei (BSD) / Open source / C#

2D / Präsentation: ink- und gestenbasiertes Präsen-
tationsprogramm für Tablet-PCs



SmartSketches (Gides++) 2004 [33]

Frei / Closed Source / -

3D / CAD-Modellierung 3D: Skizzieren mit Vor-
schlagssystem

SmoothSketch 2006 [47] [17]

- / Closed Source / -

3D / 3D Modellieren: Modellieren von 3D-
Geometrien mit Freiformflächen

Tablet PC SDK 2005 [54]

Frei / Closed Source / -
/ Software-Entwicklung: Entwicklungsumgebung für
Tablet-PC Software



Teddy 2003 [41]

frei / Closed Source / Java
3D / 3D-Modellierung: Freiformmodellierung

4.3 Beschreibung Funktionsumfang

Der Funktionsumfang der Software wurde an Hand der in den Abschnitten 2 und 3 erläuterten Funktionalitäten und Oberflächen-Prinzipien beschrieben. Die Schwerpunktsetzung der Software liegt oft in einem von beiden Bereichen: Entweder werden bekannte Funktionen in ungewöhnlicher Form für stiftbasiertes Arbeiten umgesetzt, oder neue Funktionalitäten für die Erstellung, Verwaltung und Interpretation von Skizzen werden mit klassischen Oberflächen-Prinzipien erprobt. Besonders die akademischen Prototypen fokussieren meist auf eine enge, sehr spezielle Fragestellung und implementieren alle als Grundlage dafür notwendigerweise benötigten Funktionen nur beiläufig nach bewährten, aber überkommenen Prinzipien.

Zeichnen (Skizzieren)	X	3D Journal	Design Review	Blobmaker	Chateau	Denim	Digital Clay	Cocktail Napkin	Esquise	Freak Draft View	Freak Sketch Client	iGesture	InkSeine	K-Sketch	Lea	LiveNotes	MathPad	NoteLab	NotePad
Bearbeiten der Zeichnungselemente	-	-	-	X	X	X	-	X	-	X	X	-	X	X	X	-	X	X	-
Auswahl der Zeichnungselemente	-	-	-	X	-	X	-	X	-	X	X	-	X	X	X	-	X	X	-
Verwalten mehrerer Skizzen	-	-	X	-	-	X	-	X	X	-	-	-	X	X	-	X	X	X	-
Strukturierung innerhalb Skizze	-	-	-	-	-	X	-	X	X	-	-	-	-	-	X	-	-	-	-
Überlagerung (Ebenen)	-	-	-	-	-	X	-	X	X	-	-	-	-	-	-	-	-	-	-
Zeichnungsteile verknüpfen (Marker)	-	-	X	-	-	-	-	X	-	-	-	-	-	-	-	-	-	-	-
Einbinden bestehender Daten (Ref.)	-	-	X	-	-	-	-	-	-	X	-	-	-	-	-	-	-	-	-
Zeichnungsassistent	-	-	-	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-
„Beautification“	-	-	-	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-
geom. Erkennung	-	-	-	-	X	-	-	X	X	-	-	-	-	-	X	-	X	-	-
Symbolerkennung	-	-	-	-	-	-	-	X	X	-	-	-	-	-	X	-	X	-	-
Handschrifterkennung	-	-	-	-	-	-	-	-	X	-	-	-	X	-	-	-	X	-	-
Geom. Rekonstruktion	X	-	-	X	-	-	X	-	-	-	-	-	-	-	-	-	-	-	-
Modellieren	X	-	-	X	-	-	-	-	X	-	X	-	-	-	X	-	-	-	-
Import	-	-	-	-	X	-	-	-	-	-	-	X	-	-	-	-	-	-	-
Export	-	-	-	-	X	-	X	-	X	-	-	X	-	-	-	-	-	-	-
Gestenerkennung	-	-	-	-	X	-	-	-	-	-	-	X	X	-	-	-	-	-	-
Kreismenü	-	-	-	-	X	-	-	-	-	-	-	-	X	-	-	-	-	-	-
Vorschlags-System	-	-	-	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Feedback	X	-	-	X	X	X	X	X	X	-	X	-	-	-	-	-	-	-	-
Navigation in Skizze / Modell	-	-	X	-	X	X	X	-	X	X	X	-	-	-	-	X	X	X	-
Navigation mehrere Skizzen	-	-	X	-	-	X	-	X	-	-	-	-	X	-	-	X	X	X	-
Umgang (v.a. Darst.) mit Ungenauigkeit	X	-	-	X	X	X	X	X	X	-	-	-	-	-	X	-	-	-	-
Modi	-	-	X	-	X	X	-	-	-	X	-	-	-	-	-	-	-	-	-
Teilfenster / Views	-	-	X	-	X	X	-	X	-	X	-	-	X	-	X	-	X	X	-
Maximierung Eingabebereich	-	-	-	-	X	-	X	-	X	-	X	-	-	X	-	X	-	-	-

Datei: 08 tech-rep tauscher skizzen.pdf

5 Fazit: GUI-Entwicklung im Rahmen des Forschungsprojekts

Dieses abschließende Kapitel fasst im Hinblick auf die Entwicklung einer eigenen prototypischen skizzenbasierten Schnittstelle für das Forschungsprojekt „Digitaler Gebäudebestand als Grundlage für Projektentscheidungen im Altbau“ idealerweise zu implementierende Funktionalitäten und Oberflächenelemente zusammen. Außerdem werden GUI-Prototyping-Methoden und GUI-Notationen vorgestellt, mit denen das bisher in annotierten Diagrammen vorliegende Oberflächen-Konzept im Hinblick auf die Implementierung weiterentwickelt werden kann.

5.1 Funktionalität des idealen prototypischen Systems

Grundsätzlich müssen Computersysteme die Funktionalitäten der traditionellen Medien adaptieren können und darüberhinaus zusätzliche Vorteile aufweisen, damit ihr Einsatz gerechtfertigt ist. Die Softwaretools müssen sich daher an den Funktionalitäten der traditionellen Medien messen lassen. Konkret bedeutet das für die aus Kapitel 2 ableitbaren Schwerpunkte, dass

1. für die Erstellung und Bearbeitung von Grafiken die Funktionalitäten des traditionellen Stifts als Zeichenwerkzeug
2. für die Verwaltungsmöglichkeiten inkl. Datenaustausch das Mediums Papier (einschl. Büchern und Ordnern)
3. für die Interpretation und semantische Anreicherung inkl. dreidimensionalem Modellieren die Fähigkeit der menschlichen Intelligenz, Dinge mit Bedeutung zu verstehen

als Maßstab dienen.

1. Obwohl die Qualitäten des traditionellen Stifts (Haptik, Ergonomie, Ästhetik) sehr schwer zu adaptieren sind, ist die Übernahme essentieller Funktionalitäten sinnvoll, um einen durchgängigen Workflow ohne Wechsel der Werkzeuge und Medien zu ermöglichen. Bei den Funktionalitäten zum Ändern und Löschen von Zeichnungsteilen ist das digitale Medium der Stift-Papier-Lösung überlegen.
2. Das größte Potential der digitalen Technologie zur Unterstützung freihändigen Schreibens und Zeichnens kann in den gegenüber der konventionellen Papiervariante erhöhten Verwaltungsmöglichkeiten gesehen werden. Wenn gut benutzbare

Funktionen zum Sortieren, Neuarrangieren, Ordnen, Benennen und Katalogisieren zur Verfügung stehen, wird der intelligente Nutzer in die Lage versetzt, seine Freihandnotizen und -zeichnungen nicht nur wieder aufzufinden, sondern darüberhinaus auch mit ersten Bedeutungen zu versehen.

3. Die automatisierte Interpretation als Ersatz für die explizite Angabe von Bedeutung durch den menschlichen Benutzer gestaltet sich für komplexe vielschichtige grafische Eingaben schwierig. Probleme der Benutzbarkeit ergeben sich besonders durch die Folgen von falscher oder unvollständiger Interpretation. Als Kompromiss zwischen der autonomen uninterpretierten Eingabe seitens des Benutzers und einer vollständigen Interpretation durch das System bietet sich eine beschränkte Interpretation an, bei der das System nach festgelegten Regeln interpretiert und der Nutzer in Kenntnis dieser Interpretations-Regeln seine Eingabe gestalten muss.

Für die skizzenhafte Grundriss-Erfassung im Forschungsprojekt wird deshalb eine Lösung vorgeschlagen mit einem Kompromiss im Bereich der Interpretation und semantischen Anreicherung der Skizze, einem Schwerpunkt auf der Anbindung der Skizze an ein Ordnungssystem und den notwendigen Voraussetzungen im Bereich der Ergonomie und intuitiven Bedienung (s. Kapitel 5.2).

5.2 Oberfläche und Interaktion im prototypischen System

Von den unter Kapitel 3.2 aufgeführten Hardwarevarianten ist der stiftbediente Eingabebildschirm state of the art und wird als optimal für das Arbeiten vor Ort gesehen. Bei der Oberflächenentwicklung müssen die daraus ableitbaren Restriktionen (z.B. durch Verdecken von Teilen des Bildschirms mit der Hand) beachtet werden.

Entsprechend Kapitel 3.3 muss für die Bedienung mit dem Stift insbesondere eine durchgängig einheitliche Lösung für das Wechseln der Eingabe-Modi gefunden werden. Die Maximierung des Eingabebereichs soll mit herkömmlichen Methoden (Ausblenden von Interaktionsbereichen) umgesetzt werden. Die zusätzliche Nutzung angepasster Interaktionselemente - Kreismenüs und Stiftgesten - kann je nach erforderlichem Aufwand optional implementiert werden.

Für die Eingabe skizzenförmiger grafischer Inhalte (Kapitel 3.4) müssen Navigationsbefehle einerseits als spezieller Modus in ein durchgängiges Konzept des Moduswechsels eingebunden werden und andererseits mit den verwandten Bearbeitungsoperationen abgestimmt sein, um die intuitive Bedienbarkeit sicherzustellen. Feedback über die erfolgte Interpretation soll durch an- und ausschaltbare Einblendung der interpretierten Geometrie erfolgen. Bei für die Modellbildung kritischen Eingaben muss das Feedback zwingend erfolgen ohne Option des Ausschaltens. Ungenauigkeit und Ambivalenz werden durch Beibehalten der Repräsentation der Freihandeingabe dargestellt. Hinsichtlich der Interpretation uneindeutige Eingaben werden nicht durch komplexe Interpretationsalgorithmen abgefangen sondern durch die Möglichkeit, die Eingabe zu korrigieren.

5.3 GUI prototyping Methoden und GUI Notationen

GUI prototyping hat einerseits zum Ziel, die Anordnung der Interaktionselemente auf dem Bildschirm visuell zu überprüfen und andererseits ein interaktives Mockup für ein vorgezogenes Usability-Testing der Oberfläche herzustellen, damit noch während der Implementation Korrekturen vorgenommen werden können.

GUI- und Webseiten-Mockups werden in der Praxis oft noch mit Pappscheiben umgesetzt, um ein einfaches und flexibles Usability-Testing zu ermöglichen. Die digitalen Alternativen reichen von Grafikdesign-Programmen wie Illustrator, über Skizzierprogramme bis zu spezialisierter GUI-Design-Software [5].

Die verwendete Software müsste idealerweise Möglichkeiten bieten, Interaktionselemente grafisch anzuordnen, Nutzerinteraktionen einschließlich der Software-Reaktion zu spezifizieren und ggf. zu simulieren sowie den Oberflächenentwurf in einer Form zu exportieren, die für die Implementation weiterverwendet werden kann.

Weiterverwendbare Notationen der GUI-Spezifikation müssen sich an den verwendeten GUI-Bibliotheken orientieren. Diese sind zwar teilweise plattformübergreifend, aber mit wenigen Ausnahmen durchgehend Programmiersprachen-spezifisch, so dass die Entscheidung für eine Bibliothek und damit eine Notation der GUI nicht unabhängig von der Implementationssprache erfolgen kann. Dann können auch GUI-Toolkits mit WYSIWYG-Editoren verwendet werden.

Bei dieser Herangehensweise kann zwar der entwickelte GUI-Prototyp für die Implementation weiterverwendet werden, jedoch können interaktive Mockups nur in der später verwendeten Programmiersprache erstellt werden. XML-basierte GUI-Notationen wie die XML User Interface Language (XUL) des Mozilla Projekts oder die in Views enthaltene XML-Notation [3] könnten hierfür als Austauschformat dienen und die Nutzung beliebiger Skriptsprachen für das Interaktions-Mockup erlauben.

6 Quellen

6.1 Literatur

- [1] ARAUJO, B. R. D., J. A. JORGE, M. C. SOUSA, F. SAMAVATI und B. WYVILL: *MIBlob. A Tool for Medical Visualization and Modelling using Sketches*. In: *31st International Conference on Computer Graphics and Interactive Techniques*. Los Angeles, 2004.
- [2] BEETZ, J.: *Ein parametrischer Ansatz für die Echtzeit-Interpretation von Handskizzen in den frühen Phasen des Architekturentwurfs*. Bauhaus-Universität, Weimar, Diplomarbeit, Lehrstuhl für Informatik in der Architektur Aufl., 2003.
- [3] BISHOP, J.: *Developing Principles of GUI Programming Using Views*. In: *SIGCSE 04*, Norfolk, Virginia, USA, 2004.
- [4] BLAZIC, A.: *Konzeption und Entwicklung eines intuitiven skizzenbasierten Grafikwerkzeugs für Architekten*. TU Ilmenau, Ilmenau, Diplomarbeit, Fakultät für Elektrotechnik und Informationstechnik Aufl., 2005.
- [5] C2: *Gui Prototyping Tools*. URL <http://c2.com/cgi/wiki?GuiPrototypingTools>, 2008. zuletzt abgerufen 15.6.2008.
- [6] COMPANY, P., A. PIQUER, M. CONTERO und F. NAYA: *A survey on geometrical reconstruction as a core technology to sketch-based modeling*. *Computers & Graphics*, 29(6):892–904, 2005.
- [7] DAVIS, R. C., T. S. SAPONAS, M. SHILMAN und J. A. LANDAY: *SketchWizard. Wizard of Oz Prototyping of Pen-Based User Interfaces*. In: *UIST 07*, Newport, Rhode Island, USA, 2007.
- [8] DIENEL: *Persönliches Wissensmanagement von Ingenieuren/innen. Aufstieg und Niedergang des Notizbuchs*. URL <http://www.thinksupport.de/pdf/notizbuecher.pdf>. zuletzt abgerufen 15.6.2008.
- [9] FITTS, P. M.: *The information capacity of the human motor system in controlling the amplitude of movement*. *Journal of Experimental Psychology*, 47(6):381–391, 1954.

- [10] GROSS, M. D. und E. L. DO: *Ambiguous Intentions. A Paper-like Interface for Creative Design*. In: *Proceedings of ACM Conference on User Interface Software Technology (UIST) '96*, Seattle, 1996.
- [11] HAMMOND, T. und R. DAVIS: *LADDER, a sketching language for user interface developers*. *Computers & Graphics*, 29(4):518–532, 2005.
- [12] HINCKLEY, K. und S. Z. U.A.: *InkSeine. In Situ Search for Active Note Taking*. In: H.G. (Hrsg.): *CHI 2007*, San Jose, California, 2007. ACM.
- [13] IGARASHI, T. und J. F. HUGHES: *A Suggestive Interface for 3D Drawing*. In: SIGGRAPH, A. (Hrsg.): *UIST 2001*, S. 173–181. ACM Press, New York, 14th Annual Symposium on User Interface Software and Technology, 11.-14.11.01, Orlando, Florida, US Aufl., 2001.
- [14] IGARASHI, T., S. MATSUOKA, S. KAWACHIYA und H. TANAKA: *Pegasus. A Drawing System for Rapid Geometric Design*. In: H.G. (Hrsg.): *Making the impossible possible*, S. 24–25, New York, 1998. ACM Press.
- [15] JUCHMES, R., P. LECLERCQ und S. AZAR: *A freehand-sketch environment for architectural design supported by a multi-agent system*. *Computers & Graphics*, 29(6):905–915, 2005.
- [16] KARA, L. B.: *An evaluation of user experience with a sketch-based 3D modeling system*. *Computers & Graphics*, 31(4):580–597, 2007.
- [17] KARPENKO, O. und J. F. HUGHES: *SmoothSketch. 3D free-form shapes from complex sketches*. *ACM Transactions on Graphics*, 25(3):589–598, 2006.
- [18] LANK, E. und E. SAUND: *Sloppy selection. Providing an accurate interpretation of imprecise selection gestures*. *Computers & Graphics*, 29(4):490–500, 2007.
- [19] LAVIOLA, J. J.: *An initial evaluation of MathPad. A tool for creating dynamic mathematical illustrations*. *Computers & Graphics*, 31(4):540–553, 2007.
- [20] MASRY, M. und H. L. D. KANG: *A freehand sketching interface for progressive construction of 3D objects*. *Computers & Graphics*, 29(4):563–575, 2005.
- [21] MATTING: *Vom Skizzenblock in den Rechner. General Keys Digitaler Notizblock*. URL <<http://blog.focus.de/techtoys/archives/855>>, 2007. zuletzt abgerufen 13.6.2008.
- [22] PAPERIUM: *Paperium*. URL <<http://www.paperium.com>>, 2007. zuletzt abgerufen 13.6.2008.
- [23] PARAGRAPH (Hrsg.): *PenOffice™ 2.6 Owners Manual*. o. O., 2005.
- [24] PLC, D.: *Destiny IO digital pen*. URL <<http://www.destinyplcshop.com/acatalog/notetaking-solutions.html>>, 2008. zuletzt abgerufen 07/2008.

- [25] SCHWEIKARDT, E. und M. D. GROSS: *Digital Clay. Deriving Digital Models from Freehand Sketches*. In: *Digital Design Studios. Do Computers Make A Difference? (ACADIA 1998 conference)*, Quebec, Canada, 1998.
- [26] SHNEIDERMAN, B.: *User interface design. Effektive Interaktion zwischen Mensch und Maschine*. mitp-Verlag, Bonn, 3. Aufl., dt. Ausgabe Aufl., 2003.

6.2 Software

- [27] *Open Ink*. URL <<http://openink.sourceforge.net>>. zuletzt abgerufen 05/2008.
- [28] *SketchUp*. URL <<http://sketchup.google.com/>>, 2007. zuletzt abgerufen 05/2008.
- [29] AUTODESK: *Design Review*. URL <<http://www.autodesk.de/adsk/servlet/index?siteID=403786&id=10580577>>, 2008. zuletzt abgerufen 07/2008.
- [30] AUTODESK: *Sketchbook Pro*, 2008. zuletzt abgerufen 03/2008.
- [31] BEAT und KURMANN: *iGesture*. URL <<http://www.igesture.org/>>, 2007. zuletzt abgerufen 05/2008.
- [32] BEETZ, J.: *Lea*. URL <<http://www.ds.arch.tue.nl/General/Staff/jakob/diploma/>>, 2003. zuletzt abgerufen 03/2008.
- [33] CARDOSO und SILVA: *SmartSketches (Gides++)*. URL <<http://immi.inesc.pt/tfdc/gides++/>>, 2004. zuletzt abgerufen 05/2008.
- [34] CHIU und BROWN: *Slidewriter*. URL <<http://sourceforge.net/projects/slidewriter>>, 2006. zuletzt abgerufen 05/2008.
- [35] DAVIS: *K-Sketch*. URL <<http://www.k-sketch.org>>, 2008. zuletzt abgerufen 05/2008.
- [36] DAVIS und L. SAPONAS: *Sketch Wizard*. URL <<http://dub.washington.edu:2007/sketchwizard/>>, 2007. zuletzt abgerufen 05/2008.
- [37] GILCHRIST: *Notepad*. URL <<http://sourceforge.net/projects/notepad>>, 2006. zuletzt abgerufen 05/2008.
- [38] GROSJEAN und CLAMAGE: *Piccolo.NET User Interface Library*. URL <<http://www.cs.umd.edu/hcil/jazz/download/index.shtml>>, 2005. zuletzt abgerufen 05/2008.

- [39] GROSS, M. D. und E. L. DO: *Electronic Cocktail Napkin*. URL <<http://depts.washington.edu/napkin/>>, 1996. zuletzt abgerufen.
- [40] HINKLEY: *InkSeine*. URL <<http://research.microsoft.com/inkseine/>>, 2008. zuletzt abgerufen 05/2008.
- [41] IGARASHI: *Teddy*. URL <<http://www-ui.is.s.u-tokyo.ac.jp/~takeo/java/smoothteddy/index.html>>, 2003. zuletzt abgerufen 05/2008.
- [42] IGARASHI, T.: *Pegasus*. URL <<http://www-ui.is.s.u-tokyo.ac.jp/~takeo/java/pegasus/pegasus-e.html>>, 1998. zuletzt abgerufen 05/2008.
- [43] IGARASHI, T.: *Chateau*. URL <<http://www-ui.is.s.u-tokyo.ac.jp/~takeo/chateau/chateau.htm>>, 2001. zuletzt abgerufen 05/2008.
- [44] INFAR: *SAM*. URL <http://infar.architektur.uni-weimar.de>, 1998.
- [45] JATUPOJ: *Sketchboard*. URL <<http://sketchboard.sourceforge.net>>, 2004. zuletzt abgerufen 05/2008.
- [46] JORGE, J. A. und B. R. DE ARAUJO: *Blobmaker*. URL <<http://vimmi.inesc-id.pt/~brar/blobmaker/index.html>>, 2003. zuletzt abgerufen Caligraphic interface for free-form modelling.
- [47] KARPENKO, O.: *SmoothSketch*. URL <<http://www.cs.brown.edu/people/jfh/papers/Karpenko-S3F-2006/main.htm>>, 2006. zuletzt abgerufen 3D free-form shapes from complex sketches.
- [48] KRAMER: *NoteLab*. URL <<http://java-notelab.sourceforge.net/>>, 2007. zuletzt abgerufen 05/2008.
- [49] LAVIOLA: *MathPad*. URL <<http://www.cs.brown.edu/~jjl/mathpad/>>, 2007. zuletzt abgerufen 05/2008.
- [50] LIN: *Denim*. URL <<http://dub.washington.edu/denim>>, 2007. zuletzt abgerufen 05/2008.
- [51] LUCID GROUP: *Esquise*. URL <http://www.arch.ulg.ac.be/Lucid/Projects_EsQUIsE_SMA.php>, 2007. zuletzt abgerufen 05/2008.
- [52] MASRY, M.: *3D Journal*. URL <<http://ccsl.mae.cornell.edu/research/sketch/index.html>>, 2005. zuletzt abgerufen 05/2008.

- [53] MEDIA, O.: *Ript*. URL <<http://ript.com>>, 2008. zuletzt abgerufen 05/2008.
- [54] MICROSOFT: *Tablet PC SDK*. URL <<http://www.microsoft.com>>, 2005. zuletzt abgerufen 05/2008.
- [55] NAIDON und CORRIERI: *Pencil*. URL <<http://www.les-stooges.org/pascal/pencil/index.php>>, 2008. zuletzt abgerufen 05/2008.
- [56] PARAGRAPH: *PenCommander*. URL <<http://www.phatware.com>>, 2005. zuletzt abgerufen 05/2008.
- [57] PARAGRAPH: *PenOffice*. URL <<http://www.phatware.com>>, 2005. zuletzt abgerufen 05/2008.
- [58] SCHNEIDER, S.: *Freak Sketch Client*, 2007.
- [59] SCHWEIKARDT, E. und M. D. GROSS: *Digital Clay*. URL <<http://code.arc.cmu.edu/lab/html/project10.html>>, 1998. zuletzt abgerufen 07/2008.
- [60] SWIGART: *SiGeR*. URL <<http://msdn.microsoft.com/en-us/library/aa480673.aspx> <http://sourceforge.net/projects/siger/>>, 2005. zuletzt abgerufen 05/2008.
- [61] THUROW, T.: *Freak Draft View*, 2007.
- [62] WANG: *Livenotes*. URL <<http://sourceforge.net/projects/livenotes>>, 2005. zuletzt abgerufen 05/2008.