



The South African Institute of Computer Scientists  
and  
Information Technologists

# **Proceedings**

**of the**

**1996 National Research and  
Development Conference**

**Industry meets Academia**

Interaction Conference Centre, University of Natal,  
Durban .  
26 & 27 September

**Edited by  
Vevek Ram**

©1996 Copyrights reside with the original authors who may be contacted directly

**ISBN 0-620-20568-7**

Cover printed by Natal Printers (Pty) Ltd, Pietermaritzburg

Copying by the Multicopy Centre, University of Natal, Pietermaritzburg

Binding by Library Technical Services, University of Natal, Pietermaritzburg

The views expressed in this book are those of the individual authors

## FOREWORD

This book is a collection of papers presented at the National Research and Development Conference of the Institute of Computer Scientists and Information Technologists, held on 26 & 27 September, at the Interaction Conference Centre, University of Natal, Durban. The Conference was organised by the Department of Computer Science and Information Systems of The University of Natal, Pietermaritzburg.

The papers contained herein range from serious technical research to work-in-progress reports of current research to industry and commercial practice and experience. It has been a difficult task maintaining an adequate and representative spread of interests and a high standard of scholarship at the same time. Nevertheless, the conference boasts a wide range of high quality papers. The program committee decided not only to accept papers that are publishable in their present form, but also papers which reflect this potential in order to encourage young researchers and to involve practitioners from commerce and industry.

The organisers would like to thank IBM South Africa for their generous sponsorship and all the members of the organising and program committees, and the referees for making the conference a success. The organisers are indebted to the Computer Society of South Africa (Natal Chapter) for promoting the conference among its members and also to the staff and management of the Interaction Conference Centre for their contribution to the success of the conference.

On behalf of the Organising Committee

Vevek Ram

Editor and Program Chair

Pietermaritzburg, September 1996

## **Organising Committee**

### **Conference General Chairs**

Mr Rob Dempster and Prof Peter Warren (UNP)

### **Organising Chair**

Dr Don Petkov (UNP)

### **Secretariat**

Mrs Jenny Wilson

### **Program Chair**

Prof Vevek Ram (UNP)

### **Program Committee**

Prof Peter Wentworth, Rhodes  
Dr Milan Hajek, UDW  
Prof Derek Smith, UCT  
Prof Anthony Krzesinski, Stellenbosch  
Dr Don Petkov, UNP  
Mr Rob Dempster, UNP  
Prof Peter Warren, UNP

# Table of Contents

Foreword	i
Organising Committee	ii
List of Contributors	vi
<b>Keynote Speaker</b>	
<i>The Role of Formalism in Engineering Interactive Systems</i> M D Harrison and D J Duke	1
<b>Plenary</b>	
<i>Industry-Academic-Government Cooperation to boost Technological Innovation and People Development in South Africa</i> Tjaart J Van Der Walt	15
<i>Checklist support for ISO 9001 audits of Software Quality Management Systems</i> A J Walker	17
<i>The IS Workers, they are a-changin'</i> Derek Smith	29
<b>Research</b>	
<i>Examination Timetabling</i> E Parkinson and P R Warren	35
<i>Generating Compilers from Formal Semantics</i> H Venter	43
<i>Efficient State-exploration</i> J. Geldenhuys	63
<i>A Validation Model of the VMTP Transport Level Protocol</i> H.N. Roux and P.J.A. de Villiers	75
<b>Intelligent Systems</b>	
<i>Automated Network Management using Artificial Intelligence</i> M Watenboeck	87
<i>A framework for executing multiple computational intelligent programs using a computational network</i> H L Viktor and I Cloete	89
<i>A Script-Based prototype for Dynamic Deadlock Avoidance</i> C N Blewett and G J Erwin	95
<i>Parallelism: an effective Genetic Programming implementation on low-powered Mathematica workstations</i> H. Suleman and M. Hajek	107
<i>Feature Extraction Preprocessors in Neural Networks for Image Recognition</i> D Moodley and V Ram	113

## **Real-Time Systems**

*The real-time control system model - an Holistic Approach to System Design* 119  
T Considine

*Neural networks for process parameter identification and assisted controller tuning for control loops* 127  
M McLeod and VB Bajic

*Reference Model for the Process Control Domain of Application* 137  
N Dhevcharran, A L Steenkamp and V Ram

## **Database Systems**

*The Pearl Algorithm as a method to extract information out of a database* 145  
J W Kruger

*Theory meets Practice: Using Smith's Normalization in Complex Systems* 151  
A van der Merwe and W Labuschagne

*A Comparison on Transaction Management Schemes in Multidatabase Systems* 159  
K Renaud and P Kotze

## **Education**

*Computer-based applications for engineering education* 171  
A C Hansen and P W L Lyne

*Software Engineering Development Methodologies applied to Computer-Aided Instruction* 179  
R de Villiers and P Kotze

*COBIE: A Cobol Integrated Environment* 187  
N Pillay

*The Design and Usage of a new Southern African Information Systems Textbook* 195  
G J Erwin and C N Blewett

*Teaching a first course in Compilers with a simple Compiler Construction Toolkit* 211  
G Ganchev

*Teaching Turing Machines: Luxury or Necessity?* 219  
Y Velinov

## **Practice and Experience**

*Lessons learnt from using C++ and the Object Oriented Approach to Software Development* 227  
R Mazhindu-Shumba

*Parallel hierarchical algorithm for identification of large-scale industrial systems* 235  
B Jankovic and VB Bajic

## **Information Technology and Organizational Issues**

<i>A cultural perspective on IT/End user relationships</i> A C Leonard	243
<i>Information Security Management: The Second Generation</i> R Von Solms	257
<i>Project Management in Practice</i> M le Roux	267
<i>A Case-Study of Internet Publishing</i> A Morris	271
<i>The Role of IT in Business Process Reengineering</i> C Blewett, J Cansfield and L Gibson	285

## **Abstracts**

<i>On Total Systems Intervention as a Systemic Framework for the Organisation of the Model Base of a Decision Support Systems Generator</i> D Petkov and O Petkova	299
<i>Modular Neural Networks Subroutines for Knowledge Extraction</i> A Vahed and I Cloete	300
<i>Low-Cost Medical Records System: A Model</i> O A Daini and T Seipone	301
<i>A Methodology for Integrating Legacy Systems with the Client/Server Environment</i> M Redelinghuys and A L Steenkamp	302
<i>Information Systems Outsourcing and Organisational Structure</i> M Hart and Kvavatzandis	303
<i>The relational organisation model</i> B Laauwen	304
<i>The Practical Application of a New Class of Non-Linear Smoothers for Digital Image Processing</i> E Cloete	305
<i>A Technology Reference Model for Client/Server Software Development</i> R C Nienaber	306
<i>The Feasibility Problem in the Simplex Algorithm</i> T G Scott, J M Hattingh and T Steyn	307
<b>Author Index</b>	309

## List of Contributors

**Vladimir B Bajic**  
Centre for Engineering Research,  
Technikon Natal,  
P O Box 953  
Durban 4000

**C N Blewett**  
Department of Accounting  
University of Natal  
King George V Avenue  
Durban 4001

**Justin Cansfield**  
Department of Accounting  
University of Natal  
King George V Avenue  
Durban 4001

**Tom Considine**  
Apron Services (Pty) Ltd  
P O Johannesburg  
International Airport  
1600

**Eric Cloete**  
School of Electrical Engineering  
Cape Technikon  
Box 652  
Cape Town

**I Cloete**  
Computer Science Department  
University of Stellenbosch  
Stellenbosch  
7600

**O A Daini**  
Department of Computer Science  
University of Botswana  
Gaborone  
Botswana

**Nirvani Devcharan**  
Umgeni Water  
Box 9  
Pietermaritzburg  
3200

**P J A de Villiers**  
Department of Computer Science  
University of Stellenbosch  
Stellenbosch  
7700

**Ruth de Villiers**  
Department of Computer Science and  
Information Systems  
UNISA  
Box 392, Pretoria, 0001

**G J Erwin**  
Business Information Systems  
University of Durban-Westville  
Private Bag X54001  
Durban 4000

**G Ganchev**  
Computer Science Department  
University of Botswana  
PBag 0022  
Gaberone, Botswana

**J Geldenhuys**  
Department of Computer Science  
University of Stellenbosch  
Stellenbosch  
7700

**Louise Gibson**  
BIS, Dept Accounting & Finance  
University of Durban  
Pvt Bag X10  
Dalbridge 4014

**Mike Hart**  
Department of Information Systems  
University of Cape Town  
Rondebosch  
7700

**M. Hajek**  
Department of Computer Science  
University of Durban-Westville  
Pvt Bag X54001  
Durban 4000

**A C Hansen**  
Dept of Agricultural Engineering  
University of Natal  
Private Bag X01  
Scottsville 3209

**J M Hattingh**  
Department of Computer Science  
Potchefstroom University for CHE  
Potchefstroom 2520



**Boris Jankovic**  
Centre for Engineering Research  
Technikon Natal  
P O Box 953,  
Durban 4000

**Paula Kotze**  
Department of Computer Science and  
Information Systems  
UNISA  
Box 392  
Pretoria, 0001

**J W Kruger**  
Vista University  
Soweto Campus  
Box 359  
Westhoven 2124

**A C Leonard**  
Dept of Informatics  
University of Pretoria  
Pretoria  
2000

**Ben Laauwen**  
Laauwen and Associates  
P O Box 13773  
Sinoville  
0129

**Mari Le Roux**  
Information technology, development: project  
leader  
Telkom IT 1015  
Box 2753  
Pretoria 0001

**P W L Lyne**  
Dept of Agricultural Engineering  
University of Natal  
Private Bag X01  
Scottsville 3209

**Rose Mazhindu-Shumba**  
Computer Science Department  
University of Zimbabwe  
Box MP167  
Harare, Zimbabwe

**Meredith McLeod**  
Centre for Engineering Research,  
Technikon Natal,  
P O Box 953  
Durban 4000

**D Moodley**  
Computer Management Systems  
Box 451  
Umhlanga Rocks  
4320

**Andrew Morris**  
P O Box 34200  
Rhodes Gift  
7707

**R C Nienaber**  
Technikon Pretoria  
Dept of Information Technology  
Private Bag X680  
Pretoria 0001

**E Parkinson**  
Department of Computer Science  
University of Port Elizabeth  
Box 1600  
Port Elizabeth 6000

**Don Petkov**  
Department of Computer Science and  
Information Systems  
University of Natal  
PBag x01  
Scottsville 3209

**Olga Petkov**  
Technikon Natal  
Box 11078  
Dorpspruit 3206  
Pietermaritzburg

**N Pillay**  
Technikon Natal  
Box 11078  
Dorpspruit 3206  
Pietermaritzburg

**V Ram**  
Department of Computer Science and  
Information Systems  
University of Natal  
PBag x01  
Scottsville 3209

**Melinda Redelinghuys**  
Department of Computer Science and  
Information Systems  
UNISA  
Box 392  
Pretoria, 0001

**Karen Renaud**  
Computer Science and Information Systems  
UNISA  
Box 392  
Pretoria, 0001

**H N Roux**  
Department of Computer Science  
University of Stellenbosch  
Stellenbosch  
7700

**T G Scott**  
Department of Computer Science  
Potchefstroom University for CHE  
Potchefstroom  
2520

**T Seipone**  
Department of Computer Science  
University of Botswana  
Gaborone  
Botswana

**Derek Smith**  
Department of Information Systems  
University of Cape Town  
Rondebosch  
7700

**Anette L Steenkamp**  
Department of Computer Science and  
Information Systems  
UNISA  
Box 392  
Pretoria, 0001

**T Steyn**  
Department of Computer Science  
Potchefstroom University for CHE  
Potchefstroom 2520

**H. Suleman**  
Department of Computer Science  
University of Durban-Westville  
Pvt Bag X54001  
Durban 4000

**A Vahed**  
Department of Computer Science  
University of Western Cape  
Private Bag X17  
Bellville 7530

**A Van der Merwe**  
Computer science and Informations Systems  
UNISA  
P O Box 392  
Pretoria,0001

**Tjaart J Van Der Walt**  
Foundation for Research and Development  
Box 2600  
Pretoria, 0001

**K Vavatzandis**  
Department of Information Systems  
University of Cape Town  
Rondebosch  
7700

**Y Velinov**  
Dept Computer Science  
University of Natal  
Private Bag X01  
Scottsville 3209

**H Venter**  
Department of Computer Science  
University of Port Elizabeth  
Box 1600  
Port Elizabeth 6000

**H L Viktor**  
Computer Science Department  
University of Stellenbosch  
Stellenbosch  
7600

**R Von Solms**  
Department of Information Technology  
Port Elizabeth Technikon  
Private Bag X6011  
Port Elizabeth 6000

**A J Walker**  
Software Engineering Applications  
Laboratory  
Electrical Engineering  
University of Witwatersrand  
Johannesburg

**Max Watzenboeck**  
University of Botswana  
Private Bag 0022  
Gaberone  
Botswana

**P Warren**  
Computer Science Department  
University of Natal  
P/Bag X01  
Scottsville 3209

# A REFERENCE MODEL FOR THE PROCESS CONTROL DOMAIN OF APPLICATION

Nirvani Dhevcharran, Prof. A L Steenkamp and Dr V Ram

## Abstract

The increasing complexity of large process control systems impedes efforts to construct, co-ordinate and monitor these systems effectively. The object oriented paradigm and associated mechanisms greatly assist in software development in such domains. A reference model suitable for the process control domain and embodying object orientation, is described.

## Introduction

The development of software systems is a complex undertaking, involving both technical and managerial considerations. In the case of large and complex application domains, development projects are conducted by following a team approach. The diverse perspectives of team participants call for a frame of reference which facilitates understanding of the specific aspects of software development.

A project for researching all relevant aspects when developing software systems, the Object Oriented Information Systems Engineering Environment (ISEE), has been formulated by Steenkamp[1995]. A general framework of four reference models are investigated: Development Process Reference Model, Quality Reference Model, Technology Reference Model and Target System Reference Model.

Process control deals with the technical, economic and safety dimensions of applications such as those found in chemical, oil and gas refineries and municipal water and sewage treatment. Each of these industries has data to be acquired, data to be disseminated, decisions to be processed, communications to be performed and reports to be generated. The challenge is to develop a control system that addresses the tremendous but unique complexity of each application without reinventing a thousand new wheels each and every time as described by Beam[1993].

According to Brown [1992], a reference model is a conceptual and functional framework which helps experts to describe and compare systems. It allows experts to work productively and independently on the development of standards for each part of the reference model. A reference model is thus not a standard itself; it should not be used as an implementation specification, nor as the basis for the conformance of actual implementations.

This paper concentrates on developing a reference model for the process control domain of application within the context of The Target System Reference model. The following aspects have been identified for the reference model: Environment, Information, Software Engineering and Systems Engineering.

The environment aspect : The Environment Aspect occupies the highest level within any methodology. For process control systems this aspect considers the structure of the process control environment in terms of its interface to the real world as well as its interface to physical devices.

The systems engineering aspect : The main focus of this aspect is on those engineering principles relevant at the operating and networking level. There are a number of principles involved in the process control domain, viz., distribution transparency, dependability, performance optimisation, scaling and language integration.

The software engineering aspect : This aspect expresses the target system in terms of the interacting object metaphor. The intra-object structure, inter-object relationships, the functionality of the objects

and the dynamic behaviour of objects are addressed. More specifically, the timeliness issue, the dynamic internal structure, the reactivity issue, the concurrency issue and the distributed issue of the process control domain are described. Further, formalisms are required to model the structure of the system, of the data and of control.

The information aspect : The manner in which information is used within the process control domain is modelled. Typically, information has to be represented, manipulated and interpreted. The representation of data in this domain is normally in a real time database. This data has to be logically and temporally consistent. The latter arises from the need to preserve the temporal validity of data items that reflect the state of the environment that is being controlled by the system.

The process control domain is intrinsically very complex. It operates in real-time with a number of processes operating concurrently. The Object Oriented (OO) paradigm for software development supports the development of more flexible, more easily maintainable and less error-prone process control systems.

By making optimal use of the powerful OO features and further by establishing a reference model for the process control environment, a structured framework for software development is created.

### Reference Model for the Process Control Environment

Figure 1 represents the process control meta model conceptually in terms of the relevant meta primitives for each of the four aspects that have been identified.

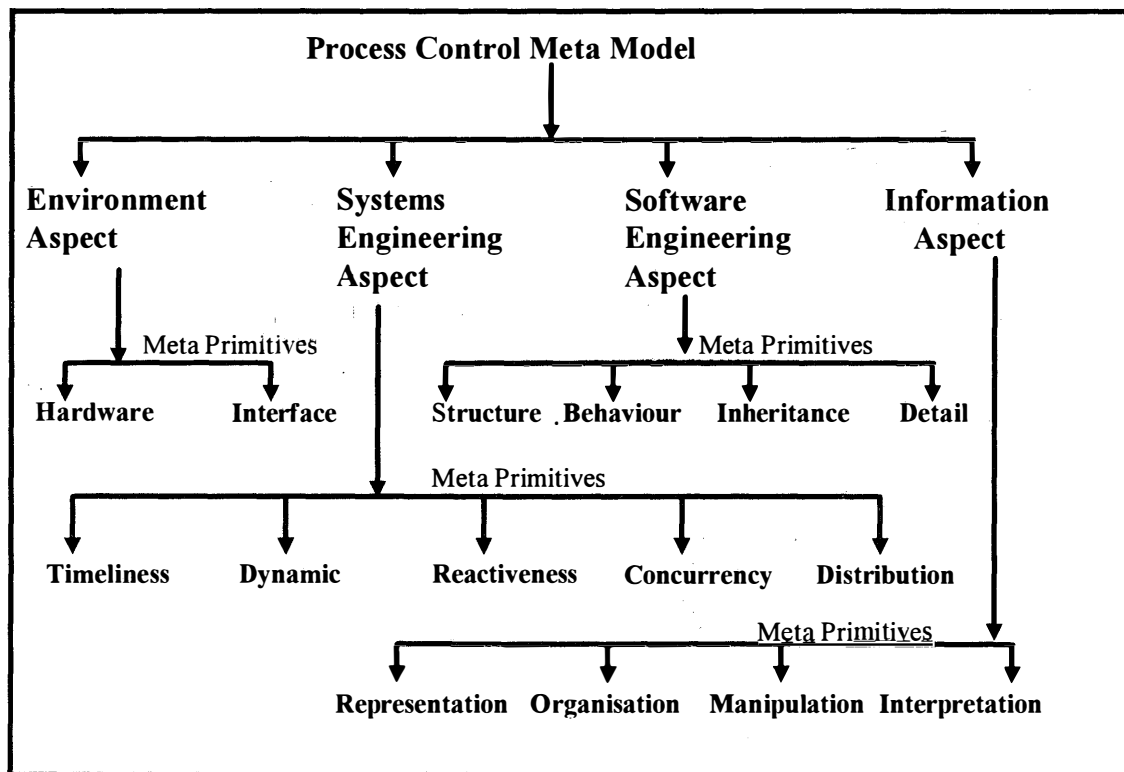


Figure 1. Process Control Meta Model

### Meta Primitives of the Environment Aspect

Careful consideration must be given to configuring the environment within which a process control target system will reside in (hardware considerations) as well as to its interfaces (interface considerations) to the real-world. It is necessary to focus on the environment because it forms the basis for constructing a model of the behaviour of a system. The meta primitives are depicted in Figure 2.

Hardware considerations : Process control applications operate in real-time. Time is a critical factor in these systems and there are a few hardware issues that can affect the response/timing of an event, e.g. event scheduling and synchronising of clocks. A difficult problem is the determination of tight timing information on instructions and code sequences for contemporary computers; pipelining, caching and a host of other performance-enhancing features seen to hinder timing predictability. By selecting appropriate granularities for the higher level language elements, these and other hardware issues, such as exactly how and where to incorporate worst-case effects of memory and bus contention, can be handled practically.

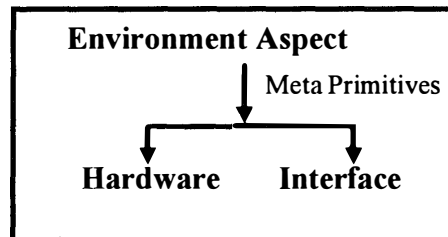


Figure 2. Environment Meta Model

Considerations : The interfaces to a process control application can be either humans or devices. An example of a device is a PLC. There is continuous interaction between the application and the PLC which the system must be capable of handling. Simultaneously, one or more operators may be attempting to access the application. There is therefore, a need for the application to be capable of handling all the PLC as well as the operators transactions, i.e., need for concurrency. The interface between the operator and the system is on-line and interactive. Hence, the response time must be fast. If the same data is being accessed/updated by both the PLC and the operator, the access/update must be synchronised. If there is no synchronisation there is a potential for inconsistent data to be stored on the system.

### Meta Primitives of the Systems Engineering Aspect

This aspect focuses on those engineering principles relevant at the operating system and networking level [Steenkamp1995]. The space/time trade-off, representing a particular set of options in the form of parameters, must be determined.

The salient properties which form the meta primitives, depicted in Figure 3, characterising the systems engineering aspect of process control systems are:

Timeliness of function : This is the most common attribute of real time systems of any kind. By definition, a real time system is required to perform its function “on time”, whatever that happens to mean in a particular context.

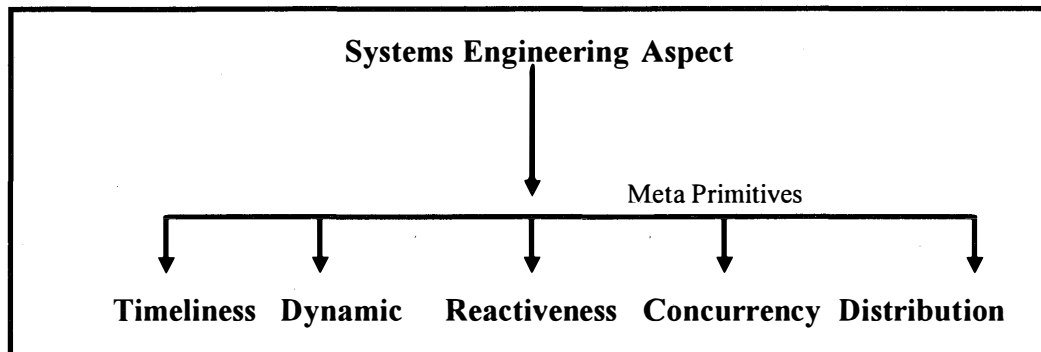
Dynamic internal structure : Many real time systems are required to exercise control over an environment whose properties vary with time. This requires that the system components dealing with

the particular aspects of the environment must be dynamically reconfigured to match the dynamic of external environments. Because of limited resources (memory, processor capacity), this typically entails the dynamic creation and destruction of software components.

**Reactiveness :** A reactive system is one that is continuously responding to different events whose order and time of occurrence are not always predictable.

**Concurrency :** This is a feature of the real world in which a real time system is embedded. At any given time, multiple simultaneous activities can be taking place in the real-time system. When this is combined with the need for real time response, the usual result is that the real-time system itself must be concurrent.

**Distribution :** A distributed computing system is one in which multiple computing sites co-operatively achieve some common function. Distribution is either inherent ( as is the case for communications systems) or it may be driven by the need to increase throughput, availability or functionality [Selic1994].

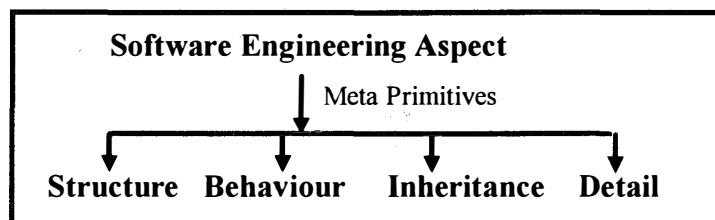


**Figure 3. Systems Engineering Meta Model**

#### **Meta Primitives of the Software Engineering Aspect**

This section is loosely based on Selic [Selic1994], who has developed a real-time OO modelling technique called ROOM.

This is the one of most important aspects within the process control domain. The success of a software system is highly dependent on the selection of the appropriate software engineering method and supporting techniques. The meta primitives of the software engineering aspect are depicted in Figure 4.

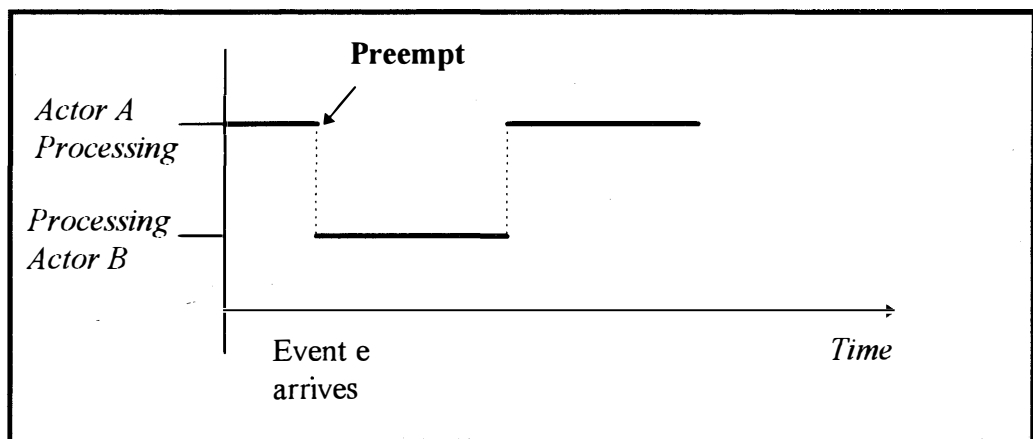


**Figure 4. Software Engineering Meta Model**

**High level structure modelling :** At this level the concept of reusability is fundamental to the OO paradigm. ROOM has adopted this reusability strategy at the high level structure dimension. A

special concept defined as an actor is its basic modelling concept. In order to achieve reusability, each actor has to have one specific purpose with well defined interfaces to other components. An actor can have more than one interface as it can communicate with different types of actors. With this structured architecture, actors can co-exist with other actors concurrently in its domain. The process control domain is composed of concurrent objects and it is this functionality which makes ROOM a suitable choice. An actor makes use of encapsulation to maintain its primary purpose. The reasons for using encapsulation is that it is necessary to ensure that the coupling between actors is restricted to only the interactions across the interfaces.

High level behaviour modelling : This domain is characterised by the presence of two complex and difficult phenomena, concurrency and distribution. An event which is basically a message that is related to time, i.e. it has temporal properties, is generated from outside the ROOM environment. There are two general approaches to the handing of events: run-to-completion and pre-emptive. ROOM has opted for the run-to-completion approach. The two approaches are depicted in Figure 5.



**Figure 5. Pre-emption and Run-to-Completion**

In order to achieve a short execution time of  $e_1$ , protracted event processing sequences can be broken up into a number of shorter chunks. However, this can significantly complicate the implementation.

In terms of priorities, the ideal would be to have one single priority. However with the distributed process control environment this is not possible. It is thus recommended to have some level of urgency to be provided. This would cater not only for the high priority tasks but also the low priority tasks. Frequently, the low priority tasks are disregarded during busy periods.

High level inheritance : ROOM views inheritance primarily as an abstraction mechanism that helps to deal with complexity by allowing detail to be introduced gradually. With the abstraction based approach, abstract classes play a pivotal role. The definition and preservation of the integrity of abstract classes is the primary concern in dealing with inheritance. Each abstract class should symbolise a well-defined abstraction with a clear meaning.

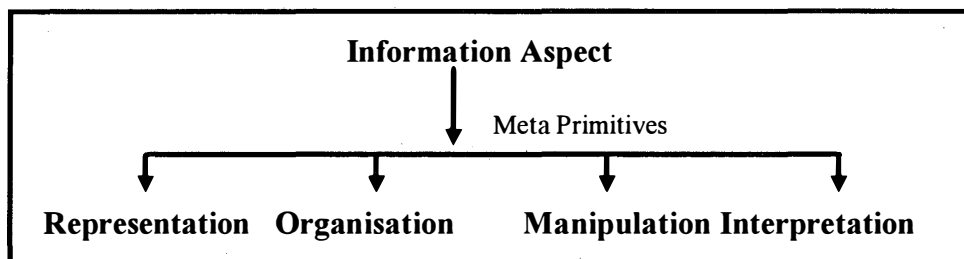
In ROOM, inheritance plays a fundamental role, since all designs are specified as classes. Three different class hierarchies are supported: the actor hierarchy, the protocol hierarchy and the data object hierarchy. The inheritance rules of data objects are determined by the Detail Level language used in modelling. Actor classes allow both high level structure and high level behaviour to be subclassed, providing for a much higher form of reuse than is available in traditional OO programming languages. Only single inheritance is supported for actor classes and exclusion and overriding of attributes is allowed [Selic1994].



The detail level : This section examines the issues and techniques used to specify the Detail Level aspects of a ROOM model. The detail level deals with fine grained actions and fine-grained objects. All elements of concurrency and high level architectural issues are filtered out and dealt with at higher abstraction levels. As a result the complexity of the specifications at this level is greatly reduced. Detail level actions capture the behaviour that occurs during transitions of a state machine from one state to another. Fine-grained objects are used either to capture the extended state of a state machine or as information units that can be transferred between actors [Selic1994].

### Meta Primitives of the Information Aspect

Process control systems utilise real time databases. A real time database system provides database features such as data independence and concurrency control while at the same time enforcing real-time constraints that applications may have. The meta primitives of the information aspect are depicted in Figure 5.



**Figure 5. Information Meta Model**

Representation of information : There are 3 key issues that affect the representation of information, data independence, binding and efficiency

There is generally a single globally accessible shared data area, whose organisation is defined by the data definition facilities of the chosen programming language; every task in the system that shares data is compiled with the definition of the entire shared area and thus has access to all the data at any time. This scheme has proven to be problematic. A key objective of data management technology is the achievement of data independence; that is each task should be unaware of the storage structure and access mechanism of the data accessed. If data independence is achieved, stored data can be re-organised with no impact on the code of each task. Data management technology achieves data independence by interposing one or more layers of system software between the application task and access to the data itself. In this way, a call is made to a system software task which then uses a data description to access the data item. These "additional" layers of code may introduce inefficiencies into the implementation, which causes problems in a process control system.

The efficiency and independence objectives are clearly in conflict. The two objectives can be studied together by introducing the concept of binding. In the case of a single globally shared data area whose definition is compiled into each process, the names are bound to locations at compile time or at link/load time. However, the use of a procedure to access a data item will delay the binding until run time. It is the execution of the binding at run time that causes the use of data management technology to be less efficient yet more independent [Ward1985].

Data organisation : There are two faces to data organisation implementation; the view that is presented to the designer and programmer and the actual physical organisation of the data on the chosen storage media. The designer and the programmer will want to perform operations on the data structures according to an external view of how the data is grouped and organised; this can be separated from the internal data structure chosen for the data. It is the business of any data

management software to provide mechanisms to manipulate the external data structure which can then be translated into operations on the internal data structure.

However support for real-time data base systems must take into account the following: firstly, not all data in a real-time database are permanent; some are temporal and secondly, since timeliness is sometimes more important than correctness, in some situations, precision can be traded for timeliness.

Manipulation of information : Process control transactions are characterised along three dimensions: the manner in which data is used by the transactions, the nature of the time constraints and the significance of executing a transaction by its deadline or more precisely, the consequence of missing specified time constraints.

Further temporal consistency requirements of the data can lead to some of the time constraints for the transaction.

Real-time database systems employ all three types of database transactions, i.e., write-only transactions, update transactions and read-only transactions. This classification can be used to tailor the appropriate concurrency control schemes.

Some transaction time constraints come from temporal consistency requirements and some arise from requirements imposed on system reaction time. The former can typically take the form of periodicity requirements: For example, every 5 seconds, sample the reservoir level.

System reaction requirements typically take the form of deadline constraints imposed on aperiodic transactions. For example, if  $\text{Water\_Flow} > 100$   
add Chlorine within 10 seconds.

In this case, the system's action in response to the high  $\text{Water\_Flow}$  must be completed by 10 seconds.

Transactions can also be distinguished based on the effect of missing a transaction's deadline. As discussed in the section on the Temporal paradigms the terms of hard, soft and firm transactions have been discussed. This categorisation shows the value imparted to the system when a transaction meets its deadline. The processing of real-time transactions must take their different characteristics into account.

Interpretation of information : In the process control domain, a real time database consists of a set of data objects representing the state of an external world controlled by a real-time system. The data objects are interpreted as two types: continuous and discrete. Continuous data objects are related to external objects continuously changing in time. The value of a continuous data object can be obtained directly from a sensor (image object) or computed from the values of a set of image data objects (derived object) within a regular period. Continuous data objects are related with the following additional attributes: a timestamp indicating when the current value of the data object was obtained, an absolute validity duration is the length of time during which the current value of the data object is considered valid and a relative validity duration is associated with a set of objects used to derive a new data object. Discrete data objects are static in the sense that their values do not become obsolete as time passes, but they are valid until update transactions change the values.

## Summary

The environment, information, software engineering and system engineering aspects cannot exist in isolation. There is a close relationship between them as their boundaries are not well defined and there is clearly an overlap between each aspect's discipline.

The Reference Model proposed is meant to serve as a guideline in the development of process control applications. Due to the applications' inherent complexity, it is necessary to have a frame of

reference that encompasses all the relevant issues in its development. Further, by using the object oriented paradigm, there is an increase in the reuse of software in the automation software area, especially if the same applications with probably even the same configuration are to be used at various plants.

## **References**

- Beam K: Object Technology Object-Oriented Programming. I/S Analyser, December 1993, Vol. 31, Iss. 12, pp. 1 - 15.
- Brown AW, Earl AN & McDermid JA: Software Engineering Environments. McGraw Hill, 1992.
- Selic B, Gullekson G & Ward PT: Real-Time Object Oriented Modelling. John Wiley & Sons, 1994.
- Steenkamp L: Object Oriented Information Systems Engineering Environment. May 1995, pp. 1-20.
- Ward PT & Mellor SJ: Structured Development for Real-Time Systems. Volume 1: Introduction and Tools, Yourdon Press, 1985.