

Managing the Work Flow of the Upgrade Procedure for Long Tunnels: An Artificial Intelligence Approach

M. Cristani

Dipartimento di Informatica, Università di Verona
Cà Vignal 2, strada Le Grazie 15, I-37134 Verona,
`cristani@sci.univr.it`

Abstract

In a recent paper on Civil Engineering and Artificial Intelligence the importance of the support to upgrade procedure of long tunnels by intelligent systems has been stressed. One major question to answer to, in this case, is how to manage the work flow of the upgrade procedure.

In particular, many data are processed by agents participating in the procedure, by means of specific computer programs, and in some cases even by hands, and the support to the information interchange is, in the project authors of this paper are working on, an ontological layer where the main concepts employed in describing tunnels, and format standards for data interchange, are provided, in such a way that the interchange process may proceed on the web.

In this paper we focus on the definition of requisites' and design specifications of a work flow control system for managing the upgrade procedure of long tunnels on the web.

1 Introduction

In this paper we shall provide an architecture proposal for managing the work flow of upgrade procedure as conceived in the context of the UPTUN project.

The work flow we shall manage is condensed in Figure 1.

The paper is organized as follows: Section 2 presents related work, Section 3 is devoted to the introduction of terminology and basic definitions, Section 4 defines the work flow based on the terminology of Section 3 and provides a sharable model in RDF. The architecture proposal, founded on the above defined schema is provided in Section 5. Section 6 takes conclusions and sketches further work.

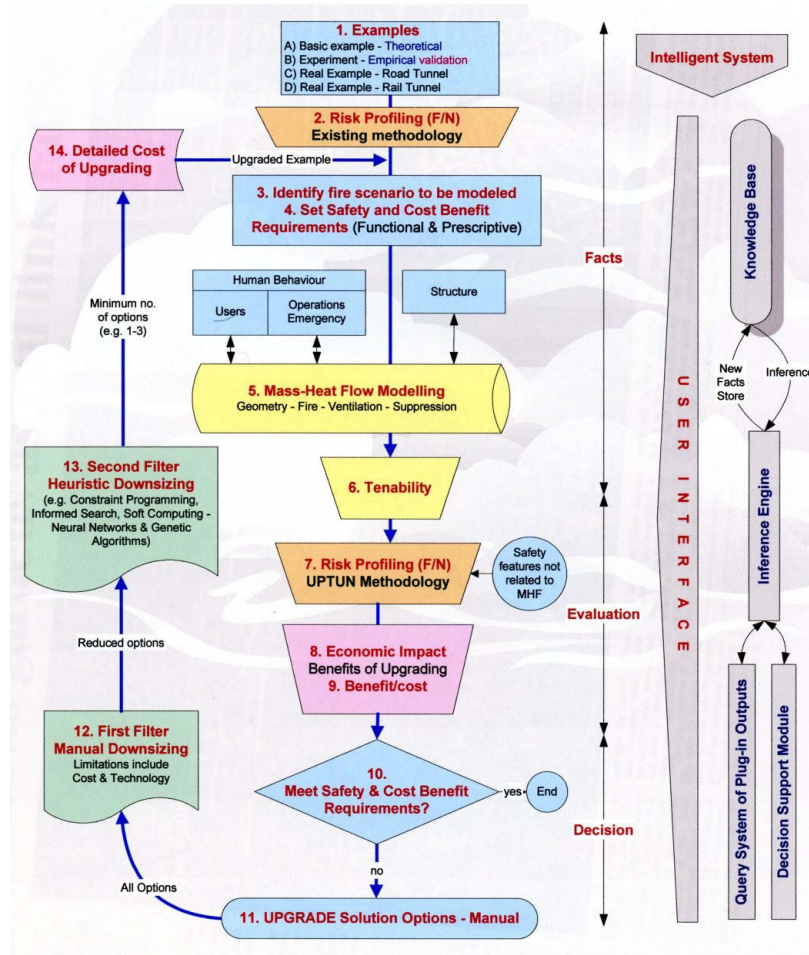


Figure 1: The work flow of the upgrade procedure of long tunnels.

2 Related work

Before to go into the analysis of up-to-date literature we want to address some general issues. First of all, let us motivate the analysis itself. As a starting point, we looked at the book [36]. In particular the following quotation is a rather useful viewpoint on the problems, methods and techniques that constitute an initial step in the development of an autonomous discipline of Artificial Intelligence in Engineering.

Expert system methodology has been applied in the past to a number of problems of planning, design, diagnostics etc. However, the problems of engineering design have not been adequately addressed, since these problems have to be addressed in an integrated manner with knowledge from different domains and sources.

Artificial Intelligence and Expert Systems for Engineers
C.S. Krishnamoorthy and S. Rajeev

2.1 Modelling design collaboration

Collaboration can be defined as [31]:

The agreement among specialists to share their abilities in a particular process, to achieve the larger objectives of the project as a whole, as defined by a client, a community, or society at large.

The activities of design which involve several individuals and specific viewpoints constitute one of the frontiers of the investigations about the application of artificial intelligence to engineering. This is particularly true for civil and structural engineering, where the collaborative and cooperative activities are becoming even more frequent than in the past, due to the need for team design, which is necessary because of the complexity of modern civil engineering projects.

In [8], Brazier et al. dealt with the problem of representing the knowledge level of individual designers in collaborative environment. The fundamental assumption of this paper is that the construction of a level where knowledge is represented also in epistemological terms, namely where agents have tools to represent the viewpoints of other agents, and where viewpoints have an explicit representation as well, let us cope with the complexity of relevant tasks of cooperation in engineering.

Note that the supported phase is again the design one, and that such an approach bring directly to the impossibility of representing collaboration as different from cooperation. The second evident difficult of the approach proposed in [8] is that it does not take in direct account several aspects which require a specification for engineering activities and for civil engineering activities in particular. The major focus is on the opportunity, usefulness and reliability of a collaboration model for engineering design.

Kalay, in [31], deals with one of the specific aspects of engineering needs: multidisciplinary. In civil and structural engineering several different abilities are involved: Geology, Structural Engineering, Fire Engineering, Economics, Politics, and many others.

Quoting directly:

The growing complexity of the built environment requires increased professional specialization to master each one of the many issues involved in designing buildings. This specialization is transforming the construction process into a highly de-centralized operation, involving an ever-growing number of collaborating professionals.

The base of the proposed approach is a system model named P3, formed by a set of tools. These tools comprise three separate, but strongly connected databases:

- An object database (ODB)
- A project database (PDB)
- A context database (CDB).

Each database provides one part of the information needed to completely comprehend the data, hence is not, in and of itself, sufficient to represent the entire data. Rather, it is their connectedness which makes the overall representation semantically rich.

A similar problem is posed by [23], whose proposal differs from [8] essentially for three assumptions:

- The notion of interdisciplinarity assumed in [23] is founded on the idea of a collaboration schema; there is no degree of freedom for the methodology, steps, and data exchange;
- The involved disciplines are homogeneous in terms of epistemological assumptions, and the methods employed do not differ much;
- The specific focus is in the opportunity for the members of the team to provide critiques to the project, an activity which requires homogeneity, at a given degree.

The construction industry has a long tradition of collaborative working between the members of a construction project team. At the design stage, this has traditionally been based on physical meetings between representatives of the principal design disciplines. [5] examines some of the issues associated with the use of distributed artificial intelligence systems within the construction industry. It describes the potential for the use of agent technology in collaborative design and then goes on to present the key features of an agent-based system for the collaborative design of portal frame structures.

	Same time	Different time
Same place	Face-to-face collaboration	Asynchronous Collaboration
Different place	Synchronous Distributed Collaboration	Asynchronous Distributed Collaboration

Table 1: Table of the four basic ways of collaboration [5].

In a very simple analytical synthesis [5] proposes a model for the four basic ways in which a collaboration can be carried out. Table 1 we quote Figure 2 at page 91 of [5].

The paper presents also a case study of the application of the collaboration concepts and in particular of the notion of intelligent agent to the problem of portal frame design. The analysis is deeply performed along with an interesting implementation.

The most important value of [5] is the proposal of an architecture for intelligent systems acting in a distributed environment which can support collaboration in civil and structural engineering. This problem is subdivided into three main streams:

- The individuation of a reliable model of agent;
- The choice of a reliable ontological model of the domain (in the case study, the frame portals);
- The assignment of a specific collaboration protocol.

The latter has the definite purpose of supporting semantic negotiation among the agent, a problem also posed by [31]. For the problems posed by communication in interactive engineering design, in particular, misunderstanding, see[53].

A deeper analysis of this aspect has been carried out in [4], with the declared purpose of finding a complete solution for the problems about negotiation posed in [5].

[11] focuses on the notion of Discourse Model, a model of concurrent engineering whose epistemological assumption is the need for a transparent, but knowledge-aware model of cooperation for human agents in a environment which allows asynchronous and distributed activities. The paper presents an application of the model to real-world cases, in an implementation of a distributed environment.

A very interesting case of a reliable application is the system known as SEED-Conf [22]. This modern and complex system performs interesting innovative tasks, by means of an architecture made of four basic components:

- the Design Information Repository;
- the Design Knowledge Manager;

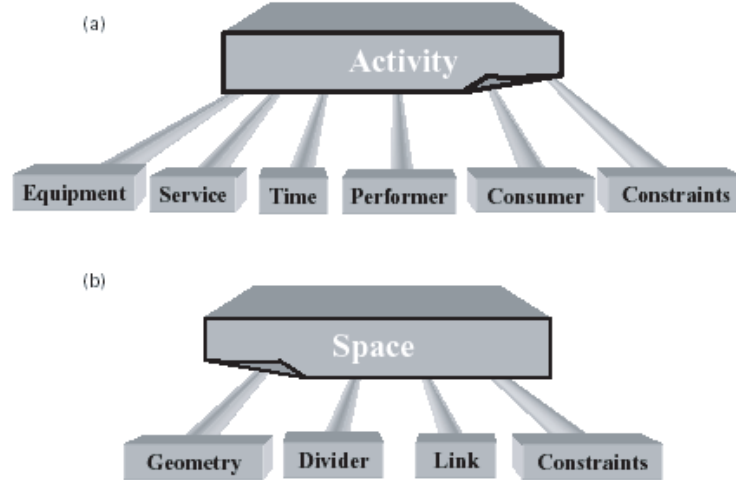


Figure 2: The refinements to Simoff's Ontology of Building Design [59].

- the Classification Reference Manager;
- the Geometric Modeler.

Another reference is the paper of Rajeev's et al. [70], where authors deal again with the problem of supporting critique in cooperative problem solving, as applied to engineering domains.

2.2 Ontology in civil engineering

In [59] Richards and Simoff employ a consolidated ontological model of Design Ontology, developed originally in [66]. Schematically the authors recur on a conceptualization of building design based on the notions of activity and space. These two concepts are refined then as in Figure 2.

Authors of this investigation propose the employment of two tools:

Formal Concept Analysis (FCA) a method for providing mathematically well-founded definitions of the conceptual analysis which is provided in any practical context for databases. The core part of the method consists in using the notion of context and provide a formalization of concepts as crosstables. A concept is a set of objects along with a set of attributes applied to those objects;

Ripple Down Rules (RDR) a system for the representation of relations among objects in contexts. RDR are applied to provide explanation to decisions made in expert way, automatically.

The most extended case of Ontology in applied engineering is the YMIR Ontology [1] proposed by Alberts. The YMIR ontology is vast and complex and would deserve a separate treatment. We can simply say, in this context few basic things on this methodology.

First of all, after having declared an intention of being general, as imaginable, the author concentrates mainly on the notions needed for mechanical engineering.

2.3 Representing spatial knowledge

The most important aspects of knowledge from the description of Civil and Structural Engineering activities are, definitely, the spatial and temporal ones.

This holds, in particular, for the investigations on Design activities, as profusely described in [20]. The nature of the investigation of Ekholm and Fridqvist is the need for a general model of knowledge on space used and usable by individuals living in an environment. This notion depends entirely on the function of the environment. The model they propose is based upon three basic notions:

- Space as a classification domain, where objects are considered;
- Spatial relations;
- Spatial domain for datasets.

The latter is intended to refer to the usefulness of constructing a schema for all the objects involved in spatial knowledge for civil engineering.

The form of the schema they provide appear in Figure 3.

Though it is clear that the field of application of their proposed method is not widespread being definitely constrained to the case of architecture design, we can employ the same ideas in other closely related domains, like structure projecting.

A more specific investigation is the one of Knight et al. [32], who provide a formalization of the model of space fire engineer implicitly use for knowledge management. This model of space is based on the ways we maintain information about a fire:

Fire as a cloud : we see fire as the hot moved on along with a smoke cloud;

Fire as a hot air wind : we see fire as the motion of air, heated by the fire itself.

The ways of looking at fire are not the only aspects of knowledge engineering dealt with insystem (KBS) component of Smartfire, a modelling tool for use by members of the re safety engineering community who are not expert in modelling techniques. The KBS captures the qualitative reasoning of an experienced

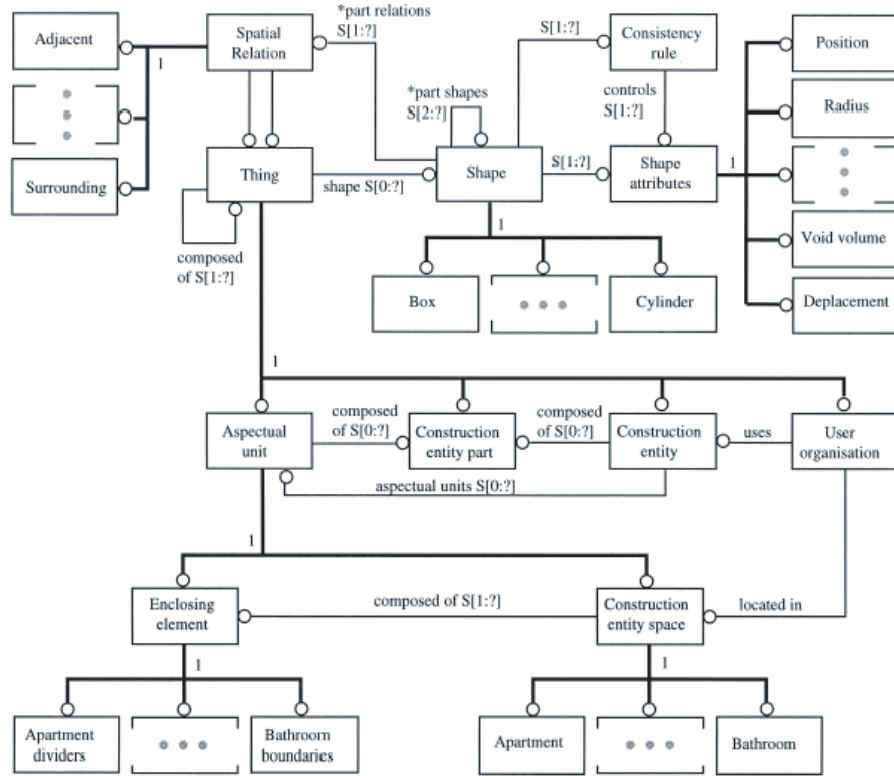


Figure 3: A schema for all the objects involved in spatial knowledge for civil engineering (Fig. 8, pag. 326 [20]).

modeller in the assessment of room geometries, so as to set up the important initial parameters of the problem. Fire modelling expertise is an example of geometric and spatial reasoning, which raises representational problems. The approach taken in this project is a qualitative representation of geometric room information based on Forbus' concept of a metric diagram. This takes the form of a coarse grid, partitioning the domain in each of the three spatial dimensions. Inference over the representation is performed using a case-based reasoning (CBR) component. The CBR component stores example partitions with key set-up parameters; this paper concentrates on the key parameter of grid cell distribution.

2.4 Knowledge Bases

In [7], authors deal with the problem of developing a system that assists engineering design. They argue, in particular, that this task requires an important modeling effort to be accomplished. Computational treatment of these problems is carried out through the application of software engineering and knowledge engineering techniques. They desire to take in account the recent evolution of these fields. In particular, both object-oriented modeling and knowledge modeling present useful individual characteristics that are complementary. The article analyzes the need for integration of those useful modeling aspects, and presents an integrated scheme for the development of knowledge-based design systems (KBDSs).

The schema of Object Design Class they modeled is pictorially represented in Figure 4.

The method pursued is illustrated in three basic points, which are directly inspired by [6].

- Application domain is organized around real-world object classes: design objects and environment objects (environment objects are objects that are not part of design solutions but they and design objects interact, and the definition of design objects depend upon their characteristics). This is based upon the OMT methodology [61].
- The domain knowledge of an application is modeled and structured on the basis of concepts that are of general validity in different domains and phases of engineering design. In particular, the ontological assumptions of the model distinguish among physical, behavioural, functional, environmental interaction, and state characterization aspects.
- The model of the problem-solving needed, is carried out by using the layers for control knowledge in the CommonKADS methodology. This serves as a definition of basic design operations, the relations between them, their organization in design tasks, and the design strategies that are applicable.

The paper then goes deeply in the analysis of the methods applicable to the design model, and applies to the case of preliminary design of foundation

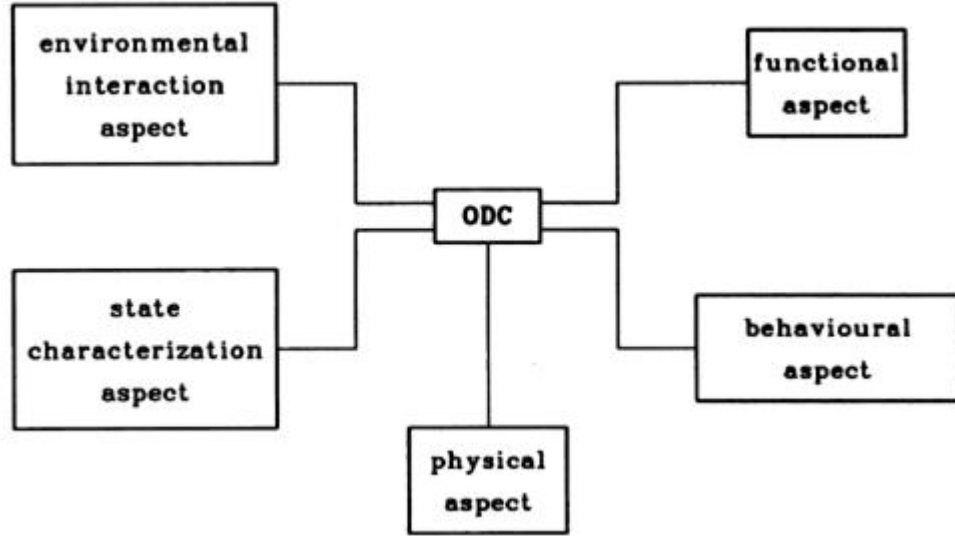


Figure 4: The Object Design Class (ODC) modelling of Bravo-Aranda et al. ([7], pag. 765, Fig. 1)

systems. The result illustrated in the paper can be summarized in three points: between them.

1. Object-oriented modelling of the application domain allows to establish the real-world object classes, their attributes, and relations, that is, to construct a macroscopic model of what there is in the application domain.
2. The use of the conceptual framework defined supports the analysis of domain knowledge through meaningful concepts, and has the ability of making reusable the resultant elements of knowledge.
3. Control knowledge modelling allows to describe the problem solving behaviour required in the application:
 - (a) by identifying the inferences that are to be perform and their knowledge requirements;
 - (b) by establishing the structure of the problem solving process.

3 Terminology and definitions

We employ here few basic terms, that may be left to informal or implicit description with no difficult for understanding the rest of the paper and for which

we anyhow provide formalization.

The name *agent* is assigned to any expert who may act on the web within the upgrade procedure. An agent perform *actions* consisting in doing things that affect the procedure on the web itself. Following the classical planning terminology, we observe that since we have no independent event, action performance is the only type of *postcondition* of an action and, analogously, the performance of other actions is the only type of *preconditions*. A sequence of actions, including alternatives is named an *activity*

The only existing object in our model of knowledge is a *document*. A document may be in standardized *syntactic* form, namely, it is a data document that is serialized in XML, or *semantically* standardized, namely, it is a part of the description of one tunnel based upon the ontology of tunnels firstly introduced in [14].

Agents relate to documents in few predefined ways. In fact an agent can be

- an *author*, who has various privileges, the most obvious of which is deciding who may do actions on the document;
- an *editor* who has the privilege of modifying the document;
- a *reader* who may read the paper itself.

One of the authors is also the *responsible* of the document, who is committed to the deadlines, and is the only agent who has the power of establishing the agenda of authors and editors.

A document will be in a given *state* of its life, which shall depend upon the actions that have been already performed and those which may be performed and shall be performed.

4 The work flow

First of all we shall list in details all the documents involved in an upgrade procedure for long tunnels.

We have three top categories of these documents:

- The *knowledge documents* (KD), containing either standardized ontology-driven tunnel descriptions or attachments to these descriptions, in particular explanations. These documents are in XML serializing RDF translation of Protégé files;
- The *data documents* (DD) containing data in proprietary languages serialized in XML;
- The *reports* (REP), attached to activities.

All the above mentioned documents will be in XML, including the reports, that will be in standard HTML.

The structure of the general activity provided with the upgrade procedure is defined as the *upgrade* activity and is formed by the sequence of the activities mentioned above. A general schema of the implemented procedure can be specified as implemented in three actual levels:

1. The TOP level, in which we have four groups:
 - (a) Preliminary documents (PD);
 - (b) MHF Documents (MHF);
 - (c) Economic documents (ED);
 - (d) Decision documents (DED);
2. The FIRST level, in which we have all the FIRST CLASS documents, and all the SUMMARY documents of SECOND level STRUCTURAL documents, if provided;
3. The SECOND level, in which we have all the SECOND CLASS documents, and all the SUMMARY documents of THIRD level STRUCTURAL documents, if provided;
4. The THIRD level, in which we have all the THIRD CLASS STRUCTURAL documents.

More specifically:

- We have no FIRST level preliminary documents, and possibly two SUMMARY documents, called the *General Tunnel Description* (TD), that is a REPdocument, and the *Upgrade requirements' analysis* (URA) an REPdocument;
 - The TD document summarises the *Risk Profile Report* (RP) (a REPdocument) (a SECOND CLASS STRUCTURAL DOCUMENT), the SECOND level summary (or unsummarised documents) of the Safety Features (SFD) (if it is a summary it would look like a REPdocument), the *Safety level document* (SLD) (a KDdocument) and the Fire Scenarios (FS) (again SECOND CLASS STRUCTURAL DOCUMENTS, whose generic type is KD).
 - The URA document, instead, summarises the *Prescriptive and functional requirements* document (PFR) (a KDdocument), the *Cost/Benefit Requirements* document (CBR) (a REPdocument), the *Human Behaviour Analysis* (HBA) (a REPdocument) and the *Structure Document* (SD) (an REPdocument again). The SFD summarises the *Safety Features' Inventory* (SFI), a KDdocument, and the *Requirements' analysis of safety features* (RASf), a REPdocument.

Document Acronym	Generic Type	Specific Type	Level	Format
TD	REP	PD	FIRST	SUMMARY
URA	REP	PD	FIRST	SUMMARY
SFD	REP	PD	SECOND	SUMMARY
RP	REP	PD	SECOND	STRUCTURAL
SLD	KD	PD	SECOND	STRUCTURAL
FS	KD	PD	SECOND	STRUCTURAL
PFR	KD	PD	SECOND	STRUCTURAL
CBR	REP	PD	SECOND	STRUCTURAL
HBA	REP	PD	SECOND	STRUCTURAL
SD	REP	PD	SECOND	STRUCTURAL
SFI	KD	PD	THIRD	STRUCTURAL
RASF	KD	PD	THIRD	STRUCTURAL
MHF	DD	MHF	FIRST	STRUCTURAL
TA	REP	MHF	FIRST	STRUCTURAL
RP2	DD	MHF	FIRST	STRUCTURAL
BA	REP	ED	FIRST	STRUCTURAL
EA	REP	ED	FIRST	STRUCTURAL
UP	REP	DED	FIRST	STRUCTURAL

Table 2: Document Summary

- Mass-Heat Flow Analysis is deployed into FIRST level documents, the *MHF NewVendis Data* (MHF) that will be a DDdocument, the *Tenability analysis* (TA), a REPdocument and the *Risk Profiling* (RP2), again a DDdocument.
- The Economic documents are just two: the *Benefit analysis* (BA) and the *Cost/Benefit analysis* (EA), both REPdocuments .
- There is only one Decision Document, in the currently deployed hypotheses: the *Requirement Fulfillment Document* (UP), an REPdocument.

In Table 2 we list the above defined documents with the associated properties:

- Document Generic Type (KD, DD, REP);
- Document Specific type (PD, ED, MHF, DED);
- Document Level (FIRST, SECOND, THIRD);
- Document Format (SUMMARY, STRUCTURAL).

The dependencies among documents, in particular the dependency established when a document summarises another one is described pictorially in Figure 5.

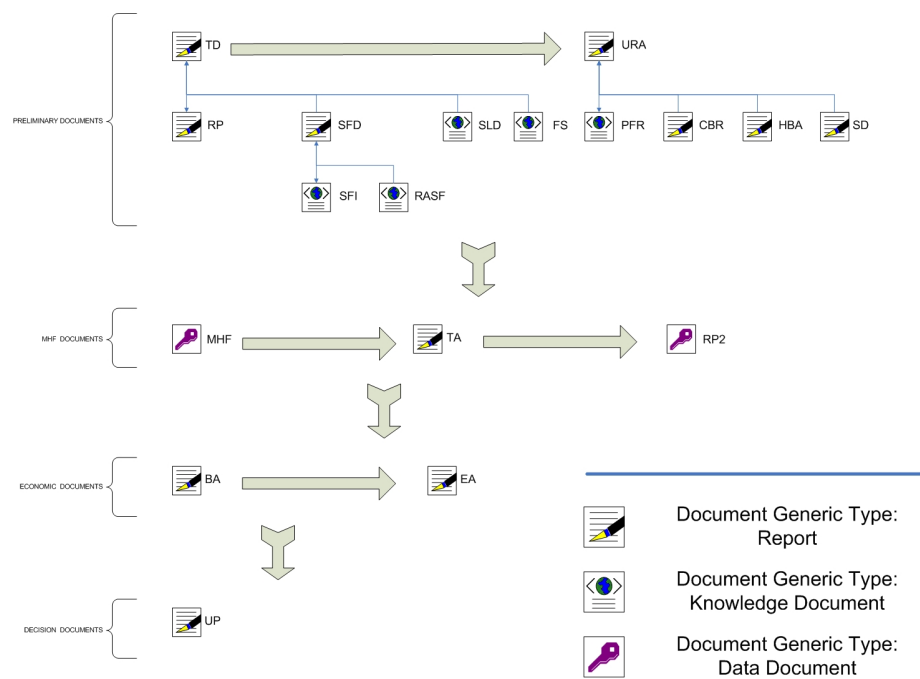


Figure 5: The document hierarchy of INTELLITUN. Arcs represent the relation of summarisation.

5 An architecture proposal

The Deployment of IntelliTUN will take place in three subsequent prototypes.

5.1 First prototype

The first prototype is meant to give a preview of what could be IntelliTuns web interface. This will give an immediate feeling of what will be the interaction with the system, while leaving a good level of separation between the interface and the back-end logic.

5.1.1 What the first prototype will do

- User roles and management (authentication, subscription)
- Document work flow
- Document indexing and tagging (hand-made, basic input fields)
- Flat document storage
- Document search (according to basic indexing)

5.1.2 What the first prototype will not do

- User invitation via e-mail
- Addressbook import
- OntoTun interaction
- Advanced work flow management (reminders, to-do lists)
- Automatic document indexing
- Document versioning
- Advanced document search

5.1.3 Architectural details and explanation

The IntelliTun web interface will be based solely on Open Source components and software. Zope is the Web Application Server of choice, along with Python as main programming language. Open standards such as XHTML and, generally, XML will be preferred in any possible case. Since Zope won't hold the whole system and there will be many external processes and components to interact with it, mainly for user management, users will be stored in a standard SQL database, such as MySQL. This will be accomplished with commonly available Python-MySQL and Zope-MySQL products. Document work flow will be

implemented with the help of OpenFlow, an italian Open Source Zope product for work flow management. User interface will be designed using XHTML and CSS, producing W3C compliant pages. Where needed, pages will be rendered via an XSL transformation. This will be useful in the following prototypes, e.g. where interaction with OntoTun will produce XML documents with the fields which the indexing engine hasn't been able to fill automatically. Documents will be stored inside the ZODB, Zope's object database. ZODB's advanced features, such as dynamic object type definition and tagging, won't be used by this prototype. The document archive will be initially flat, meaning that all documents are stored inside a single folder and indexed with Zope's basic indexing components (Dictionary).

5.2 Second prototype

With the second prototype user and work flow management will be improved. There will be also a first interaction with OntoTun, deriving forms for manual document indexing and tagging directly from the OntoTun specification, via XSL transformations. Users will be prompted by the system about the upcoming deadlines, both via e-mail and via messages on the web interface. Documents will be stored with versions, allowing to track the various releases. Using Zope's content management capabilities, users will be able to share documents between members of a group (previously defined by an administrator).

5.2.1 What the second prototype will do

- User invitation via e-mail
- OntoTun interaction (form generation)
- e-mail reminders
- To-do lists
- Document sharing
- Document versioning
- Run-time document work flow modification
- Document search results sorted by relevance

5.2.2 What the second prototype will not do

- Automatic document indexing and tagging
- Document history logging
- Run-time new work flow creation

- Addressbook import
- Document ranking

5.2.3 Architectural details and explanation

User invitation will be managed via an external component which will interact with both Zope and MySQL, reading mail templates from the ZODB, filling them with data coming from MySQL, where users' personal data will be stored. This component will be implemented from scratch as a Python script invoked as a cron job, which will parse a Mailbox file searching for accepted invitations and will send new invitations, assignments and reminders. Since only the ZODB-connected part (where we read mail templates to fill) is strictly related with the project architecture, this is a simple "just-do-the-job" component, which can be easily replaced with better Open Source solutions (provided they can connect to the ZODB and find there what they need). Both e-mail reminders and to-do lists are automatically generated by the system at user-interaction time, they can also be issued manually by the system administrator to a user or a group. System administrators will be also able to move documents between work flows, if needed. Search results will be sorted by relevance, according to their tags.

5.3 Third prototype

The third prototype will be as close as possible to a fully working system, indexing and tagging documents automatically and leveraging the "finished product" experience for users.

5.3.1 What the third prototype will do

- Automatic document indexing and tagging
- Document history logging
- Run-time new work flow creation
- address book import
- Document ranking
- ZEO Scalability

5.3.2 What the third prototype will not do

- Automated reasoning;
- Peer to Peer Knowledge Management.

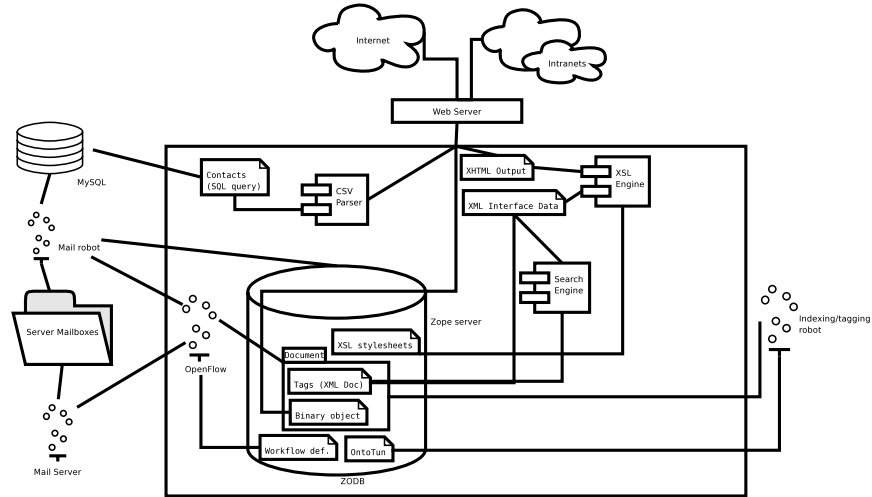


Figure 6: The architecture of the third prototype.

5.3.3 Architectural details and explanation

External components will be invoked by Zope and will access the ZODB every time a new document is uploaded to the system. They will index and tag it according to the OntoTun specification, and they will generate the XML document with all the missing fields (those they haven't been able to fill). That document, via XSL transformation, will produce a form for manual user input, to fill the missing fields. Every operation on a document will be tracked in an external SQL table, leaving a detailed log of all operations (not just revisions, as in the second prototype). Administrators will be able to define new work flows, using OpenFlow syntax or a simplified interface to it (that must be implemented in that case). Every administrator will have his/her personal address book, which can be imported from common mail clients (Outlook, Evolution, Mozilla Thunderbird, ...) via CSV file (they must export their address book from the mail client and upload it to the server). address books will be stored inside the SQL server. Search results will be ranked according to their relevance (tags and keywords), version history and previous user interactions with the search engine. The whole system will be re-considered in its non-Zope parts (MySQL, mail robot, tag-index robots) to allow taking full advantage of Zope's ZEO scalability.

The third prototype architecture is pictorially represented in Figure 6.

6 Conclusions and further work

This paper dealt with the problem of deploying a web application able to manage the upgrade procedure of long tunnels. The study aims at the definition of a document workflow whose implementation is provided in Zope, a Web Servlet that can be used for web applications (Content Management Systems).

The managed workflow is based upon a general schema of the holistic procedure for upgrading long tunnels analysed in the European Project UPTUN, and attempts at providing the very same sequence of analytical steps.

Acknowledgments

I gratefully thank the European Community, under Grant GRD1-2001-40739 (the UPTUN Project), for supporting the investigation which we have documented in this paper.

The UPTUN-project Cost-effective, sustainable and innovative upgrading methods for fire safety in existing tunnels is being carried out in the framework of the Competitive and Sustainable Growth Programme (project GRD1-2001-40739, Contract G1RD-CT-2002-0766), with a financial contribution of the European Community.

I would also like to thank Michele Albrigo for his very useful help in both the development of the prototypes and their description.

References

- [1] L. Alberts and F. Dikker. Integrating standards and synthesis knowledge using the ymir ontology, 1994.
- [2] American Society of Civil Engineers, Reston, VA. *Quality in the Constructed Project (A Guide for Owners, Designers, and Constructors)*, second edition, 2000. 0-7844-0506-9, 266 pp.
- [3] C.J. Anumba. Data structures and dbms for computer-aided design systems. *Advances in Engineering Software*, 25:123–129, 1996.
- [4] C.J. Anumba, Z. Rena, A. Thorpe, O.O. Ugwub, and L. Newnhamc. Negotiation within a Multi-agent System for the Collaborative Design of Light Industrial Buildings. *Advances in Engineering Software*, 34:389–401, 2003.
- [5] C.J. Anumba, O.O. Ugwu, L. Newnham, and A. Thorpe. Collaborative Design of Structures Using Intelligent Agents. *Automation in Construction*, 11:89–103, 2002.
- [6] G. Bravo-Aranda. *Modelización de Problemas de Diseño en Ingeniería: Solución mediante Sistemas Basados en el Conocimiento*. PhD thesis, Universidad de Sevilla, 1995. in spanish.

- [7] G. Bravo-Aranda, F. Hernandez-Rodriguez, and A. Martin-Navarro. Knowledge-based System Development for Assisting Structural Design. *Advances in Engineering Software*, 30:763774, 1999.
- [8] Frances M.T. Brazier, Lilia V. Moshkina, and Niek J.E. Wijngaards. Knowledge Level Model of an Individual Designer as an Agent in Collaborative Distributed Design. *Artificial Intelligence in Engineering*, 15:137–152, 2001.
- [9] C.J. Burgoyne and S.R.H. Sham. Application of expert systems to prestressed concrete design. *Civil Engineering Systems*, 4:14–19, 1987.
- [10] Octavio Cabezut-Boo and Antonio Sanchez-Aguilar. Towards an Ontology of Waste Water Treatment Plants: the Identification Phase. *Environmental Modelling and Software*, 14:401–408, 1999.
- [11] Michael P. Case and Stephen C-Y Lu. Discourse model for collaborative design. *Computer Aided Design*, 28(5):333–345, 1996.
- [12] Wai-Fah Chen and J. Y. Richard Liew, editors. *The Civil Engineering Handbook*. CRC Press, second edition, 2002.
- [13] C.K. Choi and I.H. Choi. An expert system for selecting types of bridges. *Computers and Structures*, 48(2):183–192, 1993.
- [14] Matteo Cristani. An ontology of tunnel safety features. Research Report RR 10/2003, Dipartimento di Informatica, Università di Verona, 2003.
- [15] Matteo Cristani, Gabriel A. Khoury, and Carmelo E. Majorana. The control of upgrade activities for long tunnels by an intelligent system. In B.H.V. Topping, editor, *Proceedings Of The Seventh International Conference On The Application Of Artificial Intelligence To Civil And Structural Engineering*. Civil-Comp Press, 2003.
- [16] R. Davis and J.J. King. An overview of production systems. In E. Elcock and D. Michie, editors, *Machine Intelligence*. Horwood, England, 1982.
- [17] Johan de Gelder. Conceptual modelling of building regulation knowledge. *Artificial Intelligence in Engineering*, 11:273–284, 1997.
- [18] C.L. Dym and E.T. Levitt. *Knowledge-Based Systems in Engineering*. McGraw-Hill, New York, 1991.
- [19] Michael Eisfeld and Raimar Scherer. Assisting Conceptual Design of Building Structures by an Interactive Description Logic based Planner. *Advanced Engineering Informatics*, 17:41–57, June 2003.
- [20] A. Ekholm and S. Fridqvist. A Concept of Space for Building Classification, Product Modelling, and Design. *Automation in Construction*, 9:315–328, 2000.

- [21] S.J. Fenves, U. Flemming, C. Hendrickson, M.L. Maher, and G. Schmitt. Integrated software environment for building design and construction. *Computer-Aided Design*, 22(27), 1990.
- [22] S.J. Fenves, H. Rivard, and N. Gomez. Seed-config: a tool for conceptual structural design in a collaborative building design environment. *Artificial Intelligence in Engineering*, 14, 2000.
- [23] Renate Fruchter, Mark J. Clayton, Helmut Krawinkler, John Kunz, and Paul Teicholz. Interdisciplinary communication medium for collaborative conceptual building design. *Advances in Engineering Software*, 25:89–101, 1996.
- [24] M. Green, editor. *Knowledge Aided Design*. Academic Press, London, 1993.
- [25] I. Hibs and J.S. Gero. An evolutionary process model of design. *Design Studies*, 13(3):272–290, 1992.
- [26] Tyler Gregory Hicks. *Handbook of Civil Engineering Calculations*. McGraw-Hill Professional, 31 August 1999.
- [27] J. Hinze and A. Tracey. The contractorsubcontractor relationship: the sub-contractors view. *Journal of Construction Engineering and Management ASCE*, 120(2):274–287, 1994.
- [28] M. Hudson. *Rock Engineering Systems; Theory and Practice*. Ellis Horwood, London, UK, 1990.
- [29] D. Jain, H. Krawinkler, K.H. Law, and G.P. Luth. A formal approach to automating conceptual design, part ii: Application to floor framing generation. *Engineering with Computers*, 7(91), 1991.
- [30] G.S. Kalamaras. A Computer-based System for Supporting Decisions for Tunneling in Rock under Conditions of Uncertainty. *International Journal of Rock Mechanics and Mineral Science*, 34(3/4):588, 1997.
- [31] Yehuda E. Kalay. Enhancing Multi-disciplinary Collaboration through Semantically Rich Representation. *Automation in Construction*, pages 741–755, 2001.
- [32] B. Knight, S. Taylor, M. Petridis, J. Ewer, and E.R. Galea. A knowledge-based system to represent spatial reasoning for fire. *Engineering Applications of Artificial Intelligence*, 12:213–219, 1999.
- [33] C.N. Kostem and M. L. Maher, editors. *First Symposium on Expert systems in civil engineering*, Seattle, Washington, 1986. ASCE.
- [34] C. S. Krishnamoorthy, S. Karimalla Raja, S. Rajeev, and H. Shiva Kumar. Development environment for knowledge-based system for engineering design. In *Second International Conference on the Applications of Artificial Intelligence to Civil and Structural Engineering*, pages 165–174, Oxford, England, 1991.

- [35] C.S. Krishnamoorthy, H. Shiva Kumar, S. Rajeev, and S. Suresh. A knowledge-based system with generic tools for structural engineering. *Structural Engineering Review*, 5(2):121–131, 1993.
- [36] C.S. Krishnamoorthy and S. Rajeev. *Artificial Intelligence and Expert Systems for Engineers*. CRC Press, CRC Press LLC, 1996.
- [37] C.S. Krishnamoorthy, C. Srinivasa Rao, and S. Rajeev. Synthesis in engineering design. *Journal of Structural Engineering*, 4(118):125, 1992.
- [38] H. Shiva Kumar and C.S. Krishnamoorthy. A Process Model for Knowledge-based Concrete Bridge Design. *Engineering Applications of Artificial Intelligence*, 8(4):435–447, 1995.
- [39] H. Shiva Kumar and C.S. Krishnamoorthy. A frame work for case-based reasoning in engineering design. *Artificial Intelligence in Engineering Design, Analysis and Manufacturing*, 9:161, 1995.
- [40] H. Shiva Kumar, S. Suresh, C.S. Krishnamoorthy, S.J. Fenves, and S. Rajeev. Gencrit: A tool for knowledge-based critiquing in engineering design. *Artificial Intelligence in Engineering Design, Analysis and Manufacturing*, 8:239, 1994.
- [41] G.P. Luth, D. Jain, H. Krawinkler, and K.H. Law. A formal approach to automating conceptual design, part i: Methodology. *Engineering with Computers*, 7(91), 1991.
- [42] Ian A. Macleod. Computer representation issues for civil engineering design. *Structural Engineering Review*, 8(2/3):99–103, 1996.
- [43] M.L. Maher. Engineering design synthesis: a domain independent implementation. *Artificial Intelligence in Engineering Design, Analysis and Manufacturing*, 1(3):207, 1987.
- [44] M.L. Maher. Expert systems for structural design. *Journal of Computing in Civil Engineering*, 1(4):270–, 1987.
- [45] M.L. Maher. Synthesis and evaluation of preliminary designs. In J.S. Gero, editor, *Artificial Intelligence in Design*. Springer-Verlag, Berlin (Germany), 1989.
- [46] M.L. Maher. Process models for design synthesis. *AI Magazine*, Winter 1990:49, 1990.
- [47] M.L. Maher, D. Sriram, and S.J. Fenves. Tools and techniques for knowledge-based expert system for engineering design. *Advances in Engineering Software*, 6, 1984.
- [48] H. Mason. ISO 10303-1 overview and fundamentals principles of STEP. Technical Report TC184/SC4/PMAG, 1992.

- [49] C.J. Moore. *An expert system for the conceptual design of bridges*. PhD thesis, Cardiff School of Engineering, University of Wales, Cardiff (UK), 1991.
- [50] C.J. Moore and J.C. Miles. The development and verification of a user oriented KBS for the conceptual design of bridges. *Civil Engineering Systems*, 8:81–86, 1991.
- [51] C.J. Moore and J.C. Miles. The importance of detailed KBS evaluation for systems intended for implementation in industry. *Computing Systems in Engineering*, 2(4):365–378, 1991.
- [52] C.J. Moore, J.C. Miles, and D.W.G Rees. Decision support for conceptual bridge design. *Artificial Intelligence in Engineering*, 11:259–272, 1997.
- [53] D.V. Morse and C. Hendrickson. Model for communication in automated interactive engineering design. *Journal of Computing in Civil Engineering*, 6(1):85–105, 1990.
- [54] H.G. Natke, H. Doll, P. Hildebrandt, and M Schuetze. Bridge condition assessment using an expert system. *Structural Engineering Review*, 7(3):165–180, 1995.
- [55] S. Rajeev and C.S. Krishnamoorthy. Discrete optimization of structures using genetic algorithms. *Journal of Structural Engineering*, 118(5):1205, 1992.
- [56] S. Rajeev, C.S. Krishnamoorthy, S. Suresh, and H. Shiva Kumar. Design evaluation using knowledge-based techniques. *Journal of Structural Engineering*, (151), 1992.
- [57] S. Rajeev, S. Suresh, and C.S. Krishnamoorthy. Design criticism: a generic component in knowledge-based systems for engineering design. In B.H.V. Topping, editor, *Second International Conference on the Applications of Artificial Intelligence to Civil and Structural Engineering*, Oxford, 1991. Civil-Comp Press.
- [58] R.B. Reddy, A. Gupta, and R.P. Singh. Heuristic, symbolic logic and knowledge-based approach to the design and construction of buildings. *Computers and Structures*, 43(6):1191–1197, 1992. Technical Note.
- [59] Debbie Richards and Simeon J. Simoff. Design Ontology in Context - A Situated Cognition Approach to Conceptual Modelling. *Artificial Intelligence in Engineering*, 15:121–136, 2001.
- [60] S. Roderman and C. Tsatsoulis. Panda: A case-based system to aid novice designers. *Artificial Intelligence in Engineering Design, Analysis and Manufacturing*, 7(2):125, 1993.

- [61] J.B. Rumbaugh, M. Premerlani, W. Eddy, and F.W. Lorensen. *Object-oriented modeling and design*.
- [62] Paolo Salvaneschi, Mauro Cadei, and Marco Lazzari. A Causal Modelling Framework for the Simulation and Explanation of the Behaviour of Structures. *Artificial Intelligence in Engineering*, 11:205–216, 1997.
- [63] R. Sause and G.H. Powell. A design process model for computer integrated structural engineering. *Engineering with Computers*, 6:129, 1990.
- [64] R. Sause and G.H. Powell. A design process model for computer integrated structural engineering: design phases and tasks. *Engineering with Computers*, 7:145, 1991.
- [65] C. E. Shannon. Programming a computer for playing chess. *Philosophical Magazine*, 41:256–275, 1950. Series 7.
- [66] S.J. Simoff and M.L. Maher. Deriving ontology from design cases. *International Journal of Design Computing*, <http://www.arch.usyd.edu.au/kcdc/journal>, 1998.
- [67] Chee-Kiong Soh, A.K. Soh, and Kum-Yew Lai. A simple knowledge system environments for engineering problem solving. *Computing Systems in Engineering*, 6(6):485–496, 1995.
- [68] W.J. Spences, R.M. Atkins, and P. Podlaha. Development of an expert system for the preliminary design of bridges. *Civil Engineering Systems*, 6:51–57, 1987.
- [69] D. Sriram, M.L. Maher, and S.J. Fenves. Knowledge-based expert systems in structural design. *Computers and Structures*, 20(1-3):1, 1985.
- [70] S. Rajeev S. Suresh and C.S. Krishnamoorthy. A criticism-based model for cooperative problem solving. *International Journal of Computing Systems in Engineering*, 4(2-3):201–210, 1993.
- [71] H. Takeda, P. Veerkamp, T. Tomiyama, and H. Yoshikawa. Modeling design processes. *AI Magazine*, (Winter 1990):37–48, 1990.
- [72] A.B. Tee and M.D. Bowman. Bridge condition assessment using fuzzy weighted averages. *Civil Engineering Systems*, 7:49–57, 1991.
- [73] H. Ping Tserng and Will Y. Lin. Developing an Electronic Acquisition Model for Project Scheduling Using XML-based Information Standard. *Automation in Construction*, 12:67–95, 2003.
- [74] S.J.M. van Eldonk, L.K. Alberts, R.R. Bakker, F. Dikker, and P.M. Wognum. Redesign of Technical Systems. *Knowledge Based Systems*, 9:93–104, 1996.

- [75] Žiga Turk. Phenomenological foundations of conceptual product modelling in architecture, engineering and construction. *Artificial Intelligence in Engineering*, 15:83–92, 2001.
- [76] M.R. Wigan. Engineering tools for building knowledge-based systems on micro systems. *Micro Computers in Civil Engineering*, 1(52), 1986.
- [77] F.S. Wong and W. Dong. Fuzzy information processing in engineering analysis. In D. Sriram and R. Adey, editors, *First International Conference on the Applications of Artificial Intelligence in Engineering Problems*, Southampton University, U.K., 1986.
- [78] F. Zhao and M.L. Maher. Using analogical reasoning to design buildings. *Engineering with Computers*, 4:107, 1988.