

University of Pretoria etd

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 Markup 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 lewensiklus ontwerpinligting in kompakte gevalle op 'n mikrorekenaar moontlik. *CBR*-tegnieke word gekombineer met *Microsoft ActiveX* objektetegnologie in die ontwerp van 'n outonome komponent wat sommige van die programmatuurbehoefes 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 dataleef tyd en laat verskillende gebruikers oor die lewensiklus van die fasiliteit relevante programmatuur aanwend om die ontwerpinligting te verwerk.

Gedurende die ontwikkeling van *AEDDES* (Architectural Evaluation and Design System), het die span ontdek dat die strukturering van ontwerpinligting, op so 'n wyse dat dit vir programmatuurstelsels oor die lewensiklus van 'n fasiliteit bruikbaar is, aansienlik kompleks 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 rekenaargesteunde argitektoniese ontwerpnavorsing 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 is eerder die daarstelling van 'n gerieflike en neutrale ontwerptaal en skep van intelligente en aanpasbare elektroniese ontwerp gereedskap na te strewen wat die betrokke ontwerpinligting kan gebruik.

'n Omvattende studie word gemaak van verskeie belangrike vervaardigingsindustrie tegnieke, *CBR* en objektetegnologie buite die domein van argitektuur om die wenslike karakteristieke 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 argitektuuroopdragging en ontwerp nog nie die status van 'n wetenskap bereik het nie, en waarskynlik nooit sal bereik nie. Dit word bevestig deur die brake in KI en kennisbestuur wat dieper 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 rekenartaal 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

Abstract	2
Ekserp	3
Acknowledgements	5
Contents	6
List of figures	11
List of tables	12
Definition of terms	13
List of abbreviations	16
Chapter 1: Introduction and overview	19
Chapter 2: Motivation	23
2.1 Problem statement.....	23
2.2 Sub-problems	23
2.3 Bounds and constraints	24
2.4 Research method.....	24
Chapter 3: Review of literature	26
Introduction	26
3.1 Knowledge management	29
3.1.1 Introduction	29
3.1.2 The nature of knowledge.....	30
3.1.2.1 <i>Socialisation</i>	32
3.1.2.2 <i>Externalization</i>	32
3.1.2.3 <i>Combination</i>	32
3.1.2.4 <i>Internalization</i>	32
3.1.3 The current situation.....	33
3.1.3.1 <i>Desirable emerging technologies to enable knowledge management</i>	34
3.1.4 Knowledge management architectures.....	36
3.1.4.1 <i>Hypertext based systems</i>	39
3.1.4.2 <i>Search engines such as Alta Vista Discovery</i>	40
3.1.4.3 <i>Essential elements of a knowledge management architecture</i>	41
3.1.5 Microsoft's approach to KM	43
3.1.5.1 <i>Messaging and collaboration</i>	44
3.1.5.2 <i>Complete Intranet</i>	44
3.1.5.3 <i>Communities, teams and experts</i>	45
3.1.5.4 <i>Portals and search</i>	45
3.1.5.5 <i>Content management</i>	46
3.1.5.6 <i>Real-time collaboration</i>	46
3.2 Knowledge based design.....	48
3.2.1 Introduction	48
3.2.2 Artificial intelligence and design	50
3.2.2.1 <i>Life cycle enabled design ontology</i>	53
3.2.3 Problem-solving architectures.....	56
3.2.3.1 <i>Top-down strategies</i>	57
3.2.3.2 <i>Bottom-up strategies</i>	57
3.2.3.3 <i>Middle-out strategies</i>	58
3.2.4 Case-based design	59

3.2.4.1 Introduction	59
3.2.4.2 Advantages of a Case-Based Reasoner?	61
3.2.4.3 The disadvantages and caveats of Case-Based Reasoning	62
3.2.4.4 Case-based Reasoning compared with other methods	63
3.2.4.5 Types of Case-Based Reasoners.....	64
3.2.4.6 Generic models	65
3.2.4.7 Associative models	66
3.2.4.8 Exemplar models.....	66
3.2.4.9 The design precedent	67
3.2.5 Case-based Reasoning indexing and retrieval	68
3.2.5.1 Introduction	68
3.2.5.2 The indexing problem	68
3.2.5.3 Choosing an indexing vocabulary.....	69
3.2.5.4 Methods for index selection	70
3.2.5.5 Retrieving cases from the case library	72
3.2.5.6 The use of fuzzy sets for case indexing.....	74
3.2.5.7 The use of fuzzy sets to formulate dynamic linguistic variables for case retrieval.....	76
3.2.5.8 Fuzzy set linguistic modifiers.....	78
3.2.6 Conclusion.....	83
3.3 The systems view of the world.....	84
3.3.1 Introduction	84
3.3.2 What is manufacturing?	84
3.3.2.1 Manufacturing capability.....	86
3.3.2.2 Manufacturing processes	86
3.3.2.3 Low-quantity production.....	86
3.3.3 Concurrent engineering (CE)	87
3.3.3.1 Strategies for concurrent engineering.....	88
3.3.3.2 Concurrent engineering enabling technologies	90
3.3.3.3 Flow management.....	93
3.3.3.4 Theory of Constraints (TOC).....	96
3.3.4 Taguchi techniques for quality engineering	97
3.3.4.1 The meaning of quality.....	97
3.3.4.2 Taguchi loss function	98
3.3.5 The Fuzzy Front End (FFE)	99
3.4 Objects.....	101
3.4.1 Introduction	101
3.4.2 Origins of the object approach	101
3.4.2 Why is the use of objects advisable.....	102
3.4.3 Object-oriented programming	103
3.4.3.1 Encapsulation	103
3.4.3.2 Objects	104
3.4.3.3 Class	106
3.4.4 The model approach to architectural design.....	106
3.4.5 Frameworks for object components	107
3.4.5.1 Aims of object components.....	108
3.4.5.2 Technical components.....	108
3.4.5.3 Compound documents.....	108
3.4.5.4 Business components	109
3.4.6 OLE/ COM from Microsoft	109
3.4.6.1 Persistence of objects.....	110
3.4.6.2 Data exchange	110
3.4.6.3 Enabling relationships between documents	110
3.4.6.4 In-place activation	110
3.4.6.5 The object-component model	111
3.4.6.6 Support for distributed objects.....	111
3.4.6.7 OLE/ COM basic services.....	111
3.4.6.8 The main OLE interfaces	113
3.5 Kansei engineering and new product development.....	114
3.5.1 Introduction	114
3.5.2 What is Kansei Engineering (KE)	114
3.5.3 Types of Kansei Engineering	115

3.5.3.1 Type 1: Category Classification.....	116
3.5.3.2 Type 2: Kansei Engineering Computer System (KES).....	116
3.5.3.3 Type 3: Kansei Engineering Modelling.....	117
3.5.3.4 Type 4: Hybrid Kansei Engineering.....	118
3.5.3.5 Type 5: Virtual Kansei Engineering.....	118
3.5.4 Main Kansei Engineering steps.....	118
3.5.5 The Semantic Differential Method.....	119
3.5.5 Conclusion.....	120
3.6 Quality Function Deployment (QFD).....	121
3.6.1 Introduction.....	121
3.6.2 What is QFD?.....	122
3.6.3 The affinity diagram.....	124
3.6.4 Kano's model of user satisfaction.....	129
3.6.4.1 Dissatisfiers.....	129
3.6.4.2 Satisfiers.....	129
3.6.4.3 Delighters.....	129
3.6.5 QFD software.....	130
3.7 Theory of inventive problem solving (TRIZ).....	134
3.7.1 Introduction.....	134
3.7.2 TRIZ.....	134
3.7.3 Steps in using TRIZ.....	135
3.7.3.1 Formulate the problem: the prism of TRIZ.....	135
3.7.3.2 Search for previously well-solved problems.....	135
3.7.3.3 Look for analogous solutions and adapt to solution.....	136
3.7.3.4 Socially responsible TRIZ.....	136
Summary.....	136
Chapter 4: Precedents to the present research.....	139
Introduction.....	139
4.1 The PREMIS Facilities Management System.....	139
4.1.1 Introduction.....	139
4.1.2 Intrinsic design principles.....	139
4.1.3 A typical application.....	140
4.1.4 Critique of PREMIS.....	141
4.2 The AEDES prototype system.....	142
4.2.1 Integrated life cycle process.....	144
4.2.1.1 The characteristics.....	144
4.2.1.2 Evaluation during the process.....	144
4.2.2 Concurrent multimedia environment.....	144
4.2.2.1 The need for multimedia in the architectural profession.....	144
4.2.2.2 Multimedia in AEDES.....	144
4.2.3 Life cycle requirement validation.....	145
4.2.3.1 Multi-media QFD.....	145
4.2.3.2 Break-out tools.....	147
4.2.4 Ad-hoc queries and reports.....	148
4.2.4.1 Electronic traceability.....	148
4.2.4.2 Object manipulation.....	149
4.2.4.3 Flexible queries and reporting.....	149
4.2.4.4 Implicit linking technique.....	150
4.2.5 Major components.....	150
4.2.5.1 Relational database.....	150
4.2.5.2 Software shell.....	151
4.2.5.3 Database forms.....	151
4.2.5.4 Help system.....	151
4.2.5.5 QFD diagram software.....	151
4.2.5.6 Starter kit packaging.....	151
4.2.5.7 Materials database.....	152
4.2.6 Conclusion.....	152

4.2.7 Critique of AEDES.....	152
Summary	153
Chapter 5: Aims of ARGOS	154
Introduction	154
5.1 Concept selection	155
5.1.1 Introduction.....	155
5.1.2 Conceptual vulnerability	156
5.1.3 Overview of the method.....	156
5.1.4 Concept screening	159
5.1.4.1 <i>Select possible solutions to the particular design problem</i>	159
5.1.4.2 <i>Prepare the evaluation matrix</i>	159
5.1.4.3 <i>Rate the concepts</i>	159
5.1.4.4 <i>Rank the concepts</i>	160
5.1.4.5 <i>Combine and improve the concepts</i>	160
5.1.4.6 <i>Select one or more concepts</i>	160
5.1.4.7 <i>Reflect on the results and the process</i>	160
5.1.5 Concept scoring.....	161
5.1.5.1 <i>Prepare the selection matrix</i>	161
5.1.5.2 <i>Rate the concepts</i>	162
5.1.5.3 <i>Rank the concepts</i>	162
5.1.5.4 <i>Combine and improve the concepts</i>	162
5.1.5.5 <i>Select one or more concepts</i>	163
5.1.5.6 <i>Reflect on the results and the process</i>	163
5.1.5.7 <i>Some important factors</i>	163
5.1.6 Enhanced QFD and concept selection.....	164
Summary	164
Chapter 6: Implementation details.....	166
Introduction	166
7.3 Life cycle Information infrastructure	166
6.1.1 Introduction.....	166
6.1.2 XML as a design language	167
6.2 Packaging and retrieval of design knowledge.....	180
6.2.1 Introduction.....	180
6.2.2 Constraints.....	182
6.2.3 The design of the ARGOS intelligent component.....	185
6.2.4 Classification and knowledge organisation in a packaged environment.....	187
6.2.5 The co-existence of ARGOS with other software.....	189
6.2.5.1 <i>Concept selection</i>	190
6.2.5.2 <i>Spreadsheets</i>	190
6.2.5.3 <i>Computer languages</i>	190
6.2.5.4 <i>Process analysis</i>	191
6.2.6 The design of the ARGOS object.....	191
6.3 World Wide Web Implementation	193
6.4 Hypothetical use of ARGOS	194
6.5 Empirical response tests.....	199
Summary	201
Chapter 7: Summary, Conclusions, Recommendations and Assessment	203
Introduction	203
7.1 Summary	203

7.1.1 Out-of-industry methodologies	203
7.1.2 Life cycle design knowledge.....	204
7.1.3 ARGOS intelligent component	204
7.2 Conclusions	205
7.3 Recommendations for further work.....	206
7.4 Assessment	207
References.....	208
Appendix A: Implicit linking in PREMIS	214
Appendix B: PREMIS search criteria definition.....	216
Appendix C: Interface an ActiveX control to an Excel spreadsheet....	217
Appendix D: XSL stylesheet to convert XML into VML for web page display.....	218
Appendix E: Visual Basic code to implement a minimal web browser	222
Appendix F: Visual Basic code to implement ARGOS intelligent component.....	224

List of figures

Figure 1: Ability to influence system characteristics (Sparrius 1998: 1.1).....	20
Figure 2: Product innovation methodologies (Collated by author).....	26
Figure 3: The classical view of knowledge hierarchies (Author).....	31
Figure 4: The knowledge cycle and process. See text. (GartnerGroup 1998:2).....	32
Figure 5: Concept extraction for representation (GartnerGroup 1998:8).....	35
Figure 6: Visualisation of representation (GartnerGroup 1998:9).....	36
Figure 7: A typical Alta Vista Discovery screen (Author).....	40
Figure 8: Knowledge management architecture (GartnerGroup 1998:3).....	41
Figure 9: Knowledge Management technology model (GartnerGroup 1998:4).....	42
Figure 10: Knowledge Management vendors (Based on GartnerGroup 1998:16).....	43
Figure 11: The modules of a Knowledge Management evolution (Leibmann 1999:7).....	44
Figure 12: Typical hierarchical relational database structures used in a Facilities Management system (Author).....	55
Figure 13: IAI, Industry Foundation Classes Release 2.0 Object Hierarchy (Author).....	56
Figure 14: Case-Based Reasoning compared to concept selection (Collated by author from Kolodner (1993:18), Ulrich et al. (1995) and Pugh (1996)).....	63
Figure 15: Terms of the linguistic variable age in a building context (Author).....	77
Figure 16: Two ways to define manufacturing, a technical or an economic process (Groover 1996:3).....	85
Figure 17: Decoupling of time, cost and quality by means of Concurrent Engineering (Berndes 1996).....	88
Figure 18: Strategies for concurrent engineering (PSI-strategy) (Berndes 1996).....	89
Figure 19: Different flow types in a process (Author).....	94
Figure 20: Throughput in a manufacturing process (Author, based on Goldratt 1993:207).....	95
Figure 21: Costs associated with greenhouse film (Ross 1988).....	99
Figure 22: Using a Global Unique Identifier (GUID) to link graphic objects to other data (Author).....	105
Figure 23: The Kansei engineering process (Nagamachi 1999).....	115
Figure 24: The translation of Kansei into physical car traits (Nagamachi 1999).....	116
Figure 25: Type 2: Kansei Engineering Computer System (KES) (Nagamachi 1999).....	117
Figure 26: Components of a hybrid Kansei Engineering System (Nagamachi 1999).....	118
Figure 27: Adjectives applicable to coffee cups when using the semantic differential method (Nagamachi 1999).....	120
Figure 28: Schematic representation of the QFD House of Quality (Cohen 1995:12).....	123
Figure 29: Kano's customer satisfaction diagram (Cohen 1995:37).....	130
Figure 30: QFD/Capture product planning matrix screen (Author).....	131
Figure 31: Relational database tables used in the AEDES prototype QFD software (Author).....	132
Figure 32: Typical screen of the AEDES prototype QFD software (Author).....	133
Figure 33: PREMIS Facilities Management Software Components (Author).....	139
Figure 34: The life cycle phases of a building (Author).....	142
Figure 35: AEDES QFD Process (Conradie and Küsel 1999:24).....	143
Figure 36: Three tiered collaborative data structure (Conradie and Küsel 1999:26).....	145
Figure 37: A typical AEDES screen with multi-media information (Author).....	147
Figure 38: AEDES software components (Author).....	150
Figure 39: Concept A, Oracle form with CAD in OLE container and Visual Basic attribute reader (Author).....	157
Figure 40: Concept B, ActiveX control based starter kit (Author).....	157
Figure 41: Concept C, ActiveX control based starter kit (Author).....	157
Figure 42: Concept D, ActiveX control based starter kit (Author).....	158
Figure 43: Typical starter kit drawing as used in the design of the AEDES prototype used as the starter kit reference concept (Author).....	158
Figure 44: The concept screening matrix for the concepts A to D (Author, based on Pugh 1996; Ulrich et al. 1995:114).....	161
Figure 45: The concept scoring matrix for the concepts B, C and D (Author, based on Pugh 1996; Ulrich et al. 1995:117).....	163
Figure 46: Structured Planning/ Design Knowledge Delivery (Author).....	178
Figure 47: Display of CAD drawing in XML format by means of VML (Author).....	180
Figure 48: Taxonomy of constraint types (Hinrichs 1991:99).....	185
Figure 49: ARGOS object in 2D mode (Author).....	186
Figure 50: ARGOS object in 3D mode (Author).....	186
Figure 51: The relationship between the ARGOS, ActiveX design object and the applications software (Author).....	189
Figure 52: The relationship of the ARGOS object to other intelligent data sources (Author).....	191
Figure 53: Design of 16 bed male/ female/ paediatric in-patients section step 1 (Author).....	195
Figure 54: Design of 16 bed male/ female/ paediatric in-patients section step 2 (Author).....	196
Figure 55: Setting design properties of a paediatric ward (Author).....	197
Figure 56: The calculation of area and volume (Author).....	198
Figure 57: The reduction of volume by lowering the ceiling (Author).....	198
Figure 58: The retrieval and insertion of a Four Bed Ward detailed case (Author).....	199
Figure 59: ARGOS component response (Author).....	200
Figure 60: ARGOS blackboard size (Author).....	201

List of tables

<i>Table 1: Stages of technological knowledge (Bohn 1997:77).....</i>	<i>30</i>
<i>Table 2: Current and emerging multiple standards that form a barrier to responsive NGM information systems (NGM 1997).....</i>	<i>37</i>
<i>Table 3: The main requirements for a Knowledge Management enabling environment (Collated by author)</i>	<i>38</i>
<i>Table 4: A comparison between Case-Based, Rule-Based and Model-based Reasoning (Collated by author)</i>	<i>64</i>
<i>Table 5: Methods of obtaining Voice of Customer (Collated by author)</i>	<i>125</i>
<i>Table 6: Essential user requirements extracted for the AEDES VOC exercise.....</i>	<i>127</i>
<i>Table 7 : Sample of form used to extract constant sum paired comparisons from users.....</i>	<i>127</i>
<i>Table 8: Relative importance of user requirements within group.....</i>	<i>128</i>

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