Metaobjects as a programming tool

R.W. Lemke, M.Sc.

12526959

Thesis submitted for the degree Doctor of Philosophy in Information Technology at the Vaal Triangle Campus of the North-West University

Promoter: **Prof D.B. Jordaan**

November 2010

Acknowledgements

To Marthi, Gustav and Cia – without your love and sacrifice this would not have been possible!

To those who have gone before – Thank you. To those who follow, do not despair – there is an answer out there. Find it!

To Dawid, under whose auspices incoherent conjecture evolved into meaningful comprehension.

Summary

Computer applications can be described as largely rigid structures within which an information seeker must navigate in search of information - each screen, each transaction having underlying unique code. The larger the application, the higher the number of lines of code and the larger the size of the application executable.

This study suggests an alternative pattern based approach, an approach driven by the information seeker. This alternative approach makes use of value embedded in intelligent patterns to assemble rules and logic constituents, numerous patterns aggregating to form a 'virtual screen' based on the need of the information seeker. Once the information need is satisfied, the atomic rules and logic constituents dissipate and return to a base state. These same constituents are available, are reassembled and form the succeeding 'virtual screen' to satisfy the following request.

The pattern based approach makes use of multiple pattern 'instances' to deliver functionality - each pattern instance has a specific embedded value. Numbers of these patterns aggregate to drive the formation of a 'virtual screen' built using patterns, each pattern referencing and associating (physical) atomic logic and spatial constituents. This is analogous to painting a picture using removable dots. The dots can be used to describe a fish, and then, once appreciation has been completed, the image is destroyed and the dots are returned to the palette. These same dots can later be reapplied to present the picture of a dog, if that is requested by the information seeker. In both pictures the same 'dots' are applied and reused. The form of the fish and dog are retained as value embedded within the patterns, the dots are building blocks aligned using instructions within the patterns.

This group classifies current solutions within the 'Artefact-Pattern-Artefact' (APA) group and the pattern based approach within the 'Pattern-Artefact-Pattern' (PAP) group. An overview and characteristics of each are discussed.

The study concludes by presenting the results obtained when using a prototype developed using the PAP approach.

Opsomming

Rekenaartoepassings kan beskryf word as hoofsaaklik rigiede strukture waardeur ŉ inligtingsoeker moet navigeer opsoek na inligting. Elke skerm, elke verslag en elke transaksie het sy eie onderliggende kode. Hoe groter die toepassing, hoe groter die aantal lyne in die programkode en hoe groter is die uitvoerbare toepassingskode.

Hierdie studie stel ŉ alternatiewe patroon-gebaseerde benadering voor - ŉ benadering wat deur die inligtingsoeker beheer word. Hierdie alternatiewe benadering maak gebruik van waarde, inherent aan intelligente patrone, om reëls en logiese komponente saam te stel. Veelvoudige patrone word saamgevoeg om ŉ 'virtuele skerm' te lewer wat gebaseer is op die inligtingsoeker se behoeftes. Sodra die inligtingsoeker se behoefte na inligting bevredig is, verdwyn die atomiese reëls en die logiese komponente keer terug na ŉ basis toestand. Dieselfde komponente is weer beskikbaar om die daaropvolgende 'virtuele skerm' saam te stel om sodoende die volgende inligtingsversoek te bevredig.

Die patroon–gebaseerde benadering maak gebruik van veelvoudige patroon-voorkomste om funksionaliteit te verskaf. Elke patroon-voorkoms het ŉ spesifieke inherente waarde of betekenis. Verskeie van hierdie patrone word versamel om ŉ 'virtuele skerm' te vorm. Elke patroon besit spesifieke (fisiese) atomiese, logiese en ruimtelike komponente. ŉ Analogie is om ŉ prent te teken met verwyderbare kolle. Die kolle kan so geplaas word om ŉ vis te vorm. Sodra die waardering van die vis tot ŉ einde kom, word die prent vernietig en die kolle word teruggeplaas op die palet. Dieselfde kolle word dan weer gebruik om ŉ prent van ŉ hond te skep, indien ŉ hondebeeld verlang word. In albei gevalle word dieselfde kolle aangewend en hergebruik. Die vorm van die vis en die vorm van die hond word behou as waarde binne-in die 'patrone' - die kolle is boustene wat gerangskik word deur die instruksies in die patrone.

Hierdie studie klassifiseer huidige toepassings binne die 'Artefak-Patroon-Artefak' (APA) groep, en die patroon benadering hoort aan die 'Patroon-Artefak-Patroon' (PAP) groep. ŉ Oorsig en kenmerke van elkeen word bespreek.

Die studie sluit af met resultate wat verkry is van ŉ prototipe toepassing wat volgens die PAP benadering ontwikkel is.

Table of Contents

1. Introduction ... 1

1.1 Background ... 1

1.2 Problem statement ... 1

1.3 Main research question ... 2

1.4 Hypothesis ... 2

1.5 Method of investigation ... 3

1.6 Objectives - Contribution to the Field of IT ... 3

1.7 Conclusion ... 6

2. Literature Review ... 7

2.1 Technology, Software and Database Management Overview ... 7

2.2 From Data to Wisdom ... 12

3. Understanding Information ... 16

3.1 The continuum from Data to Wisdom Extended ... 16

3.2 Interaction between the Physical World and the Intellectual domain ... 20

3.3 Essence and Accident ... 21

3.4 Hierarchy of Understanding Expanded ... 22

3.5 Intellectual reflection ... 25

3.6 Characteristics of a hierarchical model ... 32

3.7 Characteristics of an intellectual model ... 32

3.8 Temporality and persistence within the physical world ... 32

3.9 Conceptual Model Introducing Intellectual Reflection....33

3.10 Expanding Essence and Accident....42

3.11 Migration from a hierarchical to a network model....47

3.12 The Network: Input-Process-Output (NIPO) Model - The Network Model....54

3.13 Sequential and parallel processing using Metaobjects....55

3.14 Emulating Intellectual Reflection....56

3.15 Constructs derived from the physical primitives....56

3.16 Ability to satisfy the demand for requirements....58

4. Research Findings and Conclusions....60

4.1 Methodology....60

4.2 Prototype Findings....61

4.3 Independent Assessment....64

4.4 Research Findings....64

4.5 Shortcomings....67

4.6 Future Research....67

5. References....69

List of Tables

TABLE 1: COMPARISON BETWEEN THE HIERARCHICAL AND NETWORK MODELS 57
TABLE 2: MEASURES IN THE ENTITY-RELATIONSHIP MODEL 61
TABLE 3: RESULTS FROM A METAOBJECT BASED APPLICATION 61

List of Figures

FIGURE 1: FROM DATA TO WISDOM (SOURCE: FLEMING, 1996) 13
FIGURE 2: HIERARCHY OF UNDERSTANDING 16
FIGURE 3: RELATIONSHIP BETWEEN ARTEFACTS AND PHYSICAL PATTERNS 19
FIGURE 4: INTELLECTUAL REFLECTION 20
FIGURE 5: HIERARCHY OF UNDERSTANDING EXPANDED 22
FIGURE 6: WISDOM, KNOWLEDGE AND INFORMATION WITHIN THE CONSTRUCTS OF TIME 23
FIGURE 7: FLEMING'S BRIDGE BETWEEN DATA AND WISDOM 23
FIGURE 8: EXTENDED BRIDGE BETWEEN DATA AND WISDOM 24
FIGURE 9: PHYSICAL WORLD 24
FIGURE 10: INTELLECTUAL DOMAIN 24
FIGURE 11: INTERACTION BETWEEN SENSES AND THE INTELLECTUAL DOMAIN 25
FIGURE 12: INTELLECTUAL REFLECTION 26
FIGURE 13: ARTEFACTS AND PHYSICAL PATTERNS 27
FIGURE 14: EMBEDDED VALUE AND PRIVATE PATTERNS 28
FIGURE 15: HIERARCHY OF UNDERSTANDING: PHYSICAL DOMAIN 29
FIGURE 16: HIERARCHY OF UNDERSTANDING: INTELLECTUAL DOMAIN 29
FIGURE 17: PHYSICAL WORLD AND THE INTELLECTUAL DOMAIN 29
FIGURE 18: EMULATING INTELLECTUAL REFLECTION 30
FIGURE 19: EQUATING 'BODY-MIND' TO 'DATA-EMBEDDED VALUE' 31
FIGURE 20: TRANSITIONING BETWEEN STATES USING A PATTERN 34
FIGURE 21: MEMORY, INFORMATION AND LOGIC IN THE INTELLECTUAL PROCESS 35
FIGURE 22: STRUCTURE, RULE AND LOGIC IN THE COMPUTING TRANSITIONING PROCESS 36
FIGURE 23: CONSOLIDATION OF INTELLECTUAL AND PHYSICAL MODELS 38
FIGURE 24: CONCEPTUAL MODEL ILLUSTRATING RULE DELIVERY 39
FIGURE 25: CONCEPTUAL MODEL ILLUSTRATING LOGIC DELIVERY 39
FIGURE 26: CONCEPTUAL MODEL ILLUSTRATING STRUCTURAL TRANSFORMATION 41
FIGURE 27: RELATIONSHIP BETWEEN ESSENCE AND ACCIDENT 43
FIGURE 28: TRANSITIONAL AND STABLE CONSTRUCTS 44
FIGURE 29: STATE MODEL AS A CONTINUUM 44
FIGURE 30: IRREDUCIBLE MINIMUM 45
FIGURE 31: TRANSITION AND STABILITY 45
FIGURE 32: HIERARCHICAL TRANSFORMATION 46
FIGURE 33: STATE MODEL WHICH CAN EMULATE INTELLECTUAL REFLECTION 47
FIGURE 34: ADAPTATION OF HIERARCHICAL RELATIONSHIPS 48
FIGURE 35: DIRECT PERSISTENCE TO INDIRECT PERSISTENCE 49
FIGURE 36: PRESENTATION AND POST-PRESENTATION 49
FIGURE 37: PROGRAM EXECUTION 50
FIGURE 38: PRESENTATION (METAOBJECT STUB CREATION) WITHOUT POST-PRESENTATION 51

FIGURE 39: PRESENTATION AND POST-PRESENTATION USING A BROKER 52
FIGURE 40: POST-PRESENTATION 53
FIGURE 41: NETWORK: INPUT-PROCESS-OUTPUT 54
FIGURE 42: SEQUENTIAL EXPRESSION 55
FIGURE 43: PARALLEL EXPRESSION 56
FIGURE 44: TRADITIONAL PROGRAMMING APPROACH 58
FIGURE 45: METAOBJECT BASED APPROACH 59
FIGURE 46: GUI TRANSACTION SCREEN 62
FIGURE 47: AN ENTITY-RELATIONSHIP MODEL 63

1. Introduction

1.1 Background

1.1.1 Long Term Need

Information and Communication Technology (ICT) has advanced significantly over the past 100 years. This era is characterized by advances which increased exponentially across a broadening frontier of discovery. Moore's law [33] introduced a mechanism enabling the progression of hardware to be predicted. This is not so with the development of software. Brooks [4] states '*I believe the hard part of building software to be the specification, design, and testing of this conceptual construct, not the labor of representing it and testing the fidelity of the representation'*. The need for software applications far outstrips the ability to deliver them. Added to this, the maintenance complexities required to ensure operational applications retain their integrity and remain sustainable places high overheads on the resources. The long term outlook for today's ICT software solutions indicates that sustainability will degrade for implemented solutions. There is a need to look beyond the current technologies to conceptualize solutions which can equal or better Moore's law [33], and satisfy emerging demand for software solutions. This must be attained without compromising the integrity and sustainability of the solution.

1.1.2 The differing dimensions of current solutions

The accidental components [8] within technology must be aligned and constructed to deliver information such that the intellectual information thirst, the need to master knowledge of the essences, is satisfied. Information Technology's challenge is to bridge this gap through the delivery of information.

1.2 Problem statement

Duplication of logic and rule contribute significantly to the development and maintenance effort of information delivery vehicles.

"*Development of the accidental aspects supporting information technology solutions today consume significantly more resources than that required to elucidate the essence of the requirement*" [4].

Accidental aspects can simplistically be described as the *means* required to achieve a particular end. The *end* if an intellectual appreciation forms the *essence* of the need. In this context, duplication within the accidental aspects plays no small role in consuming this effort, not only in duplicating development effort, but also exponentially increasing maintenance complexity.

1.3 Main research question

- What are the fundamental pillars underlying information delivery?

1.3.1 Secondary research questions

- Is an optimized information delivery solution using the fundamental pillars of information possible?
- How can these pillars be best aligned towards providing sustainable information delivery solutions while retaining integrity?
- Will these generic logic components provide a significant improvement in the ability to develop and maintain Information Delivery Solutions in terms of reducing cost, reducing time required and requiring lower levels of expertise for a given information requirement?

1.4 Hypothesis

Computer applications use programs for the application of logic, and databases for the retention of data content based on rules which provide continuity between events. Programs and databases have discrete roles - programs contain logic applied in real time during events processed by the Central Processing Unit, and databases provide persistence for the end result, stored in a dormant mode between events. Rules identifying the attributes and relationships of components are retained directly (manifested in the physical relationships, for example, attributes within an entity) or indirectly (as metadata). This study investigates whether logic can be reduced to absolute, autonomous, atomic components stored within a universally available algorithm library. It investigates whether universally unique, non-redundant logic components can be constituted and synchronized dynamically using metadata and manifested as a solution responding to an information need, rather than an information need needing to navigate within a rigid solution. It introduces the concept of ’Value Driven Information Delivery‘ (initially described as the Pattern-Artefact-Pattern model in [26]) where the essence of the information need harnesses technology, contrasted against

’Technology Driven Information Delivery‘ applications where solutions are rigidly encoded within monolithic technology silos. It extends the Hierarchical Input Process Output (HIPO) based approach [14] [21] through the introduction of Network Input Process Output (NIPO). Logic is introduced as an instrument of change, and Rule as a mechanism promoting persistence across generations. Logic and Rule are applied as intellectual instruments towards channelling structural progressions from one state to the next, both intellectually and in the physical world.

1.5 Method of investigation

1.5.1 Literature review

All relevant concepts will be examined and described by means of a literature review. The majority of the sources to be used will be obtained from text books, scientific journals and research documents which are scientifically verifiable.

1.6 Objectives - Contribution to the Field of IT

1.6.1 The need for ‘new stuff’

Clocksin [6] has suggested the following possible reasons, amongst others, why progress in meeting the demand for software solutions has been much slower than expected (addressing AI but also relevant in the broader sense). He has listed the following:

- resource limitations,
- complexity limitations,
- the impossibility of AI, and
- conceptual limitations.

Clocksin [6] says, with regard to conceptual limitations: “*There are several types of conceptual limitations, and several authors have attempted to explain such limitations by what Brooks [5] calls a ‘New Stuff’ hypothesis: that there may be some extra form of ‘New Stuff’ in intelligent systems outside our current scientific understanding. ‘New Stuff’ theories are prevalent in popular writings about the problem of consciousness and AI”*.

Conceptualising of what could possibly be ‘New Stuff’ starts at the basic foundation of human understanding. Energy and matter are fundamental building blocks found within the physical world, and all worldly derivatives are extensions of this. This introduces the questions around understanding persistence [28] and the concept of time [30].

Persistence [28] is the propagation of rules of spatiality across generations - attributes of current, existing generations persisted into the following generation cycle. Persistence is around propagating rules defining structures - in the physical world, structure is built using atoms (electrons, protons and neutrons), in the digital world, structures are built using bits which reflect a value (on / off) as the fundamental component. The concept of time [30] is suggested to be an intellectual construct, based on the capacity of memory, to remember and internalize and the ability to apply logic, to forecast and influence the future. These two capabilities (memory and logic) have enabled mankind to conceive the constructs of time, of past, present and future.

Further rationalisation suggests that time is a logical, intellectual understanding [30] and persistence is a manufactured construct [28]. Biological persistence (propagation across generations using the genetic code) is an example of an indirect persistence mechanism modelled using the principles of the Artefact-Pattern-Artefact cycle [26]. Intellectual persistence, where value is propagated from one intellectual generation to another via the physical world is a mirror image of the physical persistent model where physical reproduction carries attributes from one biological generation to the next. Intellectual persistence is modelled using the principles expressed in the Pattern-Artefact-Pattern cycle [26]. Pursuing this theme further begets the following question:

- What is 'rule', what is 'logic' and what is 'structure'?

Cursory inspection of 'rule' indicates that an imperative for 'rule' is persistence within spatial constructs - a dependence on a spatial (persistence) capability equating to the matter dimension. Cursory inspection of 'logic' identifies an ephemeral quality where logic is released (expressed) or encapsulated (e.g. captured in source code) during a 'temporal moment' [30] equating to the energy dimension. 'Structure' is a derivative of both 'logic' - where an intellectual meaning is expressed, and 'rule' which delivers the persistence required. Equating this hypothesis to the delivery of information begets the following question:

- How can information and value be delivered across the divide between the intellectual universe and the corporeal world?

Understanding the nature of information is a prerequisite before an information delivery solution can be conceived and delivered. To this end, the hierarchy of understanding is introduced and expanded. The hierarchy of understanding introduced in Figure 2, is suggested to be a bidirectional bridge bridging the divide between raw data and wisdom. At the top end of this hierarchy lies the concept of wisdom. At the opposite end, as physical and digital base components, artefacts and data are found. A definition of an intellectual understanding which bridges this divide is suggested - intellectual reflection. Paragraph 2.2 suggests an explanation and defines components which are found between data and wisdom. Chapter 3 expands this discussion by extending the continuum between data and wisdom. A summary of this continuum is diagrammatically illustrated in Figure 5: Hierarchy of Understanding Expanded. This figure contextualises the expanded hierarchy of understanding and then provides the springboard into an explanation of the properties underlying a 'Value Driven Information Delivery' solution. The objective of this thesis is to propose and expand an understanding of a 'Value Driven Information Delivery' solution using the Network Input Process Output (NIPO) diagram (Figure 41) to contextualise the approach and describe the primitives underlying information delivery using such a solution. This approach suggests two primary primitives (rule and logic) can be combined to present a third construct (structure) as follows:

- Rules describe the rules describing relationships around persistent components[1].
- Logic is the real time expression of intellectual intent - managing and orchestrating change.
- Structure can be described as a container within which intellectual expression is manifested, expressed and realized using rules prescribing persistence and with logic driving change.

---

[1] NOTE: In the physical world, rules are built around atomic structural constituents – atoms (in turn constituted from electrons, protons and neutrons). Persistent components at the finest granularity within the context of IT are bits. Constructs within the digital world are built using bits. Structures are defined, derived and presented using rules which relate one bit to the other, one group of bits to another – directly (contextually) or indirectly. Bytes, Words, Fields and Tables are all structures built using specific rules – bits being the 'material', the Bytes, Words, Fields and Tables delivering the 'structure' [26]. The expression of logic delivers the real-time re-arrangement of bits based on the rules.

1.7 Conclusion

Rule and Logic absolutes form the fundamental constituents underlying the delivery and consumption of information. Structure is an intellectual dimension which is used to order these absolutes. This study categorizes these fundamentals into an approach which introduces the Network Input Process Output (NIPO) model.

This approach can be used for the management of value embedded within electronic spatial constructs promoting the delivery of electronic patterns. These electronic patterns can be dynamically aligned to changing intellectual needs sodoing satisfying the intellectual thirst for information for a specific predefined domain. Criteria are suggested which may be used to evaluate the advantages and disadvantages of the hierarchical and network groupings. Finally these criteria are applied, and research findings presented.

2. Literature Review

The literature review consists of two parts. The first part (Paragraph 2.1) lists Information Technology advances in three fields - Technology, Software and Databases. These instruments form the accidental aspects within which value will be carried and through which it is exposed to the knowledge worker. The intellectual need, where information and wisdom are intellectual constructs, is where the essential nub of information is manifested. This is addressed in the second part (Paragraph 2.2). The first part has a common ambition to optimise the use of technology using data. The second part attempts to understand what information is, from an intellectual and technology agnostic perspective. By analogy the difference between data and information can be illustrated as follows, summarized in the observation by Maxwell. *"But if the mental cannot be understood scientifically, how is it to be understood? There is another kind of explanation, I shall argue, which may be called 'personalistic' explanation. This is an entirely respectable kind of explanation; it works, however, in a certain sense, in the opposite direction to scientific explanation. As things become increasingly personalistically intelligible, they become, roughly, increasingly scientifically unintelligible, and vice versa. As the contents of a conscious person's head come increasingly into focus scientifically, as a brain or physical system, inevitably the mental aspects seem to disappear; as the contents of the person's head come increasingly into focus personalistically, as a mind, so the brain, the neurons, the physical system seem to disappear. The key to solving this important part of the philosophical or conceptual mind–body problem is to recognize a dualism, not of kinds of entity, but of kinds of explanation" [32]*.

These two camps represent on the one hand the 'Information Technology' neuro-surgeon working with the equivalent of a physical brain - these are bits and bytes, Central Processing Unit's and Gigabytes, and on the other, the 'Knowledge Management' psychiatrist working with the intellect using amorphous constructs - information and knowledge - the foundational components of a knowledge worker. Both aspects need to be addressed when considering the use of metaobjects as a programming tool.

2.1 Technology, Software and Database Management Overview

2.1.1 Technology Overview

Ranyard, in his presidential address to the Operational Research Society [35] eloquently provides a history of computing and Operational Research (OR). This address lists the progress of OR and the milestones reached and passed. The focus of the address revolves primarily around the progress of hardware, from the world's first general-purpose computer - the Difference Engine, designed by Charles Babbage in the 1820's. He proceeds to list further milestones, including ENIAC completed in 1946 and EDVAC completed in 1951. IBM started introducing scientific computers in 1956. In the 1960's a step improvement was reached with the introduction of transistors heralding the start of the mainframe era. The 1970's witnessed the introduction of Minicomputers and a fall in purchase prices. First generation Microcomputers were introduced at the start of the 1980's bringing a computing capability to individual users and sodoing "*computing could be taken directly to the client without the involvement of another party"*. The advent of the IBM PC introduced a *de facto* second generation microcomputer standard into the industry. The introduction of PC 'clones' introduced more cost effective PC solutions, and a host of derivatives evolved delivering the multitude of computing devices found on the market today.

2.1.2 Software Overview

Software development followed in the footsteps of hardware. Quintas [34] in a resume of software development notes that two principal problems have long been with us: Firstly that the productivity of software development has increased at a slower rate than that of hardware; and secondly software systems are delivered which are of a poor quality.

Frederick Brooks observed that the feared "monster" of the disastrous software project leads to *"…desperate cries for a silver bullet - something to make software costs drop as rapidly as computer hardware costs do*". However, *"…Not only are there no silver bullets now in view, the very nature of software makes it unlikely that there will be any - no inventions that will do for software productivity, reliability, and simplicity what electronics, transistors, and large-scale integration did for computer hardware*" [4].

Quintas [34] goes on to present the history of languages. In the paper, he suggests that the initial focus of software was "*to write programs for optimal efficiency in machine use and saving memory"* using binary code. Progression from machine language to assembler to third generation languages followed, releasing programmers from the need to have a detailed knowledge of the internals of a particular computer technology. 4GL languages allowed users with no programming knowledge the ability to describe their requirements using

”*vocabularies and syntax that are very similar to natural language*”. He goes on to introduce the role of declarative languages and the use of CASE, and the use of CASE tools to generate code. Two streams within Software Engineering are discussed, spawned from the original debate around the definition of software engineering. ”*A quarter of a century on from the eponymous NATO conference the phrase software engineering remains a repository for a number of differing and often competing interpretations as to what an engineering approach to software development should consist of. The two attributes of engineering mentioned in the above quote - theoretical foundations and practical disciplines - were and indeed still are reflected in rather different interpretations as to what constitutes software engineering. The first centres on the idea that an engineering approach must be founded on a scientific and mathematical base; the second that engineering is essentially a practical activity which needs to develop better tools to help solve practical problems and build systems.*“ Quintas suggests that ”*the most interesting and significant current technical innovation concerns object-oriented (OO) design and OO programming*” is because of the potential for reuse ”*objects can be reused in new developments*”.

Objects can be reused - the further challenge is to now detach logic which is locked within discrete objects and allow this detached logic to be reused. Further discussion will introduce the concept of value - this is the ultimate reusable component from a logical perspective. Persisting value requires that mechanisms which can address the needs for indirect persistence [28] are delivered. Proposing such a mechanism is the prime focus of this thesis.

2.1.3 Database Management Overview

The brief resumes of the history of technology evolution and programming languages cannot be complete without an overview of the evolution of the various database management approaches. The management of the persistent components encapsulating data content and structure (and their relationships) falls within the science of data management. Jim Gray [18] has identified six generations of data management:

- Zero’th generation, extending from the initial record management (4000BC - 1900AD).
- First Generation: Record Managers 1900 - 1955.
- Second Generation: Programmed Unit Record Equipment 1955 - 1970.
- Third Generation: Online Network Databases 1965 - 1980.

- Fourth Generation: Relational Databases and client-server computing 1980 - 1995.
- Fifth Generation: Multimedia Databases 1995 onwards.

Databases can be regarded as repositories containing persistent material containing rules describing structure (metadata content) and storing raw business content. The static state is retained therein. Change to these artefacts is orchestrated through events where the status of the base entry state is transformed to an amended status - this transformation takes place as algorithms express logic which consumes and sequentially transforms data from a base state to a transformed state.

2.1.4 Structure, Logic and Rule

The technology overview in 2.1.1 provided a glimpse into the platforms - the physical structures - on which the applications are loaded, and the software overview in 2.1.2 identified how logic has been channelled, starting from machine language and progressing to the Object Oriented Programming and Aspect Oriented Programming languages of today. Together with the overview of database management in section 2.1.3 these three approaches introduce the pillars within which information is both internalised and expressed. These three pillars are structure, logic and rule.

2.1.4.1 Structure - physical, digital and intellectual

The hardware and the underlying operating system and networking capabilities provide a physical and digital milieu where applications are loaded and executed. This structure is an environment which enables knowledge workers to capture and access information based on preordained requirements within which digitised logic and rule interact. The digitised structures and patterns, when exposed to the user, are transformed into meaning. Together, the physical environment and the digital domain support the delivery and capture of digital patterns containing value to the information seeker. This is achieved by combining logic and rule to deliver structures of meaning (intellectual constructs – a mind picture or ‘thought object’ [22] and memory maps) to the user. Within the physical structures, digitised structures are created and amended - these digital structures ultimately deliver the digital patterns which are exposed to the user as patterns within which the business value is ensconced. Business value is intellectually examined and information extracted driven by an intellectual appreciation. The information is then persisted as memory within the individual.

2.1.4.2 Logic

Programs contain the logic required to transform the persistent components retained within the database from one state containing value to a second state reflecting amended value. Logic is a vital capacity expressed within a temporal moment [30]. Logic is expressed in real time as a expression of directed energy. Logic cannot exist in its pure form, but needs to be 'persisted' in some manner. This is done by encapsulating this 'vital force' into source code files. The source code is also a set of rules expressed by sequentially ordering syntax built upon operators and operands and then persisted on a database. In the case of code, this is more correctly called a library - in essence a container within which the code files (source, object, executable, class, binary, dll's are examples) are stored. Logic can neither be recorded nor manifested without spatial change taking place - digital or physical.

2.1.4.3 Rule

Databases retain the persistent components required for the delivery of information which include both the content and the rules for the content (metadata) retained either directly or indirectly. Rules describing the spatial components and their relationships are retained as metadata. Logic rules are also retained - not in a database per se, but within libraries and directories containing executables and dynamic linked libraries. The delivery of information is dependent on rules being available which describe the persistent components of the spatial dimensions, and rules (algorithmic) within which logic is encapsulated - similar dimensions with differing essential purposes. Rules can only exist where there is persistence.

2.1.4.4 Atomicity of logic [29]

In this discussion, atomicity is applied to logic components. Atomicity of logic is attained when a logic component is absolute having been reduced to the finest atomicity, an irreducible minimum - it cannot be refined further. In a sense, the logic aspects of programming are normalized. "*Two derivations represent the same proof if and only if they are equivalent*" [12].

2.1.4.5 Granularity of rule [29]

In this discussion, granularity is applied when developing rules. The finest granularity of rule is attained once a rule becomes absolute and cannot be refined further. Rules describe the spatial relationships between persistent artefacts. Codd [7] introduced the principles of normalisation, used when designing structured databases. These rules have made a significant contribution towards reducing structural duplication across data elements found within formal databases.

2.1.5 Evolution of hardware and software

The evolution of hardware and software continues. Predictions on the ability to increase the delivery of software solutions and to maintain the implemented solutions, when compared to Moore’s law, are vague. Whereas Moore’s Law [33] has been validated, no similar accurate progression supporting fast delivery of software solutions has been observed. Indications are that the software backlog is increasing.

The challenge to Information Technology professionals is to provide a similar roadmap for the delivery of information solutions into the future. The delivered products would provide services with uncompromising integrity, sustained throughout the lifetime of the services rendered. To this end, data as a physical asset must be harnessed towards transporting value, the value ascending and descending through the hierarchy of understanding, discussed in Chapter 3.4, until ultimately it forms part of the intellectual armoury of the knowledge worker - value must be enmeshed in the intellectual construct of wisdom.

2.2 From Data to Wisdom

The preamble in 2.1 describes the platforms applied in delivering data, orchestrating change and formatting the relationships between components towards an intellectual end - the requirements of the knowledge worker. These platforms fall within the realm of Information Technology as a science and form the first milestone being a capability to classify, manipulate and store data using computers - electronic digital artefacts on the road to delivering value, a constuent of information and knowledge, in turn these two pillars underpinning wisdom and the intellectual mind.

A complimentary milestone when considering mechanisms to consume and express information must include the knowledge worker. This perspective pays no heed to the technologies employed, nor is it concerned about the accidental mechanisms and constructs employed to access the patterns required to satisfy the information need. What is of concern is the patterns must contain the value required from which, upon intellectual inspection, information is constructed and internally consumed for appreciation and empowerment.

2.2.1 Introduction

Alberthal postulates that "*The rising tide of data can be viewed as an abundant, vital and necessary resource. With enough preparation, we should be able to tap into that reservoir by utilizing new ways to channel raw data into meaningful information*" [1]. He continues to infer that "*Information, in turn, can then become the knowledge that leads to wisdom*" [1].

Fleming [15] observed the following:

- A collection of data is not information.
- A collection of information is not knowledge.
- A collection of knowledge is not wisdom.
- A collection of wisdom is not truth.

This observation can be pictured as in Figure 1.

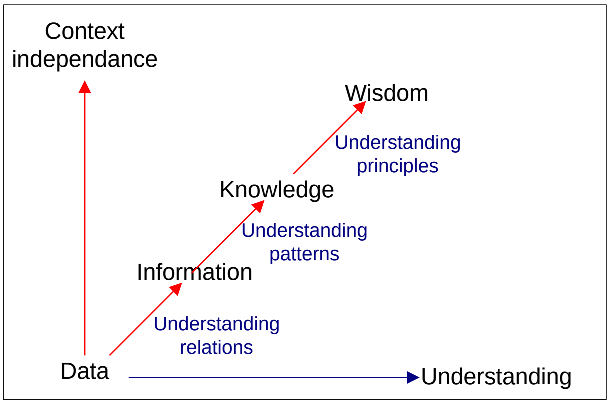

Figure 1: From data to wisdom (Source: Fleming, 1996)

'*From this observation we can conclude that information, knowledge, and wisdom are more than only collections. This in turn means that the whole represents more than the sum of its parts and the whole forms a synergy of its own. The following section will discuss the stages depicted in Figure 1*' [15].

2.2.2 Data

"*Data are raw*" [36].

"*Data is just a meaningless point in space and time, without reference to either space or time. It is like an event out of context, a letter out of context, a word out of context"* [3].

"The integral or key concept here is 'out of context' from which we can reason that it is without a meaningful relation to anything else. When a piece of data is encountered the first reaction is usually to associate it with other things in an attempt to find a way to attribute meaning to it. For example, the number 15 can be associated with integer numbers and related as being greater than 14 but less than 16, independent whether this was implied by the original number or not. If one sees a single word, such as 'time' there is a tendency to immediately form associations with previous contexts within which the word 'time' was meaningful. This context might be 'the train is on time' or 'time is running out'. This implies that when there is no context, there is little or no meaning so, a context is created. But, more often than not, that context is an assumption and the meaning is fabricated" [3].

2.2.3 Relations and Information

'That a collection of data is not information, as Fleming [15] indicated, implies that if there is no relation between collections of data, it is not information. "The pieces of data may represent information, yet whether or not it is information depends on the understanding of the one perceiving the data" [3]. How the collection of data is understood is dependent on the associations one is able to recognize within the collection. And, the recognized associations are dependent on all the previous associations that one has been able to make in the past. Information is then an understanding of the relationships between pieces of data, or between pieces of data and other information. Where a collection of data exists, so the contextual aspects introduce a relationship - these relationships together with the atomic data components thus form the structure of the construct. This construct can be equated to structure in the context of Aristotle's structure' [19] [38]' [3].

"*While information entails an understanding of the relations between data, it generally does not provide a foundation for why the data is what it is, nor an indication as to how the data is likely to change over time. Information has a tendency to be relatively static in time and linear in nature. Information is a relationship between data and, quite simply, is what it is, with great dependence on context for its meaning and with little implication for the future*" [37].

2.2.4 Patterns and Knowledge

"*Beyond relation there is pattern where pattern is more than simply a relation of relations. Pattern embodies both a consistency and completeness of relations which, to an extent,* *creates its own context. Pattern also serves as an Archetype [37] with both an implied repeatability and predictability*" [2].

'When a pattern relation exists amidst the data and information, the pattern has the potential to represent knowledge. It only becomes knowledge, however, when one is able to realize and understand the patterns and their implications. The patterns representing knowledge have a tendency to be more self-contextualizing. That is, the pattern tends, to a great extent, to create its own context rather than being context dependent to the same extent that information is. A pattern which represents knowledge also provides, when the pattern is understood, a high level of reliability or predictability as to how the pattern will evolve over time, for patterns are seldom static. Patterns which represent knowledge have completeness to them that information simply does not contain [24]' [3].

2.2.5 Principles and Wisdom

'Wisdom arises when one understands the foundational principles responsible for the patterns representing knowledge being what they are. And wisdom, even more so than knowledge, tends to create its own context. These foundational principles are universal and completely context independent. In summary the following associations can reasonably be made:

- *Information relates to description, definition, or perspective (what, who, when, where).*
- *Knowledge comprises strategy, practice, method, or approach (how).*
- *Wisdom embodies principle, insight, moral, or archetype (why)'* [3] .

2.2.6 The continuum from Data to Wisdom

"*Note that the sequence data → information → knowledge → wisdom represents an emergent continuum. That is, although data is a discrete entity, the progression to information, to knowledge, and finally to wisdom does not occur in discrete stages of development. One progress along the continuum as one's understanding develops. Everything is relative, and one can have partial understanding of the relations that represent information, partial understanding of the patterns that represent knowledge, and partial understanding of the principles which are the foundation of wisdom*" [3].

3. Understanding Information

3.1 The continuum from Data to Wisdom Extended

Figure 2 proposes a hierarchy of understanding which illustrates the progression from physical artefacts (or when in a digital form, data) to wisdom. Domains of Knowledge Management, Information Management and Information Technology are contextualized and intellectual reflection is introduced.

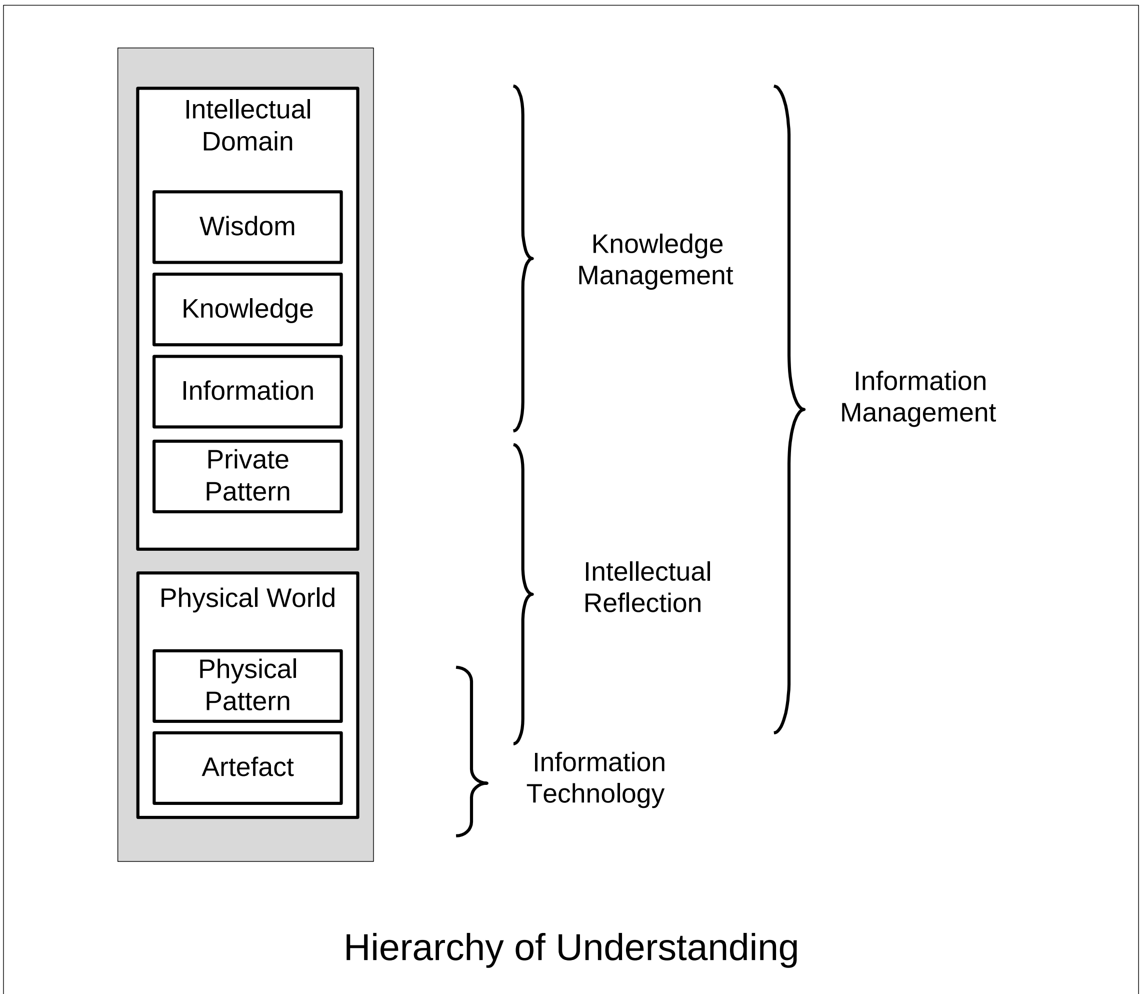

Figure 2: Hierarchy of Understanding

Figure 2 summarizes the suggested 'Hierarchy of Understanding' which starts with the presence of artefacts (items) and evolves towards an understanding of Wisdom. This hierarchy can be grouped into three zones:

- Information Technology is discussed as part of Figure 3.
- Intellectual Reflection is discussed as part of Figure 4.

- Knowledge Management - Intellectual Awareness is discussed as part of Figure 6.

The continuum from Data to Wisdom is bound on the one side by the physical world containing the accidental aspects. Information technology as an enabler qualifies as an accidental component manipulating artefacts in the physical world. These can be equated to data and binary patterns in the digital world. At the other end of this continuum, the esoteric intellectual appreciation resides - the intellectual essence falls within the domain of Knowledge Management. Essence is neither energy nor matter, but an intellectual appreciation built around an intellectual concept formed within the mental model of a capable individual. Essence is a value born of an intellectual appreciation founded in patterns.

Examining the physical world with a particular focus on the persistent components - real artefacts and physical objects, identifies a definite base or foundation for the hierarchy of meaning. The artefacts or in the case of computer software, data which can be persisted provides the spatial foundation built around ‘things’ and the relationships between things. These things and their relationships originate and manifest the rules allowing persistence to be exhibited and propagated across generations. The composite created by the ’things‘ and their relationships create a physical pattern. Data in isolation is largely meaningless. Associating data elements with one another creates a spatial composite which displays and contains a physical pattern. The pattern provides a unique ‘fingerprint’ for that particular association of elements. ’o‘, ’s‘, ’e‘, ’r‘ has no meaning - associating these four characters in a particular sequence introduces ’rose‘ - a term which has meaning - it has value shared between individuals sharing a similar public metaphor [17]. This document is a text example of a physical pattern. Similarly, graphic pixels can be consolidated to deliver an image of a rose which contains the same value of ‘rose’, and audio can be used to deliver voice messages also describing a rose. In all cases a common value is ensconced within a media type to deliver a common meaning from an originator to the recipient where the value is shared between them. The pattern of ’rose‘ to a person born without sight would differ from the private pattern of a ‘rose’ to a sighted person, although the private patterns (private mental images) would largely overlap - private pattern are introduced later.

The mind with its capability for intellectual expression makes use of the following constructs.

Wisdom: Wisdom makes use of the three constructs of time [26] [30]. The experiences of the past are combined with the stimulus and situation of the present and using a capability to apply logic, a position(s) in the future is conceived. The difference between wisdom and other predictive human capabilities (information, knowledge) is in the nature of the behaviour surrounding the expression and the recognition society appropriates to it. The nub of this definition is that Wisdom crosses three time constructs - past, present and future. Wisdom interprets, considers and expresses the essence of an understanding. Wisdom is the overarching construct crossing the constructs of the past, present and future. Wisdom can be defined as an ability to anticipate and address the future. It is built on Knowledge accumulated through experience and learning (past and present), and augmented by Information (present). Wisdom lies closest to an intellectual abstraction.

Knowledge: Knowledge can be described as a characteristic which crosses two time constructs. Persisted (remembered) experience and learning form a reference foundation which is applied within a given situation (present tense), and this learning and experience are reflected without intellectual intervention or mediation. The nub of this definition suggests that knowledge crosses two of the three time constructs - past and present. Knowledge applies the lessons of the past to the present.

Information: Information is the third of the intellectual constructs which exists only within the intellectual universe - the mind. Information is initially presented as a spatial sequence of patterns consumed using the biological senses and intellectually appreciated. Information is suggested to be a realtime stream of patterns of meaning to the beholder - intellectually appreciated. Meaningless sensory streams or causal inputs [13] to one individual may be meaningful to another - information varies between beholders and is appreciated within the intellectual capacity of that beholder. Public metaphors provide the milieu within which information is shared between intellectuals. Information is a real time mental stream of patterns delivering situational and environmental parameters enabling awareness. It is a product, an (internalised) output of the neurological senses optionally preserved chemically and electrically as memory within the neurons of the brain. It is realised meaning sourced from the physical world via the causal inputs and synthesized using persisted past mental experiences and learning. Figure 6 suggests that information is an intellectual construct, the end result of an interaction between a biological (senses) and physical interaction. The biological (senses) and physical interaction form the filling in a sandwich between the intellectual universe and the corporeal world. Causal inputs [13] stream patterns and value for internalisation and intellectual consumption. Intellectual expression, when an intellectual purpose is manifested into the physical world, streams intellectual meaning into the physical world.

Private pattern: Private patterns are the intellectual imagery available to the individual during the process of intellectual expression - the process of thinking. Private patterns are the beholders 'mind picture' built using intellectual spatial objects [26] or 'mind imagery' - a form of intellectual coherency. Private patterns are recalled memory and topical mental imagery appropriately constructed or 'imagined' for the matter under scrutiny.

Physical Pattern: A physical pattern exists where the relationships between physical (or electronic) artefacts contain meaning. Rearranging the relationship changes the meaning. As an example, the word 'stop' implies terminate. Rearranging the letters reveals 'post' - a letter or parcel delivered by the postal services. In this case the artefact is the word, and the meaning is derived from the relationships between the letters – artefacts used to construct the word.

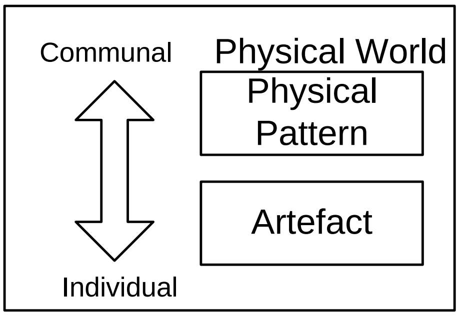

Figure 3: Relationship between artefacts and physical patterns

Figure 3 suggests that a composition of physical artefacts together with their relationships forms physical patterns.

Artefact[2]: In the broader sense of the word, any tangible or intangible construct visible to the mind. In the computing sense, a digital construct is constituted from 'bits'. These are the things or objects which constitute the inspected construct. A room with furniture, or a landscape with trees, hills and clouds. These discrete components or objects deliver an appreciation which is an essence. The individual components are the artefacts, and together they create a whole which is then reflected into the intellectual domain using causal inputs and the senses.

3.2 Interaction between the Physical World and the Intellectual domain

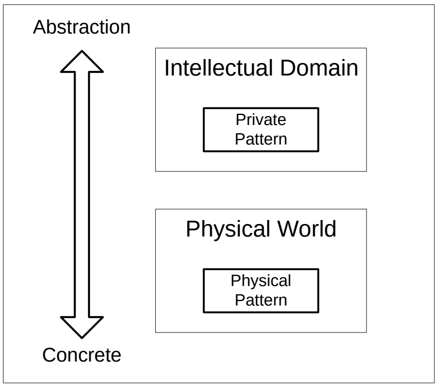

Figure 4: Intellectual Reflection

Intellectual reflection, discussed in 3.5 defines the interactions and the relationship between the Intellectual Domain and the Physical World. The bidirectional arrow in Figure 4 indicates how physical patterns formed by data and artefacts at a concrete level, and private patterns within the mind of the individual within the intellectual domain interact. Intellectual appreciation extracts the essence from a realtime experience. This is an esoteric private emotion which is found only within the intellectual domain. It is a consequence of an interaction with the physical world - of composites built upon energy and matter. Essence and accident are properties, the one founded in the intellectual domain, and the other in physical artefacts and their relationships (from which value can be derived). Intellectual reflection oscillates between the essential understanding and the internalisation of value embedded within patterns ensconced within accidental components.

---

[2] Artefact

1. something made or given shape by man, such as a tool or a work of art, esp an object of archaeological interest

2. anything man-made, such as a spurious experimental result
http://www.thefreedictionary.com/artefact - referenced on 21st September 2010

3.3 Essence and Accident

"*The other, more satisfactory basis for distinguishing essence from accident is an epistemological or methodological one. Knowledge of the essence of a thing is said to be more important than knowledge of its other attributes*" [9].

"*Accident, as used in philosophy, is an attribute which may or may not belong to a subject, without affecting its essence*" [23].

"*In philosophy, essence is the attribute or set of attributes that make an object or substance what it fundamentally is, and which it has by necessity, and without which it loses its identity. Essence is contrasted with accident: a property that the object or substance has contingently, without which the substance can still retain its identity. The concept originates with Aristotle, who used the Greek expression to ti ên einai, literally 'the what it was to be', or sometimes the shorter phrase to ti esti, literally 'the what it is,' for the same idea. This phrase presented such difficulties for his Latin translators that they coined the word essentia to represent the whole expression. For Aristotle and his scholastic followers the notion of essence is closely linked to that of definition (horismos)"* [8].

Brooks [4] on software technology: *"... essence, the difficulties inherent in the nature of software, and accidents, those difficulties that today attend its production but are not inherent*".

Brooks [4] "*The essence of a software entity is a construct of interlocking concepts: data sets, relationships among data items, algorithms, and invocations of functions. This essence is abstract in that such a conceptual construct is the same under many different representations. It is nonetheless highly precise and richly detailed*". The 'essence' to which Brooks is referring can be equated to an appreciation of the delivered value embedded within the patterns exposed to the information seeker.

3.4 Hierarchy of Understanding Expanded

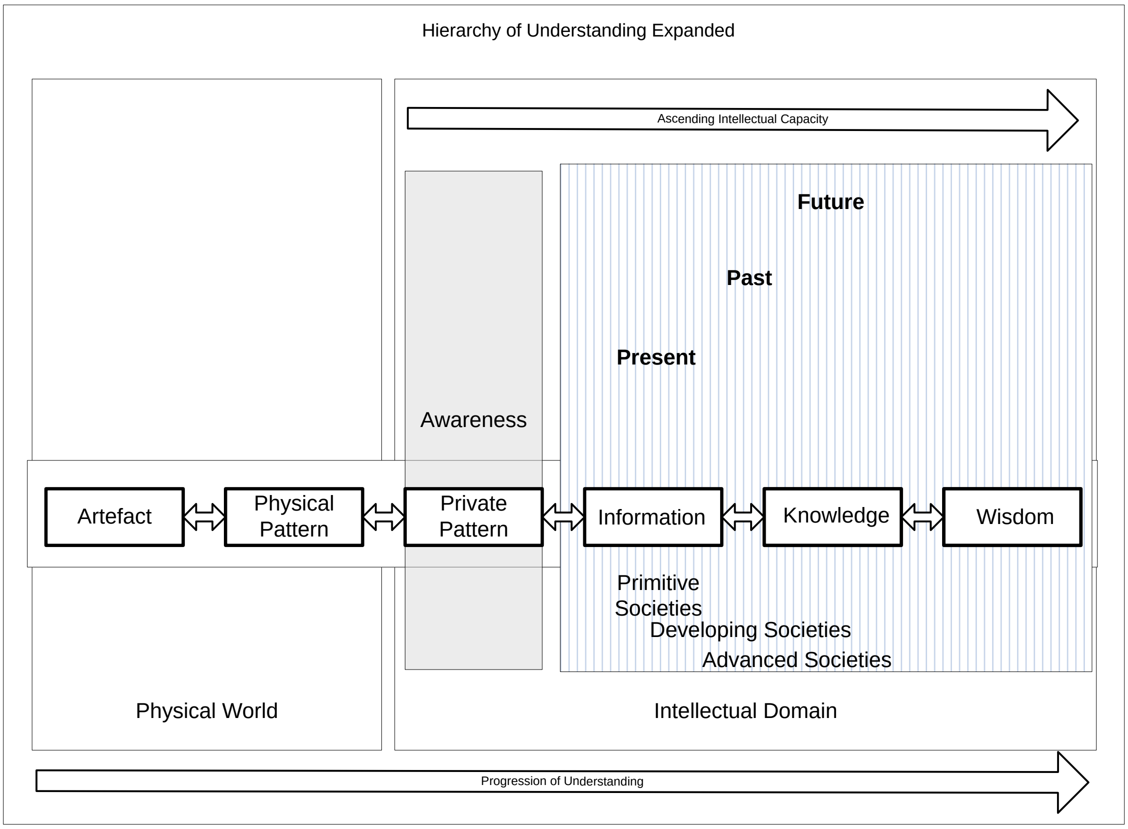

Figure 5: Hierarchy of Understanding Expanded

Figure 5 consolidates a perspective which contextualises the hierarchy of understanding within the concepts of time, concurrently illustrating the relationship between an increasing intellectual capacity and the continuum between data and wisdom. It consolidates and expands the Hierarchy of Understanding and introduces the progression of understanding - the growth of an intellectual capacity. It illustrates the relationship between the physical world and the intellectual domain, overlaid with the concept of awareness and the constructs of time. Ascending intellectual capacity is observed growing from an intellectual awareness which initially does not grasp the concept of time, for example, a young child. As intellectual awareness grows, so a capability for abstraction increases. As an understanding of causality matures, so the intellectual maturity ripens, until at the most mature level, there is a capability to accurately control events far into the future (also an intellectual construct).

3.4.1 Interaction between the Intellectual Domain and the Physical World

Figure 6 ranks the three constructs of Wisdom, Knowledge and Information found within an intellectually aware capability. Information is experienced as a real-time (present tense) conscious understanding of meaning. Information, when persisted as memory becomes knowledge, straddling two of the three time constructs - past and present. Wisdom is an extension of knowledge - additionally, it incorporates a logical capacity harnessing memory and logic to conceive and control the future. Wisdom straddles three time constructs, combining with knowledge and information, to deliver a recognised intellectual capacity to manipulate the future. The concepts of past, present and future are discussed further in [30].

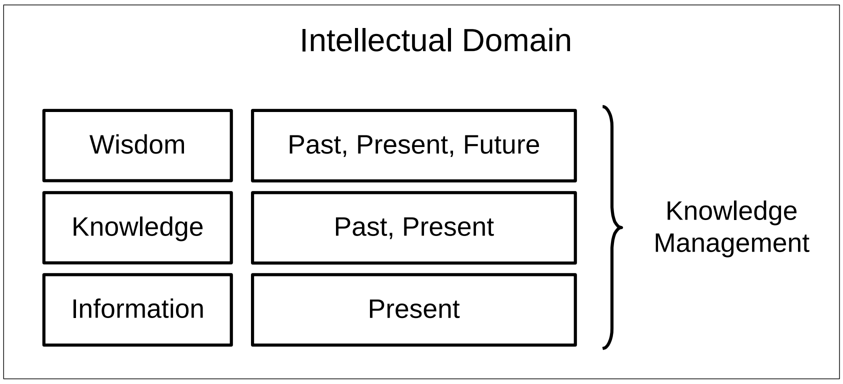

Figure 6: Wisdom, Knowledge and Information within the constructs of Time

The initial awareness of the physical environment is revealed to the consciousness through the senses as a stream of sensory impulses - causal inputs [13]. The consciousness filters this stream and extracts meaningful value from this stream - the mental interpretation of meaning as information. This information is then persisted in memory as knowledge - in physical terms, persisted neurological assets of a chemical (spatial component) and electrical (Temporal aspect) nature. The intellectual domain exists within each individual as a private pattern - a unique mental model accessible and available only to the individual. This private pattern is exposed to the physical world through intellectual reflection.

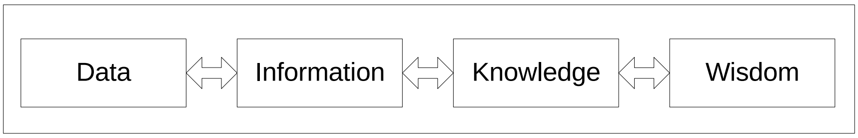

Figure 7: Fleming's bridge between data and wisdom

Figure 7 shows how the perspective reflecting the relationship between data and wisdom as reflected by Fleming in

Figure 1.

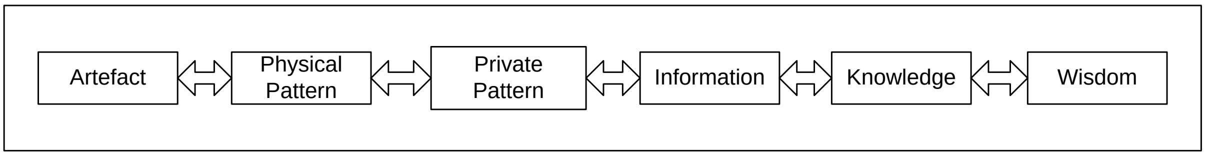

Figure 8: Extended Bridge between Data and Wisdom

Figure 8 shows the progression of

Figure 1 to Figure 2. Artefact in Figure 8 is a physical or digital construct – digital artefacts can be 'bits' – these are the digital equivalent of a physical artefact.

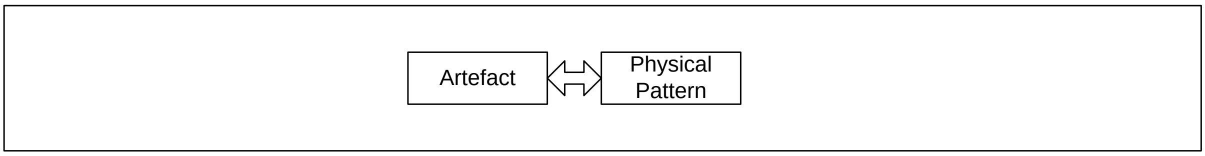

Figure 9: Physical World

Figure 9 illustrates how the artefacts and the relationship between artefacts together deliver physical patterns found within the physical world.

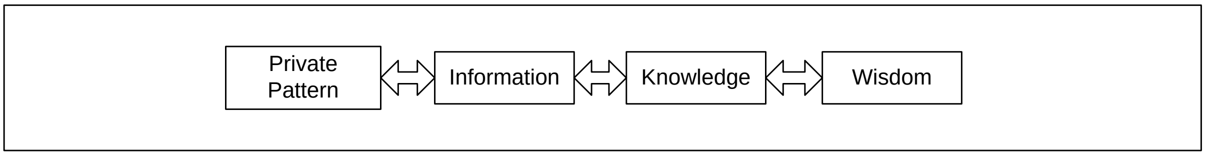

Figure 10: Intellectual Domain

Figure 10 proposes constructs found within the intellectual domain. Intellectual reflection (described further in paragraph 3.5) bridges the divide between the physical realm and the intellectual domain. The challenge for information delivery as a science is to emulate intellectual reflection.

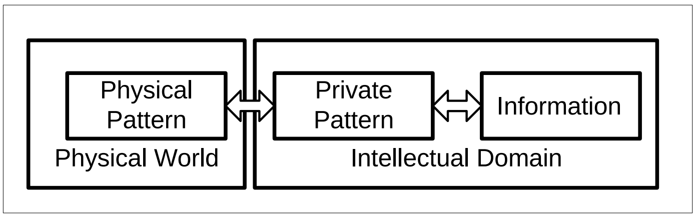

Figure 11: Interaction between senses and the intellectual domain

Figure 11 illustrates the physical pattern→ private pattern→ information interaction as a bidirectional oscillating interaction between the conscious intellect within the intellectual domain and the presented imagery within the physical world which is available to the senses of the individual [26]. This imagery is limited to stimuli which are exposed to the senses and where causal inputs [13] can be affected - this relies on the availability of senses which can react to specific stimuli. Value embedded within the physical pattern is internalised to the private intellectual pattern of the knowledge worker as information to be intellectually internalised as memory.

3.5 Intellectual reflection

The interaction between the intellectual domain and the physical world is suggested to be intellectual reflection and is discussed in detail over the following pages.

Intellectual reflection is defined as follows:

Intellectual reflection is an immediate mechanism which receives and delivers the conscious stimulus underlying structured thought. The structure of intellectual reflection is volatile, shapeless and fleeting. It can be seen as a dynamic and short living abstraction within which an information thirst is formed and subsequently quenched. Understanding this allows a response to be designed which reacts fluently, dynamically realigning with the changing information need supporting the intellectual thirst. Intellectual reflection provides the highway along which the demand for patterns containing value can be identified, anticipated, delivered and released. Intellectual reflection can be described as a bi-directional conduit between the corporeal world, containing the accidental aspects, and the intellectual world within which the conceptual need - the essence - is manifested [10].

The characteristics of Information must be understood if a solution supporting information delivery is to be developed. Understanding information is dependent on understanding how value is transferred from one intellectual domain to another, crossing intellectual domains separated by the physical world, using patterns. Patterns contain relationships between artefacts real or imagined. Note that intellectual appreciation does not depend on patterns directly - rather it makes indirect use of patterns as conduits to transfer *value* across intellectual domains. This section addresses the mechanisms applied when delivering and expressing patterns, from the intellectual and physical perspectives. It will reference the concept of a temporal moment [30] as the period when change takes place, and use this concept to classify the atomicity and granularity of an event. It introduces and expands on definitions and the roles of rule, logic and structure. These form the fundamental constituents of intellectual reflection.

The relationship between the physical world and the intellectual domain is illustrated in Figure 12. The interaction between the Intellectual Domain and the Physical World is where intellectual reflection takes place. Intellectual reflection is a process which bridges the physical world and the intellectual domain.

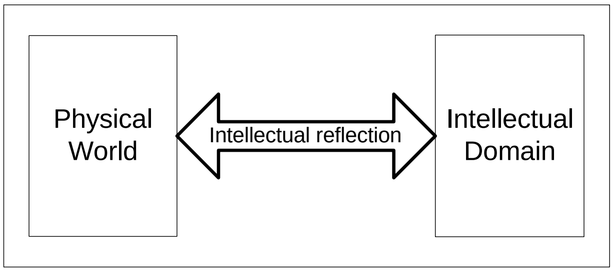

Figure 12: Intellectual Reflection

Physical world artefacts and the relationships between the artefacts have value embedded therein. The physical world interacts with the intellectual domain through causal inputs [13], and allows causal inputs with embedded value to reach the conscious plane, where the value is extracted and internalised as information. This information is analysed and appraised intellectually. Intellectual reflection oscillates between the intellectual and physical domains. Intellectual decision harnesses the physical capabilities (voice, hand and limb actions) towards expressing the intent of the intellectual need. The expression of bodily manoeuvres are persisted in some form or another within corporeal artefacts. Examples include carving, writing and painting. Intellectual expression manifests and is persisted in a spatial reordering, through biological expression and behaviour (use of limbs, hands and feet, speech, gestures) of physical artefacts (writing, spoken word, physical gestures) which produced refreshed physical patterns which contain the value originating within the intellectual domain of the originator, transferred into a physical medium and later received into an intellectual domain.

This process describes Intellectual reflection. It is this mechanism which must be understood and emulated if an information delivery solution matching the information needs of an information user is to be met. The mechanics underlying the use of Metaobjects would be positioned between artefacts and the physical patterns which are constituted from these artefacts.

The essence of an understanding describes the intellectual appreciation either individually for a private value or communally. The capability to internalize and appreciate, or alternatively to externalize and communicate the essence is dependent on the nature of the vehicles utilized to transport the appreciation - these form the accidental aspects.

Intellectual reflection is the bridge where value is transferred between an intellectual domain and the physical world. As value is embedded within the relationships between physical artefacts, it resides thus between physical patterns and artefacts as illustrated in Figure 13

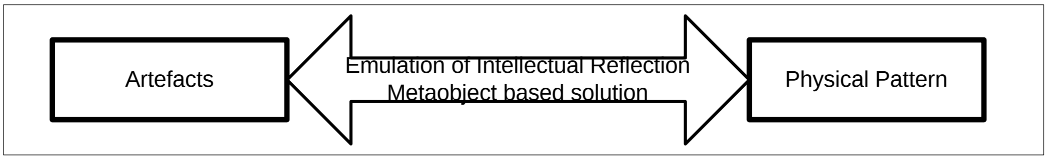

Figure 13: Artefacts and Physical Patterns

A physical pattern reflects the perspective from a physical world perspective. The same construct, when viewed intellectually, deliver embedded value ensconced within the physical artefacts. This dual role is indicative of the dualism which characterise the intellectual and physical dimensions - intimately interwoven but mutually exclusive, the one a physical hierarchy, the other a value - an abstraction of the mind.

Extending this, the embedded value can be filtered to deliver value to the individual as the intellectual understanding manifested when internalising the physical pattern. This internal view extracts the causal inputs, converts it into information and internalises it as a private metaphor within the intellectual armoury of the individual. The information is persisted by the individual as a private pattern retained in memory in neurological electrical and chemical persistent states.

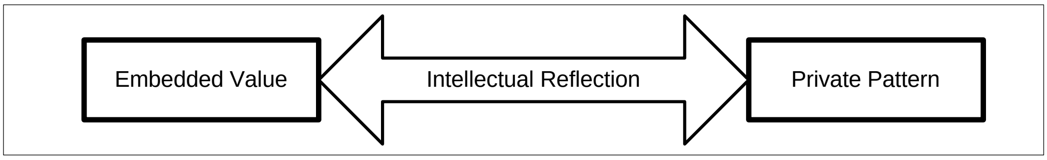

Figure 14: Embedded Value and Private Patterns

The oscillating nature of intellectual reflection is characterised by an interaction between the physical world and the intellectual domain. Figure 13 identify the artefacts and their relationships forming physical patterns available for evaluation and expression (through causal inputs and physical expression). Figure 14 reflects the intellectual view of this oscillating relationship. It consumes the embedded value within the physical pattern and internalises it, persisting it as memory which extends the private pattern of the individual. Expression of intent takes place when the intellectual will of the individual is manifested in the physical behaviour and actions (body movement and expression) of the individual impact the spatial relationships between the physical artefacts within his span of control.

The dual nature of this oscillating relationship must be reflected when emulating the process of intellectual reflection. Metaobjects appear to exhibit these characteristics. When viewed from a physical perspective, as spatial components, they are equivalent to a normal program call with a handle, function and signature. When they are viewed from an intellectual perspective - a value perspective, the handle, function and signature become meaningless, but the relationships form a pattern which has meaning. This would be in terms of information delivery within the conventions of the metaobject application semantics. Metaobjects are electronic manifestations which emulate intellectual reflection in that they use electronic artefacts (or data) to deliver physical patterns of meaning to the computer user. In this sense, they bridge the gap between data and the delivery of electronic patterns (within which value is embedded) which when examined and internalised become private (intellectual) patterns and ultimately information to the user.

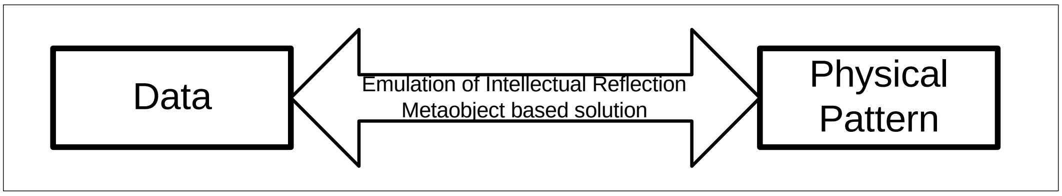

Figure 15: Hierarchy of Understanding: Physical domain

Figure 15 illustrates the gap between data in its raw form and physical patterns where value has been ensconced.

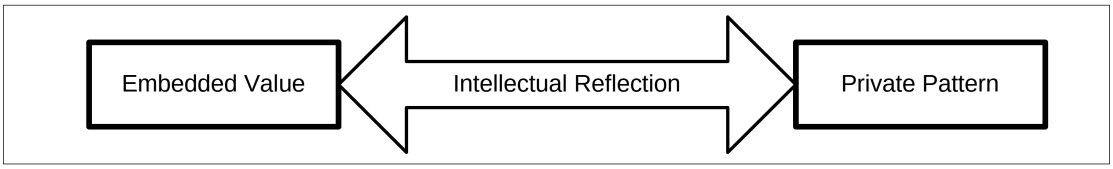

Figure 16: Hierarchy of Understanding: Intellectual Domain

Figure 16 illustrates the gap found within the intellectual domain, where embedded value needs to be stored as a private pattern (within the memory of the individual)

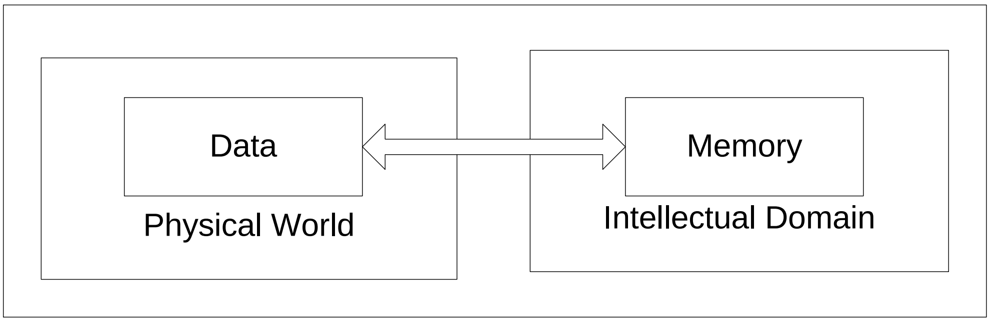

Figure 17: Physical world and the intellectual domain

Figure 17 displays the base states within both the physical and intellectual domains. In the physical world data contains contextual value. The intellectual domain once value delivery has been completed contains the transferred value embedded within neurological material as memory. Intellectual reflection is a mechanism which carries this value between the two states. Information delivery solutions emulating intellectual reflection would need, when in a physical role to be able to manipulate the physical world.

Understanding information delivery, the underlying mechanisms and the properties of intellectual reflection is a necessary prerequisite when introducing the metaobject based approach (or value driven information delivery solution) which forms the focus of this thesis. With this as background, properties of the domains across which intellectual reflection must oscillate are expanded. These are across the domains where the accidental properties reside (physical domain) and where essence is appreciated (intellectual domain).

3.5.1.1 Flow between the Corporeal World and the Intellectual Domain

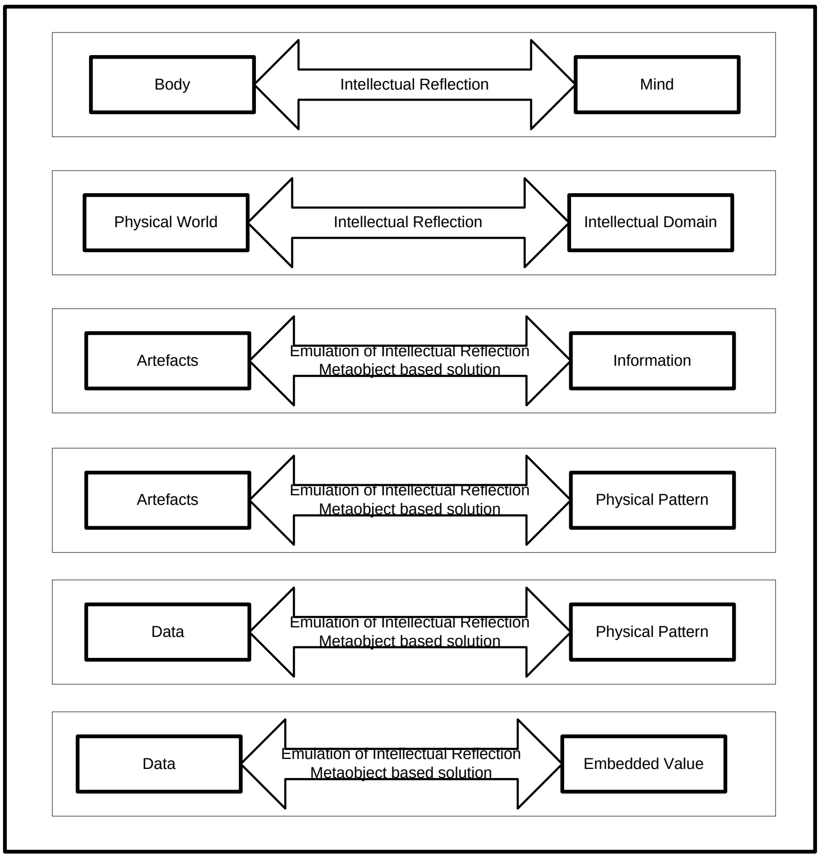

Figure 18: Emulating Intellectual Reflection

Figure 18 presents a succeeding progression starting with the ‘Body-Mind’ relationship and terminating with a ‘Data-Embedded Value’ comparison, in which the ‘Body-Mind’ relationship is emulated.

1. The first layer suggests that intellectual reflection is the conduit between the body and the mind.

2. The body, being an organic construct, is of the physical world and the mind of the intellectual domain.

3. Information has been proposed to be an intellectual construct, extracted from an immediate stream of sensory stimuli containing patterns with embedded value. The source of the sensory stimuli would be the artefacts (and their relationships to other artefacts) exposed to the senses.

4. The fourth level is derived by revisiting the definition of information, which suggests that information is value embedded within patterns of meaning to the beholder. This implies that the value internalised can be directly equated to the physical patterns available for sensory inspection.

5. Finally this relationship between artefacts and physical patterns is viewed from a computing perspective - it indicates that data reflects physical patterns within which embedded value is ensconced.

6. Data must have its embedded value released using a mechanism similar to that of intellectual reflection, which links the physical world with the intellectual domain. This is the domain of metaobjects.

Figure 19 illustrates that the characteristics and properties underlying the body-mind relationship need to be emulated if the mechanism underlying information delivery is to be emulated. Data, a physical manifestation would need to be dynamically realigned towards delivering physical patterns (or rather presenting the value using physical patterns) in the same manner intellectual thoughts realign physical artefacts towards an intellectual objective.

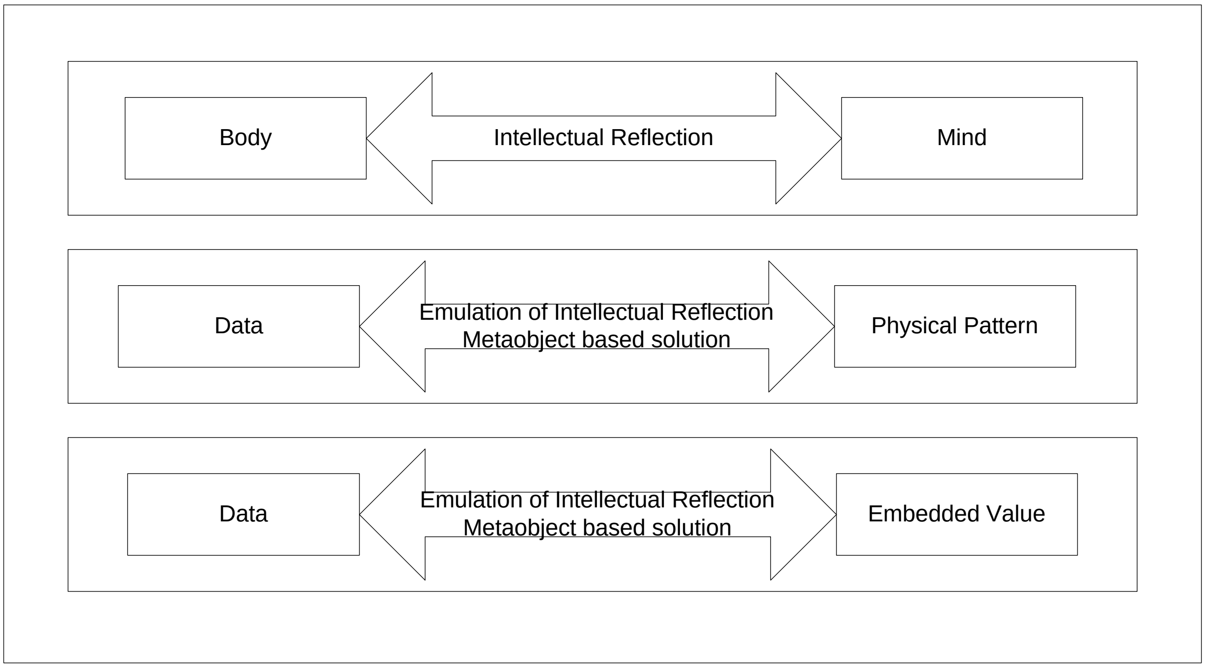

Figure 19: Equating ‘Body-Mind’ to ‘Data-Embedded Value’

Data and embedded value need to be dynamically associated, so that data when delivered immediately delivers the patterns with value, intellectually required. Conversely, the intellectual requirement can immediately be transposed into value which can be immediately transferred into physical patterns. This implies a dynamic realtime re-alignment of data, or in other terms, a bi-directional, real time oscillating interaction.

A model of the physical world is hierarchical in nature. A model of the intellectual universe has networking properties Figure 16. The characteristics of each are introduced below:

3.6 Characteristics of a hierarchical model

A hierarchical model has the following characteristics:

- Hierarchical
- Direct
- Physical

These characteristics introduced in [26] and [27] are utilised in Table 1 as criteria when evaluating a hierarchical approach against that of a network approach.

3.7 Characteristics of an intellectual model

It is prudent to understand the characteristics of the thought process if an information delivery solution to satisfying the intellectual needs is to be developed. These criteria were identified when examining the characteristics of a metaphor [27] and are recommended criteria for an information delivery framework [27]. These characteristics are utilised in Table 1 when evaluating the metaobject approach.

- Network
- Ephemeral
- Indirect
- Anonymous
- Autonomous

3.8 Temporality and persistence within the physical world

Two intellectual constructs underpin this understanding.

- Temporal Moment [30]

- Direct and Indirect Persistence [28]

These constructs provides a theoretical foundation necessary for the further development of a real world information solution which can satisfy intellectual needs. These criteria will also be utilised later when evaluating the hierarchical and proposed (network) models.

3.9 Conceptual Model Introducing Intellectual Reflection

The preamble in Paragraphs 3.1 to 3.8 introduces definitions of an intellectual appreciation constituted from information, knowledge and wisdom - the domain of essence. Conversely, it suggests that data and the relationships between data form physical patterns containing value - this value is presented for intellectual inspection after which it enters the intellectual armoury of the beholder as information if deemed of value.

3.9.1 Modelling Intellectual Reflection - the application of rule and logic in structuring the delivery and dissemination of information

Modelling intellectual reflection is a daunting challenge. Previous work [4] [26] aimed at understanding the intellectual aspects underlying information need and information delivery has succeeded admirably in delivering the accidental components supporting these objectives, but the need to satisfy the essential aspects remains.

3.9.2 Model illustrating state transformation

Figure 20 presents a conceptual diagrammatic representation illustrating the transition a spatial construct, or pattern, undergoes when change is effected. State 1 illustrates a pattern - a spatial construct constituted from components with associated relationships.

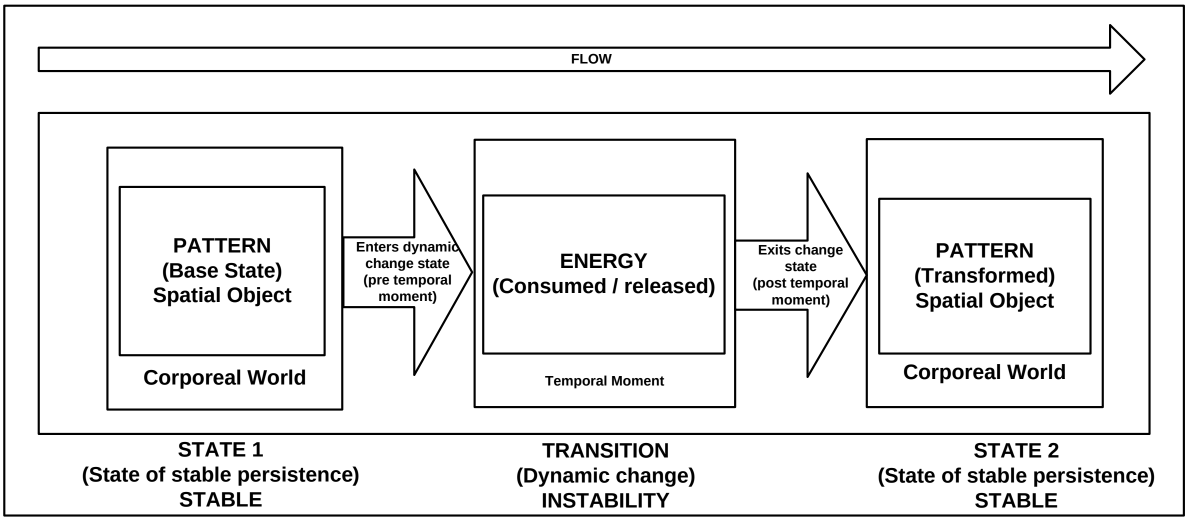

Figure 20: Transitioning between states using a pattern

A grouping of artefacts or ‘spatial objects’ together form a pattern. During State 1, the spatial objects, and thus the pattern, is in equilibrium, and energy consumed and released does not impact on the structure of the pattern. Note that at the material level - (the Aristolian concepts of material, structure, form and essence are discussed in [20] [26] [39]), atomic components may be exchanged - this does not impact the structural stability of the pattern. The introduction of energy initiates the transformation event where the pattern is amended. Spatial reorientation of the pattern takes place as the energy is expended or consumed. Once the energy transfer is exhausted and spatial transformation effected, the second state of stability is entered. The period where energy freely manifests during this transition determines the length of the temporal moment. This term will be discussed in the following paragraphs - in essence, a temporal moment for an atomic transition is incomprehensibly small whereas the temporal moment for the universe is incomprehensibly large [30].

Figure 20 displays a transition of a pattern (a spatial object formed from components and the relationships between the components) from a base state to a transformed state. Figure 21 expands Figure 20 and extends the change of state from a base state to an altered state for a single cycle of memory within the intellectual domain.

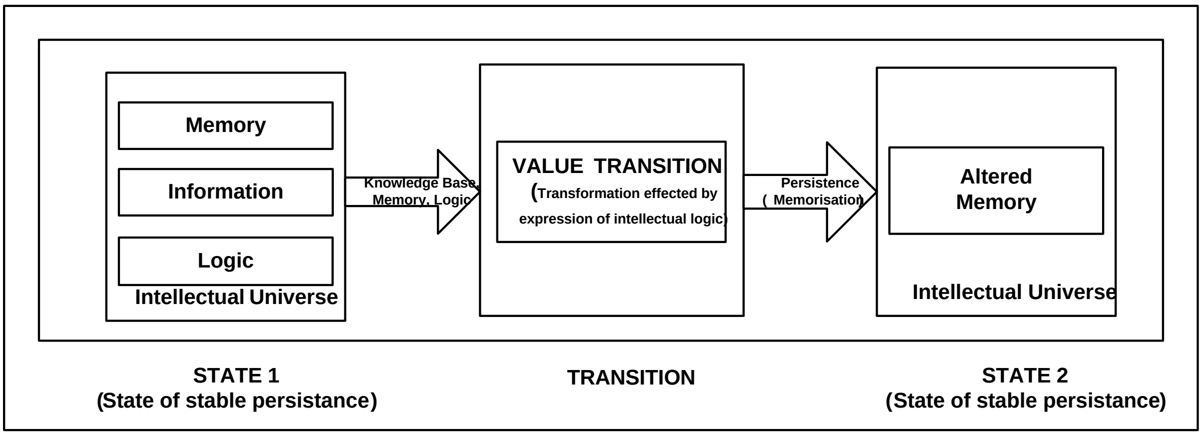

Figure 21: Memory, information and logic in the intellectual process

Figure 21 includes the three foundation components required to manipulate an intellectual spatial object. The base state includes ‘Memory’ - a persistent pattern embedded within neurological material. ‘Information’ is value containing the rules, which are interrogated during the temporal moment [30] of the event. The third constituent is ‘Logic’ – a capability released during the temporal moment.

Rule and logic are delivered inputs which do not form part of the ‘Memory’ structure directly - they are consumed during the transition. Metabolic energy is utilised during the temporal moment when the logic and rule interact during the transformation process to add value to the memory construct. The output of a particular event is altered persistent components or spatial objects. State 2 is attained once the persistent altered components (=altered memory) attain equilibrium - no free energy is available. The persisted components reach equilibrium and await the following event, where they will in turn become an input towards further transformation.

3.9.3 Model illustrating how rule, logic and structure in the computing process

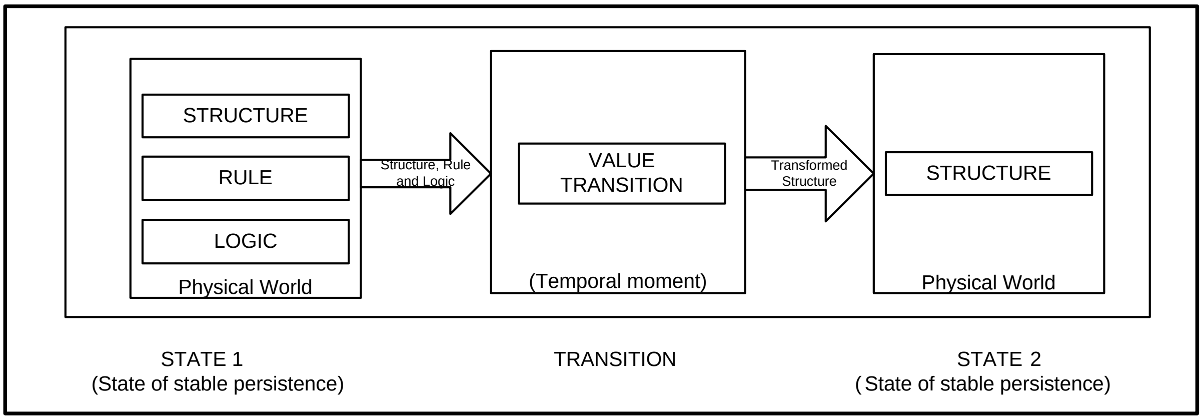

Figure 22: Structure, rule and logic in the computing transitioning process

Figure 22 applies the conceptual model in discussed in Figure 21 to computer processing. It indicates how transition is dependent on a supply of logic, embedded together with the rules which will be applied to structurally alter a structure formed by data content constituted from bits. Bits are the material which will be altered and the new structure persisted. State 2, a stable phase found after the transition period, would then reflect an altered pattern being the restructured data. Each cycle of a Central Processing Unit would pass through a transition from State 1 to State 2. Note that the relationship State 1 - Transition - State 2 reflects a pattern - moving from a stable state through a transition state, to the next stable state, but with an altered spatial outcome.

Note that the transition process does not necessarily require a structure as input, although rules and logic are required in every instance. A structure can be instantiated (=created) during the transition process, as an expression of logic uses rules to manifest an outcome - a structure can be created using rules and logic as input only, and it is not necessarily dependant on a pre-existing structure. For example, a button can be instantiated using logic (=instruction) with rules (=parameters). It can exist autonomously of other components and is thus autonomous and independent of other components. This capability allows an artefact to have a temporal moment totally independently of other artefacts or components - this capability of artefact autonomy is a pre-requisite if intellectual reflection is to be emulated. The contextual value when such a discrete component is published in isolation is dependent on the historical (past), situational (present) and intellectual (future) perspectives which surround the button as an artefact. Historical context can be delivered either using direct persistence, or indirect persistence [28].

3.9.4 Consolidation of intellectual and computer models

Figure 23 displays the flows required for an intellectual 'change of state' to transpire - the process of intellectual learning and it's interaction with the computer world, and reliance upon electronic input.

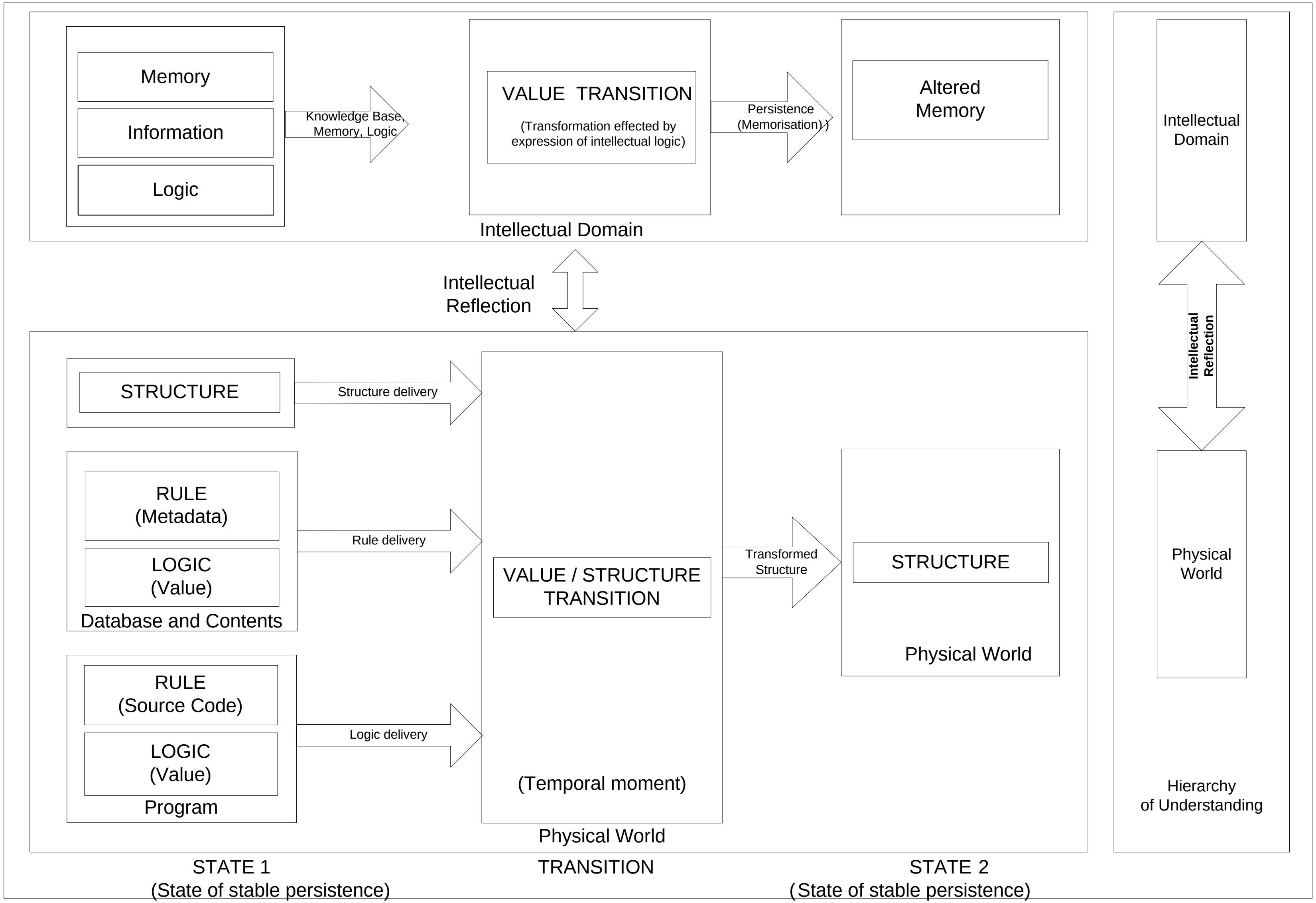

Figure 23: Consolidation of intellectual and physical models

3.9.5 Model illustrating delivery of rule as part of the transition process

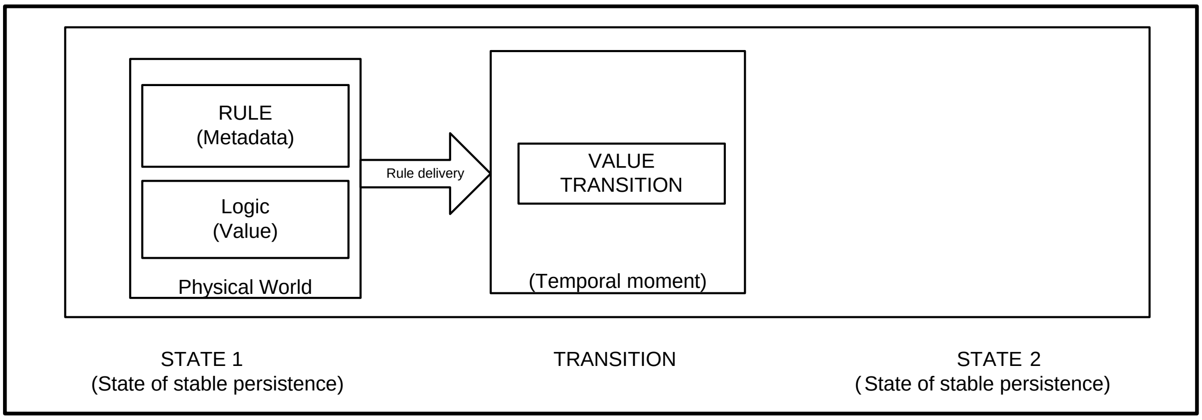

Figure 24: Conceptual model illustrating rule delivery

Figure 24 provides a view of rules delivered into a transitioning process. Rules describing spatial order are a mandatory constituent required when conducting a transition event. The rules will describe the order and sequence of the electronic constituents which are to be transitioned and persisted. The rules specifically describe the required persistent 'to-be' (relational / spatial) aspects of the structure been transitioned (or created). These rules when interpreted, deliver a stream of logic which differs in intent from the logic discussed in Figure 25 in that the focus is not to excise intellectual intent, but rather resides at a lower level. This logic ensures that integrity of persistent artefacts is maintained. It orchestrates structural transformation without compromising structural integrity.

3.9.6 Model illustrating the delivery of logic as part of the transition process

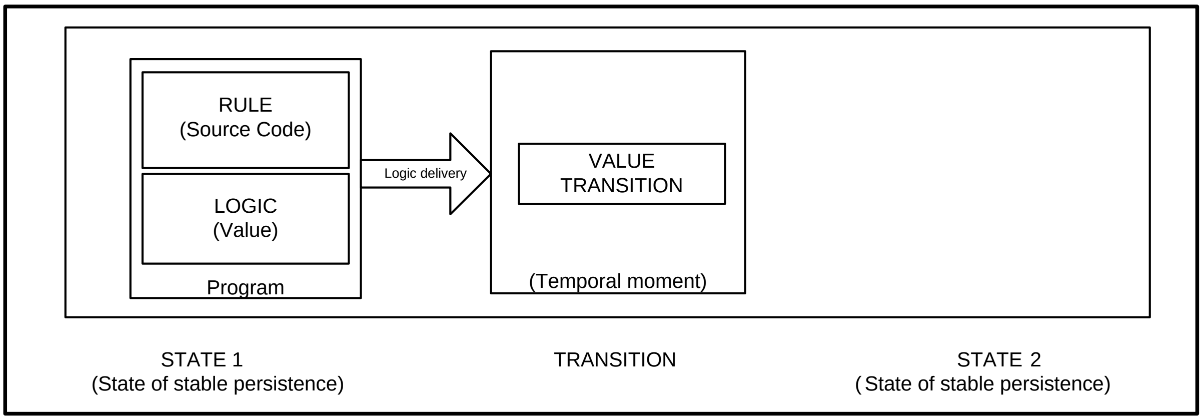

Figure 25: Conceptual model illustrating logic delivery

Figure 25 describes how intellectual intent, locked within computer syntax and expressed using binary code is delivered during the transitioning function. The logic is a value which is manifested during the Central Processing Unit cycles as the one instruction sequentially processes bits, and as instructions in turn are sequentially processed [38]. In this case, the rules originally encapsulated in source code, and ultimately expressed as binary code are sequentially interrogated, and the value containing the logic is energised and released - as logic has the same properties as energy it manifests in its pure form only when in the presence of free energy. The Central Processing Unit uses energy as a power source - this is the energy which energises the logic. The box with the arrow 'Logic Delivery' in Figure 25 illustrates the delivering of instructions within which the temporal agents are encapsulated - the logic embedded within the algorithm. A computer source program is a persisted pattern (electronic file) constituted from symbols with syntactical coherence, developed intellectually to satisfy an information need. The various derivatives (object code, dll's and executables) ultimately manifesting as binary (=executable) code are derivations of this meaning. The ultimate 'essence' is in the logic, contained as value embedded within these patterns, electronically persisted and propagated from source code, ultimately residing in executable code. This logic is released when the Central Processing Unit, powered using electricity, sequentially steps through the instructions, each instructions relationship to the previous and next developing a stream of value which is then manifested and ultimately persisted as part of the process of transforming a spatial construct - by amending content found in a persistent store of one type or another - for example, as Graphic User Interface (GUI) on a screen, bits on the disk or in internal memory. Note that logic has similar properties to energy - it can only be expressed within a temporal moment - at all other times it is encapsulated within a spatial construct.[3]

[3] This is the most complex aspect of the model– logic, having temporal properties similar to energy, can only exist in its 'unadulterated' or' pure' form within the temporal moment. It must be persisted within a spatial construct when dormant, outside the temporal moment. This spatial construct (the executable) must 'unpack' or 'deliver' the logic when real time transformation takes place. Unpacking of logic is dependent on rules; logic itself is manifested as a set of rules.

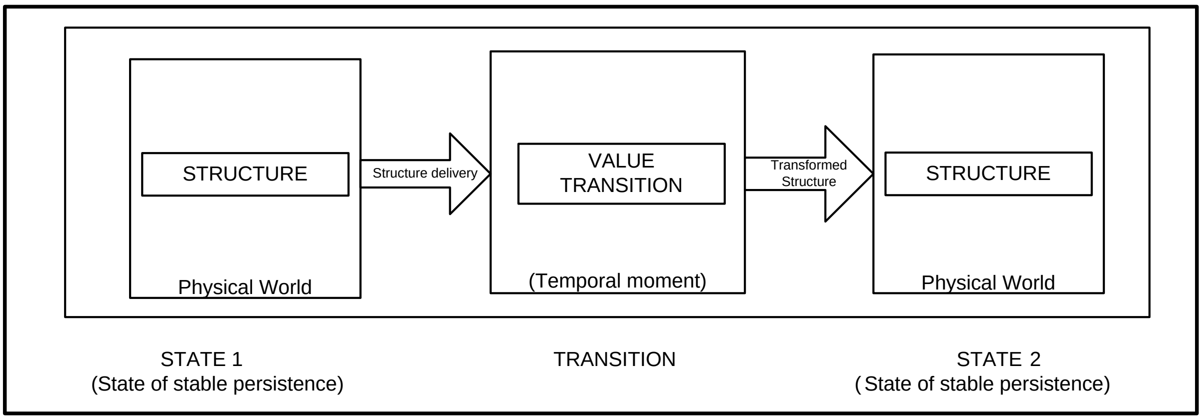

Figure 26: Conceptual model illustrating structural transformation

Figure 26 shows how a structure which is in a base state exposed to transition, after which it stabilises. The physical example is relatively easy to understand - an amorphous piece of plasticene when moulded, changes shape - a simple morphological transformation. To illustrate - a structure in a base state, when exposed to rules (pinch, press, rub, and roll) and logic (make a mouse shape) is transformed to a second state of stability albeit now in the shape of a mouse - an intellectual appreciation. The material is the same - the structure is different. Two mechanism's for transiting from a base state to a transformed state exist - direct persistence and indirect persistence [28].

3.9.6.1 Direct Persistence [28]

Direct persistence when presenting an image builds directly upon an existing image - the image is a physical composite, and change is undertaken by changing the image components directly, each edited component is related to the composite as a whole, and the contextual positioning determined accordingly. Components which make use of direct persistence cannot exist independently of the composite as a whole.

3.9.6.2 Indirect Persistence [28]

Indirect persistence introduced in 3.7 is necessary if the characteristics of an intellectual lifecycle are to be emulated. Indirect persistence allows a component to be created independently of the composite. The component is spawned and inherits 'genetic material' [26]. Genetic material becomes a spatial construct carrying the value from the previous generation to the progeny. This is the link to a spawned components predecessor. No other linking mechanisms to preceding generations exist. Once spawned, the component would inherited its identity and purpose from the genetic material - situational awareness [26] would then drive it's behaviour within the context it has been created during the temporal moment [30] of that component. A 'situational pre-cognisance' [26] capability enables the component to prepare next generation components to be seeded during which the component enters a unique 'metaobject identification and seeding' function, discussed in Figure 41, illustrates this common function. Software components created using indirect persistence do not require a structure as a pre-condition - they can be created independently of other components. The requirements for this category of component is that once spawned using genetic material, logic and rules are then applied during the active transition period to deliver the encapsulated functionality. When discrete functionality is required outside the dedicated capability of the component, the component would instantiate the appropriate metaobjects which would be expressed - the broker would intercept these metaobjects, interpret the value and invoke the program - so initiating the next generation.

3.10 Expanding Essence and Accident

Djikstra [11] introduces the term 'separation of concerns' in the context of ordering one's thoughts when dealing with various aspects simultaneously as part of the process of intelligent thinking. *"..... It is what I sometimes have called "the separation of concerns", which, even if not perfectly possible, is yet the only available technique for effective ordering of one's thoughts, that I know of. This is what I mean by "focusing one's attention upon some aspect": it does not mean ignoring the other aspects; it is just doing justice to the fact that from this aspect's point of view, the other is irrelevant. It is being one- and multiple-track minded simultaneously*".

Being one- and multiple-track minded simultaneously would require a one-track focus on a particular concern, and concurrently, a global consolidation of universal concerns affected by the focus under scrutiny. This interaction would have oscillating characteristics, flip-flopping from the one to the other. One state focusing on the concern to hand, the second state evaluating the effect of that concern against others, and then intellectually digesting the consequences, by reviewing the focus initially considered. This 'intellectual reflection' can be considered to have similarities to a metaphor 'sandwich' - the source being the concerns under focus, the intellectual effort the filling in the sandwich, and the target the revised understanding of the concern [25].

Gentner summarises this as follows: "*...structure-mapping theory is that an analogy is a mapping of knowledge from one domain (the base) into another (the target) which conveys that a system of relations that holds among the base objects also holds among the target objects*" [16].

The relationship between essence and accident is pictured in Figure 27.

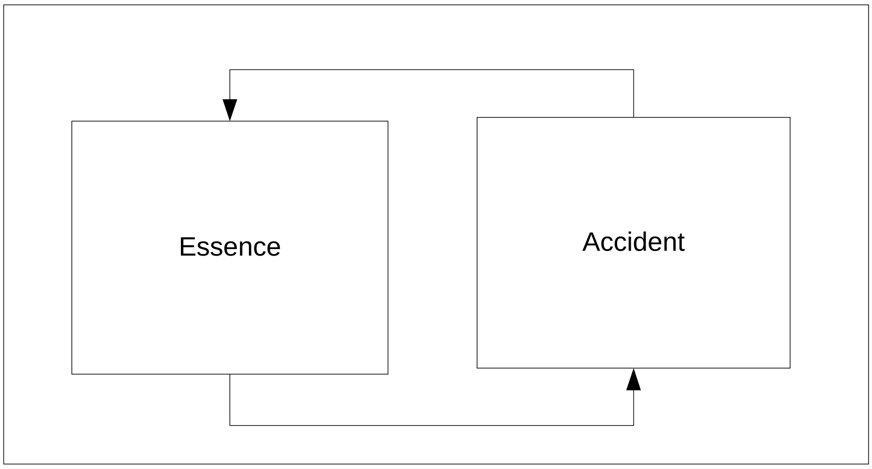

Figure 27: Relationship between Essence and Accident

Essence is suggested to be an intellectual appreciation, delivered as value embedded within the patterns exposed to the information seeker. The accidental components are real or imagined persistent phenomena which act as containers for this value - the value is embedded within the patterns presented by the relationships between the artefacts.

Figure 27 illustrates simplistically the relationship between the realm of intellectual appreciation, where essence resides and the accidental environment (source of sensory stimulus - causal input sources [13]) which is the origin of the intellectual experience. The accidental component can be a 'rose'. The intellectual essential experience is the value appropriated to the internalised value - it could be, for example, wonder of the beauty of creation underlying the physical artefact.

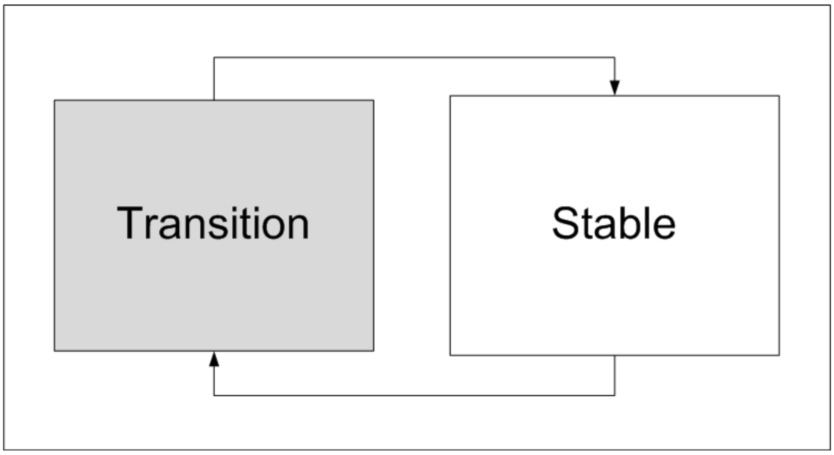

Figure 28: Transitional and Stable constructs

Expanding Figure 27 by introducing the intellectual 'real time' appreciation of the essence based on a persistent physical object is illustrated in Figure 28. If the premise that time is an intellectual construct is explored, then reality can be described as a real time immediate experience. All other experiences are remembered or intellectual constructs of the mind. The stable phase within Figure 28 illustrates the persistent components and the transition phase the phase where change takes place - transition is accompanied by free energy being released or consumed. In the case of neurological processes, this would be the metabolic energy driving the brain as an organ. Intellectual thought is a derivative - it uses the metabolic energy to transition neurological patterns - retrieved to access patterns past, and stored to preserve patterns future.

Transitional phases have the following characteristics:

- They make use of free energy - they are either endothermic or exothermic.
- They are unstable.
- They manifest change.

Figure 29 extends this view and illustrates how one state can be the result of a transition, after which it becomes the origin of the next.

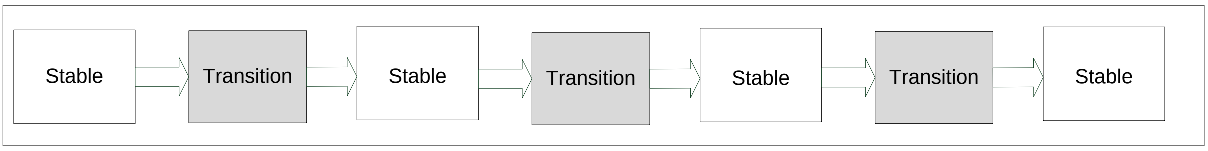

Figure 29: State model as a continuum

Reducing Figure 29 simplifies the model to two absolutes which form an irreducible minimum - a stable state and a state-in-transition. This is diagrammatically illustrated in Figure 30.

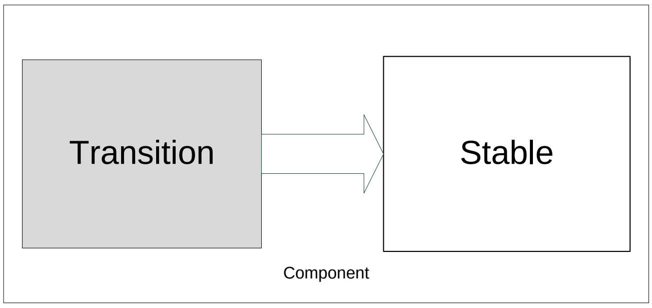

Figure 30: Irreducible minimum

Figure 30 represents a component which when created, undergoes transformation after which it enters a stable state. This state will be maintained as long as there is an absence of free or released energy. The moment further change is initiated, then the destabilisation will result in a second transition to a stable state, and if change continues, a third, then a fourth and further transition will be effected. These transitions will impact the structure of the component in that there will be a realignment of the spatial constituents constituting the component. In the hierarchical model, transitions affecting a component take place within that component - each succeeding state is a subset of the preceding state. On the one hand, there is no 'delinking' of a succeeding state from the preceding state - direct persistence [28] is the mechanism used to propagate change across generations, and on the other hand the temporal moment of the component will be determined by the sum total of the duration of the temporal moments of the siblings. These characteristics and others are used when evaluating the hierarchical transformation model against the network transformation model discussed in Table 1.

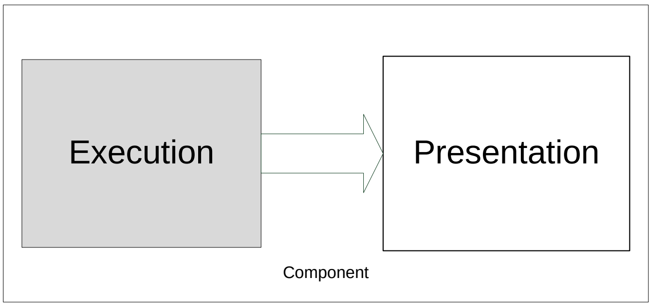

Figure 31: Transition and Stability

'Component' in Figure 31 contains two blocks. The Execution block describes where transition takes place - in computing, this would be where program processing and (digital) spatial reorientation takes place. This phase is succeeded by a phase of stability titled ‘presentation’ - this indicates a point where no transition takes place and the focus of the transition stabilises. Each cycle in the processor is an example of the transition - stable phase. A trigger is required to initiate the next execution phase which will in turn be succeeded by a second presentation state. Note that the presentation - transition - presentation cycle reflects the change of state for a component from state 1 to state 2.

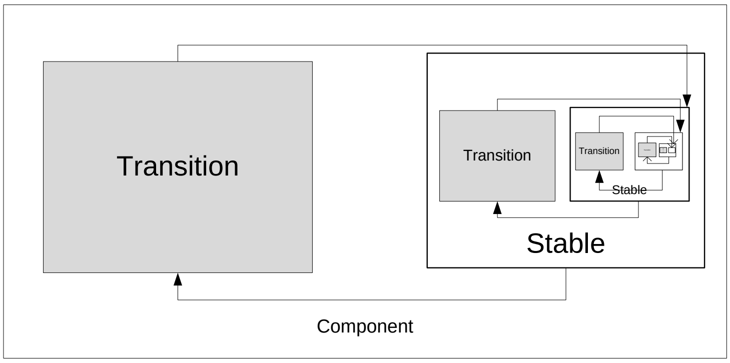

Figure 32: Hierarchical transformation

A hierarchical transformation model which does not employ indirect persistence will not be capable of satisfying the requirements describing the characteristics of an intellectual lifecycle, as listed in 3.7, required when emulating intellectual reflection. As is conceptually illustrated in Figure 32, a hierarchical model expands internally until the lowest node is reached, within an overarching temporal moment [30] of the invoking application together with exploding dependencies as the application deploys. Each iteration adds to the complexity of the preceding processing. Ultimately the hierarchical model becomes rigid and monolithic.

Figure 33 introduces a conceptual model with indirect persistence [28] which accommodates the requirements of an intellectual lifecycle, and sodoing has a capacity to emulate intellectual reflection. Each component is autonomous and anonymous. It is invoked, executes and terminates, without contact with a following generation of components. Linkages between components are driven using metaobjects which are the intermediary between succeeding generation. Metaobjects make use of genetic material [26] to span generations, and allow components to have temporal moments which are restricted to the life of the ‘transition’ or processing part of the component lifecycle.

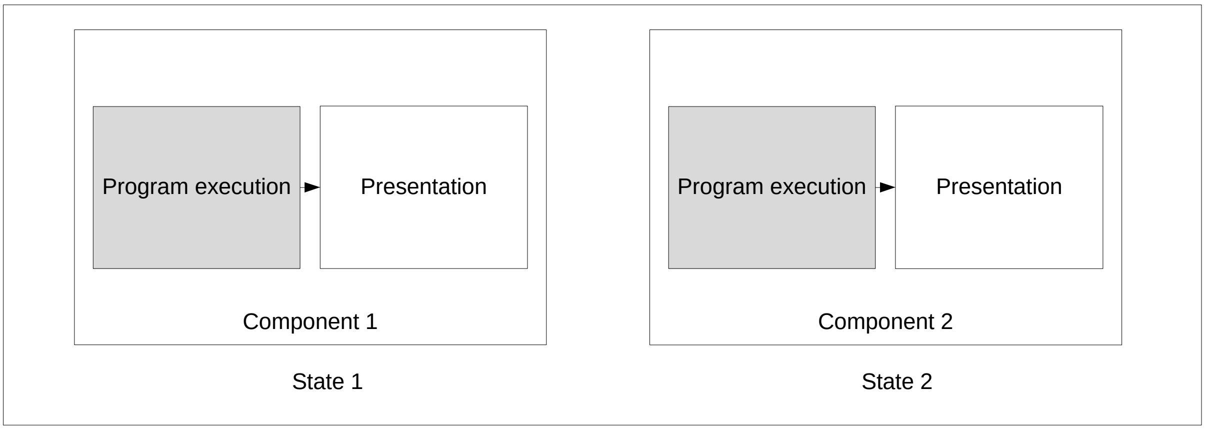

Figure 33: State Model which can emulate intellectual reflection

Figure 33 presents a state change which straddles two generations but no line indicating a relationship is displayed. Component 1 experiences transition, creates metaobjects, reaches a stable state and terminates. Component 1 can terminate (or continue within its temporal moment) once Component 2's metaobjects have been spawned. Component 2, a sibling of Component 1 is created from metaobjects created before Component 1 terminates. There is no direct coupling between the two components. Once Component 2's founding metaobject has been created Component 1 can terminate[4]. At this point, Component 2 would not need to have been created. The broker would consume the metaobject for Component 2 when required, after which Component 2 would be manifested, even though Component 1 has terminated. Figure 34 and Figure 35 labels the relationships between components and discusses how they would need to be adopted in order to move from a hierarchical to a network architectural foundation. This capability to seed a sibling indirectly is the strength of understanding and applying indirect persistence [28].

3.11 Migration from a hierarchical to a network model

Coupling between components implementing a network approach allows the components to be anonymous of one another. The following diagrams illustrate how a hierarchical model

[4] A component after delivering its functional capability expresses metaobjects which manifest as sub-components embedded within the originating component, awaiting selection. When selected, the sub-components then deliver their functional contribution and in turn, spawn their own metaobjects. The sub-components appear to be directly related to the parent component. The reality is that they are components of the following generation with no direct relationship to the parent.

can evolve towards a network model. Two fundamental changes need to be implemented to attain this goal:

- Decoupling of spatial dependencies - achieved using indirect persistence. This introduces a capability to re-use logic - the primary strength of a metaobject based approach.
- Reduction of temporal durations - reducing granularity and atomicity to irreducible minimums for a particular structural requirement sodoing reducing the duration of the temporal moment for individual constituent components.

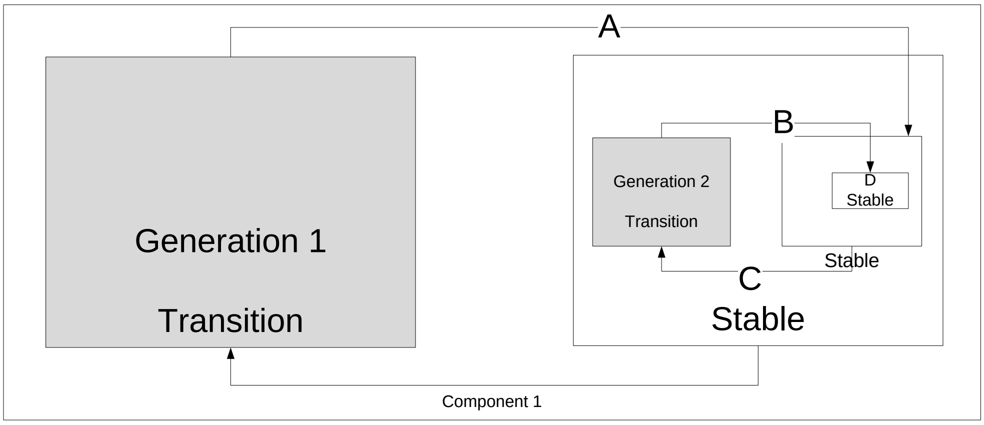

Figure 34: Adaptation of Hierarchical Relationships

Figure 34 indicates a specific invocation (marked A) required to transition a component within a hierarchical structure from an initial state to another. A further change of state for the same component within a hierarchical model would be directly invoked, with the stability and subsequent presentation dependant on the initiating state. One state is built upon the preceding state - applying the rules of direct persistence - in this model it is not possible to invoke a subsequent state without a preceding state being present. This relationship indicates a direct dependence by the succeeding state on the preceding state. Destabilisation of the foundation state results in a knock-on effect to all the downstream states. This ‘direct persistence’ relationship or dependency must evolve towards an indirect persistence model, in that the succeeding state should not have any dependencies on the preceding state if intellectual reflection is to be emulated. This can be done by spawning a next generation *metaobject* component using indirect persistence and then terminating the calling component, as opposed to changing the state within a single component.

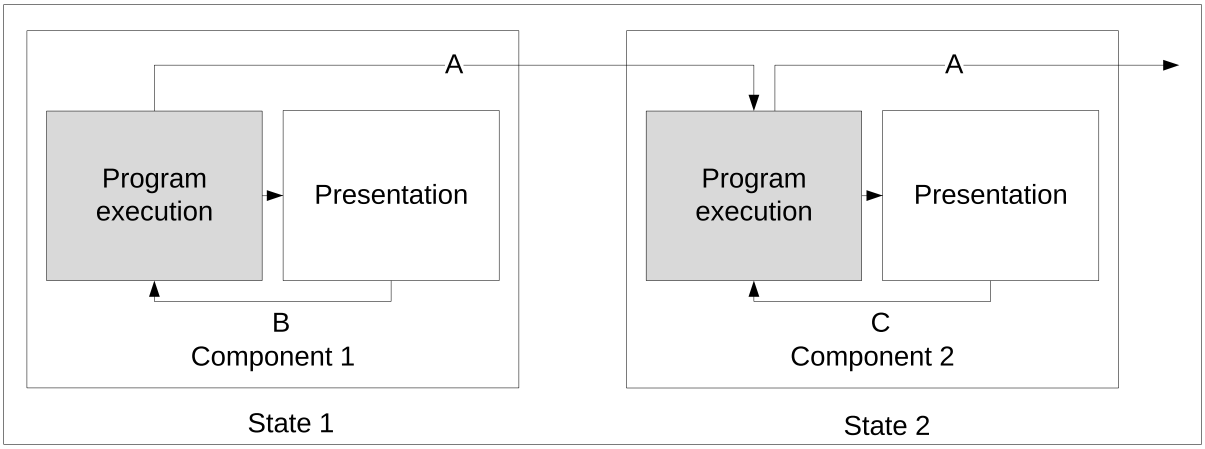

Figure 35: Direct Persistence to Indirect Persistence

Figure 35 illustrates diagrammatically how Figure 34 must be rationalised when promoting indirect persistence and changing away from direct persistence. The relationship indicated by A must change, and component 2 must in some way be indirectly coupled to component A rather than through the direct manner - direct persistence must be replaced by indirect persistence for the two components to become autonomous of one another. A spatial component must be introduced to refine the energy - matter - energy cycle [26] to a finer level of granularity. The following figure illustrates a mechanism to achieve this.

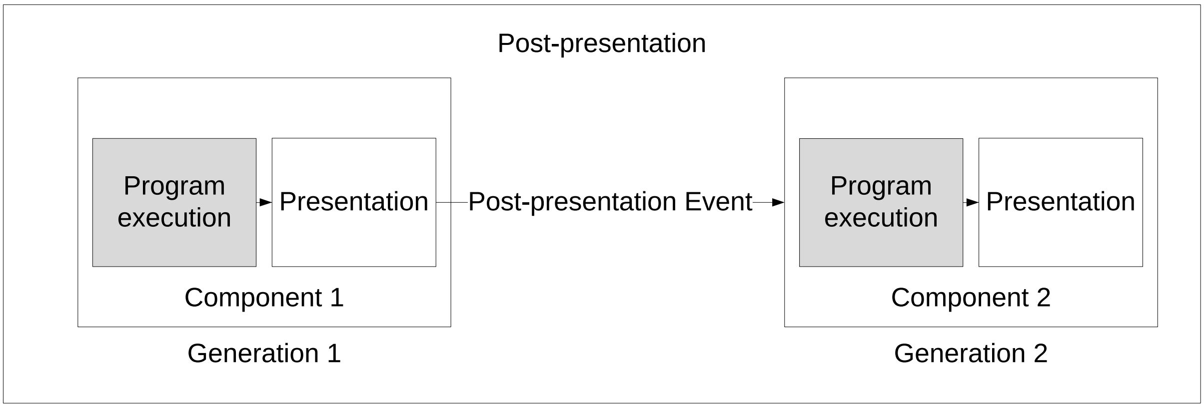

Figure 36: Presentation and post-presentation

Figure 36 expands Figure 35 by substituting A in Figure 35 with the post-presentation capability. B and C are removed. Expanding program execution identifies two categories - the discrete functionality bespoke to the function, and a metaobject stub creation function which would, when selected, express the appropriate metaobject (dedicated for purpose to the required function).

3.11.1 *Program execution*

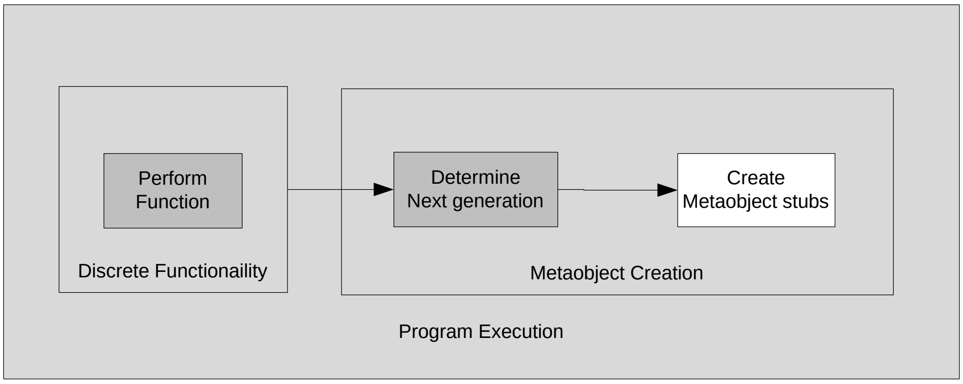

Figure 37: Program execution

Figure 37 expands on the actions required to publish a metaobject. Every 'program execution' process has two fundamental processing legs - the first leg would be to execute the discrete functionality, the second would be to create and express the metaobject stub which would be presented, and when selected, become the 'baton' to carry the discrete next generation values to a succeeding generation, where the components constituting that generation are manifested. Program execution contains two functional components:

- Discrete Functionality

  The 'reason-for-being' of the component is exercised in 'Perform Function' in Figure 37. This is the role of the temporal component, which when reduced to an irreducible minimum, is a single task unique to the function. The temporal moment [30] of the component is defined as the period from the start of the processing period of this function to the end of the processing period for this function.

- Metaobject Stub Creation

  Once the discrete functionality unique to the component executing has been expressed, a metaobject stub creation function is implemented. This function identifies potential permutations which could manifest as a consequence of the discrete functionality, and builds metaobject stubs for each of the possible options. These metaobject stubs are then expressed (displayed as part of the presentation) and await selection after which the metaobject is invocated and consumed by the broker. An example is a button on a form. Once the button is selected, the button function is invoked.

The metaobject stub creation function can spawn one or many metaobject stubs as the requirement dictates. Figure 37 illustrates how components initiate next generation processing by performing a 'metaobject stub creation' process which assembles metaobject stubs and when selected during the presentation state, launches them as metaobjects. The broker consumes these metaobjects and then invokes the subsequent components which then become next generation components. The parent component uses its own 'genetic material' when creating the metaobject stubs. This genetic material is adopted using a 'situational awareness' and a situational pre-cognisance capability identified at the time when the metaobject stub is created. These are discussed further in [26].

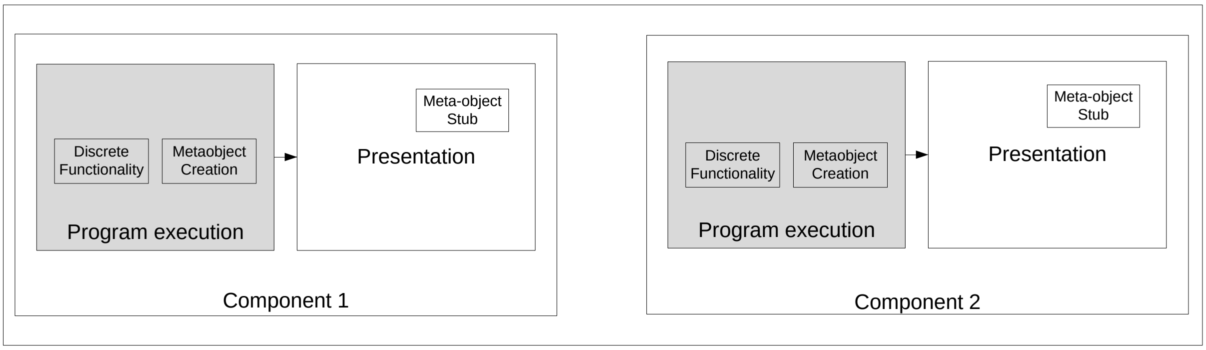

Figure 38: Presentation (Metaobject stub creation) without Post-presentation

Figure 38 illustrates components with the discrete and metaobject stub creation functionality. These are the presentation functions embedded within the component. Note that the post-presentation link has been removed. Retaining this would imply a direct link to the following component - an anonymous indirect mechanism must replace this direct link.

3.11.2 Post-presentation Functions

The role of the broker, discussed below is used to achieve this. The following figure demonstrates how the post-presentation event introduced in Figure 36 is replaced using a broker. Details on the functions of the broker and post-presentation activities are discussed in Figure 40.

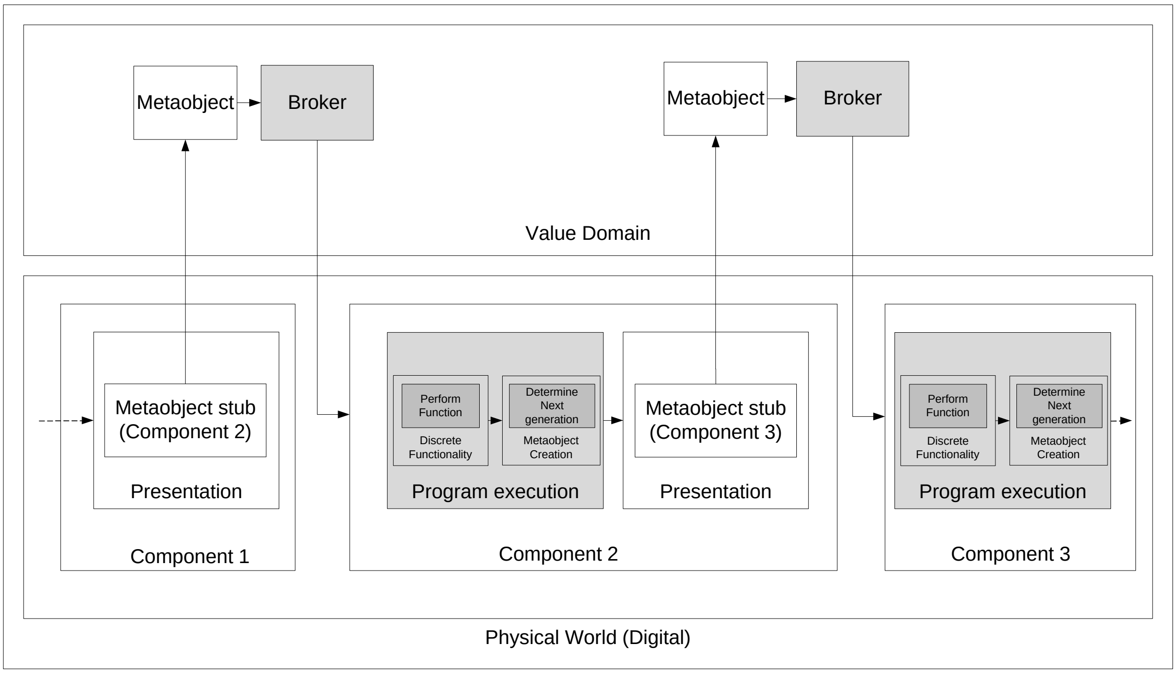

Figure 39: Presentation and Post-presentation using a Broker

The presentation function is the stable period which takes place once processing for a discrete function has been completed. In Figure 39 ‘Component 2’ displays two large blocks - ‘Program execution’ and ‘Presentation’. The ‘Presentation’ block contains the values (including metaobject stub for component 3) produced by Component 2 as output from ‘Program execution’. Embedded within the presentation block is a second block. ‘Component 3’ is a metaobject ‘stub’ awaiting invocation. The metaobject stub, spawned by Component 2, when selected, launches a metaobject which is consumed by the broker and manifested as an independent component - Component 3 on the right hand side of the diagram. The ‘Metaobject stub (Component 3)’ block within ‘Presentation’ of Component 2 represents an icon (or other base manifestation) awaiting a trigger - selection can be a user input, or an interrupt) and activation. In this example, ‘Metaobject stub (Component 3)’ is the only selectable option displayed as an output of Component 2. This example illustrates a panel with a single button. This trigger, for example, a user clicking on a button, launches the metaobject which triggers the broker to invoke ‘Component 3’ which then executes as indicated in the ‘Program execution’ block within ‘Component 3’ on the right hand side of the diagram - a component in the following generation.

Decoupling one generation from a succeeding generation requires that interactions between the preceding and succeeding generations remain anonymous. This implies that an intermediary act as a ‘post box’, and that messages be passed from one generation to the next. These messages are metaobjects, and the post-box is the broker. Metaobject stubs are developed prior to the presentation (stable) phase of the parent component being entered - they form part of a ‘metaobject creation’ function where a metaobject stub is created. When an icon (linked to a metaobject stub) is selected, the metaobject which is then launched is consumed by the broker and a function expressed. The discrete functionality for that expressed component is then invoked and the function executes, after which the function in turn, may create succeeding metaobject stubs. These metaobjects stubs await selection and then processing by the broker which once again launches components and the cycle continues. The logical generic functions of the broker are illustrated in Figure 40.

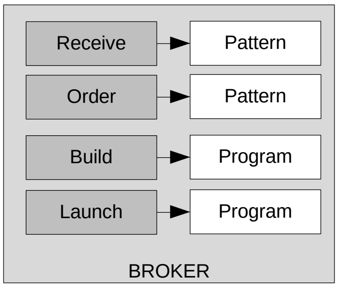

Figure 40: Post-presentation

Figure 40 identifies a universal post presentation activity performed for all metaobjects launched. This is the role of a Broker. The Broker manages transitions between generations of components. It is the agent which enables indirect persistence to be achieved. It acts as an agent which converts a pattern containing value and which is technology agnostic into a parameterised program invocation statement - this statement would need to be aligned with the constraints of the physical domain - technology specific. Metaobjects which are inputs into the broker are technology agnostic, the output from the broker is technology specific. The Broker promotes anonymity between generations. It enables one generation to become anonymous of the next generation. The identity and parameters are accommodated within the invocation statement. The Broker receives a pattern containing the embedded value required in order for a component in the following generation to be created. The pattern is interpreted and converted into a traditional program invocation. This output is used by the broker to invoke the succeeding generation - the next generation component is then instantiated.

3.12 The Network: Input-Process-Output (NIPO) Model - The Network Model

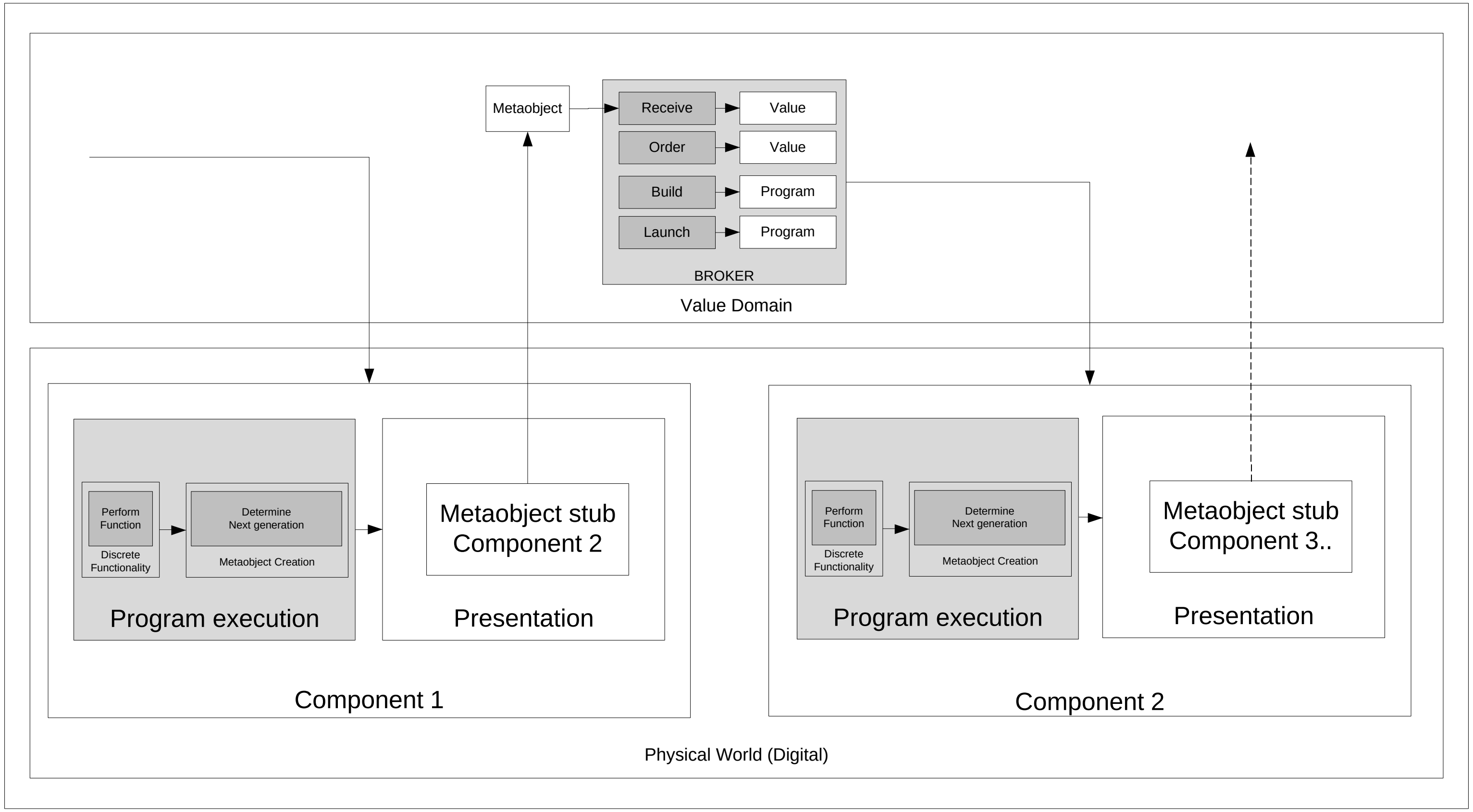

Figure 41: Network: Input-Process-Output

Figure 41 summarises the discussion on presentation and post-presentation of metaobjects. The physical metaobject and broker are contained in a block titled ‘Value domain’. This block represents the environment where value drives the rearrangement and reordering of artefacts. These artefacts are ‘published’ with the genetic material defining attributes of the structural instance required, logic requirements (the method which is to be invoked) the rules (key, table field and formatting detail) and parameters (content and situational parameters). This is executed dynamically and is instantaneous for a standard application not requiring customised requirements. These metaobjects exhibit an ephemeral property. Applications requiring customisation would require that the customisation parameters be retrieved before invocation of the component. Note that these two options (ephemeral and persisten models) are discussed in [26] in a chapter titled ‘Persistence philosophies’ on page 93.

3.13 Sequential and parallel processing using Metaobjects

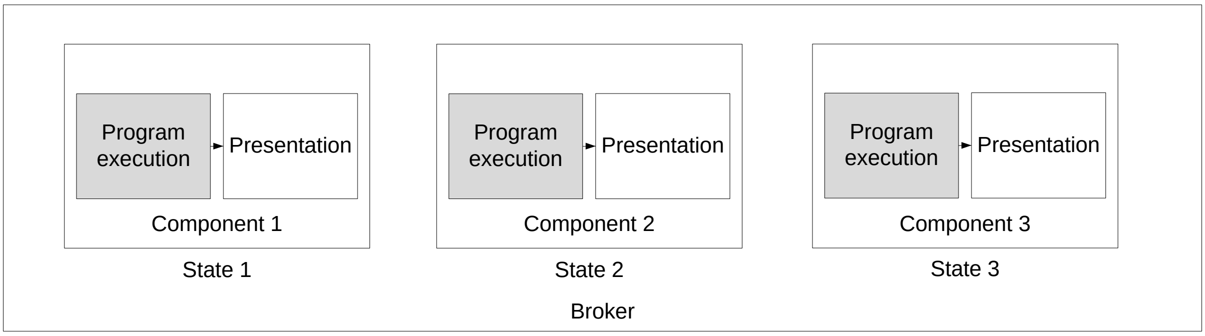

Figure 42: Sequential Expression

A single component in a parent generation can create sibling components. Sibling components can in turn become parents, sodoing propagating across multiple generations as illustrated in Figure 42.

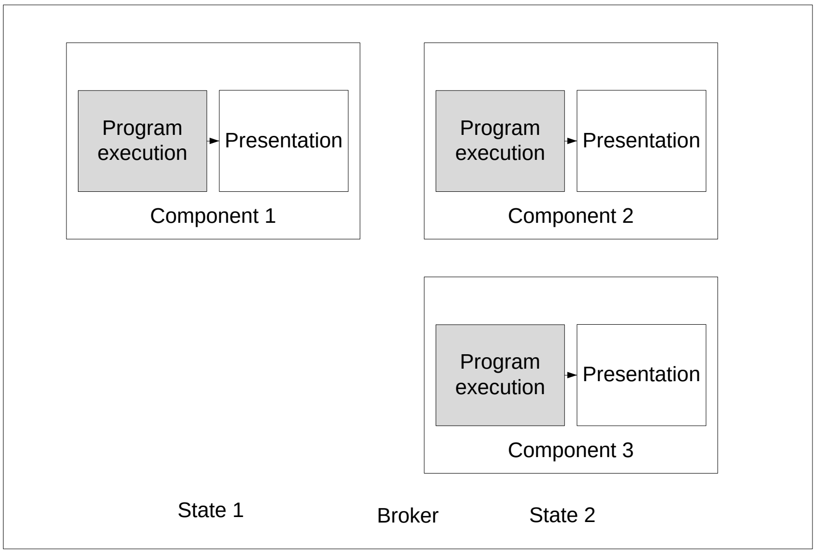

Figure 43: Parallel Expression

Similarly, components can be created which form part of a single succeeding generation where these components execute in parallel, as illustrated in Figure 43. The diagrams do not indicate a relationship using a line (No spatial or temporal (sequence) relationship exists between the components - they are autonomous of one another). The broker uses the metaobject as input to determine the conditions under which the function must be manifested and when the situation warrants it, the metaobject is consumed and the component invoked.

3.14 Emulating Intellectual Reflection

Paragraph 3.4 has expanded on the hierarchy of understanding and 3.9 suggests that intellectual reflection can be considered as a construct to be applied when bridging the divide between the intellectual domain and the physical world. Paragraph 3.12 schematically presents a conceptual mechanism which can be considered when emulating intellectual reflection.

3.15 Constructs derived from the physical primitives

When examining the hierarchical and network models supporting information delivery, the following categorization can be identified.

The key differentiator between the two is that indirect persistence is found only within the Network Model. Figure 41 illustrates how indirect persistence crosses both the physical world and the value domain. Traditional program processing with persistence across generations is implemented indirectly through the publishing of metaobjects. Each discrete component spawning metaobjectstubs awaiting invocation, each invocation launching a metaobject, each metaobject invoking a discrete component via the broker, as the requirement dictates. The broker receives individual metaobjects containing value embedded within patterns. When required, the broker receives this value and invokes the component. Components with embedded metaobject stubs await input in the presentation phase, which then triggers functional processing and the spawning of next-generation metaobjects (where required) which are then streamed to the broker. The broker interprets and translates the value embedded in the metaobjects into traditional programs which are then invoked.

The solution satisfying an intellectual need must have the capability to oscillate between a value driven intellectual approach and utilise physical structures (programs) towards the intellectual purpose in real time. This dualism must be able to dynamically accommodate changing intellectual requirements reflected using standard information technology artefacts. The network model using metaobjects is a candidate which appears able to bridge this divide. It appears to have the ephemeral transient networking, metaphoric capacity required to enable information retrieval and consumption, manipulating value embedded within structures where rules defining logic and content can be orchestrated.

Capability	Reference	Hierarchical Model	Network Model
Hierarchical	[27]	Y	N
Direct	[27]	Y	N
Physical	[27]	Y	N
Network	[27]	N	Y
Ephemeral	[27]	N	Y
Indirect	[27]	N	Y
Anonymous	[27]	N	Y
Autonomous	[27]	N	Y
Optimised Temporal Moment	[30]	N	Y
Direct Persistence	[28]	Y	N
Indirect Persistence	[28]	N	Y
Technology agnostic	Figure 40	N	Y
Pattern-Artefact-Pattern Lifecycle	[26] Page 62	N	Y
Artefact-Pattern-Artefact Lifecycle	[26] Page 56	Y	N

Table 1: Comparison between the Hierarchical and Network Models

3.16 Ability to satisfy the demand for requirements

3.16.1 Traditional solutions without Indirect Persistence

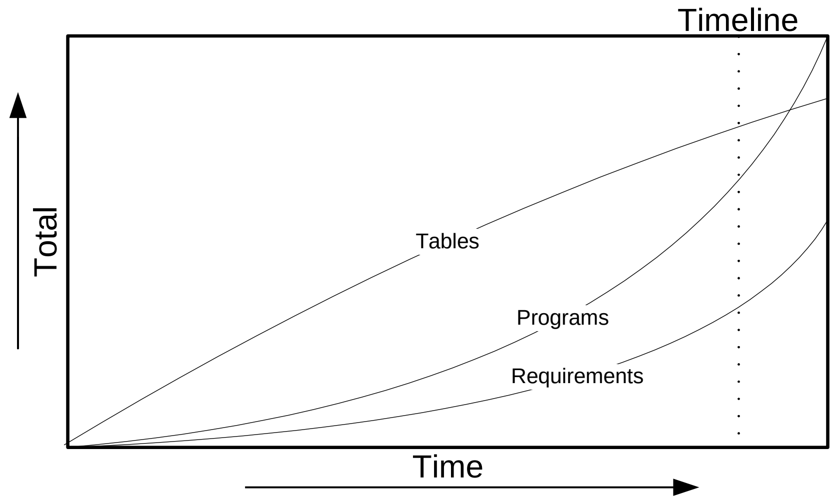

Figure 44: Traditional Programming Approach

Software solutions without indirect persistence are unable to optimally reuse logic. As a consequence, the executable code within a domain grows proportionally as requirements increase. Additionally complexity forces additional integration and interfacing algorithms to be added to the mix. Executable code growth is linked to the number of requirements. The gap between the number of executable and the number of requirements grows exponentially as the number of requirements increases.

Tables can be developed based on the principles of normalisation. As the requirements for a domain grow, so the number of tables required for the domain, initially high, will decrease over time.

Figure 44 conceptually illustrates the relationships between tables, programs and requirements. The number of tables ('Tables' curve) initially grows speedily, but tapers off over time. The number of programs ('Programs' curve) grows at pace, but instead of tapering off, the number increases as interactions with external domains proliferate and internal complexities arise. Requirements ('Requirements' curve) in turn grow over time within a domain.

3.16.2 Metaobject solutions with Indirect Persistence

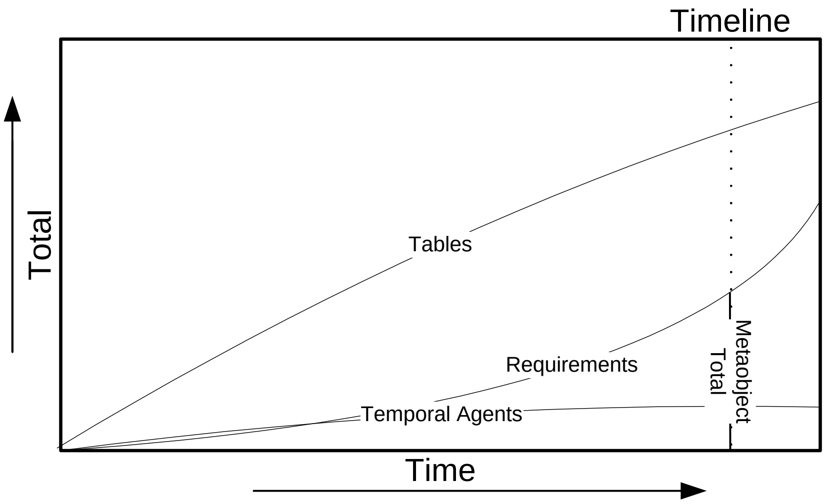

Figure 45: Metaobject Based Approach

Figure 45 illustrates metaobject based solutions which use temporal agents and indirect persistence to deliver information. Metaobject solutions with indirect persistence have a capability to reuse algorithmic components (Temporal Agents). This replaces traditional algorithmic modules with temporal agents. As the number of temporal agents is finite, once the requirements for a domain have been met, a baseline in the number of temporal agents is reached. The temporal agents which make up the suite of re-useable algorithmic components are then available for reuse whenever the logic embedded within that temporal agent is required. This is true across all business domains both within and outside the enterprise.

The total number of metaobjects existing will be defined at the point in time when the requirement must be executed when making use of the ephemeral model described in [26]. In the ephemeral model, no metaobjects exist historically, and no metaobjects other than those spawned in the manifesting generation can be created[5].

[5] This is only true for the ephemeral model. The persistent model will require allows customisation and a capability to not only jump generations, but to go back and restart previous generations. By analogy, a grandchild in metaobject terms, has the capability to change the colour of its grandparents eyes.

4. Research Findings and Conclusions

4.1 Methodology

An experimental approach was taken. Three experimental phases were initiated:

- Code generation using traditional techniques and practices.
- Data driven applications.
- Metaobject approach using metadata including metaobjects making use of metalogic and metarule components.

4.1.1 Methodology Applied

A literature study was conducted and a critical evaluation of current and emerging technologies are presented. A definition of a conceptual alternative is presented. The research is aimed at developing the concept and periodic re-evaluation and determination of future effort was undertaken.

The following principles were adopted.

- The ’raison d'être‘ of this research was to remove redundancy in all its forms thereby promoting information delivery.
- No short cuts would be tolerated.
- Time was not an issue.
- Pursue a concept, not a technology.

This study followed an informal protocol, based on qualified, intuitive reasoning pursuing an intellectual objective established at the outset. The instruments were from an accidental perspective, a Personal Computer, a Compiler, and a Database. The essence was the intellectual goal - to find a mechanism to improve the effectivity of information delivery.

The major portion of the study was spent developing the three prototypes. This development cycled between the following two extremes:

- Conceptualisation of the concept.

- Expression of the concept through experimentation.

4.2 Prototype Findings

An entity-relationship model with the measures shown in Figure 47 has been used to test the prototype. The Entity-relationship model is presented in Figure 47 with examples of one-to-many relations, many-to-many relations, subtypes, and hierarchies.

Number of Entities	122
Number of Keys	251
Number of Key Elements	350

Table 2: Measures in the entity-relationship model

Table 2 presents the metrics of a prototype used to demonstrate the concept discussed in this thesis.

<table>
<tr><td>Development Effort</td><td>4 minutes: Time Started: 11:13:30 Time Ended: 11:17:30</td></tr>
<tr><td>Number of ’Founding‘ Metaobjects[6] created</td><td>9013</td></tr>
<tr><td>Total number of metaobjects in prototype (Assume 20 siblings for each iteration, 5 iterations)</td><td>
<table>
<tr><th>Founding Metaobjects</th><th>Level 1</th><th>Level 2</th><th>Level 3</th><th>Level 4</th><th>Level 5</th></tr>
<tr><td>1</td><td>20</td><td>400</td><td>8,000</td><td>160,000</td><td>3,200,000</td></tr>
<tr><td>9013</td><td>180,260</td><td>3,605,200</td><td>72,104,000</td><td>1,442,080,000</td><td>28,841,600,000[7]</td></tr>
</table>
</td></tr>
</table>

Table 3: Results from a metaobject based application

Table 3 summarises the numbers of metaobjects which would be required assuming there are an average 20 metaobjects constituting an event and there are an average of 5 levels or ‘generations’ spawned per metaobject. Figure 46 below is an example of a screen-print from the prototype. The five callouts identify the metaobjects which form this screen.

[6] A ‘founding’ metaobject is capable of spawning metaobject ‘children’. The number of descendents is related to the number of spatial elements and relationships which exist, and the characteristics of a particular pattern.
[7] It must be emphasized that this is the estimated number of metaobjects. They are never directly accessed – only indirectly. Secondly, in their natural state, they are ephemeral and transient, not requiring disk space. They exist only in the present tense - they become ‘real’ only when customization requires that they be stored as persistent objects for later retrieval.

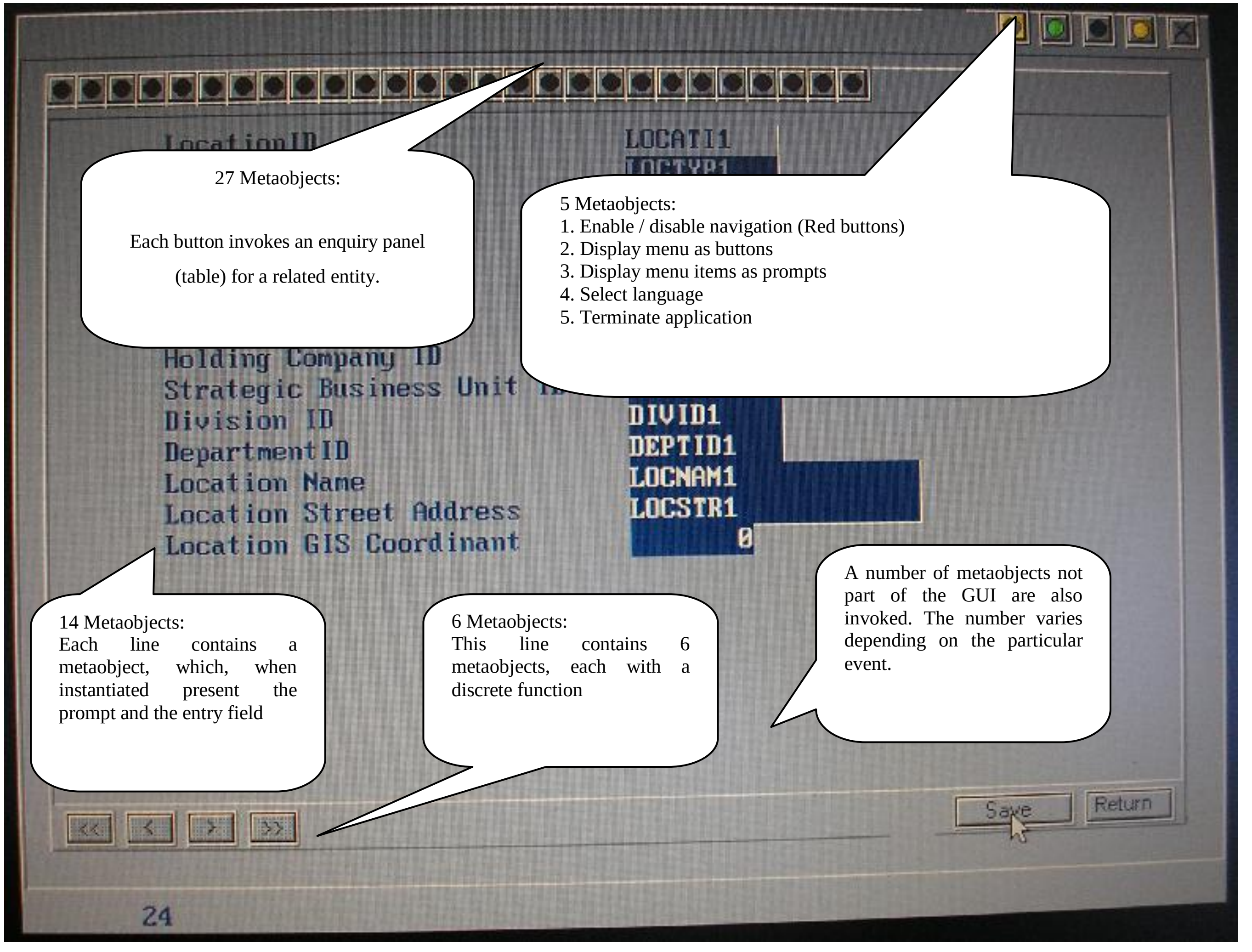

Figure 46: GUI Transaction Screen

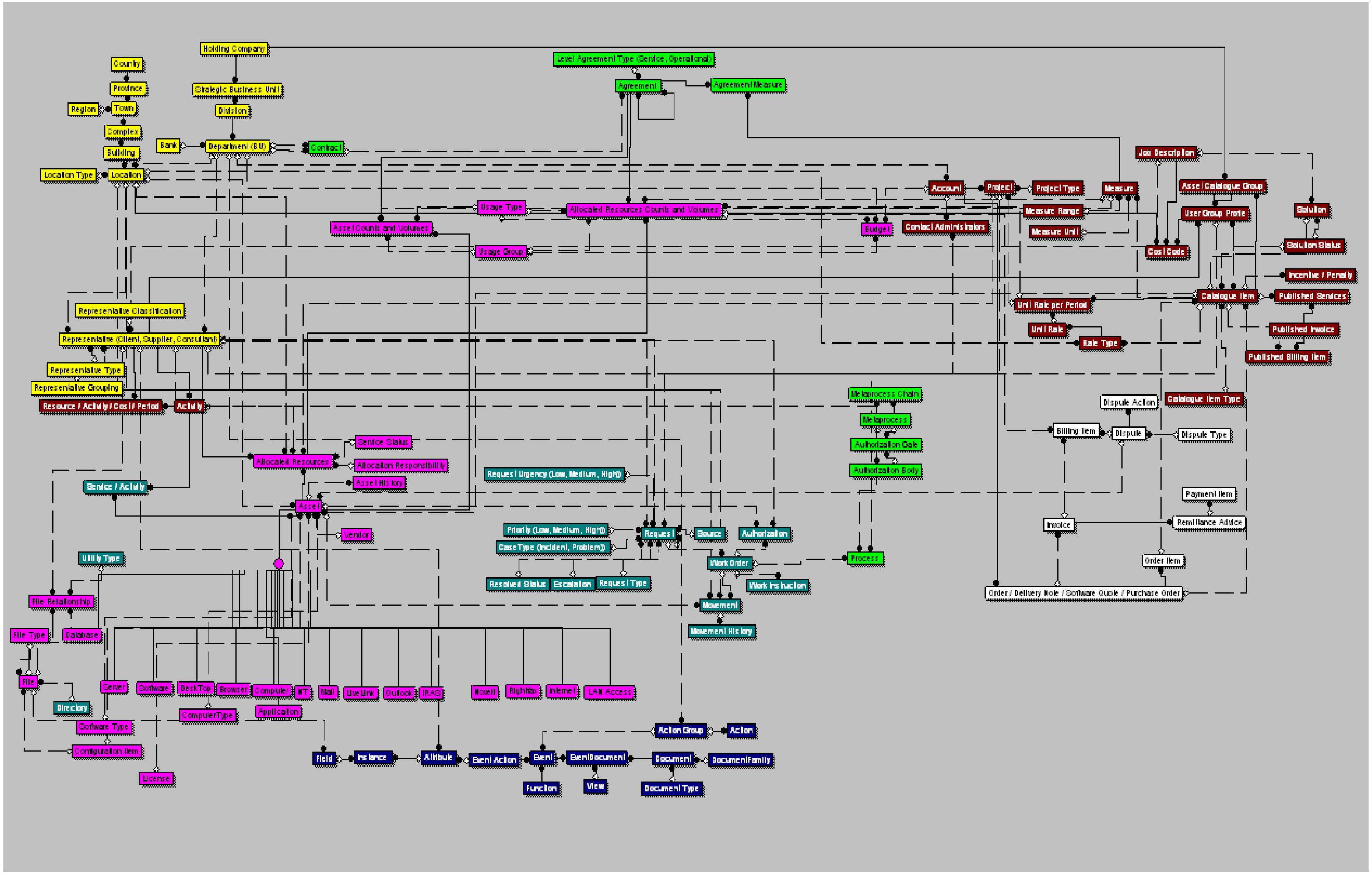

Figure 47: An entity-relationship model

4.3 Independent Assessment

Lombard [31] evaluated WARTHOG - the name given to the concept detailed in this document against the following variables:

- Code generation.
- Application size.
- Need for skilled programmers.
- Delivery speed.
- Complexities.

Her findings are that WARTHOG might indeed be a new and unique approach to application development in terms the association of meta-objects which results in less code generation, smaller application size, smaller need for skilled programmers, faster delivery speed and fewer complexities. In her final chapter she makes the following statement: "*If the claims of WARTHOG hold true in terms of code generation, delivery speed, application size and complexities then this new concept and application development tool could be the silver bullet that has been long awaited to solve the problems of software developers*" [31].

The prototype has demonstrated and confirmed the concepts discussed earlier. Implementing these concepts in an appropriate technology would require a moderate development effort. Only once this has been completed will the potential of this concept become tangible, and benchmarking will then prove its relative worth against comparable technologies.

4.4 Research Findings

4.4.1 Primary research question

- What are the fundamental pillars underlying information delivery?

Findings:

- Two fundamental physical pillars underlying the delivery of information have been identified:

  Logic: The expression of logic is an instrument which allows change to be orchestrated. It is a real time capability manifested within the temporal moment of a component. It is the expression of value, previously embedded within algorithms, and encoded into program code.

  Rules: Rules are used to describe persistent capabilities across generations. Rules guide logic towards administered change.

  A single intangible construct in the delivery of information has been identified:

  Structure: Structure is a pattern containing value which has meaning to the beholder. In information delivery terms, structure in its stable state includes the layout of an interactive electronic image or form on-screen allowing value to be exchanged or it can include the rearrangement of content within a database. A structure is the end result of the rearrangement of bits, bytes, words, columns, rows, GUI artefacts and a host of additional electronic artefacts arranged into a pattern which contains meaning presented to a beholder, or alternatively allows meaning to be expressed by a user for persisting.

4.4.2 Secondary research questions

- Is an optimized information delivery solution using the fundamental pillars of information possible?

  Findings:

  Based on the work to date, it does appear that an optimised information delivery solution is possible. Theoretical research and practical results from a prototype are very promising. Emulating indirect persistence, a property found in nature where value is transferred across generation, allows logic to be reused. The use of reusable logic, together with rules describing the spatial relationships which are achieved when metaobjects are employed reduces the physical size of applications and also appears to have significant potential to significantly reduce the development time for new requirements.

- How can these pillars be best aligned towards providing sustainable information delivery solutions while retaining integrity?

  Findings:

  Structures derived from logic and rule can best be constructed if redundancy is removed in both domains. Redundancy of rule has been achieved using the principles and rules of normalisation. Redundancy of logic, the focus of this thesis can be achieved by emulating indirect persistence. This is achieved by using atomic logic modules which are indirectly referenced using metaobjects orchestrated through a broker. Metaobjects introduce value driven information delivery solutions.

- Will these generic logic components provide a significant improvement in the ability to develop and maintain Information Delivery Solutions in terms of reducing cost, reducing time required and requiring lower levels of expertise for a given information requirement?

  Findings:

  A significant improvement in development time can be demonstrated. The study has found that using metaobjects as a programming tool introduces a meaningful reduction in development effort. Indications are that these reductions will wash through to reduce the maintenance burden. Development effort for metaobject based solutions would move towards focusing on gaining an understanding and delivering the business value needed to attain and maintain a competitive edge. Programming and database administration would decrease significantly. The use of a metaobject based approach together with the implementation of a metadata management capability across the enterprise will facilitate automation of the integration and interfacing complexities currently existing within legacy and COTS (Commercial-Off-The-Shelf) environments when implementing information delivery solutions.

  Customization at the finest level of atomicity and granularity can be undertaken allowing bespoke functionality where required. The prototype can deliver a traditional application using a standard pattern within approximately 5 minutes - as illustrated in the prototype findings. Refer to Paragraph 4.2 for detail). Lines of code (LOC) and function points as productivity metrics would need to be revisited if metrics for the productivity of a metaobject based approach is desired.

4.5 Shortcomings

No shortcomings of significance can currently be discerned in the concept directly. Certainly there are a host of developmental challenges which remain which will need to be addressed when the concept is expressed. These are believed not to be shortcomings, but would need to be resolved using appropriate design practices. As an example, where an implementation makes use of the 'situational mode' repository philosophy discussed in [26], then a real time availability of the repository would be required requiring both 100% up time and a minimum bandwidth level (given the repository is accessed remotely during the session). Current experience indicates that this risk can be mitigated by creating a local copy of the repository before initiating a metaobject based session - size of the repository does not appear to be a limitation. In the same breath, peripheral concerns have been identified which would need to be resolved. These issues include security, release and configuration management, and these too would need to be addressed individually during the design phase.

4.6 Future Research

This thesis provides a philosophical and theoretical introduction into a metaobject based information delivery solution.

Further steps include the following options:

- The approach documented within this study is published.
- It is recommended that a 'Proof of concept' be undertaken where the fundamental concepts can be realised and demonstrated. Further initiatives can then be considered based on the findings of the 'POC'.
- Metaobjects are technology agnostic. The 'Temporal Agents' referenced by metaobjects however, must exist within a particular technical platform which needs to be selected, as is an appropriate database technology.
- Further research and documentation at a more detailed level is recommended. The

following topics are suggested:

- Pattern and workflow research.
- Exploiting metadata.

Additionally, once this concept is validated,

- further research and documentation at a more detailed level is recommended. The following topics are suggested:

  - Impact of innovative and disruptive technologies on industry.
  - Impact of metaobject based tools on existing best practice and disciplines within the Information Technology Industry.

5. References

[1] Alberthal, L. 1995. Remarks to the Financial Executives Institute, October 23, Dallas, TX.

[2] Bateson, G. 1988. Mind and Nature: A Necessary Unity, Bantam.

[3] Bellinger, G. Systems Thinking: An Operational Perspective of the Universe.

[4] Brooks, F. 1987. No Silver Bullet Essence and Accidents of Software Engineering. *Computer Magazine*, April.

[5] Brooks, R. 2001. The relationship between matter and life, *Nature* 409:409-411.

[6] Clocksin, W. Artificial Intelligence and the Future. Philosophical Transactions: Mathematical, Physical and Engineering Sciences, 361(1809), Information, Knowledge and Technology, Aug. 15, 2003, pp. 1721- 1748.

[7] Codd, E. 1970. A relational model of data for large shared data banks. *Communications of the ACM*, 13(6):377-387, June.

[8] Cohen, S. Aristotle's Metaphysics, Stanford Encyclopedia of Philosophy.

[9] Copi, I. 1954. Essence and Accident. *The Journal of Philosophy*, American Philosophical Association Eastern Division: Papers to be Presented at the Fifty-First Annual Meeting, Goucher College. 51(23), December 28-30, pp. 706-719.

[10] Cousin, D. Aristotle's. 1935. Doctrine of Substance. *Mind*, New Series, 44(174). Apr.

[11] Dijkstra, E. W. 1982. On the role of scientific thought.

[12] Dosen, K. 2003. Identity of Proofs Based on Normalization and Generality. *The Bulletin of Symbolic Logic*, 9(4):477-503, Dec.

[13] Dusenbery, D. 1992. Sensory Ecology: How organisms acquire and respond to information, New York: W.H. Freeman & Co.

[14] Fagan, M. E., 1974. Design and code inspections and process control in the development of programs, IBM report IBM-SDD TR-21, pp. 572, December.

[15] Fleming, N. 1996. Coping with a Revolution: Will the Internet Change Learning?, Lincoln University, Canterbury, New Zealand.

[16] Gentner, D. 1988. Metaphor as Structure Mapping. *The Relational Shift Child* Development, 39:47-59.

[17] Gerritsen, L. 2006. Metaphors of the organization discource: in private and public worlds. PhD Thesis. Tilburg.

[18] Gray, P. & Meister, D. 2006. Knowledge sourcing methods. *Information and Management*. 43:142-156.

[19] Guthrie, W. K. C. 1990. *A History of Greek Philosophy*. Cambridge University Press. pp. 148.

[20] Hartman, E. 1976. Aristotle on the identity of substance and essence. *The Philosophical Review*, 85(4): 545-561, Oct.

[21] HIPO. 1974. A Design Aid and Documentation Technique, IBM Corp., Manual No. GC20-1851-0. White Plains, N.Y.: IBM Data Processing Div., October.

[22] Hooper, C. 1917. The Meaning of The Universe. *Mind*, New Series, 26(102): 129-145, Apr.

[23] http://en.wikipedia.org/wiki/Accident_(philosophy), accessed on 17th October 2010.

[24] http://www.systems-thinking.org/kmgmt/kmgmt.htm, accessed on 12th September 2010.

[25] Lakoff, G. & Johnson, M. 1980. Conceptual Metaphor in Everyday Language. *The Journal of Philosophy*, 77(8), pp. 453-486, Aug.

[26] Lemke, R. 2008. From metaphors to intelligent patterns: milestones on the road to code re-use. M Sc Dissertation. North West University.

[27] Lemke, R. & Jordaan, D. 2008. Criteria for the evaluation of an information delivery framework. Unpublished. North West University.

[28] Lemke, R. & Jordaan, D. 2010. Persistence: Direct and Indirect. Unpublished. North West University.

[29] Lemke, R. & Jordaan, D. 2010. Temporal Agents and Spatial Objects - An Overview. Unpublished. North West University.

[30] Lemke, R. & Jordaan, D. Time: Real or Imagined. Unpublished. North West University.

[31] Lombard, A. 2006. A Conceptual Comparison Of Warthog And Conventional Application Development Tools. MEd. Dissertation. Unpublished. Vaal University of Technology.

[32] Maxwell, N. 2000. The Mind-Body Problem and Explanatory Dualism, *Philosophy*,

75(291), pp. 49-71, Jan.

[33] Moore, G. E. 1965. Cramming more components onto integrated circuits, *Electronics Magazine*.

[34] Quintas, P. 1991. Software Engineering: The International Policy Challenge, OECD/ICCP Series, incomplete.

[35] Ranyard, J. 1988. A History of OR and Computing. *The Journal of Operational Research Society*. 39(12), Dec.

[36] Rob, P., Coronel, C. & Crockett, K. 2008. Database Systems: Design, Implementation & Management. International Edition, Course Technology, pp. 808.

[37] Senge, P. 1990. The Fifth Discipline: The Art & Practice of the Learning Organization, Doubleday-Currency.

[38] Von Neumann, J. 1948. Electronic Methods of Computation. *Bulletin of the American Academy of Arts and Sciences*, 1(3):2-4, Mar.

[39] Williams, D. 1958. Form and Matter, II. *The Philosophical Review*, 67(4):499-521, Oct.

APPENDIX A - Entity-relationship model (Figure 47 enlarged)