

EIN GENERISCHES WERKZEUG ZUR ANBINDUNG VON WISSENRESSOURCEN AN INTERNETBASIERTE PLANUNGSUMGEBUNGEN

Erik-Oliver Blaß / Uwe Forgber
Institut für Industrielle Bauproduktion (ifib), Universität Karlsruhe (TH)
erik.blass@ifib.uni-karlsruhe.de
uwe.forgber@ifib.uni-karlsruhe.de
www.ifib.uni-karlsruhe.de

Kurzfassung: *Mit der technologischen- und methodischen Entwicklung internetbasierter Formen der Zusammenarbeit wächst der Bedarf, im laufenden Arbeitsprozeß problemspezifische Informationen und Wissen auszutauschen. Es gilt dabei nicht mehr nur den Austausch von Arbeitsergebnissen zu unterstützen, sondern eine kontinuierliche und reibungslose Kooperation räumlich verteilter Partner zu ermöglichen. Mit dieser Problemstellung beschäftigt sich insbesondere der Forschungsbereich Computer Supported Cooperative Work (CSCW). Die in diesem Zusammenhang stehende Entwicklung internetbasierter Planungsumgebungen ermöglicht darüber hinaus die Integration allgemeinverfügbarer Wissensressourcen in den problemspezifischen Entscheidungsprozeß.*

Im folgende Artikel wird eine generische Struktur internetbasierter Werkzeuge vorgestellt, welche die Integration von Wissensressourcen in internetbasierten Planungsumgebungen unterstützt. Dabei wird davon ausgegangen, daß Fachliteratur in herkömmlicher Printform zukünftig durch digital aufbereitete, didaktische Informationseinheiten abgelöst wird.

1 Einleitung

Das Planen, Bauen und Betreiben von Gebäuden wird derzeit durch eine Vielzahl bauspezifischer Softwarepakete (CAD, AVA (Ausschreibung), FM (Facility Management) etc.), sowie den üblichen Kommunikationstechnologien (Telefon, Fax, e-Mail, Videokonferenzen) unterstützt. Der im Rahmen der Projektbearbeitung und des Baumanagements erforderliche Austausch an Dokumenten und aufgabenspezifischen Informationen erfolgt im Rahmen von Projekttreffen, via Briefpost, Kurier, Fax, e-Mail

oder sogenannten *Electronic Document Management* (EDM) Systemen. Fachspezifische Informationen sowie explizites, domänenspezifisches Wissen werden im Rahmen persönlicher Gespräche oder einschlägiger Literatur vermittelt.

Die zielgerichtete Kombination dieser verschiedenen, die Projektbearbeitung unterstützenden Möglichkeiten, obliegt dabei den individuellen Planern bzw. bei größeren Planungsvorhaben der Instanz des Projektmanagements. Aus der Abwicklung von Großprojekten ist allgemein bekannt, daß ein rascher Austausch an Dokumenten, Informationen und fachspezifischem Wissen sowie die möglichst enge Kooperation der beteiligten Planer Grundgrößen bei der wirtschaftlichen Abwicklung eines Projektes sind.

Die internetbasierte Integration verschiedener, die Projektbearbeitung unterstützenden Möglichkeiten (inhaltliche Bearbeitung, Kommunikation) wird kommerziell unter der Bezeichnung EDM Systeme angeboten. Darüber hinaus ist die Unterstützung der Projektorganisation sowie der Kooperation aller Projektbeteiligten Gegenstand verschiedener Forschungsvorhaben [4] [6] [2]. Das *Institut für Industrielle Bauproduktion (ifib)* entwickelte in diesem Zusammenhang eine prototypische internetbasierte Planungsumgebung [4]. Dabei zeichnet sich die betriebssystemunabhängige Planungsumgebung durch folgende Elemente aus:

- Integration aller für die Projektbearbeitung erforderlichen Ressourcen
- Projektorganisation unter dem Gesichtspunkt der Kooperationsunterstützung
- Untergliederung des Projektes in unterschiedliche Sichten (*Kontextbereiche*) bei gleichzeitiger Transparenz der Gesamtzusammenhänge (*Navigator*).

Abbildung 1 veranschaulicht mit dem Navigator die Gesamtschau auf das Projekt. Die verschiedenen *Kontextbereiche* sind durch projektspezifisch definierte *Wechselwirkungen* miteinander verbunden. Durch Mausklick ist hinter jedem *Kontextbereich* eine sogenannte kontextspezifische Arbeitsumgebung (*KAU*) zu erreichen.

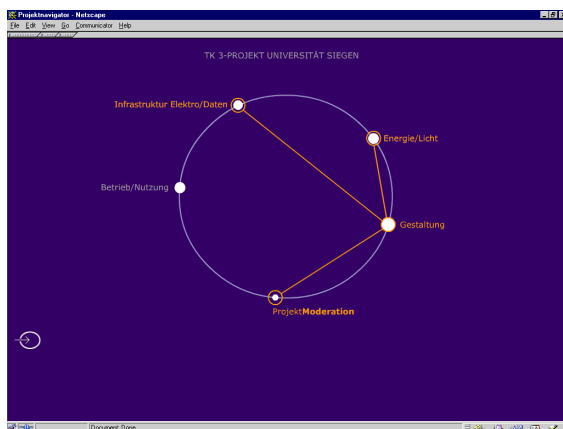


Abbildung 1: *Projektnavigator*

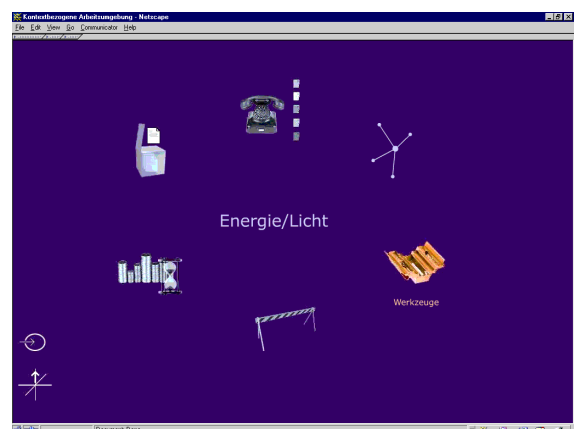


Abbildung 2: *KAU Energie/Licht*

Die im *KAU* vorhandenen Funktionselemente *TeamKommunikation*, *Ziele/Anforderungen*, *InfoContainer*, *Wechselwirkung* und *Werkzeuge* stellen die zur Projektbearbeitung erforderlichen Ressourcen und Kommunikationsinstrumente zur Verfügung (Abbildung 2).

Die Anbindung externer, internetbasierter Werkzeuge (z.B. URLs oder die nachfolgend beschriebene Werkzeugstruktur) erfolgt über das Funktionselement *Werkzeuge*. Durch Mausklick wird dabei ein zusätzliches Browser-Fenster aufgerufen. Die Anforderungen an die Anbindung einer generischen Werkzeugstruktur an die Datenbasis der Planungsumgebung werden unter 1.1 beschrieben.

1.1 Anforderungen Softwareimplementierung

Entwickelt werden soll ein allgemeines bzw. flexibles Softwarewerkzeug, das im Kontext der internetbasierten Planungsumgebung *INTESOL* angewendet werden kann. Allgemein und flexibel bedeutet in diesem Zusammenhang, daß die Software in verschiedenen Anwendungssituationen (Spezialisierung auf z.B. Energieplanung, Projektorganisation etc.) ohne Änderungen eingesetzt werden kann, bzw. mit geringen Codeänderungen an neue Situationen anpassbar ist.

Das Werkzeug soll innerhalb eines Browser-Fensters arbeiten und eine graphische Benutzeroberfläche bieten [3]. Die Software sollte möglichst plattformunabhängig sein und damit auf verschiedenen Computersystemen laufen.

Grundsätzlich soll das Werkzeug zwei Arbeitsweisen anbieten:

- 1.) Aufruf aus der Planungsumgebung heraus, d.h. die Datenbasis auf der das Werkzeug operiert wird aus der Planungsumgebung heraus geladen; oder
- 2.) Standalone, der Benutzer wählt mit Hilfe der graphischen Benutzeroberfläche die Adresse der Datenbasis selbst aus. Die zur Bearbeitung des spezifischen Projektes erforderlichen Werte werden ebenso über ein graphisches Interface ausgewählt.

Die primäre Aufgabe dieses Werkzeuges ist es daher, mit Hilfe eines Expertensystems/wissenbasierten Systems interaktiv (der Benutzer ist in den Entscheidungsprozeß eingebunden) eine Menge von Planungsfakten (Werten) nach bestimmten Regeln auszuwerten.

Zusätzlich soll der Benutzer die Möglichkeit besitzen, diese Werte aus der Planungsplattform auszulesen, zu verändern und zurückzuschreiben. Eventuelle Verständnisprobleme bzw. Wissenslücken des Benutzers sollen durch bei Bedarf aufrufbare, erklärende CBT-Prozesse [5] aufgefangen werden.

Das Werkzeug soll unabhängig von der Realisierung der Datenbasis selbst sein, d.h.:

- unabhängig von Datenbanksystemen wie z.B. Notes, Interbase, Oracle oder Postgres, ...

Zudem muß die komplette Software mit zukünftigen Datenstandards wie z.B. den *Industry foundation Clases (IFC)* - oder unvorhergesehenen - zusammenarbeiten können. Das Werkzeug soll dabei soweit dynamisch sein, daß es auch unabhängig von Größen wie der tatsächlichen Regelmenge, den tatsächlichen Werten, der eingesetzten CBT-Prozesse usw. arbeitet, und der Benutzer so ein *generisches* Werkzeug erhält, das er auf seine Bedürfnisse hin mit obigen Inhalten personalisieren kann, ohne - oder nur mit wenig – Änderungen am vorhandenen Sourcecode. Entwickelt werden soll zunächst ein Prototyp.

2 Entwurf

Ziel bei der Entwicklung des Prototypen ist ein *Labormuster* [1], d.h. ein Prototyp, der die technische Machbarkeit eines solchen internetbasierten Werkzeuges demonstrieren soll.

Wie angedeutet muß das Werkzeug aus der vorhandenen internetbasierten Planungsumgebung heraus aufgerufen werden, d.h. also innerhalb eines Browser-Fensters laufen.

Die hinzukommende geforderte Plattformunabhängigkeit bedeutet eine Realisierung der Benutzerschnittstelle des Werkzeugs als Java-Applet.

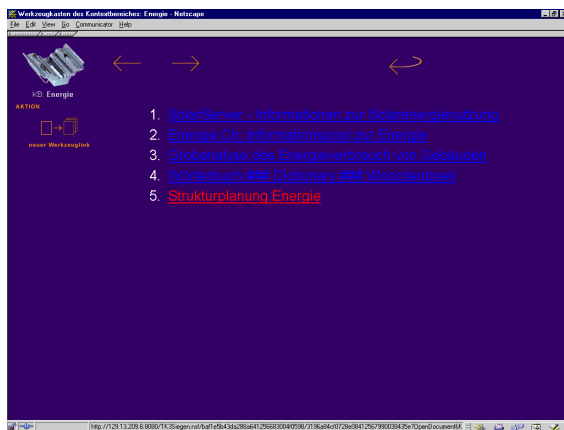


Abbildung 3: Werkzeugkasten Energie

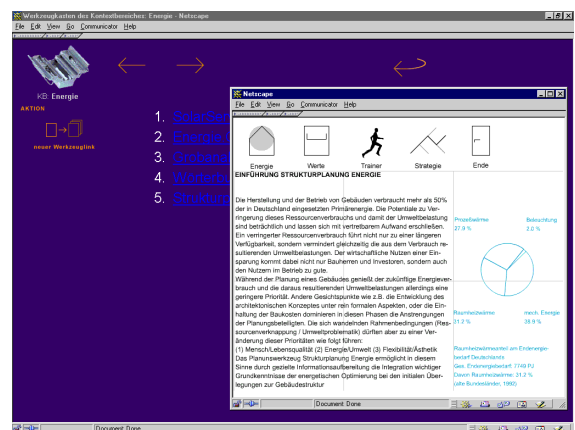


Abbildung 4: Ansicht Werkzeug-Applet

Die erklärenden CBT-Prozesse werden mit *Macromedia Flash* produziert, einer Multimediaentwicklungsumgebung für das Internet. Diese CBT-Prozesse können so mit den gängigen Browsern und dem darin integrierten Flash-Plugin abgespielt werden. Die CBT-Prozesse werden im WWW mit eigenen Adressen (URLs) bereitgestellt, so daß sie auch unabhängig vom Aufruf des Werkzeugs direkt zur Verfügung stehen.

Als Entwicklungsumgebung des Expertensystems das innerhalb des Werkzeugs operiert, wird *JESS*, die Java-Version des *C-Language integrated Production Systems (CLIPS)* gewählt, da bereits Erfahrungen in CLIPS vorliegen, und JESS frei verfügbar ist.

Das Regelwerk für das Expertensystem wird während der Ausführung des Werkzeugs von einer URL nachgeladen, so daß das Regelwerk auch unabhängig vom Werkzeug existieren und geändert werden kann.

Damit das Java-Applet, um Werte zu verarbeiten, mit beliebigen Datenbankservern über das Internet kommunizieren kann, (Werte aus der Planungsumgebung lesen, CBT-URLs aufrufen usw.) wird eine neue Kommunikationsmethode entwickelt.

Der vorgestellte Ansatz basiert auf dem Protokoll *Remote Method Invocation (RMI)*, einer Möglichkeit zur Kommunikation zwischen Java-Programmen. Für den Entwurf zukünftiger Versionen des Werkzeuges sollte - anstatt der proprietären Java/RMI basierten Methode - Java-IIOP/CORBA benutzt werden, da dies für die Implementierung die Unabhängigkeit von den gewählten Programmiersprachen der

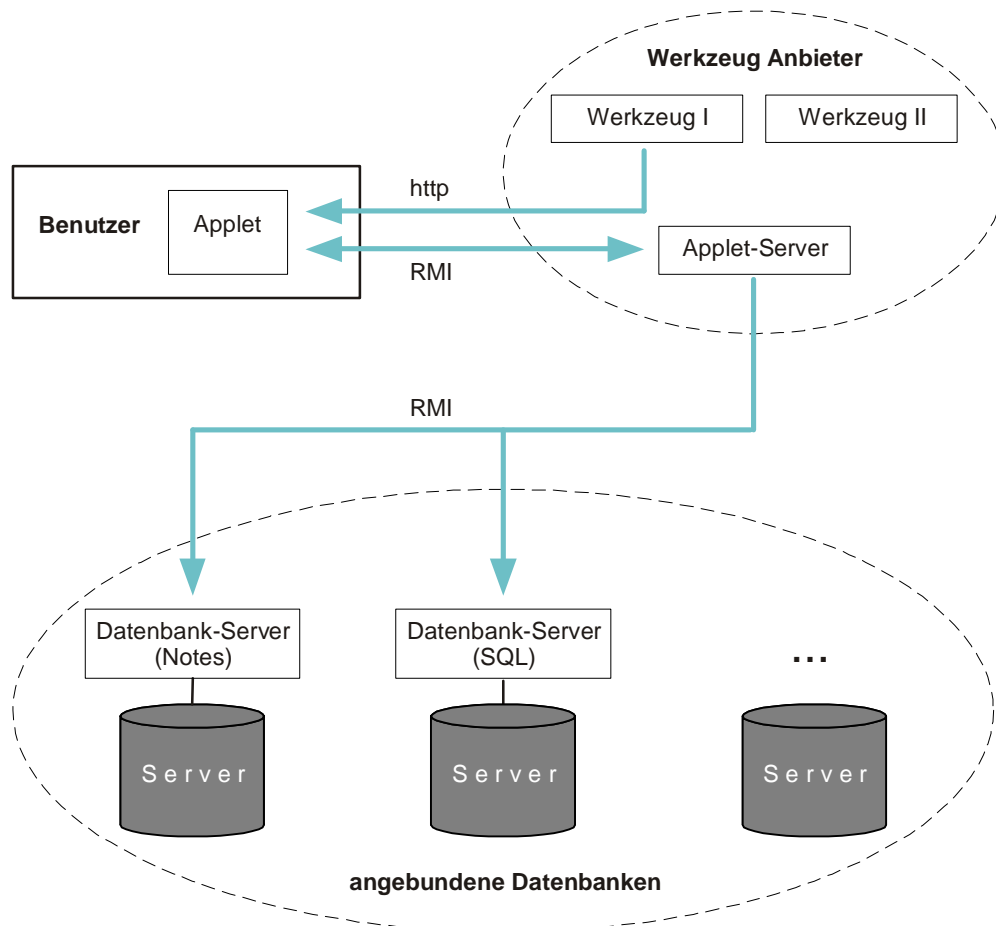


Abbildung 5: Skizze Werkzeugaufbau

jeweiligen Kommunikationspartner ermöglicht: Die Datenbankserver (s. Abbildung 5) können so z.B. auch in C/C++ geschrieben werden.

Zum Zeitpunkt der Entwicklung des Prototypen stand diese Alternative allerdings noch nicht ausreichend zur Verfügung.

2.1 Kommunikationsmethode

Der Benutzer kann zunächst die Werkzeug-Applets eines Anbieters wie gewohnt per http-Protokoll laden und starten.

Da aus Sicherheitsgründen Applets nur mit dem Rechner kommunizieren können, von dem sie geladen wurden, muß der Werkzeuganbieter zudem noch einen Stellvertreter/Proxy (s. Anhang, im CORBA Jargon: *Gatekeeper*) bereitstellen, der gewünschte Operationen auf Datenbasen unterstützt, diese intern aber an die vom Benutzer gewählten Datenbanken weiterleitet. Das Applet verbindet sich zum *Gatekeeper*, der dann selbst *stellvertretend* für das Applet die Datenanfragen weiterleitet, Ergebnisse von den Datenbankservern werden an das Applet weitergereicht.

- Dies geschieht in Java/RMI.

Entsprechenderweise müssen die angesprochenen Datenbanken einen *Datenbank-Server* implementieren, der die Anfragen von *Gatekeepern* in interne Datenbankweisungen übersetzt und die Ergebnisse wieder an die *Gatekeeper* zurückliefert.

Beispielhaft ist dies für Lotus Notes und die einfachsten Abfragen wie Werte lesen, speichern, URL lesen, ... bereits implementiert, da Lotus Notes eine recht einfache Schnittstelle zu Java anbietet.

Eine vorläufige Version für SQL-Datenbanken liegt in groben Zügen ebenso vor und zeigt, das die komplette Kommunikation unabhängig von der Realisierung der Datenbanken ist.

Die aktuelle Anbindung *Gatekeeper* <-> *Datenbank-Server* ist ebenso in Java/RMI realisiert, sollte jedoch wie oben bereits erwähnt in CORBA implementiert werden, zumal viele Datenbanken CORBA-fähig sind.

3 Zusammenfassung

Der vorliegende Artikel beschreibt die Implementierung einer generischen Werkzeugstruktur mit Anbindung an eine bauplanungsspezifische, internetbasierte Planungsumgebung. Dabei ist das Ziel die Anbindung externer Ressourcen explizit aufbereiteten Wissens. Das vorgestellte Werkzeug wurde prototypisch für die Wissensdomänen Strukturplanung Energie und Projektmoderation, einer

Grundvoraussetzung der Integralen Planungsweise, im Rahmen des Verbundforschungsvorhabens INTESOL [4] Implementiert. Die Entwicklung weiterer Versionen des Werkzeugs und seine Anwendung in verschiedenen Domänen ist geplant.

Literatur / WWW Referenzen

- [1] Balzert H.: *Lehrbuch der Software-Technik*. Heidelberg: Spektrum Verlag 1996.
- [2] COMMIT: *COnstruction Modelling and Methodologies for Intelligent information inTegration*. Projektdokumentation auf dem Internet unter: <http://www.salford.ac.uk/iti/projects/commit/commit.html>
- [3] Dinger F., Forgber U.: *Graphische Kommunikationsunterstützung internetbasierter Projekträume*. Technische Universität Darmstadt: XI. Forum Bauinformatik, 22. bis 24. September 1999.
- [4] Kohler N., Forgber U., Müller C.: *Zwischenbericht des Projektes RETEx II / INTESOL für das Jahr 1997*. Universität Karlsruhe (TH): Institut für Industrielle Bauproduktion (ifib) 1998.
- [5] Schanda F.: *Computer-Lernprogramme. Wie damit gelernt wird. Wie sie entwickelt werden*. Was sie im Unternehmen leisten. Weinheim: Beltz Verlag 1995.
- [6] ToCEE: *ToCEE- Towards a Concurrent Engineering Environment in the Building and Engineering Structures Industry*. Projektdokumentation auf dem Internet unter: <http://wwwcib.bau.tu-dresden.de/tocee/>

Anhang

Beispielinterface für den *Stellvertreter/Gatekeeper*

Der *Gatekeeper* bietet den Applets die folgenden Operationen an:

- Benötigte Werte lesen
- Bekannte Werte lesen
- Werte zurückschreiben
- URL für CBT Prozesse lesen

usw.

In Java:

```
public interface applet_interface extends java.rmi.Remote
```

```
{  
returnWert benWerteAppletLesen (String username, String passwort, String Rechner,  
String WerkzeugName) throws java.rmi.RemoteException;  
returnWert bekWerteAppletLesen (String username, String passwort, String Rechner,  
String ProjektName, Vector ben_werte) throws java.rmi.RemoteException;  
String applet_schreiben(String username, String passwort, String Rechner, String  
ProjektName, Hashtable Werte) throws java.rmi.RemoteException;  
returnWert URLAppletLesen (String username, String passwort, String Rechner, String  
ProjektName, Vector ben_werte) throws java.rmi.RemoteException;  
returnWert BelegungenAppletLesen (String username, String passwort, String Rechner,  
String ProjektName, Vector ben_werte) throws java.rmi.RemoteException;  
...  
}
```