The use of software systems to implement Case-Based Reasoning enabled intelligent components for architectural briefing and design

By

Dirk Cornelis Uys Conradie

Submitted in fulfilment of part of the requirements
for the degree of Philosophiae Doctor (Applied Sciences)
Faculty of Engineering, the Built Environment and
Information Technology
University of Pretoria
South Africa

Study Leader/Promotor: Prof. D. Holm

October 2000

Abstract

This thesis describes the development of a prototype *Case-Based Reasoning* (CBR) enabled intelligent component system, called Architectural General Object System (ARGOS), to facilitate the storage of design information in lightweight cases that can be used on the desktop computer over the total life of the facility. It uses *CBR* techniques combined with *Microsoft ActiveX* controls (object technology) to provide a useful autonomous component to implement some of the software requirements of such a system within the context of the global design and construction environment. These technologies ensure a platform independent environment and integration into the Internet. The use of *XML* (Extensible Mark-up Language) as a design language is explored to facilitate the storage of design data in a persistent and neutral manner independent from the software that originally created it. This ensures a long data life and the enables different actors over the life cycle of a facility to use their own relevant software to process the design information.

During the development of *AEDES* (Architectural Evaluation and Design System), the research team realised that the problem of structuring design knowledge in such a way to support relevant software systems across the life cycle of a facility is far more complex than originally anticipated. Although there are many similarities between the construction and the manufacturing industries, there are also significant and problematic differences. Architectural design tasks take place in an open world where the reasoner's knowledge is incomplete or inconsistent. Due to this the focus in computer-aided architectural design research has shifted back and forth from attempts to totally automate the entire design process to its partial support through drafting tools.

In an attempt to overcome some of the enormous complexities, that researchers struggled with over the past 35 years, a prototype intelligent autonomous design component *ARGOS* is developed in this research. It is clear that automated design methods are not tractable and it is therefore more worthwhile to pursue the creation of a neutral design language and the creation of intelligent and flexible design tools to manipulate these design fragments.

An in-depth study is made of various important out-of-industry manufacturing techniques, *CBR* and object technology and to establish clearly what the desirable characteristics of *ARGOS* should be. An important requirement is that *ARGOS* should be generic and non-prescriptive and should work in a *Microsoft Windows* compliant environment. A solution without the use of *CAD* is proposed that ensure a generic solution that could add value to many different construction industry actors in many different environments. More recently attempts are being made to introduce *post-modern* Artificial Intelligence (AI) into design and architecture. Despite all these efforts it is clear that architectural briefing and design has not reached the status of a science and it is unlikely ever to. This is confirmed by recent breakthroughs in the field of Artificial Intelligence (AI) and Knowledge Management that provide deeper insights into the cognitive processes of the designer.

This study indicates that *XML* is a viable means of expressing design knowledge and a feasible alternative for the complex Building Product Models currently proposed whilst at the same time supporting operations in the Internet environment. Design information and the ability to retrieve it is now more important than the software application that originally created it. The autonomous intelligent component *ARGOS* provides a method to encapsulate design knowledge at both tacit and explicit cognitive levels whilst at the same time providing global communication in a convenient desktop environment. *ARGOS* is designed in a parametric way that supports any design process that requires positional, volumetric and spatial relationship analysis in both 2D and 3D. Multiple autonomous copies can be placed in a container environment such as Excel. Any process written in any computer language that supports the use of ActiveX controls can be used to manipulate the *ARGOS* instances.

Ekserp

Hierdie verhandeling beskryf die ontwikkeling van 'n prototipe intelligente komponentstelsel met *Case-Based Reasoning* (CBR) vermoë. Dit word Argitektuur Objek Stelsel (ARGOS) genoem en maak die berging van lewenssiklus ontwerpinligting in kompakte gevalle op 'n mikrorekenaar moontlik. *CBR*-tegnieke word gekombineer met *Microsoft ActiveX* objektegnologie in die ontwerp van 'n outonome komponent wat sommige van die programmatuurbehoeftes in die globale ontwerp- en konstruksie-omgewing kan bevredig. Die tegnologieë verseker 'n platform-onafhanklike uitvoering en gerieflike integrasie in die Internet. *XML* (Extensible Mark-up Language) word as 'n ontwerptaal gebruik wat die berging van ontwerpinligting op 'n standhoudende en neutrale wyse moontlik maak, ongeag die programmatuur wat dit oorspronklik geskep het. Dit verseker 'n lang dataleeftyd en laat verskillende gebruikers oor die lewenssiklus van die fasiliteit relevante programmatuur aanwend om die ontwerpinligting te verwerk.

Gedurende die ontwikkeling van *AEDES* (Architectural Evaluation and Design System), het die span ontdek dat die strukturering van ontwerpinligting, op so 'n wyse dat dit vir programmatuurstelsels oor die lewenssiklus van 'n fasiliteit bruikbaar is, aansienlik komplekser is as aanvanklik vermoed. Alhoewel daar heelwat ooreenkomste tussen die konstruksiebedryf en die vervaardigingsindustrie bestaan is daar ook betekenisvolle en problematiese verskille. Argitektuurontwerp vind plaas in 'n oop wêreld waar die ontwerper (denker) se kennis onvolledig of inkonsekwent is. Derhalwe het die fokus in rekenaar-gesteunde argiteksontwerpnavorsing tussen die uiterstes van totale outomatisasie tot gedeeltelike ondersteuning deur tekenstelsels gewissel.

In 'n poging om sommige van die enorme kompleksiteite die hoof te bied waarmee talle navorsers oor die afgelope 35 jaar geworstel het, is die outonome ontwerpkomponent *ARGOS* ontwikkel. Dit is duidelik dat geoutomatiseerde ontwerpmetodes nie haalbaar is nie en dat dit dus die moeite werd om eerder die daarstelling van 'n gerieflike en neutrale ontwerptaal en skep van intelligente en aanpasbare elektroniese ontwerpgereedskap na te strewe wat die betrokke ontwerpinligting kan gebruik.

'n Omvattende studie word gemaak van verskeie belangrike vervaardigingsindustrie tegnieke, *CBR* en objektegnologie buite die domein van argitektuur om die wenslike karakterestieke van *ARGOS* te bepaal. Een van die belangrikste vereistes is dat *ARGOS* nie-voorskriftelik en in 'n *Microsoft Windows* aanpasbare omgewing ontplooibaar moet wees. 'n Generiese oplossing sonder die gebruik van *CAD* word voorgestel sodat dit kan waarde toevoeg tot stelsels wat deur verskillende gebruikers in 'n wye verskeidenheid van omgewings gebruik word. Tans word verskeie pogings aangewend om *post-moderne Kunsmatige Intelligensie* (KI) in ontwerp en argitektuur toe te pas. Desondanks al hierdie pogings is dit duidelik dat argitektuuropdraggewing en ontwerp nog nie die status van 'n wetenskap bereik het nie, en waarskynlik nooit sal bereik nie. Dit word bevestig deur deurbrake in KI en kennisbestuur wat diepere insigte in die kognitiewe vermoëns van die kreatiewe ontwerper aan die lig gebring het.

Die studie toon aan dat *XML* 'n lewensvatbare taal is om ontwerpinligting te struktureer en 'n alternatief vir die komplekse *Gebou Produk Modelle* (Building Product Models) is wat op die oomblik voorgestel word. Ontwerpinligting en die vermoë om dit te herwin het nou belangriker geword as die programmatuur wat dit oorspronklik geskep het. XML ondersteun ook die Internet. *ARGOS* het metodes om ontwerpinligting van beide stilswyende en eksplisiete kognitiewe aard te verpak. Globale kommunikasie is nou moontlik vanaf die mikrorekenaar. *ARGOS* het 'n parametriese ontwerp wat ontwerpprosesse in posisie, volume en ruimtelike verwantskaps ontledings in beide 2D en 3D ondersteun. Veelvuldige outonome

kopieë kan in ‘n houeromgewing soos *Microsoft Excel* geplaas word. Enige proses in enige rekenaartaal wat *ActiveX* objekte ondersteun kan *ARGOS* objekte manipuleer.

Acknowledgements

I would like to acknowledge the valued contributions and support of the following persons and organisations:

- *Prof. Dieter Holm* for his enthusiasm, guidance and support throughout the project;
- *Dr. Ben van Vliet* for his expert guidance and insights on the QFD process;
- *Prof. Craig Zimring*, *Janet Kolodner*, *Charles Eastman*, *Ashok Goel*, and *Marin Simina* at the Georgia Institute of Technology for expert guidance and insights into Case-Based Reasoning and the cognitive aspects of design;
- *Kirstin Küsel* for her support, enthusiasm and persistence in the attempts to understand the open world of architectural design better;
- My relatives and friends for their moral support and encouragement;
- My wife Christa for her love and endurance that has greatly contributed to the success of the work presented here.

This thesis is dedicated to my father, who encouraged me to follow a career in the sciences and set an exceptionally high standard for me to follow in dedicated service to his family and South African citizens at large over many years.

Contents

List of figures

List of tables

Definition of terms

AEDES
An acronym for Architectural Evaluation and Design System. This was an early attempt to structure design data during the briefing and design phases to assist with knowledge management across the life cycle of a facility (Conradie *et al*. 1999).

ARGOS
An acronym for Architectural General Object System, a Case-Based Reasoning enabled ActiveX intelligent component that can be used in Microsoft compliant container environments such as Microsoft Excel, Word, Access, Visio and Arena.

Artificial Intelligence (AI)
In the past definitions such as the following were used:
Luger and Stubblefield defined AI as the branch of computer science that is concerned with the automation of intelligent behaviour (Riesbeck 1996:373).
Minsky defined AI as the field of research concerned with making machines do things that people consider require intelligence (Riesbeck 1996:373).
Charniak and McDermott define AI as the study of mental faculties through the use of computational models (Riesbeck 1996:373).

The definition that is used in the present study is the one of Riesbeck (1996:374) for *Post-Modern AI*. AI is the search for answers to the eternal question why computers are so stupid. In *Post-Modern AI*, the AI becomes an invisible part of the overall system.

Building Product Model (BPM)
A BPR is a digital information structure of the objects making up a building, capturing the form, behaviour and relations of the parts and assemblies within the building. A BPR is potentially a richer representation than any set of drawings and can be implemented in multiple ways, including as an ASCII file or as a database (Eastman 1999).

Blackboard –Based Architecture
A Case-Based Reasoning architecture that offers flexible, opportunistic control capabilities. A blackboard architecture separates control knowledge from the domain knowledge contained in the knowledge sources (Rissland *et al*. 1991:77-78).

Case
A case is a contextualized piece of knowledge representing an experience that teaches a lesson fundamental to achieving the goals of the reasoner (Kolodner *et al*. 1996:36).

Case-Based Reasoning (CBR)
CBR solves new problems by adapting solutions that were used to solve old problems. The intuition of CBR is that situations recur with regularity. What was done in one situation is likely to be applicable in a similar situation. If we know what worked in a previous situation similar to the new one, we start with that in reasoning about the new situation (Riesbeck *et al*. 1989:25; Kolodner 1993:8).

Concept Selection
Concept selection is the emergence and selection of the best and strongest concepts with respect to customer needs and other criteria. Although creativity is essential throughout the entire product development process, concept selection reduces the number of alternatives under consideration. Concept selection is one of the most critical and difficult problems in design (Pugh 1996:167).

Constraint
In order to carry out some design activity, certain information must be available. In addition certain conditions, states or evaluations may apply to the data.

Critic
A critic is a piece of software that fires under certain circumstances to alert of possible design conflicts such as a fuel store that is right next to an operating theatre.

Frame
A frame is a case-like entity that records relationships between parts of a proposed solution but is more abstract than a case itself. Framing a problem generally means choosing some set of its specifications to concentrate on and deriving a framework that becomes more refined over time (Kolodner 1993:523).

Fuzzy sets
Bellman and Zadeh (1970) and Bojadziev *et al.* (1995:113) describe fuzzy sets as a special class of object in which there is no sharp boundary between those objects that belong to the class and those that do not.

Intelligent Component
In this view the problem of AI is to describe and build components that reduce the stupidity of the systems in which they function.

Knowledge Management (KM)
Knowledge management (KM), as defined by the GartnerGroup, is a discipline with new processes and technologies that differentiate it from information management. New technologies are required to capture knowledge that was previously tacit. Tacit knowledge is embodied in the minds and expertise of individuals. Once captured, knowledge must be shared to leverage its value and reused in similar situations and contexts.

Object-oriented design
According to Meyer (1988) Object-oriented design is the method which leads to software architectures based on the objects every system or subsystem manipulates rather than the function it is meant to ensure. Object-oriented design is also the construction of software systems as structured collections of abstract data type implementations (Meyer 1988).

Open World
An open world denotes any problem-solving situation in which the reasoner's knowledge is incomplete or inconsistent (Hinrichs 1991:5).

Quality Function Deployment (QFD)
QFD is a method for structured product planning and development that enables a development team to specify clearly the customer's wants and needs and then to evaluate each proposed product or service capability systematically in terms of its impact on meeting those needs (Cohen 1995:11).

Scalable Vector Graphics (SVG)
A working draft of 29 June 2000 of the W3C defines the features and syntax for Scalable Vector Graphics (SVG), a language for describing two-dimensional vector and mixed vector/ raster graphics in XML. SVG is a language for describing two-dimensional graphics in XML. SVG allows for vector graphic shapes (paths consisting of straight lines and curves), images and text. Graphical objects can be grouped, styled, transformed and composited into previously rendered objects. The feature set includes nested transformations, clipping paths, alpha masks, filter effects and template objects.

Systems Engineering (SE)
An interdisciplinary approach and means to enable the realisation of successful systems. It focuses on defining customer needs and required functionality early in the development cycle, documenting requirements, then proceeding with design synthesis and system validation while considering the complete problem.

Vector Markup Language (VML)
Microsoft developed their own XML application for vector graphics called VML. VML is more finished than SVG and is already supported by Internet Explorer 5.0 and Microsoft Office 2000. VML is not as ambitious as SVG and leaves out advanced features such as clipping and masking.

XML
XML was developed by an XML Working Group (originally known as the SGML Editorial Review Board) formed under the auspices of the World Wide Web Consortium (W3C) in 1996. XML is a set of rules for defining semantic tags that break a document into parts and identify the different parts of the document. It is also a meta-markup language that defines a syntax used to define other domain-specific, semantic structured mark-up languages (Harold 1999:3).

List of abbreviations

ADE	Application Development Environments
AEDES	Architectural Evaluation and Design System
AI	Artificial Intelligence
ARGOS	Architectural General Object System
BEARS	Building Environmental Assessment and Rating System for South Africa
BMMS	Building Maintenance Management System
BOMSIG	Business Object Model Special Interest Group
BPM	Building Product Model
CASE	Computer-Aided Software Engineering
CBD	Case-Based Design
CBR	Case-Based Reasoning
CBT	Computer-Based Training and Teaching
CE	Concurrent engineering (CE)
CKO	Chief Knowledge Officer
COM	Component Object Model
CONSENS	Concurrent Simultaneous Engineering System
CORBA	Common Object Request Broker Architecture
CPDM	Common Product Data Model
CSS	Cascading Style Sheets
DBMS	Database Management Systems
DCE	Distributed Computing Environment Group
DCOM	Distributed Component Object Model
DTD	Document Type Definition
DXF	Data Interchange Format
EQFD	Enhanced QFD
FFE	Fuzzy Front End
FM	Facilities Management
FMEA	Failure Mode and Effects Analysis
GUID	Global Unique Identifier
HOQ	House of Quality
HTML	Hypertext Mark-up Language
IAI	International Alliance for Interoperability
IDC	International Data Corporation
IGES	Initial Graphics Exchange Specification
IR	Information Retrieval
IS	Information Science
ISO-STEP	International Standards Organisation – Standard for the Exchange of Product model data
KA	Knowledge Architect
KBCAAD	Knowledge Based Computer-aided Architectural Design
KBDS	Knowledge-based Design Systems
KBS	Knowledge Based System
KE	Kansei Engineering
KE	Knowledge Engineering
KES	Kansei Engineering System
KM	Knowledge Management
KMS	Knowledge Management System
LTM	Long Term Memory
MBR	Model-based Reasoning
MIT	Massachusetts Institute of Technology
MOP	Memory Organisation Packet
NGM	Next Generation Manufacturing Company

NLP	Natural Language Processing
ODB	Object-Oriented Database or Object Database
ODBC	Open Database Connectivity
ODL	Object Description Language
OE	Operational Expense
OID	Object Identifier
OLAP	On-line Analytical Processing
OLE	Object Linking and Embedding
OMG	Object Management Group
OOCAD	Object-Oriented Computer Aided Design
OOL	Object-Oriented programming languages
ORB	Object Request Brokers
PDES	Product Data Exchange using STEP
PDM	Product Data Modelling
PREMIS	Professional Real Estate Management Information System
PROCAP	Procedural Guide for Clients, Architects and Other Professionals
QA	Quality Assurance
QC	Quality Control
QFD	Quality Function Deployment
RBR	Rule-Based Reasoning
ROI	Return on investment
SCM	Service Control Manager
SD	Semantic Differential
SE	Systems Engineering
SGML	Standard Generalised Mark-up Language
SME	Subject Matter Experts
SQC	Statistical Quality Control
SQL	Structured Query Language
SVG	Scalable Vector Graphics
TOC	Theory of Constraints
TOP	Thematic Organisational Packet
TQM	Total Quality Management
TQM	Total Quality Movement
UDE	Undesirable Effects
UIF	Universal Index Frame
UR	User Requirement
VBA	Visual Basic for Applications
VE	Value Engineering
VML	Vector Mark-up Language
VOC	Voice of Customer
VR	Virtual Reality
VRML	Virtual Reality Mark-up Language
W3C	World Wide Web Consortium
WM	Working Memory
www	world wide web
XML	Extensible Mark-up Language
XSL	Extensible Style language