# A SCENARIO FOR BUILDING REGULATIONS PROCESSING IN A NETWORKED ENGINEERING ENVIRONMENT

**Turk Ž.**

*ABSTRACT: Building regulations and other similar documents embed extensive knowledge which decisively influences construction products throughout their life cycle. I has proven very difficult to convert the whole meaning of a building regulation into a computer program. A likely scenario is that there will exists different networked computer programs (regulation servers) each of which will be able to handle some provisions of some regulation. Regulation servers may use different approaches for the representation of the building code knowledge, different conformance checking mechanisms and different interfaces to the outside world, but in order to be useful in a networked engineering environment they should be able to advertise their service and to interface to client programs in a way useful to the environment. The paper presents work in progress related to the integration of electronic processing of building codes into a concurrent engineering environment, the requirements and an overall system architecture which introduces regulation agents and mapping servers addresses some of the problems discussed above.*

*KEYWORDS: computerization of building regulations, concurrent engineering environment, building product models, regulation servers, regulation agents ...*

## 1. INTRODUCTION

The building regulations together with other regulatory documents such as codes, standards and bylaws, supposedly contain all knowledge that, if interpreted correctly, ensures that the building products are safe and reliable. They are intended to be interpreted by a human expert (engineer) and assume a certain level of general and engineering knowledge. Engineers should engineer in accordance with the building regulations. In a computer aided engineering environment, computers are expected to aid in ensuring that the resulting products conform to the regulations. The process which converts the regulations into a form where computers can manipulate, is called "computerization of building regulations".

The computerization approaches taken by research and commercial development had ambitions to answer questions of different complexity:

1. What does section M of regulation N say ?
2. What regulations apply to my problem ?
3. Does component O of the design P conform to the regulation R?
4. Does my design conform to the regulations ?

Even though computing the answers to any of the questions above is not trivial, the last question is clearly the most difficult one. A computer program answering the last question, when coupled with a program generating design alternatives, could automatically compute safe and reliable designs. Intuitively feeling the complexity of the knowledge needed to use and to correctly interpret the regulations and the related background knowledge, computer made design is considered unlikely for non-trivial cases for some time to come.

Several computerization models have been proposed which tried to computerize the meaning of the regulations (question 4), but without a significant impact on the software market. Kiliccote and Garrett (1995) argue that the complexity of the standards was underestimated and list four issues which were not addressed properly:

- Regulations are not self sufficient.
- Regulations are ambiguous.
- Regulations have exceptions.
- Regulations include higher order logic.

On the other hand, approaches with a less ambitious goal are successfully used in the practice. Numerous CAD packages embed some easy to interpret provisions in a fairly simple context and make sure that an engineer can only make design decisions which conform to the building regulation (questions 2 and 3). Various FEM packages include loading according to standards, proportioning software makes sure safety margins obey to the regulations, drafting packages include hatch patters and line types according to standards (Turk and Duhovnik, 1995). A growing number of regulation bodies publishes the regulations in an electronic format as hypertexts with efficient search mechanisms (question 1 and 2).

Just like CAD packages do not do the whole design, but more and more efficiently help the human designers, the computerization of building regulations should follow a similar pattern. Architects and engineers use a number of different CAD packages. To an extent, these packages can work together. A whole research area of CIC (Computer Integrated Construction) has been trying to make this collaboration tighter. Just as it has proven too difficult to write a single computer program which would be able to help in the design, construction and maintenance of a whole building products, it has proven practical to extract building regulations knowledge out of the CAD packages. The main reasons have been:

1. The building codes are complex and would add a lot to the complexity of a CAD package.
2. The type of knowledge in the regulations is different (declarative) to the type of knowledge in the CAD programs (procedural).
3. In a broad marketplace several different local regulations may be used.

It is therefore clear that for non-trivial cases regulation-processing modules are a fairly self sufficient building blocks of a CIC environment and can act as regulation servers. A general framework for the gradual and partial incorporation of the regulations and the related knowledge is required.

## 2. REQUIREMENTS

This section discusses the requirements different players in a CIC environment have towards regulation-servers. The players are engineers (CIC system users), regulation authors, software authors, and the authorities which check designs etc. against the regulations.

### 2.1 Engineers

a) Answers to tough questions. The ultimate goal an engineer might have is that the regulation server would be answer to level 4 questions. But if this is not possible, any help with the other three would be welcome, the more the better.

b) Up to date. Particularly in fast paced technologies such as (in construction) using new materials or changing legislation, electronic versions have the potential of delivering up-to date information and also to inform the user of the changes.

c) Regulation independence. The scope of a regulation is technical (which building components or processes it regulates), spatial (where in the world it applies) and temporal (when in time is it valid). There are different regulations addressing similar problems valid in different counties, states or countries. It would be desirable to be able to select which space-time regulation to use, just like switching between spelling checkers in a word processing program.

d) Application independence. There are several CAD packages working on information also addressed in the regulations. It should be possible that one regulation server could serve any application.

e) Open toward user. Average user's knowledge about a regulation will be, for some time, larger than a regulation server might have. Also, users wish to personalize, extend or bookmark the electronic version of the regulation for a more efficient use.

f) Open toward user community. The same facts stated for the user are also valid for a group of users which might wish to share knowledge about the regulations and it's applicability. Larger firms might wish to incorporate their rules of practice, their ways of using the regulation.

### 2.2 Regulation publishers and authors

a) Copyright and billing. Several standardization and professional bodies are in a large proportion funded by selling copies of the regulations. Making electronic copies makes copying much easier. It is also unclear how to handle computer programs which embed in a computable form, or cite as text and graphics, the contents of a regulation. It would be desirable that any use of the regulation, either as (hyper)text or it's computational format, could be billed.

b) Computerisation process. A uniformed way of authoring and publishing electronic format of the regulations is desirable.

c) Updates. Generally electronic copies are not bound to physical media or that media is very inexpensive. Therefore making changes, particularly to the explanations and supporting documents should be easy and inexpensive.

d) Feedback. The regulation texts are put to the real test, when applied in practice. By opening a communication channel between the authors and the user communities could result in improved future releases of the regulations.

### 2.3 Software authors

a) Reusability. A desirable feature is to isolate the complex coding into the representation specific part, make the regulation specific part simple and easy to create (at the expense of having many different representations), and minimize the number of client specific parts and agent specific parts to one.

b) Modularity, scaleability. It should be possible to break a computerized regulations into modules, each with its own representation. It should be possible to add modules gradually and locally - supplying the easy-to-do modules first and the more difficult-to-do later.

c) Environemnt friendliness. Building regulation servers are part of a concurrent engineering environment and should be easy to integrate into that environment.

## 3. AVAILABLE TECHNOLOGIES

On the today's WWW, the servers are mostly serving static information such as documents. The technology is quite sufficient for level 1 and level 2 questions as suggested by Vanier and Turk (1995). In the solution they proposed most of the section 2 requirements can be fulfilled.

Some web servers serve dynamic documents - typically database queries. The communicating computers are in a client-server relation where the server processes queries and serves replies to queries and the client presents the information on screen and handles the user interface. An emerging paradigm on the WWW, currently supported by the Java and Java Script languages is that functionality too is being uploaded from the network and processed locally. Instead of displaying a document sent from the server a program is uploaded which then, running on the client, presents the information, does the user interaction etc. It is expected (Gates, 1995) that in the future program components too will be downloaded from the network 'just-in-time' when needed and not when we decided to buy some software 'just-in-case' we need it. Therefore, the network can provide the following types of standards servers:

- Document servers with various search capabilities. Many national and international standardization bodies already provide regulations in this way. They also provide full text searching an hypertext capabilities. Few provide full text for economic reasons mainly.
- Servers checking conformance. Servers which are sent a part of the product model and they attempt to validate it.
- Servers serving conformance checkers, either in a form of Java and JavaScript programs or dynamic load libraries, executable files etc.

## 4. THE SCENARIO

### 4.1 Standards in a Cee Environment

Concurrent engineering (CE) is a systematic approach to the integrated, concurrent design of products and their related processes, including manufacturing and support. This approach is intended to cause developers, from outset, to consider all elements of the product life cycle from concept through disposal, including quality, cost, schedule and user requirements (Carter and Baker, 1992). Concurrent Engineering Environment (CEE) is where concurrent engineering takes place. It is believed that information technology can help make CE much easier to implement, because it can manipulate information in a much more efficient way that this is done without IT. IT is the technology used to build a CEE system. A CEE system is complex and can only be realized if wisely decomposed into manageable components. One such decomposition is depicted in Fig. 1. The building regulations and the user requirements provide the constraints which the building product, created during the building process with an aid of CEE system, should fulfill. The product is described in building documentation.

In an ideal CIC environment where a building product model exists, as shown in Fig. 1, the building regulations refer to building product and (perhaps) building process objects. A regulation object will need to know, and consequently use, a definition of a beam object, as defined in a product model, because it is supposed to evaluate that beam. In a practical environment without a central building product model, the situation is a bit more complicated, as shown in Fig. 2:

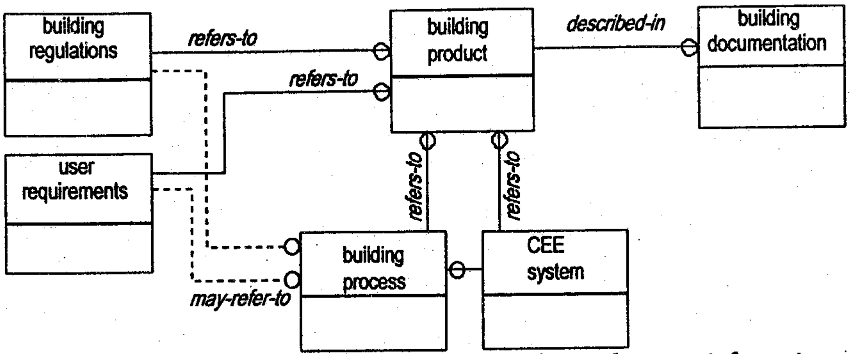


*Fig. 1: Building regulations and the relation to product and process information.*

The fundamental contradiction between Figs. 1 and 2 is that in the first schema the objects of the regulation model are the clients, while the whole - the regulation server - is the server of the various tools used in a concurrent engineering environment.

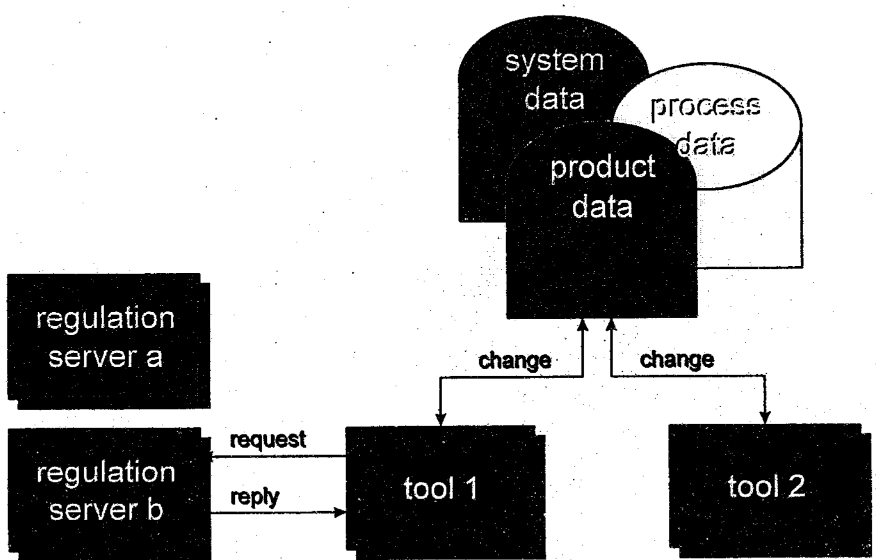


*Fig.2: Regulations processing within servers in a CIC system.*

### 4.2 Overall architecture

The overall architecture builds on the three actors which are involved in regulation processing - clients, servers and agents. Fig. 3 depicts one alternative to their relations:

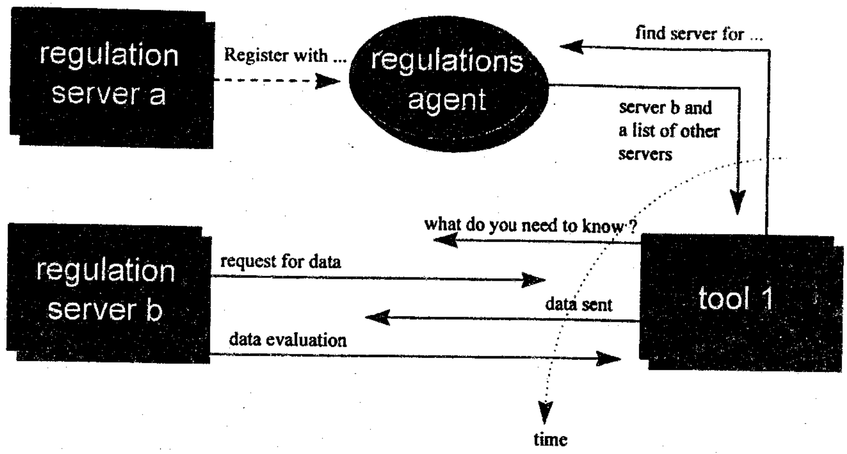


*Fig. 3: Regulation servers, regulation agents and regulation clients (tools). An engineering tool needs to consult a regulation. It first asks a regulation agent, who might be able to accomplish that. The agent replies with a list of servers which can help. The tool than asks the server what kind of information does it need. Server replies with a form a tool must fill in. The tool send the data and the regulation server replies with it's evaluation.*

The functions such a system must perform are:

- allowing the registration of a server with an agent;
- finding a regulation server which could help in checking the conformance of a product component;
- establishing the interface between the regulation server and the regulation client;
- mapping of client data into server data.

### 4.3 Regulation servers

The components of an requirements server are:

- Regulation specific part includes rule sets, procedures, objects or other components specific to a single regulation.
- Representation specific part are components, which are unique to a representation type, but may be reused with all regulations using a this regulation type. E.g. hypertext browser, rule evaluation engines ...
- Client specific parts are components which are special for each client using the regulation server.
- Agent specific parts are components specific to each agent, which will discover the standard server to the client.
- Application interface. An important feature of a server is that all servers present themselves to the outside world in a uniform way. This means they should have not client or agent specific parts but be compatible with a standardized application interface.

A server should be able to make the following information about itself available to the agents. All this information should be made available in a standardized uniform format:

- access and billing related information;
- server type (hypertext server, checker or checker server);
- server functions (access, calling parameters, bibliographic information). A server may, of course, perform more than one function.

A server should be able to communicate with the outside world on two levels:

- It should be able to publish its schema (e.g. EXPRESS or some easier to parse semantic representation).
- It should be able to exchange product data (e.g. STEP physical files).

### 4.4 Regulation agents

Regulation agents should store lists of regulation servers and the uniform descriptions of their functionality. In addition to simply finding a server and passing that information to the client and letting that client arrange a session with the server, agents could be more helpful in storing server details so that the dialogue between server and client could be shorter. Agents could also include other information such as certification data on some regulation servers, performance and reliability evaluation etc.

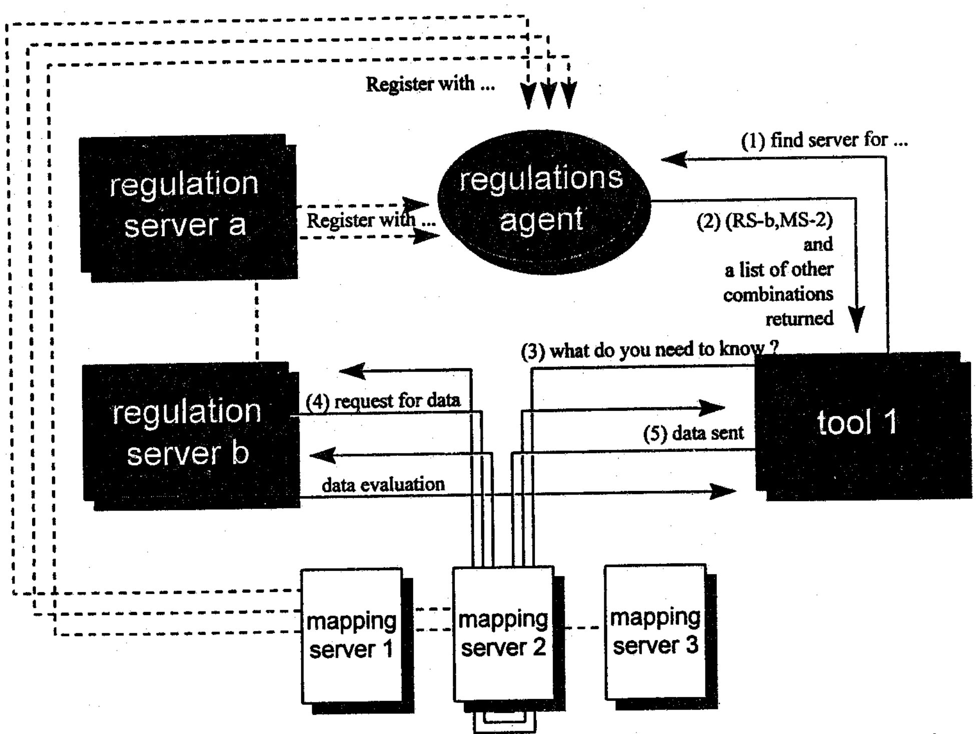


*Fig. 4: With mappings. Regulation servers, regulation agents, regulation clients and mapping server. The client also learns about the mapping servers.*

### 4.5 The mapping problem

Mapping is a procedure which transforms information from one schema into another. Many problems, which are preventing computer integrated construction from becoming reality can be traced back to a mapping problem. Using standardised conceptual models attempts to bypass that problem by enforcing a single schema. With respect to the time, when a need for the mapping is realised, two types of mappings can be distinguished. The 'static' mapping maps between the schema of two applications for which we know in advance, must need to communicate e.g. between a schema of a finite element program and a 3D modelling tool. In regulation processing we are generally need to map between an application schema and all regulation schema just to be able to determine if the regulation applies. If it applies, a complete mapping really becomes necessary. Another way of classifying the mappings is by the way they are performed. They may be generated automatically - based on a semantic representation of schemata (Turk, 1993) or the mapping application must be written by a human. A likely scenario is a combination of the two approaches.

Network environment can help in creating the necessary mappings. In the environment we have published interfaces to regulation servers (which include an schema the server uses) and applications (which may or may not have a published interface). Anyone can therefore write or generate a mapping application between each regulation-server/application pair and register it with a regulation agent. There may exist a huge set of the pairs, but in a global networked environment, each mapping will need to be defined only once. After it is published with the agent, there will be no need for anyone else to re-invent that same mapping. In time these manually written mapping servers may provide the examples for creating new ones and in providing limited automated mappings or the invention of mapping generators. Fig. 3 can thus be redrawn as Fig. 4.

The mapping database is a part of the regulation agent and stores 3-touples {regulation-client, regulation-server, mapping-server}.

## 5. CONCLUSIONS

A scenario for the networked use of standard processing and conformance checking software has been outlined. It consists of regulation servers, regulation clients, mapping servers and regulation agents. They are all located in a networked environment such as the Internet and support downloadable functionality as opposed to today's prevailing downloadable data and information. In itself the proposed solution does not solve the information transfer problem, but it uses the global networks to throw it at a much wider audience which can, collaboratively, each providing one small brick, crack it.

This work in progress is a part of the EU's ToCEE Esprit project and Slovenian COBRA research project. First prototypes are expected by the end of the year.

## 6. REFERENCES

Carter, D. and Baker, B. (1992). CE concurrent engineering: The product development environment for the 1990s, Addison Wesley, USA.

Gates, B. (1995). The road ahead, Viking Penguin Books, USA.

Kilicotte, H. and Garrett, J.H. Jr. (1995). Obstacles to the development of computable models of design standards, in Fischer, Law and Eastman (editors), Modelling of buildings through their life cycle, CIB Proceedings Publication 180, Stanford University, California, USA.

Turk, Ž. (1993). Semantic representation of product modelling data, Proceedings MITC Conference, National University of Singapore, Singapore, 1993.

Turk, Ž. and Duhovnik, J. (1995). Slovenia and computer representation of design standards and building codes, International Journal of Construction Information Technology, Vol.3, No.1, pp. 55-71, 1995.

Turk, Ž., and Vanier, D.J. (1995). Delivery of building codes using the Internet, in J.P. Mohsen (editor) Computing in civil engineering, pp. 1651-1658, American Society of Civil Engineers, ZDA, 1995.