Viable Computing Systems:

A Set Theory Decomposition of Anthony Stafford Beer's Viable System Model; Aspirant of Surpassing Autonomic Computing

Ruth Thompson

A thesis submitted in partial fulfillment of the requirements of

Liverpool John Moores University

For the degree of

Doctor of Philosophy

May 11

Abstract

This thesis articulates a novel technology: Viable Computing Systems (VCS) [1], which promotes viability within self-managing computing systems. The research momentum was rising software system complexity within the software industry today [6, 7]. Autonomic Computing [8] has been proposed as a solution to this, yet this research advances into the genre of Viable Computing Systems (VCS) by presenting a conceptual model characterizing homeostatic [9] self-governance, thereby innovating within the genre. By examining cybernetic, mathematical, biological and computing techniques, a first-stage, functional, decomposition of Stafford Beer's cybernetic Viable System Model (VSM) is presented from the viewpoint of dually modelling the relationships between the recursive levels of the VSM and between the component systems. By endorsing autonomy versus governance, this research presents a tangible formalism conceptualising homeostasis [9].

This research uniquely presents an algebraic, atomically derived, emergent model that reflects a set theory decomposition of the VSM. This is pertinent by its composition of multiple, yet independent entities sharing one or more objectives. Although the original scope of the VSM was that of human organizations, this work digresses towards its application to autonomic computing system design. The potential to deliver self-managing systems based upon the principles of the human autonomic nervous system is exposed. Since its inception, scope for progression still exists, thereby enabling the presentation of this innovative research that applies a cybernetic approach to the extension of the aforestated software architectural style.

Overall, the thesis presents an expressive grammar as a reference framework, using Beer's VSM as a vehicle to augment the state of the art of autonomic computing into the original field of Viable Computing Systems. This progresses the state of the art by offering an original framework that has the future potential to be translated into code and thus feasibly executed in a real world situation.

Case studies demonstrate a theory of how inherent learning and control is sought through system-environment interplay. By focusing on exchanges and interrelationships, the system demonstrates potential to evolve via environmental interaction. This is achieved through the conservation and management of appropriate resources provided by each entity, so exhibiting proof of concept.

Dedicated to my inspirational late father (my alpha and my omega).

To my family - my loving husband and my beautiful and patient children-

my collective raison d'être:

my gorgeous little ballerinas Georgina and Jemima

and to my sweet baby William - who has been subjected to visiting the libraries and special collections; sitting on my knee, from a newborn to a toddler whilst we wrote almost every iota, in 'virtual' isolation - together! (The hand that rocks the cradle...?).

Also, lest she is forgotten…

for

Emily Wilding Davison

Facta, Non Verba

Acknowledgements

I must thank my supervisory team: Professor Azzelarabe Taleb-Bendiab, my Director of Studies Mr. Andy Laws and my Third Supervisor, Dr. Denis Reilly. I also thank Dr. Stu Wade (retired Supervisor) for his noteworthy prior contribution to the research and am grateful to Dr. David Llewellyn-Jones for his advisory role. Additional thanks to Dr. Mark Taylor and Mr. Mark Allen. Similarly to Dr. Paul Strickland and Dr.Carl Bamford (the latter my project supervisor) - two gentlemen, in the very true sense of the word, who assisted me greatly with my first degree.

Many thanks to the technical staff from within the School of Computing and Mathematical Sciences. Their timely, professional, responses enabled me to undertake my PhD often under extreme conditions. I especially thank Mr. Paul Cartwright and his seemingly endless technical knowledge and likewise Mr. Warren Anacoura, Mr. Steve Thompson, Mrs. Trina Fitzpatrick, Mr. Dave Folksman and Mr. Neil Rowe.

Appreciations to three ladies within the School Office, who have shown consideration, assistance, and professionalism: Miss Janette Skentelbery, Mrs. Loretta Gomm and particularly Miss Lucy Tweedle. Many thanks also to Miss Emily Burningham of the LJMU Special Collections and Archives for both her helpful and timely assistance with viewings, often at very short notice and again allowing William to be in-tow.

Immeasurable thanks to my loyal and true friends who have supported me over many years, particularly my children’s modest Godmothers. To my respective English & Latin teachers at St. Hilda’s High School for Girls - Non Nobis Domine!

To the recently deceased Meir (Manny) Lehman; uninterested in celebrity, yet passionate for his cause célèbre – the advancement of software evolution, via his extensive and groundbreaking magnum opi. The canonical principles underscoring Lehmans' research corpus have been an irrefutable motivation to this study; Age can no longer wither him, nor comfort mar his infinite variety.

To my loving and eloquently principled late father, who gave me his all by imparting just values, a strong work ethic and sepia-tinged memories of a Blytonesque childhood - so inspiring a lifelong love of reading and writing. Finally to my ever precious babies *'How do I love thee…?'* Georgina, Jemima & William – I did this because of you and for you, with all my love ad infinitum, from Mummy XXXX

Contents

List of Figures and Tables

Set Theory Notations Used:

S : a system$\{_{0,1,2,3,4,5}\}$

S : a system recursion$\{^{0,n,N}\}$

i : a system identifier

A : a system set $\{a_1,a_2,a_3,a_4,\ldots_i\}|\ a_i$ *is atomic*

M : a system Management Unit

O : a system Operation Unit

E : a system Environment

t : CurrentTime parameter$|t^{+1}$*:FutureTime*$|t^{-1}$*: PastTime*

Binary Relations:

$A \Rightarrow B$: If A is true, B is also true

Binary Operators:

$A \cup B$: The set containing all of those elements within A and B

$A \cap B$: The set containing those common elements of A and B

$A - B$: The set that results when B is subtracted from A

$A \not\subset B$: The set A is not a subset of set B

Chapter 1

Introduction

This thesis presents research seeking to advance the state of the art of autonomic computing [8]. the main impetus of which is to address the increasing computing system complexity apparent within the software industry today [6].

Horn of IBM, first proposed autonomic computing as a possible solution by voicing the metaphor between the self-governance of biological systems and the requirements of computing and software systems that were growing ever more complex in terms of size and maintainability [8].

This was arguably as a result of both endogenous and exogenous ageing. Endogenous ageing occurs from within due to the act of human maintenance actually introducing greater errors and thus expanding code size, whereas exogenous ageing relates to exacerbating external influences such as the requirement to incorporate new legislation into a software system.

Emanating from the autonomic computing concept is that of Cannon and his:

'homeostatic' [9]

terminology of 1932. This represents the ability of an open bounded biological system, to maintain a stable state in a changing environment by means of self-regulation.

This research attempts to combine cybernetic [9-17], biological [9, 21] and mathematical metaphors [14, 22] to subsume autonomic computing capabilities and so produce the novel concept of a Viable Computer System (VCS) [1]. This is analogous to the subsumation of human systems by cognitive systems. The work is the first known to provide a rigorous formal description of Anthony Stafford Beer's Viable System Model (VSM) . that can apply in an autonomic computing context.

The objective is to fuse a mathematical analogue with the underpinning functionality of Beer's VSM. Toward this end, a bi-perspective set-theory blueprint has been developed as a basis of a design grammar model, resulting in a VCS architecture [1]

An initial closed-environment case study [2], derived from an experimental, environmentally adaptive system within the context of a previous genetically modified system scenario [23], has hence proven the VCS research concept to prototype level.

A second, open-bounded environment case study [3, 24], uniformly applied within the same genetically modified system scenario [23], has demonstrated a theory of systemic self-organization via homeostatic-like behaviour through reference to Sommerhoff's' directive correlation tenet [25] and Ashby's goal directedness [22] notion.

Having disseminated this research widely [1], this work not only exhibits but demonstrates proof of, the original concept of VCS.

1.1 Motivation

The main motivation for this work was the rising problems of complexity within the software industry [6], dictating the need for dynamic flexible systems capable of dealing with such complexity. An example, in Figure 1.1 depicts how the Windows family of operating systems have increased in lines of code, exponentially over the years almost in accordance with the exponential growth predicted by Moore's law [26, 27].

It is the case that IBM used to pay their developers by line of code written, arguably therefore contributing significantly to over-complexity apparent within software today [28].

Amidst this growth in complexity developers are essentially using the same procedures and practices, with the same tools, to create applications that must possess far greater functionality and processing capabilities. For this reason the VCS research seeks to address the complexity problem through viability i.e. the ability to maintain acceptable levels of operation in a potentially hostile environment. More specifically the work presents a formalized design approach to assist in the development of a VCS model.

This motivates and drives the entire research agenda, and this is such an important piece of work for computing in general. This is demonstrated by example areas where software complexity has had disastrous results, such as where large software projects have failed due to their complexity, including the NHS connecting for health project [29]. As Kweku Ewusi-Mensah stated:

> *"one-third of software development projects fail or are abandoned outright because of cost overruns, delays, and reduced functionality."*[30]

2000	NT 5.0
2001	Win2k
2002	XP
2009	Win 7

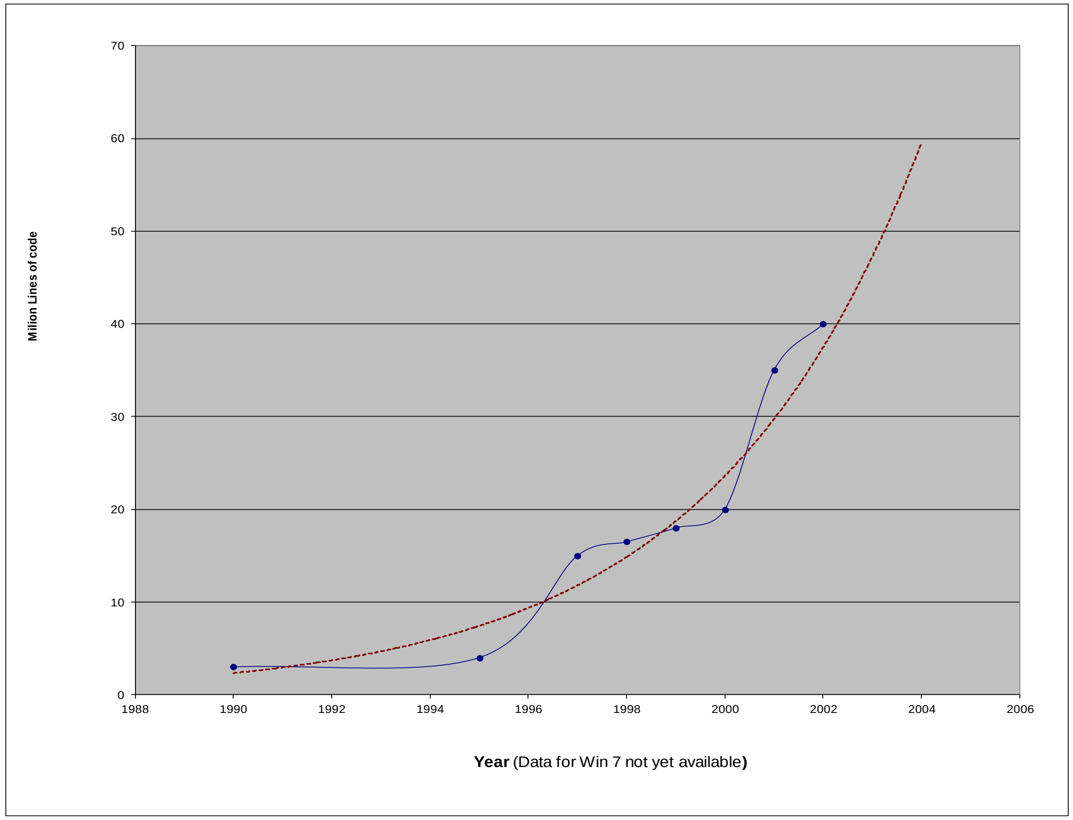


Figure 1.1: Exponential Increase in Growth within Windows Operating Systems

1.2 Statement of the Problem

We seek to articulate a novel technology progressing resource management within self-organizing systems through examination of both cybernetic and biological metaphors, qualified by mathematical and autonomic computing techniques.

This research hence ultimately aspires augmentation of the autonomic computing state of the art into the original field of Viable Computing Systems. The main research problem that is considered and attempted to address in this thesis is:

> *"How can the standard autonomic computing design be fused with Beer's Viable System Model to exhibit systems viability as first class behaviour, that is, to remain operational in the face of a hostile changeable environment through the innovation of a portable formalism?"*

To study this problem, the thesis sets out to address the following specific research questions:

- *What are the self-governing attributes and parameters of Beer's VSM that need to be modelled, to extend the autonomic computing genre?*

- *Can mathematical set theory be used to define a design grammar model to express the structural aspect of the VCS?*

- *Can the design grammar model be applied to develop a VCS architecture?*

1.3 Aims and Objectives

A distinction exists between the VCS general, high level aims, which include the desire to create a tool for improving development of complex software, as opposed to the specific objectives. The latter elaborate on the aims in a way that allows them to be tested; as follows:

The aims and objectives of the thesis fall into the two general categories of Analysis and Design, namely:-

I. Analysis:

Which aims to provide a thorough analysis of the VCS generic requirements, including: intra-systemic sub and superscript models relating to the VCS use. In more detail, this will involve the following activities:

a. A study of the state of the art of autonomic computing reference model.

b. A specific study of the attributes and relevance of Beer's VSM to the research goal of a VCS.

c. Analysis of the VCS requirements, encompassing both functional requirements, concerned with the parameters of maintaining emergence of a self governing system within an open environment such that the system must address and monitor operational requirements, concerned with how the VCS will interact with and, crucially, incorporate the said environment as part of that system under examination.

d. A formal, dual-perspective specification of the relationship between the systems and recursive levels and of the VCS, respectively.

II. Design

Which aims to develop a VCS architecture that implements the concepts and models emerging from the Analysis and demonstrates the use of the design grammar model. In more detail, this will involve the following activities:

a. *Design of a VCS that reflects the non-hierarchical infrastructure and self-governing attributes emerging from the analysis stage.*

b. *Development of an algorithmic dual-perspective design grammar model that represents the VCS requirements emerging from the analysis stage.*

c. *Representation of the design grammar model to provide architecture that supports the VCS requirements and functionality of self-governance via the monitoring of intra and extra-systemic environmental behaviour, without human agent intervention.*

d. *Evaluation of the architecture through a portfolio of case studies that demonstrate its suitability for the application to the VCS genre.*

e. *Appraisal of the relative success of the work and suggestions for extension and/or future directions for others to consider.*

1.4 Research Contributions

The foremost novel contribution to knowledge emerging from the research stems from a proposed VCS design grammar and the associated VCS architecture developed throughout the thesis.

Design Grammar Subscript Model of the Relationship between the VCS Sub-Systems [1]:

The design grammar subscript model provides a unique analysis of the VCS composition of multiple and independent entities sharing one or more objectives, which is not addressed elsewhere in the literature. It was developed using set theory syntax and modelled via reference to Beer's VSM and the VCS classification proposed by the author.

The subscript model reflects integrated management promoting each sub-system as a whole whilst examining the basic elements to be monitored and their functional and operational requirements.

Design Grammar Superscript Model of the Relationship between the VCS Recursive Levels [4].

The design grammar superscript model provides an original classification of the recursions in terms of their roles and usage context. It was similarly developed using set theory syntax and modelled via reference to Beer's VSM and the VCS classification proposed by the author.

The design grammar superscript model identifies a set of primitive relationships between the recursions via set theory syntax. It is anticipated that this may serve as a reference scheme for future researchers within the field and is the culmination of previous research published by the authors.

VCS Architecture:

The VCS architecture represents a theory of how the developed design grammar could be applied to the VSM configuration. This modified structural representation thus acts as a medium toward the development of Viable Computer Systems.

Together, with the foundation of the J-Reference Model [31], they aim to demonstrate the potential for formal modelling in the field.

VCS Case Studies:

The VCS case studies provide a novel contribution in that they demonstrate an application, and thereby the validity, of the design grammar model using a previously published, real world application scenario [23].

The respective closed-bounded [2] and open-environment [3, 24] Case studies, considered in the author's previous research publications, suggest abstract protocols to manifest the VCS concept with the thesis providing a more detailed coverage. The VCS case studies demonstrate theories of viable self-governance, complexity reduction via inherent recursion and systemic emergence.

Progressed VCS topologies are also exhibited here, demonstrating and relating how the design grammar may not only be extended, but also applied to the prior VCS research corpus [1] and reflecting the states and axioms governing the design grammar model.

Overview:

A set-theory oriented, atomically-derived, emergent model has bee evolved that reflects an algorithmic decomposition of Beer's recursive, multi-agent Viable System Model, pertinent by its composition of multiple and independent entities, sharing one or more objectives. The integrated management promotes each sub-system as a whole within a metaboundary. The relationships between sub-systems are demonstrated via syntax subscripts, while the relationship linking recursive levels is recognized via superscripts.

In this way, the resultant design grammar innovates within the genre via endorsing autonomy versus governance through exploitation of cybernetic, biological and mathematical metaphors. The VCS crucially introduces a theory of inherent learning and control through conceptual system-environment interplay. The focus has been on dually modelling the interactions and inter-relationships between the self-organizing systems and their respective environments. The VCS thus exhibits a notion of evolution or emergence, of those systemic elements achieved through conservative management of the resources provided by each entity.

The work will thus further the art of software development in the future by providing a set theory blueprint that has the potential to translate recursivity into a coded program.

An aspired research goal is, however, to specify the underlying significance of addition, or union, and any potential subtraction, multiplication and division, operators. Discerning concatenations of the elements' structures is, however, context dependant, due to the vast range that could be added or subtracted. Examples are provided within the case studies, of how these operations might be specified, in a real software environment.

It is believed that this work creatively innovates, as it is believed that third parties are yet to realize such a VCS framework; encompassing a design grammar model and related architecture.

1.5 Scope of the Thesis

The thesis sets out to determine a means to surpass the autonomic computing genre by the innovation and real world application of Viable Computing Systems through case study platforms [1]. The latter extol the manner in which VCS may be applied.

The thesis does not purport to serve as the definitive design of a VCS, but to present a theoretical, prototypical model towards the development of a feasible VCS architecture. By relating the self-governing attributes of Beer's VSM the notion of a realizable viable software system is presented and promoted. A view exits within the software industry that software complexity initiates Legacy System Syndrome [6], that is the growth of systems which are typically large and difficult to manage.

The thesis describes work applicable to, yet superseding, the class of autonomic computing systems. Modelled with algorithmic formulae, the work relates and fuses the functionality of Beer's VSM ; including its intra and extra-systemic modelling capability. The latter propones a sense of self and thereby viability, whilst similarly reflecting real time interaction with the environment in which it is embedded via attempting to represent feedback and feedforward control.

Similarly, the fractal-like, complexity-reducing, recursive architecture satisfies the requirements of autonomic computing systems as espoused in Horns self-CHOP acronym. An algebraic set-theory model forms the basis of a bi-perspective VCS design grammar that is context free.

It was, significantly, determined to incorporate the environment as part of the system-in-focus, so allowing portability to differing E-type [32] computing scenarios, as demonstrated by the VCS case studies and architecture.

The culmination of the thesis is the development of respectively closed and open environment case studies demonstrating proof of VCS concept. They also obviate the potential performance and behaviour of the modelled software components within a real world scenario, illustrations demonstrating an application framework of the extended design grammar to a novel architecture.

1.6 Thesis Structure

The thesis is structured as follows:

Chapter 2: Explains the background, terminology and fundamental concepts relating to autonomic computing and cybernetic development.

Chapter 3: Presents a literature review of related research from early foundations up to the state of the art practices of today, thereby prefacing and contextualising Beer's Viable System Model (VSM), whilst also situating and elaborating upon it. Several related Models are likewise described in the context of autonomic computing.

Chapter 4: Appraises how the VSM has been synthesised towards development of Viable Computing Systems (VCS). The research is reviewed, relating how historical consideration has directed this. State of the art developments and a descriptive overview is apprised in additon to a representative architecture of the VSM topology post-application of the design grammar model.

Chapter 5: Presents the formal model of the proposed VCS design grammar, as underpinned by the basic operations and attributes of Beer’s VSM in the context of furthering Horn’s autonomic computing self-CHOP acronym.

Chapter 6: Describes the case studies evidencing the applicability of the design grammar models, whilst assessing the relevance and use of the associated VCS architecture configurations.

Chapter 7: Evaluates by drawing overall conclusions on the novelty of the research, identifying aims and objectives for future related investigations plus a synopsis of the contribution of the thesis.

Chapter 2

Background

Introduction

Overall, the VCS research outlined within this thesis, conserves strong multidisciplinary and philosophiocal underpinnings that both inform and direct the study.

2.1 IBM and Autonomic Computing

The anticipated millennium bug problem that raised concerns within the software industry towards the year 2000, arguably instigated Horns' subsequent:

'Grand Challenge' [8]

encapsulating the autonomic computing wish list of 2001. This launch of the IBM autonomic computing initiative in March 2001, led IBM's Paul Horn to voice the rising problems of complexity within software computing systems. Through a seminal address at Harvard university, Horn spoke to the National Academy of Engineers, seeking to highlight and address the existing and worsening problem of complexity within the computing system industry, dictating the need for a discipline of dynamically flexible systems.

Arguably the most challenging aspect of developing software is that of ensuring its' reliability and viability in not only the immediate but also the long-term future. Not only are the technicalities of this problem great but also the resource usage in terms of fiscal and man-hour expenditure are enormous and consume a large ratio of an organization's annual budget. Historically, a developer's primary objective has been to produce a system that adheres to the requirements specification and not to look beyond the initial implementation of such a program towards the long-term maintenance of not only the software but also the hardware, other peripherals and the communication network.

Today's computer system elements are sophisticated, expensive and technology resources, and specialist, responsible humans are scarce. Such systems cannot be controlled centrally, but form complex self-organising computational environments, for which decentralized forms of control have to be invented. The research departments of the big IT companies have in recent years started initiatives, such as IBM's autonomic computing [8], Microsoft's .NET, Sun's N1 and HP's Adaptive Infrastructure, to focus on this kind of control. Aspiring to redress the growing problem of such legacy systems syndrome, IBM propounded its autonomic computing initiative .

This devolved a sea change in the approach towards developing and retaining the viability of a computer system, via drawing a correlation between its nature and that of biological systems. This seminal research direction was expanded upon in IBM’s October 2001 manifesto which aspired to enable organizations to efficiently accommodate and manage an ever-increasing complex environment that is comprised of software, hardware and the necessary communication infrastructure [33]. Through radically choosing terms with biological connotations, IBM drew a correlation between the autonomic nervous systems of the human body and the necessity to transpose its associated capabilities holistically to autonomic computer systems of the Twenty-First century .

Project eLiza was launched by IBM in April 2001 to integrate autonomic capabilities into its products and services [34]. This encompassed not only the computer software per se, but also reassessing the design approach to the servers, storage, middleware and IBM’s support services. November 2002 saw the 1st ACM SIGSOFT Workshop on Self-Healing Systems [35]. Followed in 2003 by the 1st International Workshop on Autonomic Computing Systems [36]. This was an official recognition of the momentum that had been generated by Horn’s biological metaphors, as by the time of IBM’s October 2001 manifesto, it was the protagonist within the research genus.

The document outlined eight key elements or characteristics that such a high-level autonomic computing system should posses, the initial self-CHOP (configuring, healing, optimizing, protecting) acronym, constituting a self-governing metaphor from animate biological system. In essence, Horn’s orations aspired a software system that would negate, or reduce, the requirement for human-agent intervention.

The autonomic concept is clearly based upon the notion of biological self-governance, specifically by the human autonomic nervous system as illustrated in Figure 3.4. To-date, the eight characteristics elucidated by IBM include:-

Self-knowledge:

Detailed knowledge of constituent components, current components status. In simple terms, therefore a system must thus know itself and comprise elements that also posses a system identity.

Self-configuration/re-configuration:

Action adjustments to a changing environment, that is configure and re-configure itself under varying and unpredictable conditions, thereby accommodate a permeable and thus open-boundaried environment.

Self-optimizing:

Monitor constituent parts and optimize accordingly. An autonomic computing system never settles for the status quo – it always looks for ways to optimize its workings.

Self-healing:

The ability to recover from malfunction. An autonomic computing system must perform something akin to healing by having the potential to be able to recover from routine and extraordinary events that might cause some of its parts to malfunction.

Self-protecting:

Detect, identify and protect itself from attack. In the context that a virtual world is no less dangerous than the physical one, an autonomic computing system must be an expert in self-protection.

Environmentally aware:

Know its environment, the context surrounding its activity and act accordingly. An autonomic computing system will know its environment and the context surrounding its activity, acting accordingly.

Co-operative:

Interact with other systems in a heterogeneous world and propone open standards. An autonomic computing system cannot exist in a hermetic environment.

Anticipatory:

Anticipate and transparently implement the resources, that is perhaps most critical for the user. An autonomic computing system will anticipate the optimized resources needed, while keeping its complexity hidden.

The question of how systems with all of these characteristics may actually be built is still very open to debate within the genre.

2.2 Cybernetic Development

Cybernetics [37] can be viewed as a cross-disciplinary approach developed in the 1930's and broadly encompassing contributions from biology, social sciences, operations research and nascent computer science.

Cannon's 1932 text '*Wisdom of the Body'* [9] first coined the term homeostasis which referred to the biological control and management of functions within living organisms.

Subsequent to this, Ashby was a major protagonist in pioneering the field of cybernetics through his key contributions such as his notion of variety [11], requisite variety [14] and Conant-Ashby theorem [17].

One of Ashby's foremost contributions to the genre from the 1940's onwards was that he was the first to formally transpose the biological connotation into a tangible format through his ultrastable homeostat machine [10]. This incorporated two mutually dependant subsystems as a holistic system. Each of the two subsystems contained constant stable states that represented the stability of the whole or holistic system. Both the system and the environment in which it exists are represented by a set of variables that represent that form of a state-determined system. As demonstrated in figure 2.1, when the stable point wanders as a result of environmental disturbance, it is homeostatically drawn back to its original 'safe' point. The environment is consequently defined as those variables whose changes affect the system and those variables that are affected by the system.

Ashby's' Law of Requisite Variety acknowledged systemic complexity as:

'*… Variety…'* [11]

that is, the number of different states a system can adopt. It continues:

'The variety in the control system must be equal to or larger than the variety of the perturbations in order to achieve control' [11]

Ashby's Law concerns controllers attempting to maintain stability within a system. The more options the controller has, the better it is able to deal with fluctuations in the system. Variety of input can only be dealt with by variety of action, or as he states:

'Only variety can destroy variety…' [11]

This seemingly counterintuitive principle has important implications for practical situations: since the variety of perturbations a system can potentially be confronted with is unlimited. Whilst systems should be prepared to deal with the current situation, they should also be prepared to be able to learn in new situations so as to be optimally prepared for any foreseeable or unforeseeable contingency. This was to be followed decades later by the equally relevant Conant-Ashby theorem stating that:

'Every good regulator of a system must be a model of that system.' [17]'

Crucial to the VCS research is that self governing homeostasis forms a fundamental building block to its' development . Forerunners to Ashby's research included the neuroscientist Warren McCulloch and logician Walter

Pitts. In 1943 they collaborated on their first and seminal paper, enigmatically entitled:

'A Logical Calculus of the ideas Immanent in Nervous Activity' [38].

Beer later declaring this McCulloch's:

'...first major work that he wrote in the field of cybernetics' [39]

In the spirit of the time, their project was an application of mathematical models to physiological and mental phenomena. Their novelty primarily lay, however, in equating the operations of reason with those of binary-logic neurons. As such, they believed that mind stood for the embodiment of the site of command and control. The rigorous logic of the paper for the first time began to elucidate the difference between brain structures and mental contents.

This model also made it clear for the first time in neurophysiology, that there must be inhibitors in nervous nets as well as excitors.

In McCulloch's terms, when a system gains the ability to construct its own sensors, or:

"this ability to make or select proper filters on its inputs" [40],

it becomes organizationally closed. In the context of the VSM , this dictates that the system then self-governs, by gaining control over the kinds of actions it has available to influence the environment of the system in focus.

The system subsequently controls the distinctions it makes on its external environment, acquiring the ability to construct its own effectors, thus gaining control over the kinds of actions it has available to influence the world. The self-construction of sensors in the context of the internal state of the system, including purposeful, goal-directed, active perception, and effectors to accomplish a variety of tasks including exploration and manipulation of the environment, incorporating the design and construction of tools towards this end. This thus leads to an epistemic autonomy, where the organism or device itself is the major determinant of the nature of its relations with the world at large.

In biological systems, regenerations of parts and reproductions of whole organisms are the central concepts that define their structure [41-44]. Regenerative processes encompass the flows of energy, material parts and functional relations that are necessarily continually recreated from system-actions to promote viability. This enables informationally-open systems to continually reproduce their internal functionality and thereby maintain their identities over periods of time. The regeneration of relations between material parts forms the basis of both self-construction and repair, a related form being to de-emphasize the role of biological symbols, e.g. the autopoietic models of Maturana and Varela [41, 44, 45].

In the case where structures and functional systems are continually regenerated by internal mechanisms, some degree of material and functional closure is achieved. This closure, in turn creates domains of relative structural and functional autonomy wherein invariant structures and functional relations are preserved by internal rather than external processes. Viability and emergence is a consequence of this closure of production of material parts and relations, or structural and functional self-causation.

In essence, the system reproduces its parts and thereby it’s whole. Closure creates an internally-controlled functionality that is self-produced and an outside realm of relatively contingent processes that are not produced by the self-production loop. Closure and autonomy are partial for systems that are in constant interactions with their environments. For an informationally-open system there must always be some contingent interactions with the external world.

Pask has proposed this organizational closure as one of the constitutive conditions for consciousness:

> *"A process is potentially conscious if it is organizationally-closed, informationally open, and if information is transferred across distinctions that are computed as required to permit the execution of the process."* [46].

The self-construction of sensors and effectors thus leads to an autonomy, and so the VSM is the major determinant of the nature of its relations with the world at large [47, 48]. This central theory of structural closure [36] and its consequent, functional autonomy elicit many of the closely related notions of semantic closure; that is, relationships to sensory and motor linkages with the external world [49], autopoiesis, or self-production [44, 50, 51], self-modifying systems [52] and anticipatory systems that, by definition, contain a predictive model of itself and/or its environment. The latter thereby enables it to change state according to the model's predictions [53]. The recurrent:

"nets with circles" [38]

concept of McCulloch and Pitts was able to show that such nets are equivalent to a Universal Turing Machine [54]. The latter was considered to be the origin of the stored program computer used by von Neumann [55].

The particular configuration of the mathematical and later to be termed *cybernetic*, tools arose in the context of an emergent culture of research into communication, control and simulation. This had been inspired by the Operational Research (OR) initiatives of World War II, whereby the Allies invented a new scientific field to assist complex military organizations to cope with rapid technological change. The approach emerged in the United Kingdom, where both engineers and officers felt uncertain on how to use emerging radar technology to maximum effect. Countermeasures and counter-countermeasures, developed in response to the new weapons that were developed during the war, drove a cycle of continuous innovation not only of technology, but also of tactics, training and the protocols required to maintain equipment [56]. This general dynamic and the contribution of OR in its mastery, led to it being characterized as the:

'…wizard war…' [56]

World War II pitted systems against systems [57]. Managing this evolution became a primary scientific multidisciplinary concern, with OR being proposed as a solution. A sense of the range of problems, military and civilian, that OR practitioners explored in the immediate postwar era, is outlined further by several, including Kirby [58].

The concepts in McCulloch and Pitts' paper were to be built upon and referenced by others, particularly contemporaries such as Wiener. He felt the article to be so innovative that he alerted his colleague von Neumann to the importance of this text [59].

The latter applied the closed loops concept to the model for his design of the memory of one of the earliest electronic computers, his binary, as opposed to decimal, Electronic Discrete Variable Automatic Computer, (EDVAC) machine [60]. These diverse studies arguably inspired, and in some cases were motivated by, circa ten symposiums termed the Macy Conferences, or:

'Cybernetics Group' [61]

that were held from 1946 until 1953 in the United States of America. This era marked a renaissance of research into, and the foundlings of a general science of, the workings of the human mind. The Macy Conferences were the impetus for one of the first organized studies of interdisciplinarity, spawning breakthroughs in systems theory, cybernetics, and what has latterly become known as cognitive science.

The participants were an array of leading scientists that comprised a 'core group' of attendees that notably included:

William Ross Ashby; psychiatrist
Heinz von Foerster; biophysicist
Warren McCulloch (chair); psychiatrist
Margaret Mead; anthropologist
John von Neumann; mathematician
Walter Pitts; mathematician
Norbert Wiener; mathematician

Significantly, some time later in 1948, Wiener coined the term Cybernetics in his text of the same name [37] , defining it as:

"The science of communication and control in the animal and the machine". [37]

To present too narrow a view of cybernetics, as if it were based only on the notion of feedback, is possibly too pedestrian a stance. Cybernetics considers systems with some kind of closure that act on themselves - invariably leading to paradoxes since one encounters the phenomena of self-reference. Wiener further described cybernetics as:

'...the study of the interaction between man, machine and animals…' [37].

The world is said to be a big enclosed system. This means that at some time any given action is going to cycle back around to the beginning. Cybernetics is closely related to the ideas of *feedback* and *self-regulation*, as cybernetic systems tell themselves how to react to changes in environmental and internal stimuli. Feedback is therefore an important part of cybernetics; output is not only affected by the current input but also the previous inputs and outputs.

Beer's work on:

"managerial cybernetics" .

was a subset of Wiener's research, drawing on the neurophysiology of the human brain and nervous system.

The contemporary nature of this VCS research has necessarily led to the application of cybernetic elements such as *homeostasis, neurophysiology, feedback, recursion* and *modelling*. Through attempting to demonstrate that local agents can self-govern without a higher-level controller, in order to enable the system to influence and cause change in the environment. In effect, the VCS attempts to exert a form of theoretical control over some part of its environment in order to maintain viability.

William Ross-Ashby proved this concept in principle with his cybernetic theory and self-vetoing homeostat, as shown in Figure 2.1, that operates through complex systemic interaction according to the Law of Requisite Variety. Ashby examined how an organism may behave mechanistically and adaptively. He consequently proposed the *Homeostat* [10] as a model of a state-determined brain-like mechanism which produces adaptive behaviour in a system that interacts with its' environment.

Ashby's' model represents one of the earliest theories of self-organizing systems and consists of several homeostats connected together, each able to determine whether its conditions for stability are met or not. In the event that they are in these coupled systems, the homeostat does nothing. If, however, the conditions are not met the homeostat will send a signal to the other homeostats to change state. They will then modify and the process repeats until all homeostats meet their conditions for stability, each vetoing the states of the fellow, until system-wide homeostasis is achieved.

The configuration of Ashby's homeostat device was a mechanism consisting of four pivotal magnets, motion constraints and electrical connections and switches. It was designed to demonstrate what he had called an *'ultrastable'* system, i.e. to achieve stability after disturbance, achieved by identification of a property that he termed ultrastability. As demonstrated in Figure 2.1, this:

'...negotiated adaptation...' .

process was designated by Ashby to be:

'Self-vetoing homeostasis' [62].

In the context of this VCS research into a man-made computer system, the purpose of such systems can be construed, in general terms, to be to create some beneficial or desired change in the environment of that particular system. Based on Ashby's homeostat, the system must thus have the ability to influence and cause change in other elements that make up that environment. In essence, therefore the system must attempt to exert a form of control over some part of its environment. This:

'operational control' [10].

encompasses the control of one system by another. Internal control is still necessary, but an additional set of variables determined by the purpose of the system is selected and maintained within tolerances set by the controlling system. If operational control fails, the system may indeed survive yet it has actually failed in its purpose. Each of the possible states a system is capable of assuming, was represented by Laws et al. in Figure 2.1 as a point. Each of the points represents the set of values held by the variables that describe the system at that time, dictating that the system can be perceived as following a trajectory encompassing all possible states of the system.

With a changing system state, different points are assumed in that space, with acceptable states for the system being those representing homeostatic and operational stability. These are grouped inside a conceptual boundary into which the trajectory assumed by the system must stay. This would reflect homeostatic control by the system.

In the event that two such systems are coupled, the output of each will form the input of the other with each pursuing local homeostasis. As illustrated by Figure 2.1, the representative point for each of the systems must remain within the boundary of acceptable states for both systems to be considered to be operating normally. Each system identifies that the other is in normal operation by the variables in each system that represent environmental variables to the other. In the event that representative point in system A moves outside the acceptable boundary, that event should register in system B and cause an appropriate change of state in that system. The nature of change required is denoted by the original change in A, which must undertake a trajectory of state changes to return its representative point to an acceptable position.

The danger is that changes in A and registered in B may drive the representative point in system B over the boundary to unacceptability which, of course is registered in A and may cause the planned trajectory to be adjusted. B must now plan its own trajectory back to stability. The process continues until both systems arrive back at acceptable states, each system thereby acting as a controller of the other, vetoing any state adopted by the other system that is unacceptable to itself. This is regardless of whether it is in the acceptable range of its counterpart system.

This self-vetoing homeostasis demonstrates a form of negotiated adaptation and should result in adaptive stability in both systems, whilst assisting each in adjusting to the operation of the other. This is despite any extraneous disturbances affecting either system. The concept of a systems' ability to return to homeostasis regardless of the perturbations that it encounters is significant when addressing redundancy within software systems and thus complexity.

One difficulty in this practice, however, is the time taken to reach dual stability. In the event that environmental disturbances arrive faster than the time taken for the homeostatic loop to complete operations, the system may oscillate interminably. A reward and punishment scheme may redress this, with initial adaptive attempts possibly being based on trial and error. Those that do result in beneficial effects are reinforced by positive feedback whilst conversely; those that have detrimental effects are discouraged. The retention of ineffective approaches prevents the system from being stranded on local optima, this information allowing for their future avoidance.

This approach promotes the growth of organization within each system, with trajectories leading to effective and timely adaptation potentially being reserved for future re-use. This provides a growing record of effective routes back to stability. The system thus learns both to adapt to new or changed circumstances, decreasing its reaction time to previously occurring disturbances.

The VCS is analogous to such a system, because it can conceptually, homeostatically, return to stability after it has been disturbed in a way not envisaged by the designer without the requirement for human agent intervention.

As McCulloch is quoted as stating in 1968:

‘’The difficulty is that we, who are not single-cell organisms, cannot simply divide and pass on our programs. We have to couple, and there is behind this a second requirement... We learn that there's a utility in death because the world goes on changing and we can't keep up with it. If I have any disciples, you can say this of every one of them ,they think for themselves’’ [39]

One of Ashby’s vital contributions to the field of systems science was to formalize homeostatic theory and thereby recognize the self-vetoing homeostat. As one of the precursors of the field of cybernetics, his codices are pivotal to the VCS research.

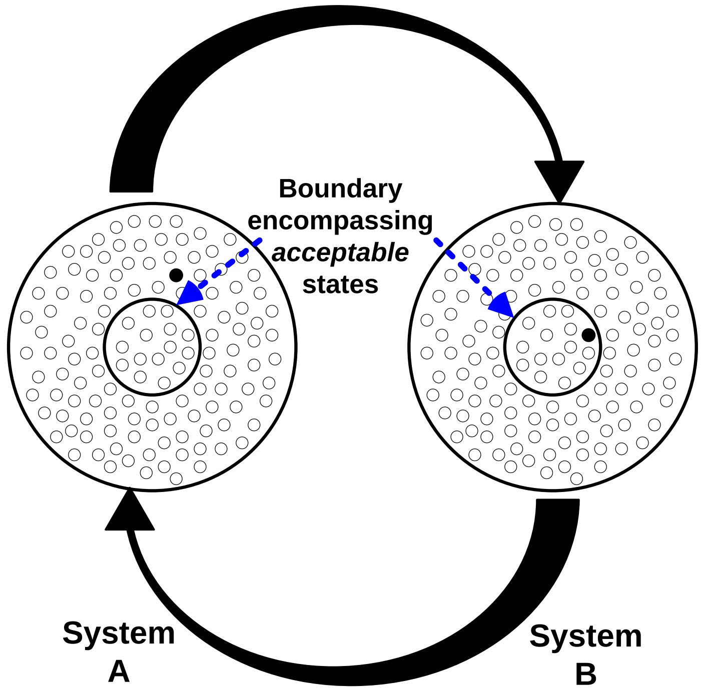


Figure 2.1: Self-Vetoing Homeostasis in Coupled Systems

2.3 Cybernetics Vis-à-Vis Autonomic Computing

The basis of autonomic computing is that systems are able to self-manage, adapting their behaviour at runtime to respond to change dynamically. Analogous to the self-governance of the human body such as a person's heart beating without the need to consciously consider, nor understand the rudiments of that action, thereby such actions being taken without conscious consideration, i.e. automatically rather than consciously, suggesting autonomy over governance.

Evidence exists of analogies with Horn's concept as far back as 1865, with Bernard's 'environment within', or:

Milieu intérieur [21].

It suggested that the stability of the internal environment is the condition to maintain free and independent life. This notion was debatably the inspiration for Cannons *homeostasis* terminology [63].

Circa 1999, Laws et al. drew a parallel between the cybernetic properties of Beer's VSM and the complexity reduction requirements of the software industry [64]. In applying the VSM to problems that were attributable to the field of self-adaptive software [65]. Laws' work precipitated the subsequent autonomic computing ideal. The 2001 J-Reference Model, Figure 3.13 [31] was subsequently produced. Displaying the existing cybernetic topology, it united with both Bratman et al.'s IRMA architecture Figures. 3.13 and 3.14 [66] with its' Beliefs, Desires, Intentions (BDI) framework, while employing Ashbian systemic variety concept to the endogenous complexity proliferation cited within Legacy Systems pre-millennium [6].

Bratman's BDI framework application, however, introduced recognized problems from Artificial Intelligence; successful application of a real-time, isomorphic model [67], depends upon a real-time BDI complete model.

This dictates a necessity for context-sensitivity, in that a system would require one-to-one mapping to the requirements within its environment. This would potentially and counterintuitively, introduce complexity, due to the scale of attempting to accommodate every possible permutation. Espejo's collaboration with Harnden [68], applied a cybernetic slant to the modelling of agent communities.

Laws et all's research continued however [31, 69-71], later running concurrent, yet significantly unparalleled, to others drawing biological homeostatic analogies such as IBM's Horn, Kephart and Chess [72]. Contemporaries including Herring [73] and latterly Stoyanov [74], applied the VSM blueprint to their autonomic computing research. Stoyanov proposed that the abstraction of observable variety and a managed communication channel was core to development of viable, autonomic computer systems, whilst outlining the importance of the runtime capability verification of interacting components.

In 2006, Laws, Bustard et al., merged autonomic computing, the VSM and Soft Systems Methodology [75].

By learning from the VSM key aspects and properties of Beer's model are replicated within a design grammar. These include: a concept of internal modelling, recursivity – so lessening redundancy and so human agent intervention, a context-free property, allowing a more fluid application to differing and thus portable, scenarios and the promotion of autonomy versus governance, again reducing the need for human interaction.

Set theory is merely a vehicle to allow the manipulation of the VCS topology. Modelling such formalism in this way could potentially also facilitate its transposition into software, to enable representation of recursivity by way of coding.

To support this proposition, this thesis provides an examination of autonomic computing, referencing Beer's cybernetic VSM as the basis for the development of a series of set theory design grammar models of the VCS. To this end, the work provides a rigorous consideration of the VCS from basic requirements and conceptual representations through to the development of a proposed VCS architecture, substantiated by the publication of both open [3, 24] and closed [2] environment case studies that demonstrate proof of concept. Specifically, a 2007 design grammar model related the VSM systems [1], and so innovated Viable Computer Systems (VCS). Inter-recursion cohesion, was later incorporated, profiling the importance of feedback in the software process [76, 77].

This research has demonstrated ecological dependence [1, 21], in order to emphasize how this reliance helps negate systemic redundancy [78, 79] and complexity. The nonreciprocal VCS reliance upon the environment enables a sense of viable self by the creation of a model of the future environment, a comparator to the model of the internal systemic capabilities. This VCS facility, Figure 4.1, in turn engenders a temporal, forecasting, capability that thereby not only advances systemic emergence, but also the retention of viability. This work has thus drawn-from and alluded-to, Lehman’s inspirational description of any software program as:

> *“A model of a model within the theory of a model of an abstraction of some portion of the world or some universe of discourse”* [32]

The closed environment case study that ensued [2], was applied to a previous genetically modifiable software system [69]. This crucially exhibited the self-governance capability of the VCS [2].

More recently von Foerster's:

"order from noise" [80]

Paradox was examined, partially inspiring an open environment VCS case study [3, 24] that applies directive correlation [25] and algorithmic hot-swapping to submit a theory of open-bounded homeostasis [3, 24] uniformly applied to the same previous genetically modifiable [69] system scenario. In so-doing, it is believed that this research has novelly progressed the state of the art from autonomic computing, towards the arena of Viable Computing Systems.

2.4 Chapter Summary

Investigations towards realizing the VCS concept, led to a diverse fusion of past and current themes and approaches. While the work outlined in this thesis shares similar objectives and roots with the above described efforts, it focuses on a novel proposition to promote the VCS, which is intended to go beyond an autonomic computing architecture. The said is accomplished by extending an autonomic system's capabilities with cognitive, deliberative and managerial function. This is achieved by combining fundamental managerial cybernetics and autonomic computing elements to form the basis of the VCS architecture. In particular, to the author's current knowledge there is no notable work that has proposed such a methodology that has the potential to be coded into software.

Chapter 3

Literature Review qua VSM and Related Models

Introduction

The VCS research increased an understanding of the basic principles of cybernetics, whilst also recognizing that it can be both applied and used, to model, different systems - electronic, computing or even biological.

Attempting to engender Lehman's '*model of a model*' [67] concept has been a core objective of the VCS research. Likewise, as Beer's VSM .is based upon effective organization within the human brain, the study hoped to adopt similar methods used by the central and visceral, or autonomic nervous systems that are used to manage the mechanisms of the organs and muscles. The foci were accordingly identified and directed specifically on those research areas that were felt to be pertinent to the study objective.

This chapter thus not only reviews previous research that has relevance to that of the VCS, as outlined herein, yet also begins with some earlier related research. It then proceeds to consider more recent work, which is more closely associated with this study.

3.1 Feedback Control in the Software Process

As early as the release of IBMs OS 360 [81] in March 1966, Lehman and Belady's research at Yorktown Heights, spanning 1964-72, revealed complexity within software computing systems, Figure 3.1.

Their eminent 1969 *Programming Process Report* was arguably, retrospectively, a landmark in terms of recognizing the existence of endogenous complexity, that is, a computer system inherently ageing from within in terms of growing code size and maintainability. They warned of an

"…overwhelming impression of growth." [81]

within IBM and the national USA programming scene of the day:

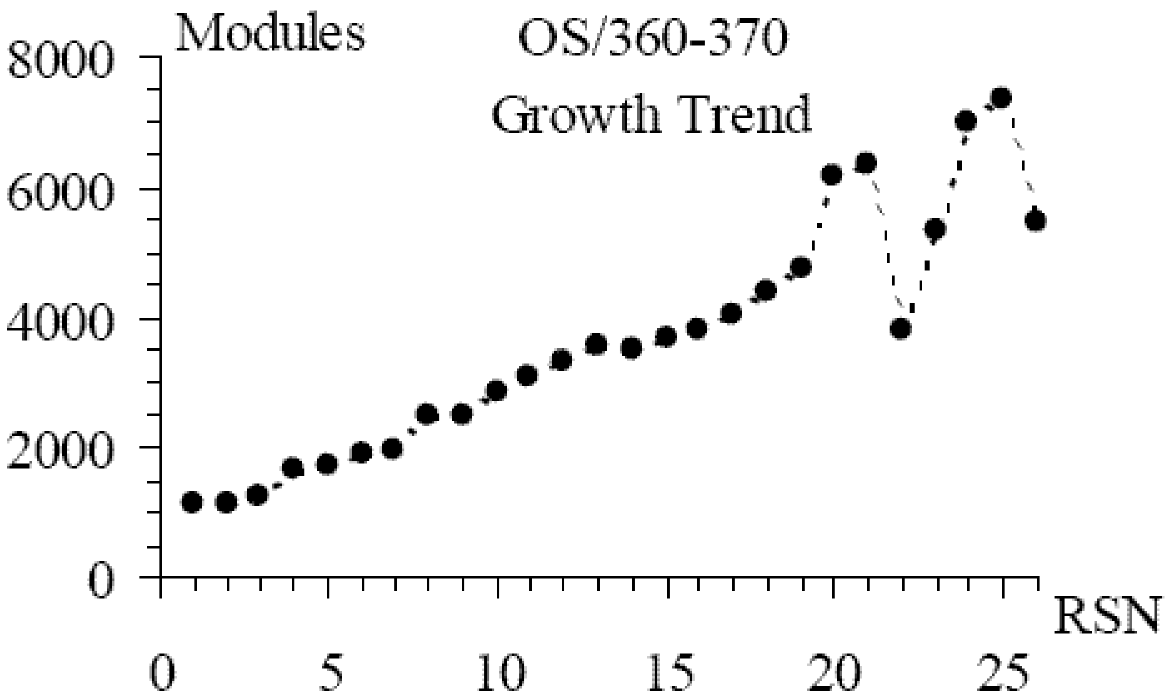
"…OS/360 represents an example of increasing size and complexity…successive releases tend to require ever more modification of an ever larger system…these trends are indicators of growing complexity…the effects of error and change are spreading ever further through the system and will soon force the initiation of an OS/360 successor…" . [81]

IBM executives had largely ignored the prediction that OS/360 was becoming too large, yet by 1971 their software developers had encountered significant complexity problems. This required the splitting of the code base into two, necessitating the implementation of the IBM OS/370.

The linear growth curve that seemed so steady in the 1960s suddenly demonstrated signs of chaos. Lehman and Belady prophetically suggested that:

"the time has come to develop a new approach to the entire process, to change the way of seeing and doing things…the problems that are encountered are due to system behaviour…"[81].

Lehman had derived emerging themes from the study: feedback in the global process, observation, measurement and models and system dynamics.



[RSN-release sequence number]

Figure 3.1: Feedback in the IBM OS/360 Software Process [82]

Lehman would confirm decades later

"It was this study and the continuing research it triggered that subsequently led my colleagues and me to the concepts of process models, evolution dynamics, program evolution and support environments…" [83]

"The importance of that model is not only in the process it depicts. It is a canonical model of software development and of development steps…"[84]

The VCS research has drawn from these studies of Lehman and Belady that have over more recent years produced and studied a number of industrial software processes using several different modelling and simulation techniques. A focus of this work has been to identify common behavioral patterns, to demonstrate the presence of strong process internal dynamics and to develop management tools for process planning and improvement [85].

Turski independently analyzed the IBM OS/360 growth curve Figure 3.1, demonstrating that, despite the ripples, the data fits closely with what he has termed the:

"inverse square growth law" [86],

Concluding that the formulas within reveal that the square growth is typical of a system dominated by its own system dynamics, thus appearing to support Lehman and Belady. Lehman's 1996 formulation, entitled his:

'Law of the Feedback System' [87],

further substantiated the VCS approach. Within this work, he concluded that, during the ongoing OS/360 research in 1971, he and Belady had uncovered substantial evidence of the role of feedback in the software process. They later proclaimed in 1996 that:

'...programming processes constitute multi-loop, mulit-level Feedback systems and must be treated as such to be successfully modified or improved' [87]

Nuseibeh and Easterbrook's apparent validation of the global analysis viewpoint, defined domain modelling as a necessary factor in requirements engineering, thus:

'A model of the domain provides an abstract description of the world in which an envisioned system will operate…they permit tractable reasoning over a closed world model of the system interacting with its' environment'[88]

The software development and maintenance process, or more appropriately described as the software evolution process [89], constitutes a feedback system. This observation was first made in a 1972 paper [82] discussing results of the IBM programming process study [81], that had observed the ripple in the plot of OS/360-370 growth shown in Figure 3.1:

"... is typical of a self stabilising process with positive and negative feedback loops. ... the rate of system growth is self-regulatory, despite the fact that many different causes control the selection of work implemented in each release…" [82].

The transformation in thinking circa the time of Lehman's 1969 IBM report, arguably led to the first NATO conference sponsored by the science commission at Garmisch, Germany. Here, the term:

"software engineering" [90]

was first publicly coined by Randall and Naur [90], espousing the idiom that was to become synonymous with not only development of software computing systems but, significantly, also its’ ongoing maintenance.

3.2 The Viable System Model

Figure 3.2, illustrates Beer’s VSM , as a top-down recursive, quintuple hierarchy of systems; System One, (S1), through to System Five, (S5), plus one ancillary, intermittent sub-system, System Three Star, (S3*). These are physically situated within recursive nestings of a management and operation unit assemblage – each constituting an S1. Beer defined an assemblage of identities as a system, because those identities are observed as acting cohesively in order to maintain control:

The system avoids or otherwise counteracts a stimulus which disrupts its activity, and embraces or seeks to increase a stimulus which favours its activity”

Designed for human organizations the VSM is underscored by managerial cybernetics [26] and applies variety engineering to manage complexity.

As illustrated in Figure 3.2, this controls varietal complexity via homeostatic loops exhibiting feedback control. In essence, variety engineering is the manipulation of varieties in whatever way is most appropriate to restore the balance between the regulator and regulated. This attenuates unnecessary variety from the parent S1 and its environments.

Managerial cybernetics advocates the design of nested, recursive organizations. For effective organization, each viable unit in the holism should be autonomous as much as possible yet also subject to some controls from its upper management or metasystem, as it is termed.

One could argue that the uniform systemic commodity that has to be managed is complexity. In cybernetics, this is given a precise definition through the concept of variety, or:

"the number of possible states of a system"[11]

and is fundamental to understanding the VSM. The idea of variety in this scenario is an important one as it assists in the understanding and to contextualize all kinds of systems, processes and behaviour as ways to either reduce the variety of something so that a system can hope to manage it better, or to increase the variety of the behaviour of a human agent, in order to have a larger impact on the entity that is hoped to be managed.

Key to the VSM and the understanding of complexity is Ashby's Law of Requisite Variety that states:

'only variety can absorb variety' [11]

As restated by Stafford Beer, this espouses that ways must be designed to increase or reduce variety and thus enhance a systems' ability to manage, and thus increase its performance, irrespective of its' context.

In terms of varietal control, the metasystem represents a more general level of the structure with the functions including the damping of oscillations between the S1 subunits that undertake operational control, and the coordination of activities of S1's to achieve synergy for the whole management control.

It is important to note that the operation unit tries to match the variety of its environment, by modifying behaviour according to an environmental response. In turn, the management unit tries to guide and improve the effectiveness of this exchange. There is a lot more variety in the environment than the operation needs to know about and a lot more variety

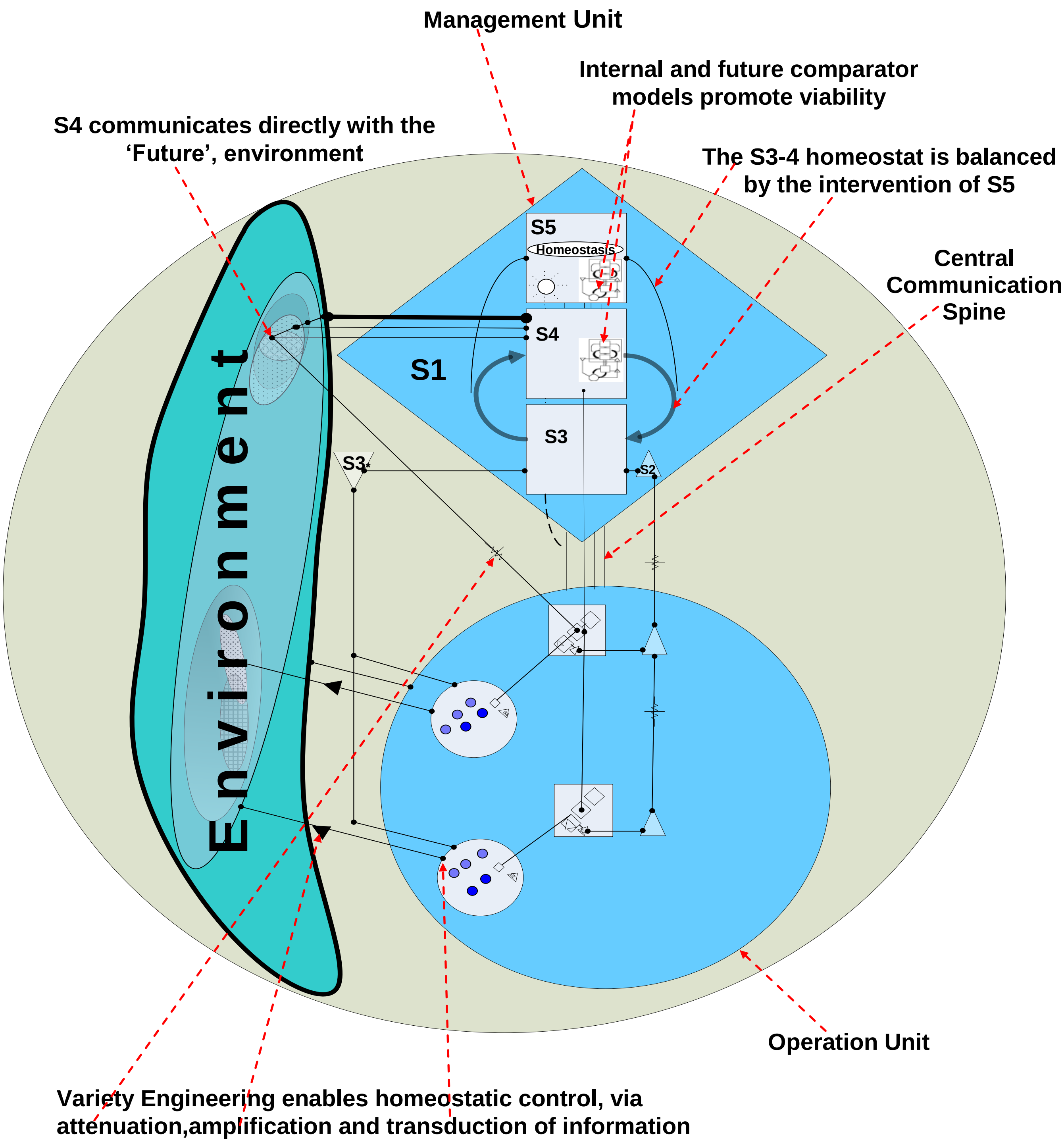


Figure 3.2: Stafford Beer’s Viable System Model

in operations than management needs to address. The VSM ideal is for efficient operations and managers to develop skills to select the information they need and ignore the rest while remaining alert to signs of change and incipient instability.

This promotes self-organization and viability to emerge via self-production, or autopoiesis of S1's. Each S1 is a viable system containing a complete viable system in fractal-like recursion. The resources require coherent distribution amongst the systems; the architecture thus provides mechanisms for all parts to work with a common holistic purpose whilst retaining viability in a changing environment.

Beer's VSM, originally powered by the cognition of it's' human-agents, can thus be defined as organizationally and operationally closed, yet informationally open . It embraces three elements of Management or metasystem, integrating the operational units, Operation, the locus of recursion and primary activities and Environment the highest recursive level of the metasystem, containing external elements directly relevant to the system. Management is thus the regulator of operations and vice-versa.

The central, vertical spine includes four communicative and one intermittent Algedonic, or alarm, channel, transducing data amongst the system-environment alliance. Each operates in a reactive, feedback and in the case of S4, feedforward, controlled mode.

Beer's VSM, Figures: 3.2 and 3.3, implements a control and communication structure via hierarchies of feedback loops. Six major systems ensure 'viability' of the system. The top-down recursive model constitutes five-systems, system one, (S1), through to system five, (S5), plus one ancillary sub-system, System Three Star, (S3*).

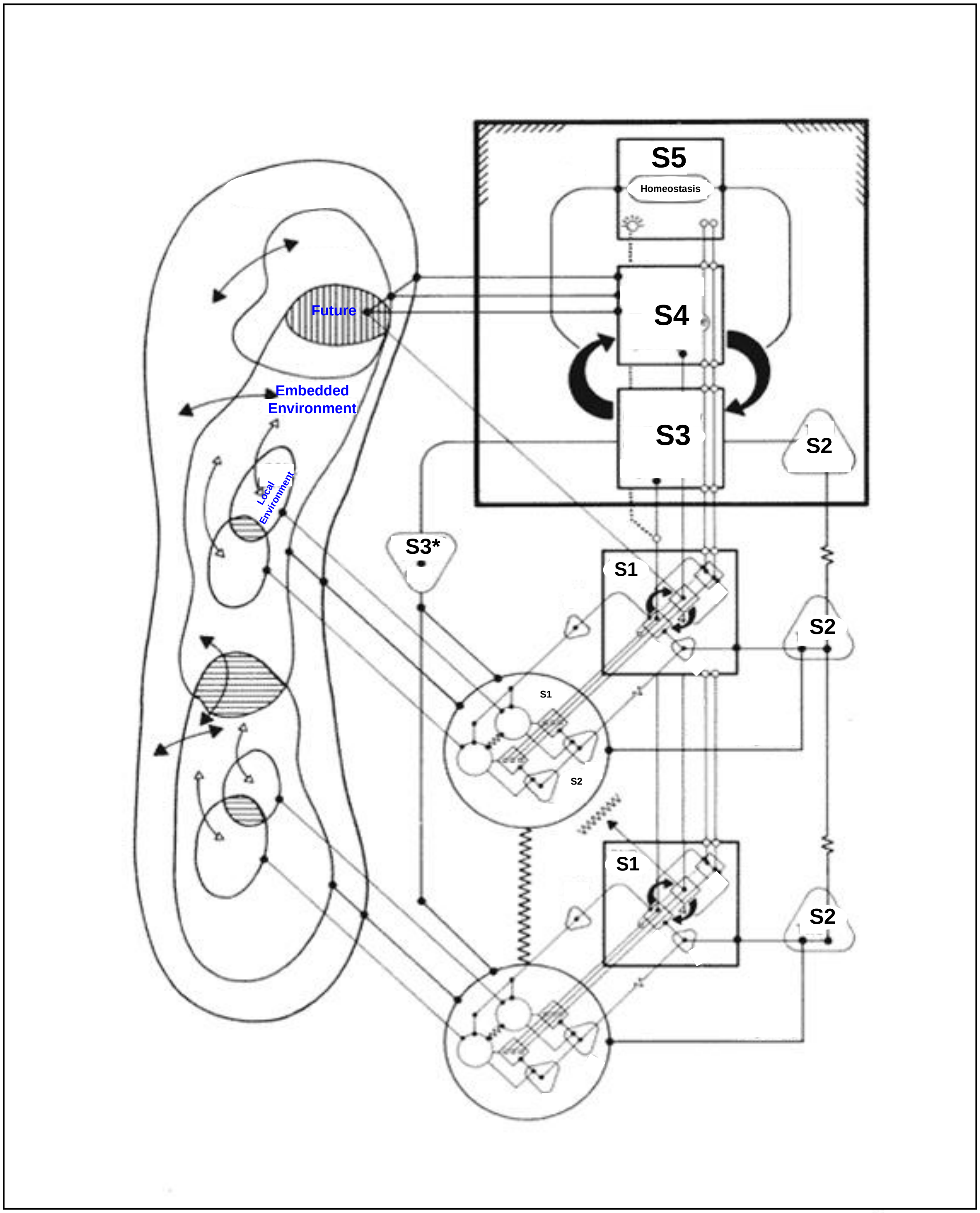


Figure 3.3: Beer's VSM System-Environment Ecology [91]

The VSM offers an extensible, recursive, model-based architecture, devolving autonomy to sub-systems, originally and typically being a human organisational *design/diagnosis* model. It emerged from the field of cybernetics which sought to design information processing and decision-making machinery using the structure of the nervous system as a guide.

One of the forerunner cyberneticians, Wiener, believed that one could study neurophysiology, develop a theory of how the brain works and subsequently apply that theory to design information processing equipment. He adopted this method to create a radar-guided anti-aircraft gun, which was installed on ships in the Pacific during World War II.

A similar approach was implemented by Stafford Beer to create the Viable System Model, a top-down recursive system constituting five-systems, system one, (S1), through to system five, (S5), plus one ancillary sub-system, System Three Star, (S3*).

Designed for application to human organizations, the VSM is powered by Managerial Cybernetics [20] and likewise unique in its use of variety engineering [92]. This process controls complexity via homeostatic loops exhibiting feedback control. This allows attenuation of variety from the parent S1, whilst amplifying the variety in terms of its environments so promoting self-organization. Viability hence becomes an emergent property, along with self-production, or autopoiesis.

These resources all rely on coherence between the elements, each part a viable system containing a complete viable system in fractal-like recursion. The original Beerian topology hence provides mechanisms for all parts to work with a common purpose, and to exist in a changing environment. Organizationally and operationally closed, yet informationally open, it embraces three elements of:

Management, or ***Metasystem****:* Ensuring integration of the operational units

Operation*:* The locus of recursion, holding the primary activities and

Environment: The highest recursive level of the metasystem, containing external elements directly relevant to the system.

Management is thus the regulator of operations and vice-versa. Three key environmental levels encompass the:

Micro*:* The local or domestic environment relevant to each particular S1.

Macro, or ***Global****:* The wider environment of the system-in-focus, in-which these are embedded.

Future or ***forecasting***: Environmental level, which identifies direct benefits or threats to the system-in-focus and accordingly allots a temporal facet to the system.

Central, vertical spine**:** encompasses four communicative and one intermittent *Algedonic*, or alarm, channel, transducing data amongst the system-environment alliance. Each operates in a reactive, feedback controlled mode.

Internal models*:* Represent the internal capabilities of the system and a changing environment, each of the two endorsing systemic viability by mapping between the ecology.

S4: Unique amongst the systemic federation, System Four promotes viability by notionally linking to the future environment, so identifying benefits and threats. The VSM is thus real time.

Figure 3.4 articulates the conceptual mapping between the VSM and the human biological self-governance metaphor promoted by the autonomic computing notion. This is followed by more detailed, individual analyses of the VSM systems and their functions, including an overview of their interrelationships and the architecture and its mechanisms and constituent parts that enable viability within a changing environment.

Autonomic Systems:-

***Implementation, S1*: **Controls the primary activities of the system by using local information whilst also consulting the higher-level control systems.

Monitoring, S2: Enables the local regulation of activities via coordination with S1, providing stability by resolving conflicts between the S1 elements through its' anti-oscillatory regulation capabilities.

Control, S3: Provides an overall picture and optimises S1 and S2 operations using currently available global information. Synergies will become effective at this level.

Audit, S3*: Provides auditing information to S3 to independently collect data about operational activities.

Anticipatory Systems: that offer self-awareness and are deliberative:-

Intelligence S4: that seeks to anticipate future changes by extrapolating operational information whilst observing the outside environment, by using planning and simulation tools to generate future strategies to be executed by the lower level control systems.

Policy S5: this provides a set of high-level policies for the whole system to adhere-to, whilst assuming the role of a homeostatic regulator to keep S3 and S4 in balance, to moderate the speed of change.

The application of the VSM to the VCS research is essentially constructivist in nature. This dictates that the VCS aims to be a construction of a reality, in which observation and interpretation play a crucial part. In this process, the agents involved make sense of the system in focus, by mapping it onto the VSM. At the same time they bring forth Harnden's notion of:

"multiple realities rather than striving for a fit with one reality" [93]

Furthermore, the key attributes of the VSM are listed, that are perceived to be relevant to the VCS research, providing a synopsis of each. The two main types encompass the deliberative and autonomic systems:

Deliberative Systems: These incorporate S4 that deliberates on future scenarios, whilst S5 determines the system identity. The process of deliberation thus interprets desires in the context of its current perspective on the environment of the system-in-focus, with environmental change being addressed by S4, guided by the S5 model, scans the environment for both detrimental events or beneficial opportunities

The outcomes of this process are the formulation of a view of the outside world which is provided to S5 in the form of the World model, and the production of future development plans that enable exploitation of advantageous opportunities whilst avoiding detrimental occurrences. Plans are then relayed to the S4 deliberation process to initiate the intention forming cycle again.

Autonomic Systems: These include S3 that notionally reasons based upon information gleaned from the S4 and S5 models. S3* audits the current status of operational S1 units and structures plans, that are then passed to a S2. The scheduling process, in cooperation with a resource bargaining process, responsible for negotiating resource deployment and usage monitoring, schedule the enactment of the plan. The schedule passes to the coordinating S2 channel for dissemination to participating S1 elements.

The potency of this method lies in the recursivity of the underlying model as illustrated in Figure 3.4. This demonstrates how the entire architecture described above, and recurs in an S1 unit in every layer. This dictates that one recursion informs the next, thus allowing an autonomous response to local conditions at each level whilst retaining the purpose of the systemic holism.

A viable system can thus be examined as consisting of three major domains: Environment, Operations and Meta System. The Environment is considered as a view of the outside world reduced to only the relevant parts of a modelled system. The Operations domain contains all system activities; when applied to computing systems, it may be regarded as covering the complete functionalities found in traditional applications. Thirdly, the Meta System is responsible for controlling harmonic integration of all operations and planning of future operations, a concept that is seldom found as an explicit architectural component in conventional computing systems.

All three domains are interconnected: elements in the Metasystem domain regulate elements in the Operations domain and observe the Environment in anticipation of changes, whereas operational elements are interacting with the Environment to fulfill their functional purpose.

The Operation undertakes all the basic work and the processes which provide a service to the Operation by ensuring the whole organizations works together in an integrated way: the Metasystem. The VSM perceives a viable system to be a collection of Operational elements which are held together by a Metasystem, or management unit. Both Operation and Metasystem must communicate and interact with their environment.

Each of the Operational units must be viable, and thus can be viewed as smaller Viable Systems embedded within the larger system.

Deliberative/

Anticipatory/

Self-Aware Systems

***Autonomic* Systems**

S5: Policy formulation, this notionally maps to the higher brain functions

S4: Intelligence: this notionally maps to the Mid Brain. The connection to the outside world via the senses, engendering future planning.

S3: Control: this notionally maps to the Base Brain which oversees the entire complex of muscles and organs whilst optimising the internal environment.

S2:Monitoring: this notionally maps to the sympathetic nervous system which monitors the muscles and organs and ensures that their interactions are kept stable.

S3*:Audit: this monitors its fellow systems, notionally looking for signs of strain.

S1:Implementationt: this notionally maps to the muscles and organs, i.e. the parts that actually DO something or the basic activities of the system. The Operation.

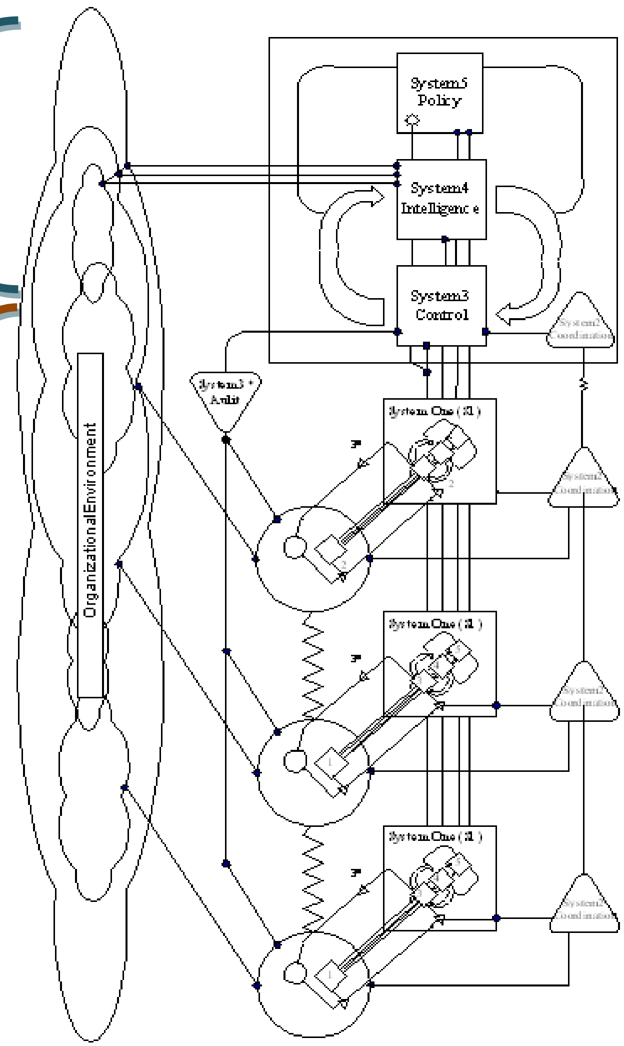


Figure 3.4 Self-Governance within Beer's VSM

The metasystem alliance undertakes three main functions within the holism, entitled the:

Inside and Now - or S3-2-1 metasystem homeostat (incorporating the S3* subsystem), which looks at the Operational units collective, attempting to ensure that they work together in mutually beneficial ways, whilst resolving conflicts.

Outside and Then – or S4-5 metasystem homeostat, that looks at the external environment, assesses the threats and opportunities and constructs plans to ensure the organization can adapt to a changing environment.

Policy – or S5, establishes the ground rules which set the tone for the whole organization and must thus have ultimate control.

3.2.1 Environment

Along with Operation and Management, The Environment is one of the three crucial elements of the VSM. A fundamental principle of the VSM is that for a system to remain viable it must be viewed as a whole that is in balance with its environment. Imbalances need to be restored in order to retain that viability and promote emergence. The internal; environment is sometimes referred-to as the Inside and Now. The wider environment of the system in focus is often referred-to as the Outside and Then. The other systems depend on the environment, whereas the environment is not mutually dependent upon any of the other systems. As the highest recursive level of the metasystem, environment is part of the triadic alliance of management, operation and its particular environment. It is important to remember that the VSM is viable only because it can maintain a separate existence within its embedded environment. Management is thus the regulator of operations and vice-versa, within the particular system-in-focus.

The internal environment consists of all the Operational units and those jobs which are dedicated to looking at them, whilst ensuring that conflicts are resolved and that their performance is optimized. The internal balance is concerned with these Metasystemic jobs and with ensuring that they have the capabilities to function properly on a continuous basis.

The key to internal balance is to view the internal system as a system of autonomous Operational elements, overseen by the S3-2-1metasystem homeostat to determine means of generating synergy. It is therefore the case that dictates from above should only be imposed when the whole systemic viability is at risk, thereby devolving autonomy to the lower level self-governing S1's.

In terms of the external environment, S4 maintains contact with the relevant parts of this, so enabling the future planning systems to develop strategies for adapting to change. The principal job for S4 is to decide what within the external environment is of direct relevance, as the VSM is able to discern two kinds of external environment: These are the *predictable* which can be monitored, in terms of identification of trends and decisions made accordingly. The second is the *identification and provision of the novel* in the relevant areas.

A viable system is thus composed of five interacting subsystems which may be mapped onto aspects of organizational structure. System 4 is concerned with the 'outside and then' - strategical responses to the effects of external, environmental and future demands on the organization.

System 5 is concerned with balancing the 'here and now' and the 'outside and then' to give policy directives which maintain the organization as a viable entity.

3.2.2 System One (S1): Implementation:

This encompasses the basic activates of the system or the operation. Beer drew a personification metaphor by analogizing this to the muscles and organs within the human body. The same structure of systems will recur although their detail and context would necessarily differ. Recursivity allows each level in the organization relative autonomy bounded by the overall purpose of the systemic whole. The VSM ideal is that each S1 should be autonomic in its own right, by cooperation and coordination within and between S1 units on the same level and sets of S1's on different recursions.. Communication channels thus operate across the hierarchy, tailored locally to each viable entity. Each S1 being a self-governing homeostat. Requisite variety applies in three distinct ways; to the blocks of variety homeostatically related, to the channels carrying information between and to the transducers relaying information across boundaries. In human systems, as Figure 3.5 exemplifies, each division would be considered as a S1.

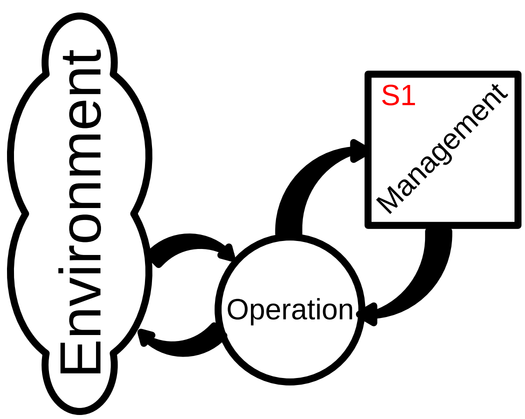


Figure 3.5: Beer's VSM System-One

S1 consists of the units that do the basic work of the fundamental operations within an organization, for example manufacturing products or delivering services. Comprised of a management and operation unit, S1 is nested within a higher parent S1. Maturana has defined S1's as the systems' autopoietic generators [94], it being autopoietic via its ability to self-produce lower-level recursions within a recommended range of one to seven S1's per recursive level. The primary systemic activities are hence executed by S1, via its capacity as an autonomous homeostat.

Interacting directly with the environment, S1 assumes primacy within the systemic federation, as it consists itself, of viable systems. It is often the case that the environments of the operations overlap with each other. They are also connected to each other by such things as flows of materials. S1 is made up of all the operations which do the things which justify the existence of the system, including the managements of these operations. It does not include senior management, which should be considered as a set of services to S1. Without S1, there would be no reason for the organization to exist.

Operation contains the primary activities of the system, i.e. the purpose of the system and is the locus of recursion within the topology. The Management unit or Metasystem ensures integration of the Operational units. The Environment incorporates external elements directly relevant to the system. Three key environmental levels encompass the micro, the macro, (or global, in-which these are embedded) and the future, forecasting level.

Systems 3-2-1 are analogous to the human autonomic nervous system. System 4 embodies cognition, System 5, the higher brain functions, include introspection and decision making .

The VSM identifies five management functions within an adaptive system, with S4 being responsible for long-range planning and in strict

Beerian application, the design of new products and services. A key feature of the VSM is the management of variety, recognizing that although information is necessary for agents to perform effectively, too much information can be a distraction. Variety attenuation is exhibited by S4 through the environmental scanning activity, informing agents of, for example, new technology, new regulations, and other competitors. From a great variety of sources of information, S4 selects the information that is most relevant and significant for the decisions the system must take. This variety attenuation is the reduction of variety down to that which the system can handle.

Conversely, variety amplification refers to the distribution of the system's messages being transmitted outside the system and plans, policies and procedures needing to be distributed within. This variety amplification is thus the expansion of variety to that level the system needs to balance external variety as required by Ashby's law of Requisite Variety. The VSM is thus useful as a guide to studying where variety is attenuated, where it is amplified, and if there is a balance in the varieties of interacting sub-systems. Accordingly, it thus complies with Ashby's Law of Requisite Variety .

Within a division, the S1's might be different manufacturing plants. S2 would coordinate interaction among the manufacturing plants. S3 would allocate funds for the operation of the different plants. S4 would consider whether new product models or new manufacturing facilities are needed. S5 considers the plans produced by S4, and decides which are enacted.

These thereby become the purpose of the system, which S1 then puts into practice. S5 would thus decide when to phase in new product models or manufacturing methods, so ensuring adaption within an organizational system. The definition of a lower level of recursion would be different production lines within a manufacturing plant, with the levels of recursion going down as far as the individual who must both carry out assigned tasks S1, S2 and S3, and consider whether he or she wants to change jobs or obtain more schooling, S4.

That human agent must take these decisions in the context of subjective personal values, without the supervision of S3. This middle management system does not supervise the S1's, or producing units in detail, but only makes a *"resource bargain"* with them. This provides the S1's with high levels of autonomy.

The model explains what structures and procedures are needed at each level of a system and hence what information and what decisions are needed in each part. As illustrated by Figures: 3.2 and 3.3, by providing a single model of activities at all levels of a system, the VSM increases awareness, and knowledge, amongst the agents of how it functions.

3.2.3 System Two (S2): Co-ordination:

Likened to the sympathetic nervous system, S2 monitors S1 or the muscles and organs, ensuring stability in their interactions. Part of the metasystem homeostat, S2 assists with conflict resolution amongst the diverse interests amidst and interactions between the S1s. S2 coordinates the activities and policies amongst each S1, as illustrated in Figure 3.6.

The local regulatory system particular to each S1, S2 is a standardizing, anti-oscillatory, body that coordinates and facilitates S3 in its' objective of integrative function. S2 is thus the locus of systemic homeostasis.

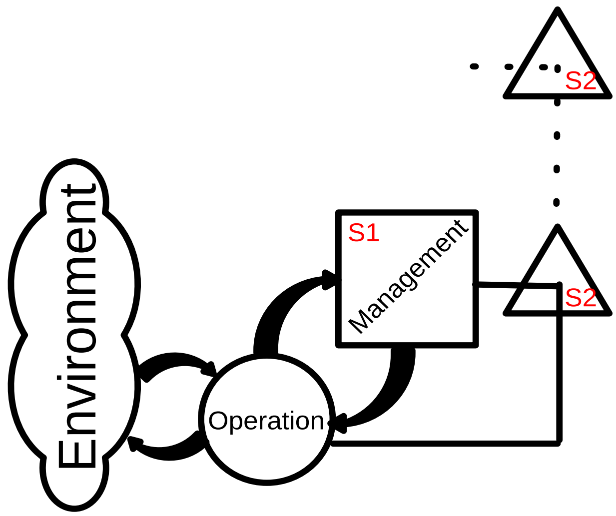


Figure 3.6: Beer's VSM System-Two

System two thereby consists of units that handle coordination and scheduling among the system ones. System two activities include allocating space and equipment and enforcing rules and procedures. Classic examples of S2 are a production plan, or a school timetable. These do not have to be imposed from senior management, but are usually arranged voluntarily between S1 elements. Senior management would only need to intervene to settle disagreements between the elements. S2 is embodied in the Regulatory Centers, represented as triangles

3.2.4 System Three (S3): Control:

This oversees the entire complex of S1s to promote synergy within the system. Beer used the muscles and organs metaphor from the human autonomic system metaphor. Whilst optimizing the internal environment. S3 exerts the internal control function, mainly using the vertical command

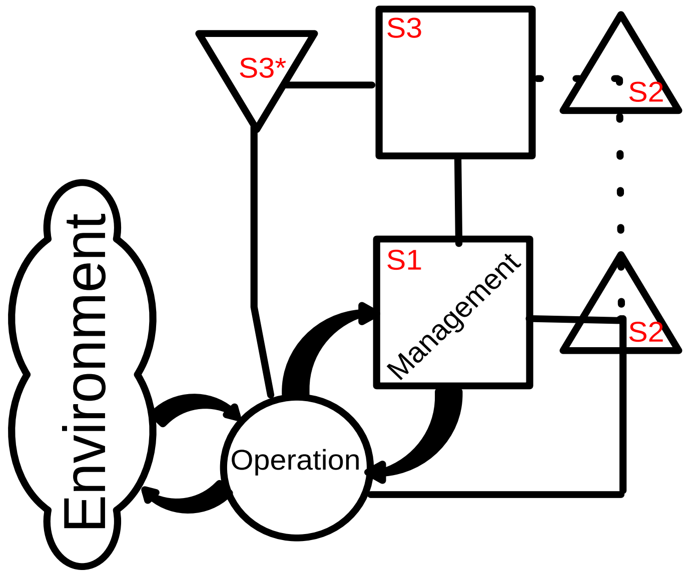


Figure 3.7: Beer’s VSM System Three .

channels shown in Figure 3.7. Control through these channels, however, may not have requisite variety to be really effective, thus S3 may need to directly monitor the operations of S1. In order to do this, S3 performs spot-checks and or audits, via S3*. This is an effective technique for maintaining S3’s requisite variety, thus explaining Beer's definition of S3 as the ‘control′ system and the senior management of internal and immediate activities of S1 and the supervision of S2. The controlling facility within the model, S3 regulates, optimizes and stabilizes internal activity. S3 undertakes strategic planning in deciding how the organization should evolve and address changes emanating from the external environment, and overall coordination and balancing of management control decisions, allocating resources among the S1’s. Beer argues that such a structure is necessary for viability in a changing, environment.

Assisted by S2, S3 provides overall structure, integrating cohesive activities of the S1's. S3 is the middle management function, except that its primary activity is to make a "*resource bargain*" with the S1's. That is, S3 three makes resources available in exchange for a commitment by the S1's to meet certain objectives that are agreed upon.

S3 is responsible for the activities of the *"inside and now*" and thereby for internal and immediate systemic control. It also supervises the coordination activities of S2, whilst essentially being the everyday control by senior management of S1.

3.2.5 System Three Star (S3*): Intermittent Audit:

Should S3 need to directly monitor the operations of S1, it may send auditors into the operations to carry out sporadic spot checks, or audits, etcetera. This is a very effective technique for maintaining the requisite variety of S3, being a necessary ancillary system to it and may be defined as an integral part of not only the 'Inside and Now' 3-2-1 homeostat, but the VSM as a whole. Stafford Beer refers to these direct monitoring operations as S3* (pronounced Three-Star). A malleable, reactive, real-time auditor, S3* is an autonomic system in its own right, monitoring and controlling requisite variety according to Ashby's law .

It is illustrated by Figure 3.8, how S3* facilitates the intermittent audit of S1 progress. By providing direct access to the physical operations of a particular S1, it allows immediate corroboration of that progress. Fundamentally, this provides additional data over and above that provided by normal reporting procedures. S3* is a special function to-which S3 is marsupial-like. An Auditor, S3* enables S3 to undertake inspections or audits.

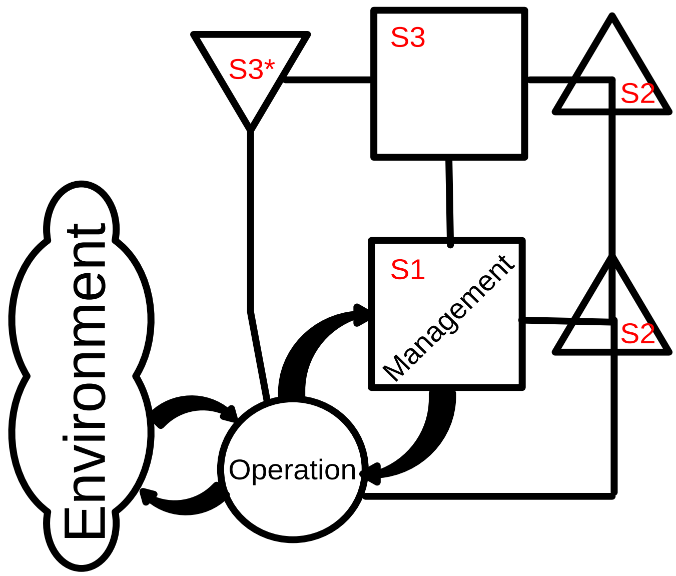


Figure 3.8: Beer's VSM System Three-Star .

This is achieved by its ability to provide direct access-to and sporadic, ad-hoc audits of S1 units. S3*'s objective is focused on checking connections and ensuring that information lags and communication failures do not occur. In this sense, it can be afforded the title of guarantor of quality, cross-checking that the S1 and s3s are mutually effective, although for the most part this should be exercised informally. S3* acts as a backup inspection facility to the validity and functionality of S1 and S3 respectively.

3.2.6 System Four (S4): Intelligence:

In the case of S4, this is a demonstration of a feedforward system by its ability to process current information of operations. This is achieved via the existence of two models within S4 and S5, which symbiotically endorse viability by mapping between the internal capabilities of the system and a changing environment. This enables S4 to contrast the model of the internal

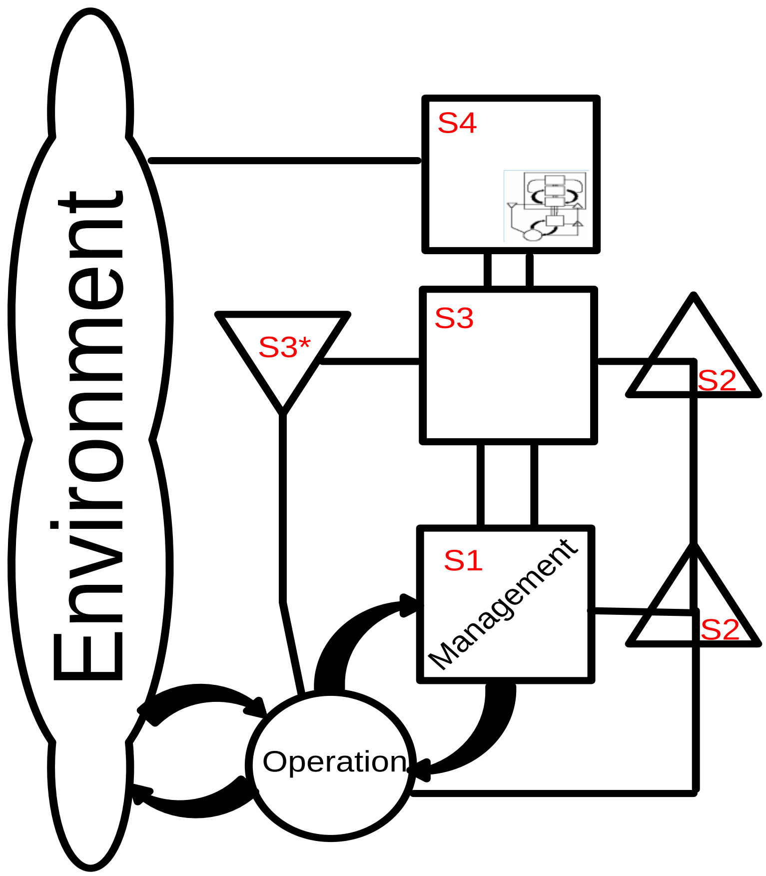


Figure 3.9: Beer's VSM System Four .

capabilities of the systemic whole, with future ideals gleaned from its link to the temporal future forecasting environment and adjust the model of the required world situation accordingly to reflect this. In this way, S4 promotes viability by mapping to the future environment, so identifying benefits and threats emanating from the environment, as shown in Figure 3.9. It is unique amongst the systemic federation, in this respect and that the system must either avoid or take advantage of. The VSM consequently attempts to achieve a real time symbiosis with the environment. Figure 3.9, shows how the architecture could enable S4 to consider the validity of product lines in the light of the environmental climate. S4 thus produces a plan model. This is the sole direct connection to the outside world, likened to the human senses and the mid-brain. S4 is thereby performing future planning, projections and forecasting, yet is arguably not autonomic in its own right. It is important to note that S4 develops strategies via future planning, in the context of a changing environment, whilst balanced with both the external and internal environments. S4 is responsible for activities *outside and then*, and is unique in directly connecting to all of the wider environments. As well as linking to the total environment, S4 requires communication channels to and from S3, as intelligent adaptation cannot be achieved without an understanding of the organization as it currently exists, which is obtained via S3. Systemic adaptations are then fed back through S3 in order to be implemented. There exists a strong interaction between S3 and S4, defined as the *S3-4 homeostat*; necessary as S4 cannot do its job of intelligent adaptation without containing a model of the whole organization and its environment. The quality of this internal model is crucial to the capability of the organization to adapt to change. S4 fulfills this intelligence function, requiring an understanding of the total environment in which the organization is embedded. This is beyond the capability of S1 units as they are concerned with only a sub-set of this total environment. Each recursive S4 also links directly to its parent and subordinate counterparts located within fellow S1's, to promote inter-recursive cohesion.

3.2.7 System Five (S5): Policy:

Member of the S3-4-5 metasystem homeostat S5 is the policy maker within the whole, procuring normative planning to be put into operation by S3, S2 and S1. S5 provides ultimate authority within the VSM S1 homeostats and the entire holism, thereby homeostatically controlling systemic complexity by metaphorically thinking about what the system does. In providing logical closure to the system, S5 monitors the S3-4 homeostat, effectively defining the identity and ethos of the organization and its purpose, maintaining the balance between the management of *here-and-now* and the management of *outside and future*. The combined structure of Systems 3, 4 and 5 as in Figure 3.10, can be said to be metasystemic to the combined structure of Systems 1, 2 and 3. The former grouping is thus logically *over and above* the latter, S3 forming an intersection of these two groupings. Hence, S5 decides systemic identity, governing principles and norms, plus decisions about S4 development plans.

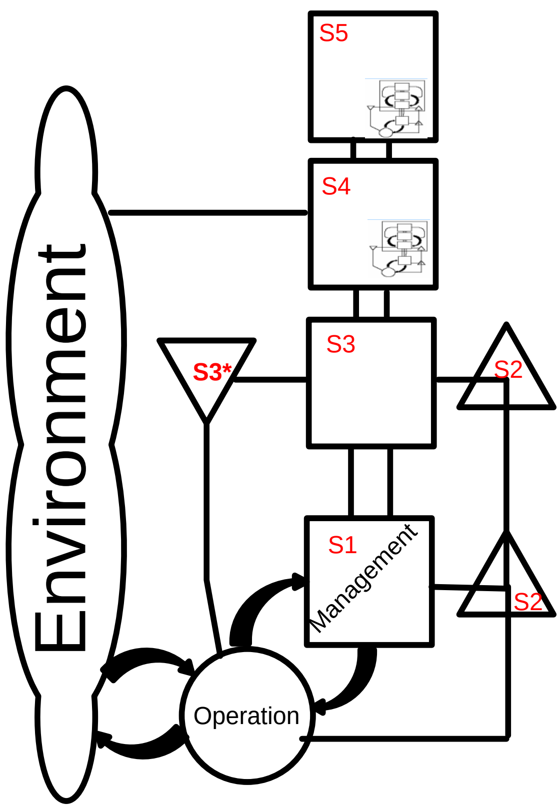


Figure 3.10: Beer's VSM System Five .

3.2.8 System Three-Four-Five (S-3-4-5): Metasystem Homeostat:

The metasystem ensures that the S1s work together in a harmonious fashion, holding them together via cohesion. The metasystem S3-4-5 homeostat balances data coming in via S4 from the wider, external environment of the system in focus (ESF), with the information emanating from the internal environment of the lower recursion (ELR) and plan accordingly.

The role of S5 within the metasystem homeostat is to oversee this process as a whole, intervening only policy is contravened. The metasystem exists to service the requirements of the S1s.

The metaphorical head of the system , comprised of S3, S4 and S5. The metasystem is the composite management vortex masterminded by S5. It presides over and beyond the S3-2-1 homeostat, of lower logical order, yet not necessarily of higher authority.

3.2.9 System Three-Two-One (S3-2-1): Metasystem Homeostat:

Within the quintuple-systemic hierarchy of the VSM, exists a coterie of systems generically termed the *Here and Now*. This encompasses together the S1's, S2's, S3's and S3*. Collectively, they comprise the management of the present operations, with the data passing around these loops and sometimes filters, being real-time.

Similarly, the rate at which measurements are taken should relate to the volatility of the situation, speeding-up when change is sensed and slowing down in the face of stability.

3.2.10 System Four-Five Metasystem Homeostat (S4-5)

In essence, as Systems 1, 2 and 3 are concerned with the *here and now* of the organization's operations, they are considered to be the *autonomic* systems within the VSM federation. Thus, S5 has to soak-up any variety left unbalanced by the operation of the S3-4 homeostat.

If the 3-4 homeostat is working well, there may be little for S5 to do, yet it will continuously receive the signal that everything is fine. This is acceptable, as long as S5 does not fall into a complacent state, and fail to wake up when action is necessary. All viable systems include a mechanism for overcoming this danger. This is referred to by Stafford Beer as the *Algedonic* signaling system.

3.2.11 Models

The VSM contains real-time models of both itself and the environment.

The former is located within S5 and acts as a comparator to the required world situation, thereby allowing for a forecasting and long-term planning capability, whilst enabling the determination of benefits and threats to the system. This also promotes emergence.

The model of the environment of the system-in-focus is produced by the s4 scanning capability and located within this function. This enables systemic viability and emergence through a temporal capacity. That is the S5 model of the internal systemic capabilities is compared to the requirements of the real-time situation represented within S4.

The communication links between the S4-5 homeostat is thus core to timely systemic forecasting and emergence, correspondingly promoting viability.

3.2.12 Communication Channels

The VSM is a recursive, non-hierarchical model, functioning with balances and closed information loops, or homeostats. These interact with a central vertical, communication spine, comprised of four, plus one Algedonic (or alarm), bi-directional channels.

The spine of the system, characterizes viability via flows of information within the system and between its environments. Comprised of four bidirectional, principal channels named *Accountability, Resource Bargain*, *Command* and *Legal and Corporate Requirements*; there exists one *Algedonic* or alarm tributary. Each operates in a reactive, feedback controlled mode. These channels represent the feedback and communication channels between the VSM systems and the environments. This pattern of communication allows the correct type of information to be transmitted in the correct format to the location where it is most needed. The central vertical communication channels represent how the VSM interacts and responds to changes in its environment.

Their function can be explained by how, in the original VSM context of human agency, senior management or the VSM metasystem, in a system controls the actions of operational management partly by striking a Resource Bargain with them. The management of each operation, or VSM S1, has to agree to carry out only certain of the actions possible to them in exchange for a share of the resources of capital, manpower and facilities which are available to the total system.

A resource bargain constitutes a powerful attenuator of the variety which operational management could generate. In exchange for resources, operational management have to be accountable for their actions to senior management. Accountability is another powerful attenuator of their variety.

In addition, senior management implement procedures to ensure that the operational management meet Corporate & Legal Requirements.

There is a two-way channel between them via a Regulatory Centre, or the S2, emphasizing the fact that management should control their operation mainly by regulation of their activity, rather than ad hoc intervention.

The one intermittent, *Algedonic*, or *Alarm* channel, derived its name from the Greek for pain and pleasure. Its propose is to transmit alarms and rewards, escalating through the levels of recursion. This would be in the scenario whereby the actual systemic performance fails or possibly exceeds capability.

It is imperative that all communication channels have requisite variety to handle transmissions. Channels must possess a higher capacity than the variety of the reports, or other entities being transmitted, in order to accommodate errors in the transmission. An example of this would be illegible handwriting. The VSM must ensure that communication along the channels has to be fast enough to keep up with the rate at which variety is generated. Should this not be the case, the system will become unstable, as the stability of the system is dynamic, rather than static.

3.2.13 Viability

The VSM may be considered as a generalization representing the self-management and retention of viability within the human autonomic system, in response to a changing environment.

All systems share the need to remain viable, i.e. the aim of continuing to exist, at least until the time when their purpose has been achieved. As this is a characteristic shared by all self-organizing systems, the VCS research determined to focus on this, and to examine what elements are necessary in order for a system to remain viable. The VSM purports to reveal the underlying structures necessary for a system to meet this criterion of viability, thus it was felt that understanding the VSM, and applying the VCS concept to it, should make it possible to understand the effectiveness.

3.2.14 Recursion

The principle of recursion applies at each and every level of recursion within the VSM, it being of a top-down nature, shown in Figure 3.3. Nested within the higher level S1, each successive S1 possesses the same self-governing principles of organization. This therefore offers an extensible, recursive, model-based architecture, devolving autonomy to sub-systems. The autonomy of subsystems dictates that they too must develop and adopt a method of viable organisation. This is due to the control function being unable to predict a response to the large complexity of all environmental disturbances.

Where appropriate, S1's can be resolved into a lower recursion level that has the same VSM structure. Theoretically, this process can be continued until the 'lowest' level recursion is representative of the individual persons within the enterprise. Levels of recursion are linked in two ways; System One is linked to the next higher level of recursion by the channels of this higher level. In addition, there are direct channels threading through levels of recursion connecting System Five to Five, Four to Four and Three to Three Star.

The VCS research concluded that every recursive call must have a termination condition, to prevent it becoming an indefinite loop. This represented by the VCS spawning mechanism omitting an S2 from the parameter N.

3.2.15 Managerial Cybernetics

Managerial Cybernetics was devised by Beer and uniquely demonstrates the role of cybernetics to the management task within the VSM.

Alternately referred-to as the science of effective organisation, Beer believed that the notion of using managerial authority to deal with organizational problems was a short-term solution.

The underlying principle of Managerial Cybernetics is thus to devolve traditional central autonomy to lover level management, thereby creating autonomous units of management that, in crude terms, know their own business better than their counterparts. In basic terms, each and every S1 has a powerful investment in their own identity. Each seeks to define its identity, to maintain it, to flourish out of a commitment to itself and a confidence in its selfhood. Each is part of a wider, holism whose primary purpose is to preserve identity that is to survive.

Managerial Cybernetics dictates that survival can only be achieved if a system is able to change and be gradually modified as the world changes. This is known as adaption and is the key to systemic survival, that is, viability in a changing environment.

3.2.16 Variety and Variety Engineering

Variety is a measure of the number of distinct states, in which a system can be. Variety engineering is the notion of balancing the varieties of systems with different variety levels to their environment, illustrated by Figure 3.11. In general the environment of a system-in-focus may engender what would be construed as huge variety.

The VSM *Operation* element will contain much less variety and the *Management* - even less variety. This is achieved through *attenuation* and *amplification*, as demonstrated in Figure 3.2; two characteristics of varietal control falling under the umbrella of the term *variety engineering*. Beer's model applies variety via its blueprint architecture of communication channels within fractal-like recursive architecture.

These channels link between the systems and their respective environments, assisted by four bi-directional communication channels and one alarm tributary, Algedonic. In order to not only control the temporal flow of information, but also to translate the said, it is necessary to transduce this data at point of departure and receipt. Similarly, the information may need to be amplified or attenuated as necessary and functionality exists within the architecture to so do.

The variety engineering process regulates systemic complexity, feedback control being key to viability by empowering the homeostatic loops with a common endeavour to attenuate the variety emanating from the parent S1, yet amplifying the variety in terms of its' respective environments.

In attaining requisite variety, the location of the command centre is determined by the data available to a concatenation of systems. This dictates the important elements and systems in real time, so promoting self-organization.

The Law of Requisite Variety states that control can only be attained if the variety of the controller is at least as great as the situation to be controlled, the variety engineering process regulates systemic complexity. This dictates the relevant elements and systems in real time, promoting self-organization.

Ashby's law of requisite variety states that:

'A controller has requisite Variety when he has the capacity to maintain the outcomes of a process within targets, if and only if he has the capacity to produce responses to all those disturbances that influence the process.' [11]

This means that situational variety, as exposed by the system in different situations, must at least be equalled by the response variety of the controller. This is based on Ashby's cybernetic law that:

'Only variety absorbs variety' [11]

In the context of the VSM, in relation to furthering this VCS research the study has drawn especially from the Conant-Ashby theorem that states:

"Every good regulator of a system must be a model of that system" [17].

This considers the management of systemic complexity. In the instance of a system interacting with its local environment, this may in turn, be managed by a management unit where each pair strives for mutual homeostatic equilibrium. Whilst the management unit seeks to control the system, that system is similarly attempting to control the environment. The complexity, or variety, exhibited by the system will typically far exceed the complexity of the management unit. Similarly, the complexity and consequent variety apparent in the environment will again generally far exceed the variety of any system that is trying to control it can display.

Each controlling element has to absorb the variety of the element it is attempting to control, else the controlled situation may assume states for which the controller had no response.

Control therefore can only be achieved in the instance where the variety of the controller is at least as great as the situation to be controlled. This emanates from Ashby's Law of Requisite Variety, which proposes that:

"only variety destroys variety" [11]

As a varietal imbalance generally exists between the proposed said environment, the system and management unit, this can be resolved by either the variety of the controlled situation being reduced or attenuated to the number of states that the controller can address, or the variety of the controller may be amplified to match or exceed that of the controlled situation.

The mutual dependence of amplification and attenuation, dictates that they are used together in order to achieve the requisite varietal balance as illustrated in Figure 3.2

3.2.17 Homeostasis

This is the Cannon [63] term that Beer felt was applicable to define the constant state of the internal VSM environment. This was achieved via the varietal processes and Managerial Cybernetics activities inherent within the cybernetic model's topology and functionality.

In the context of the VSM, an Operation unit can manage its Environment, as long as it can successfully absorb the variety from it, by attenuating the incoming variety, and amplifying its own variety back to it. Likewise, a Management unit can cope with the Operation as long as it can successfully absorb the variety from it, by attenuating the incoming variety, and similarly amplifying its own variety back to it.

In this event and if these requirements are met, the VSM can maintain Homeostasis. This means it can maintain itself in a state of equilibrium.

Should this not be the case and these requirements are not met, the system will become unstable, eventually leading to it being unable to retain viability in a changing environment.

The VCS research has advanced the state of the art, to-date with a notional representation of homeostasis within an open-bounded environment [3, 24].

3.2.18 Autopoiesis

Beer stated that only a viable system could exhibit autopoiesis at all, since autopoiesis is defined as a:

‘‘*characterization of life*’’

Maturana and Varela originated the concept of autopoiesis from biological systems [51]. They characterize the living body by its self-maintenance at the cellular level and as self-referential from the nervous system. Only the cellular level, i.e. metabolism, is autopoietic. The nervous system maintains the homeostasis of the organism and in this sense it is related to autopoiesis.

Beer recognized the importance of autopoiesis to the VSM, in terms of the self-production of S1’s and their constituents, expressing the set of necessary functions for the viability of a system. This function is maintained over recursive levels via self-reference and autopoiesis. The VSM thus possesses principles and axioms, such as variety engineering that maintains the self-referential property. Varela interprets autopoiesis as viability and believes that it is the basis of stability, the VSM systems comprised of an accumulation of components, i.e. the basis of the structure is the coupling of components.

Moreover, all unities make the closure on their structure and function autonomously; they need integrative principles which maintain cohesion. It is more important that these components and their coupling can emerge and retain viability within the VSM by their structure order intrinsically. It is believed this will maintain the balance between development and order.

As Beer applied mappings from the VSM to the human autonomic nervous system, this research deduces that it is theoretically possible to thus interpret the VSM as functions, able to be autonomically mapped to a conceptual nervous system.

3.2.19 Amplification

A key feature of the VSM is the management of variety. In the human agency context, people in organization need information to perform their jobs effectively, but too much information can be a distraction. What is needed therefore is both variety attenuation and variety amplification. An example of variety attenuation is the environmental scanning activity. Differing people in an organization must keep up with new technology, new government regulations, and what competitors are doing. From a great variety of sources of information, they select the information that is most important for the decisions the firm must make.

Variety amplification, on the other hand, refers, for example, to the distribution of the organization’s messages. Advertising messages go outside the firm. Plans, policies and procedures need to be distributed within the firm.

The VSM is very useful as a guide to studying where variety is attenuated, where it is amplified, and if there is a balance in the varieties of interacting sub-systems.

3.2.20 Transduction

Each entity in a self-organizing system has its own “language”. If one considers, for example, a company which manufactures cars, the language used by production engineers in trying to resolve a problem on the production line is quite different to the language spoken by the directors at a board meeting. These languages are likely to be mutually incomprehensible. The same applies to the language used out in the environment and that used in the operation itself.

Whenever a message crosses a boundary, therefore, it needs to be translated in order to continue to make sense. This process is called transduction. If the transducer does not have requisite variety, the message gets garbled or lost. Another familiar example is where a message is taken by somebody’s secretary, and then never gets any further.

Transducers are represented in VSM diagrams by circular dots at the boundaries between channels and other entities.

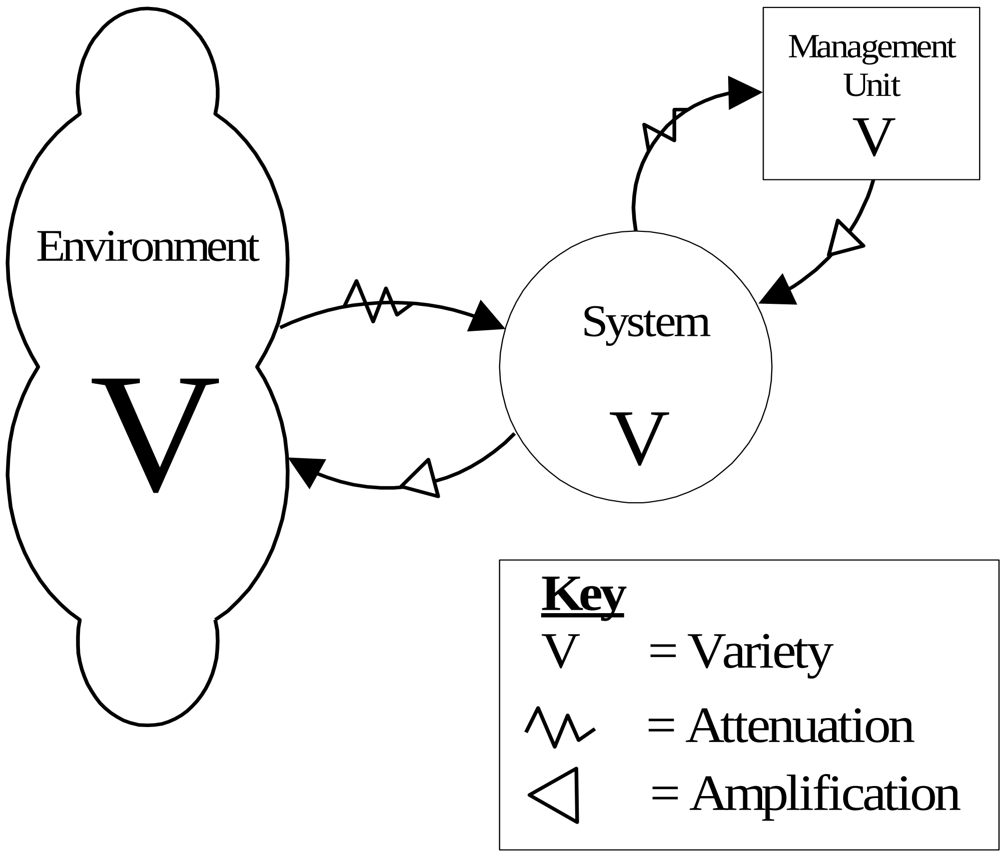


Figure 3.11: Representation of Variety Control

3.3 Bratman et al’s Intelligent Resource-Bounded Machine Architecture (IRMA)

The IRMA [95] is a classic software beliefs, desires intention (BDI) agent model drawn from the Artificial Intelligence (AI) arena during the late 1980’s as shown in Figure 3.12. Many of the elements contained in the IRMA model correlate directly to elements of the VSM. For example: Planning & Means-Ends Reasoning can be analogised to S3, Opportunity Analyser & Filtering to S4, Beliefs about the world relates to S5, as to the Desires and Deliberative Process. The Intentions can be related to S5 Policy passed to S3 for enactment.

The IRMA attempted to implement agents based on the BDI model the plan function returns plans from a plan library; a set of pre-compiled plans. An intention structure then structures various plans into larger hierarchies of plans. An intention in the intention structure in the classical BDI theory is a partial plan structured as a hierarchy of sub plans. Furthermore, sub plans may at some point be abstract, waiting to be ‘filled in’.

The significance of the IRMA to the VCS research, is that Laws et al innovatively drew a correlation between the Bratman and Beerian models, via their J-Reference Model [31] in their bid to promote autonomy versus governance.

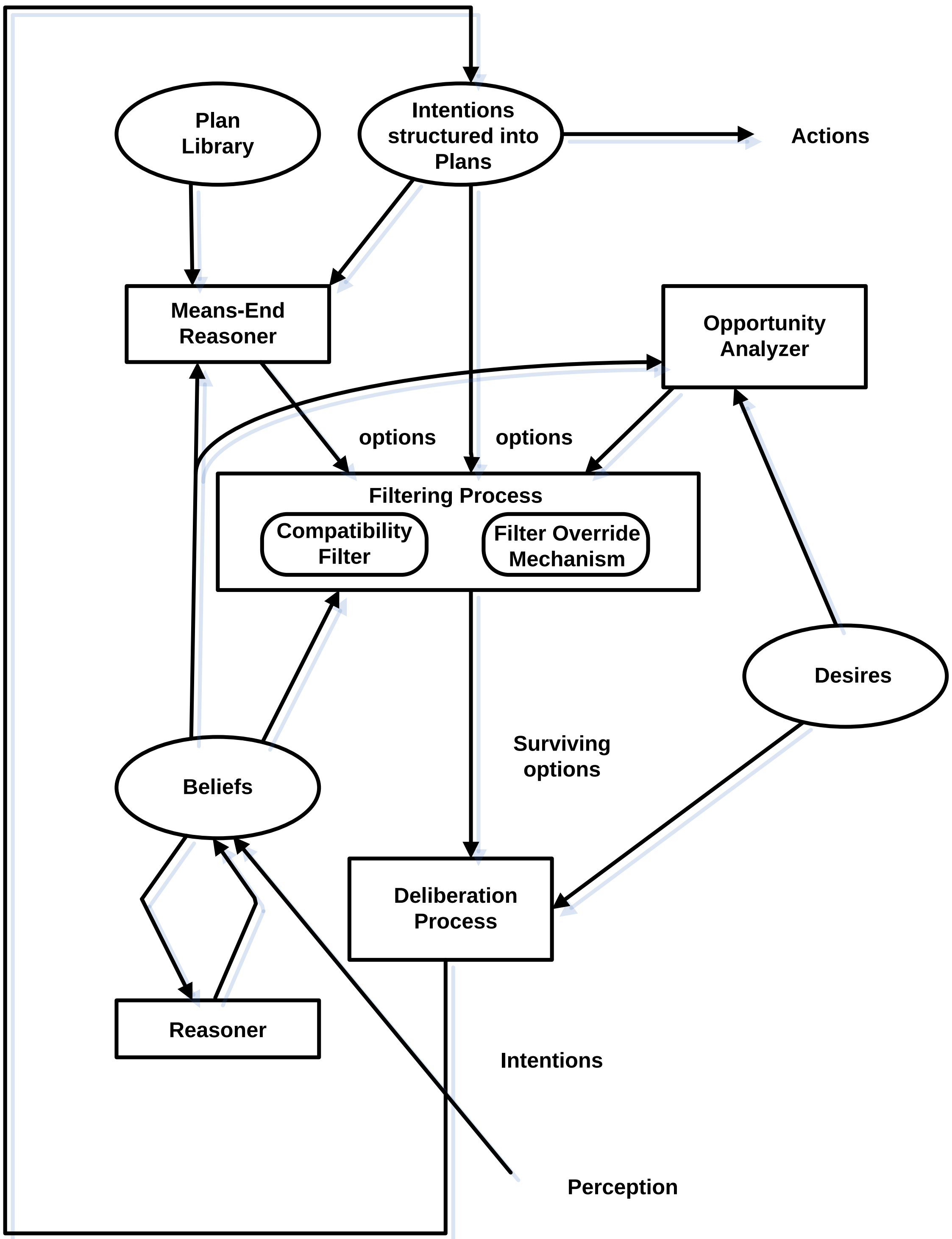


Figure 3.12: Bratman et all's IRMA Architecture

3.4 J-Reference Model

The J-Reference model, developed by Laws et al. [31], is considered as the first comprehensive adaptation from Beer's' human-agencied VSM towards the computing, software agencied context of the VCS. As illustrated in Figure.3.13, it mirrors Beers metasystemic topology by incorporating S3, S4 and S5, with S4 similarly scanning the environment.

Likewise, recursion is exhibited via the presence of a lower-level S1, emphasising importantly, that this model is of top-down decomposition, in contrast to the design grammar model that is bottom-up and atomically derived.

The presence of the two internal models, one of the systemic capabilities and the comparator model of the wider systemic environmental requirements is core to the viability of the system

Retrospectively, the J-Reference Model was arguably a greater landmark within the genre than recognised at the time, as it emanated from the self-adaptive software movement that precipitated and was debatably the forerunner of, the autonomic computing arena.

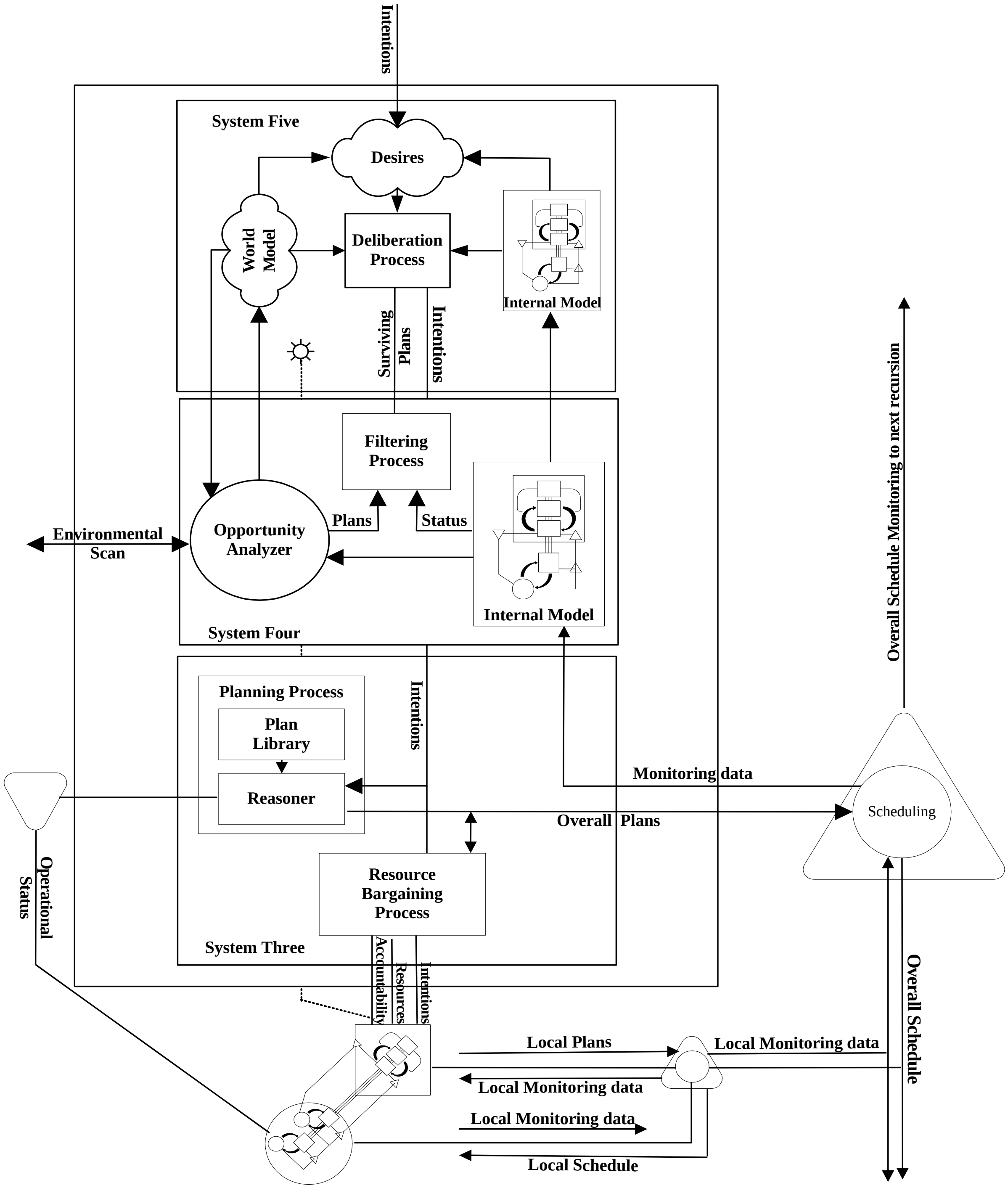


Figure 3.13: The J-Reference Model

3.4.1 IRMA Architecture Related to the J-Reference Model

One can see in Figure 3.14 how an analogy was drawn to the properties of Bratman et al's Beliefs, Desires, Intentions (BDI), IRMA, to those embedded within the J-Reference Model [31] in a bid to further the research.

Specifically, both models contain a *Plan Library* and a *Reasoner*. These are located within S3, of the J-Reference Model, likened to the IRMA, S3 is also where a *Reasoning Process* occurs. The IRMA *Opportunity Analyser* marries to its namesake in the J-Reference Model S4. Bratman's *Intentions* is reflected at the J-Reference Model S5 location, where the BDI *Beliefs* relate to the *World Model*. The *Deliberation Processes* are also linked at S5.

Further investigations determined, however, that elements are apparently missing for viability and thus could be translated to the VCS and that similarly the IRMA carried a heritage of Artificial Intelligence in that perhaps an isomorphic (complexity inducing) plan would be required, or produced as opposed to an homomorphic analogue (complexity reducing) analogue.

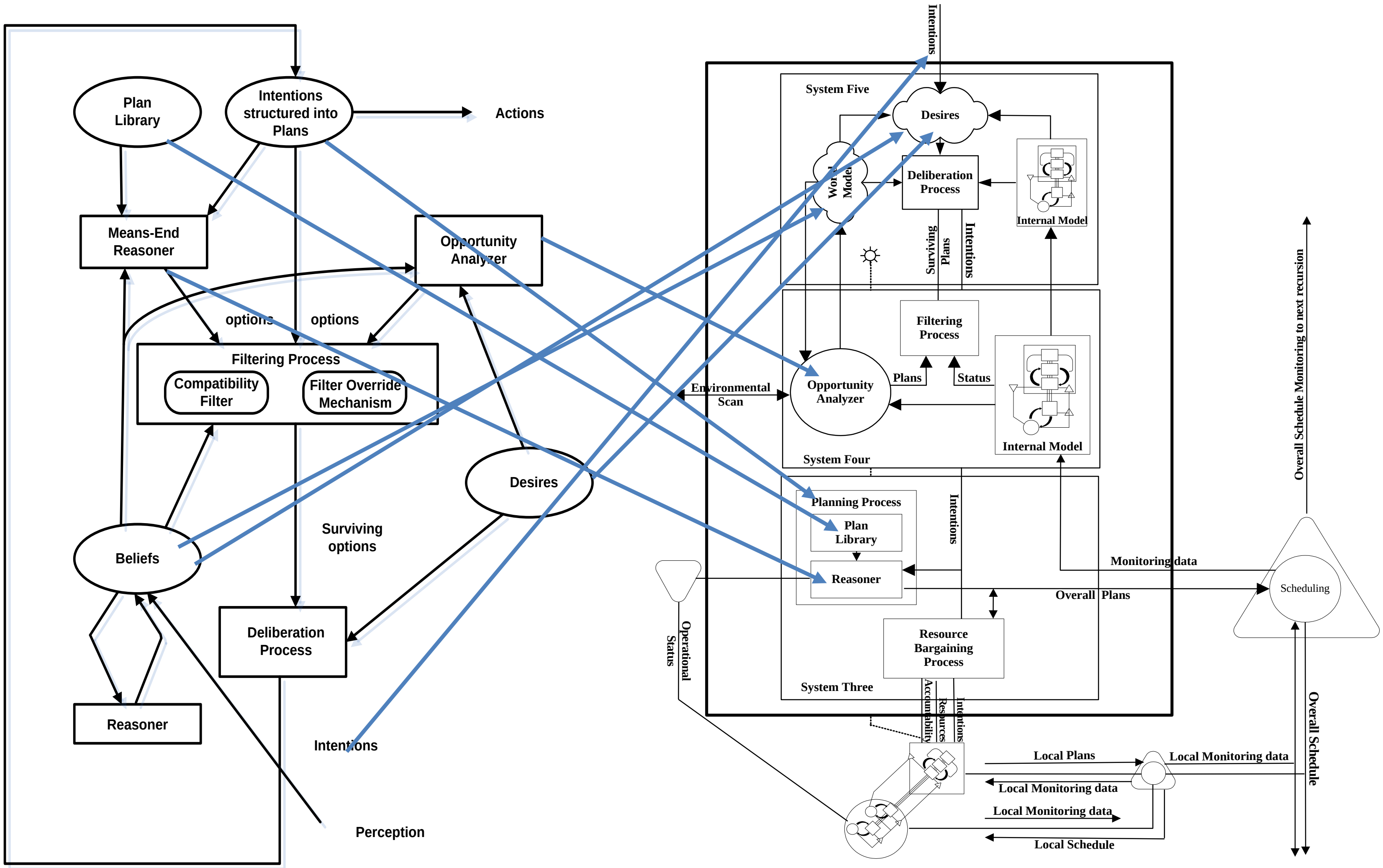


Figure 3.14: IRMA Architecture Related-to the J-Reference Model

3.5 Legacy System Syndrome

In 1995, Keith Bennett coined the term Legacy Systems Syndrome [6]. Many practitioners in the industry may have an intuitive understanding of what a legacy system is, yet it is hard to find a single definition that describes one adequately within the literature. Bennett summarised them as

"large software systems that we don't know how to cope with but that are vital to our organization." [6]

The software may have been written many years earlier, using outdated coding techniques, and yet despite its' age may continue to perform useful and very often essential work to an organizations day-today functioning. Managing and updating legacy systems may incur technical and non-technical challenges such as justifying the expense of perhaps offshore contractors who may be the only personnel available who understand the technicalities, and to the growing code size and other general maintainability. This is because it may perhaps have been written in Assembler or an early version of a third generation language. A legacy system is probably not developed using state-of-the-art software engineering, but programming pre 1968 [90] techniques and yet many perform crucial work for the organization. Legacy systems are notoriously large and generally difficult to understand by more than one or two persons within an organization. The dichotomy is, however, that they may conversely be used by many and essential for the purpose of the organization.

3.6 Autonomic Computing as per the VSM

The nexus of this research, appears to focus on the potential to produce a cognitive, organic computer system that can satisfy the self-management CHOP qualities Sterritt and Hinchley , also propound that the vision is to create software through 'Self' properties'. They acknowledge Horn's [8] initial set of four properties, in terms of autonomic computing objectives, as being self-configuring, self-healing, self-optimizing and self-protecting, along with attributes of self-awareness, self-monitoring and self-adjusting. Sterritt declares that this "self*" list has grown to: self-anticipating, self-critical, self-defining, self-destructing, self-diagnosis, self-governing, self-organized, self-reflecting, and self-simulation. He continues by promulgating his belief that many new biologically-inspired metaphors being developed and incorporated into future autonomic systems .

Originally the main protagonists in this research directive, IBM now have independent peers who are researching in a similar orientation. These include the large information technologies of Microsoft's *.Net* , Sun's *N1*, and Hewlett-Packard's *Adaptive Infrastructure*, focussing on devolving traditional central control to inventing a means to form complex self-organizing computational environments.

Decades earlier, In the context of human systems, rather than computing, Beer had partly pre-empted Horn's analogy , by introducing a generally applicable model relating to the Central Nervous System (CNS). His cybernetic: *'neurophysiological'* Viable System Model, collectively applied von Bertalanffy's Organic System Theory (OST) [96] and his later General System Theory (GST) [97], stating that an open system must be able to adapt to a changing external environment that might threaten its long-term viability. Beer lauded Cannons' biological *'homeostasis'* [9] principles, whilst similarly referencing Ashby's text of the same name Wieners' *'cybernetics'* [37] teachings and Ashby's *'ultrastability'* were underpinned by his key laws of *'Requisite Variety'* that Beer had successfully synthesized through his systemic and cybernetic doctrines. Similarly, he also praised others such as the Chilean neuroscientists, Maturana and Varela's *'autopoiesis'* [50] concept.

The multidisciplinary fusion, manifested by Beer's quintuple-hierarchy VSM topology, has been utilized by a number of researchers as a vehicle to push forwards the theoretical boundaries of development into autonomic computing.

Espejo's collaboration with Harnden [68], had applied a cybernetic slant to the modelling of agent communities, yet circa 1999, Laws et al. innovatively drew a parallel between the cybernetic properties of Beer's VSM and the zeitgeist complexity reduction requirements of the software industry [64].

In applying the VSM to problems that were later identified as pertinent to the then autonomic computing ideal, the fusion subsequently produced the 2001 J-Reference Model [31]. Displaying the existing cybernetic topology, it united with both Bratman et al.'s IRMA architecture [66] and the Beliefs, Desires, Intentions (BDI) framework, whilst employing the Ashbian systemic variety concept to the endogenous complexity proliferation [6] pre - the millennium bug issue [98]. The BDI framework application, however, unfortunately introduced recognised problems from Artificial Intelligence; for example successful application of a real-time, isomorphic model [67], depends upon a real-time BDI complete model, necessitating context-sensitivity. Laws et al's research continued however [31, 69-71], later running concurrent, yet unparalleled, to others drawing biological homeostatic analogies such as IBM's Horn , Kephart and Chess

[72]. Contemporaries including Herring [73] and latterly Stoyanov [74], applied the VSM blueprint to their autonomic computing research. Stoyanov proposed that the abstraction of observable variety and a managed communication channel was core to development of viable, autonomic computer systems, whilst outlining the importance of the runtime capability verification of interacting components [74].

In 2003, Laws et al took an apparent minority, cybernetic, perspective of self-adaptive software. Drawing a correlation between the potential of cybernetics to apply natural, inherent, adaptation strategies to software artefacts, they reported on the success of an experimental agent-based, adaptive system. Further research has been undertaken on the Beliefs Desires Intentions Agent Model, expounding that an agent has beliefs about the world and desires to satisfy, driving it to form intentions to act. These centre on beliefs about the environment and other agents, desire or goals to achieve and intentions or plans to act upon or to achieve its desires [99].

In 2005, Randles et al. [100], superseded their previous papers by a introducing a cybernetics-based viable system architectural model. This encapsulated an Enhanced Beliefs Desires Intentions (EBDI) framework, to further autonomic computing that led to a practical implementation using the implementation of a grid-based medical decision support system: *Clouds* architecture [101] with the custom designed meta-language expressed and enacted through a bespoke declarative meta-language, *Neptune* [102] mapped from the situation calculus

A 2007 publication innovated a mathematical set theory, design grammar model of the relationship between the VSM systems [1] exhibited via the subscripts, thereby surpassing the autonomic computing ideal, towards an original concept of Viable Computer Systems, or VCS. Subsequent publications on ecological dependence highlighted how this assists to negate redundancy and complexity. By referencing Lehman's:

'model of a model' [67]

notion, thereby articulating a sense of viable self. Proof of concept was demonstrated in a 2009 closed environment scenario case study [2], when applied to a previous genetically modified system scenario [23]. This evolved a distinct design grammar model, relating the recursive levels of the VSM, represented via the superscripts. An original topology was also presented, demonstrating the theory via algorithmic sorting. The inherent existence and importance of feedback control within the software process [76, 77] was exhibited by promoting inter-recursion cohesion to reduce redundancy and so complexity. This was furthered by a 2010 open-environment case study [3, 24] that demonstrated research uniformity by application to the same, previous genetically-modified system scenario [23] as the closed environment case study [2]. System One was represented as a metaphor to exhibit VCS homeostasis. This was achieved via reference-to and application of set theory notation from Sommerhoff's:

"directive correlation" [25]

tenet qua Ashby's:

"goal directedness" [22]

notion. Furthermore, a progressed architecture was presented, applying examples of the extended design grammar identities to the framework.

It is believed that this research has uniquely fused these requirements with the principles of the autonomic computing genre in the context of cybernetics, with a mathematical analogy to the VSM. In so advancing to multi-agent self-organization of software technology; a context-digression is achieved from the human-oriented Managerial Cybernetics. Research novelty lies in the blending of Beerian , Ashbian [17, 22] and Sommerhoffs' [25] concepts with autonomic computing to innovate a conceptual model-based system and formalism that, circumvents previous approaches and in so doing, demonstrates a tangible theory of open-bounded homeostasis [3, 24].

3.7 State of the Art Developments

There has been remarkable progress in the use of computing technology over the forty-plus years since Release 1 of the OS/360 and the zeitgeist establishment of software engineering as a discipline, at Garmisch. There has likewise been a growing recognition that continued software maintenance display quite different characteristics to other, tangibly engineered product. As Lehman noted in his seminal 1980 paper:

> *'…software must evolve, undergoing continuous adaptation and change. It must be treated as an ever to be adapted organism, not as a to be produced once artifact.'*[32]

Until recently, the state of the art within software engineering had been to attempt to foresee every eventuality that the program may encounter and supply a gamut of functionality to accommodate those situations. This was married with limited adaptability that may have been provided via the appliance of alternative control paths activated by run-time decisions [103]. In practice, however, deviation from any of the conditions, that had not initially been accommodated in the program model or implemented software, will necessarily lead to manual, human-agent intervention. This will inevitably be not only time-consuming, restrictive and perhaps costly, but more importantly may introduce more errors into the code base, leading to both its' physical growth and complexity. This continued endogenous ageing will quickly degrade the software and lead to a fragile, increasingly incoherent structure. This will therefore be risky to change for fear of upsetting the internal balance. Any potentially successful amendment may still incorporate more complexity due to a lack of thorough understating of the code base. Lehman more recently expanded upon his *model of a model* mantra, the latest clarifications highlighted by the square-bracketed bold text:

'A program is a model ***[by the normal definition of model]*** *of a model* ***[the wider application solution system, which must include a computer (or other program execution mechanism) but (normally will) include other devices, machinery, equipment and even human organisation(s)]*** *within a theory* ***[the specification]*** *of a model* ***[the requirements statement or description of the application to be addressed]*** *of an abstraction* ***[abbreviated definition that omits all the properties of the application and its operational domain that are considered irrelevant to a satisfactory solution and/or control and or implementation of the desired application or that have been overlooked]*** *of some portion of the world* ***[since the real world has an unbounded number of properties but humans can only process and manage a bounded number]*** *or of some universe of discourse* ***[that is a sub-portion of the real world which also has an unbounded number of properties].*** [104]

This notion advocated the development of a software classification scheme that emanated from not only the software system per se but, crucially, the environment in which it was executed and embedded.

This prerequisite, Conant-Ashby theorem [17], dictates that the most favourable regulator will be an isomorphic model of the situation to be controlled. There will, however, exist instances where such isomorphism is not possible as in highly complex systems, dictating that the regulator must possess a strongly homomorphic model of the situation.

As Laws has identified:

"…to ensure effective control or regulation of a controlled situation requires that the controller models the situation to be controlled, otherwise the situation may adopt states that are meaningless in terms of the controller" [105].

Lehman based his taxonomy upon the degree of homomorphism required to obtain a suitable mapping between a potential domain and the software model itself. Specifically, by classifying programs according to their relationship to the environment in which they are executed, sources of evolutionary pressure on computer applications and programs is identified.

This exemplifies why this results in a process of high maintenance activity, the laws of program evolution having been formulated following quantitative studies of the evolution of a number of different systems. The resulting classification practice categorizes programs into three classes, S, P, and E-type software systems [32].

In essence, *S-type* software addresses a requirements domain where a completely specifiable or isomorphic mapping is possible from a specification, the resultant program being conceptually static.

P-type software cannot be fully specified and thus approximation and assumption are introduced to the problem domain to produce a homomorphic mapping. The program is completely specified, for example by the rules of chess plus procedure rules which must indicate how the program should analyze the state of the game, whilst determining possible moves and provide a decision rule to select a next move.

E-type software is embedded in a real world situation and thereby becomes part of it and further, causes change in that environment in which it is executed. This leads to a feedback system whereby the software and certain elements within its environment to both evolve symbiotically and become mutually dependant to a degree [106]. E-type programs are inherently more change prone than their counterparts due to their nature of mechanizing a human or societal activity leading to the program effectively becoming a part of the world it models as it is embedded in it. The program as a model thus contains elements that model itself, the consequences of its execution.

The methodology was based upon the evolution of IBM's OS/360 and its successor OS/370. This research has continued and Lehman's laws are observations that are expected to hold for E-type systems, irrespective of specific programming or management practices [107].

3.8 Literature review

IBM's 2001 autonomic computing program [108] voiced the rising complexity in software systems that provoked Legacy System Syndrome [6]. Horn's self-CHOP (configuring, healing, optimizing, and protecting), acronym sought autonomy versus human governance.

Laws et al. had in 1999 proposed the cybernetic properties of Beer's VSM as a solution [64], precipitating IBM's autonomic computing initiative via the 2001 J-Reference Model [31]. Fusing Beer's model, Bratman et al.'s IRMA [109] and Beliefs, Desires, Intentions (BDI) theory, the prototype allied Ashby's variety notion. Homomorphism countered isomorphism [67] causing endogenous complexity.

The ongoing VCS work [31, 69-71], diverged from fellow homeostatic analogies, e.g. IBM's Horn, Kephart, and Chess [72], and a fellow contemporary in the form of Herring [73]. Stoyanov's [74] management of a communication channel formed viable, autonomic software systems [110].

Espejo and Harnden [68] cybernetically modelled agents, whilst in 2006, Laws, Bustard et al., merged autonomic computing, the VSM and Soft Systems Methodology [75].

3.9 Viable Computing Systems

The VCS research focused on the credo that the general theory of homeostasis possesses extensive implications for the theory of self-governing intelligent systems, therefore seeking to develop a formalism to diminish ambiguities that may be present within verbal communication.

Similar to Ashby [10], it was presupposed that the concepts vocabulary and symbols of the Bourbaki School [111] would be well-suited to the testaments and operations of this theory and the research objective. Bourbaki believed that most things could be represented via set theory and this was drawn from as a perspective of Ashby's research into homeostasis.

The latter, open-bounded environment research [3, 24] into his *goal directedness* [22], in the context of Sommerhoff's *directive correlation* [25] tenet, has sat exceptionally well with the preceding VCS investigations.

Through the fusion of a mathematical analogue with the underpinning functionality of Beer's cybernetic Viable System Model (VSM) , a set-theory blueprint has been formulated as the basis of a design grammar model. A fundamental building block in its' realization has been proposed in the form of a context free design grammar.

The evolving system will potentially be realized as a Viable Computer System (VCS), able to retain its' viability via self-organization and emergence, so managing complexity. The aspiration is that the VCS will operate in an intrinsic, reactive, forecasting mode, ready to respond to environmental stimulus post t^{n-1}.

A referential *self* model of the internal capabilities of the system i.e. the status quo t^n and a model of the wider systemic environment, dictating the required world situation t^{n+1} will be employed.

Necessary symbiosis between past, present and future events will be accommodated via transposing the sensor/effector principles from the VSM . Homeostasis and feedback control is thus core. One derivative of the rigorous formal model has been the uncovering and clarification of some grey areas in the VSM.

This research has aimed to maintain the relevance of any given variable inside the complex system, facilitating change and satisfying the imperative of a self-organizing system to continuously emerge. This research has thus focused on producing a model that will embody and handle this complexity and the associative dynamic nature of the system.

In demonstrating such proof of concept through case studies, it is felt that this research exceeds the state of the art of the autonomic computing genre partly through the principles of homeostasis [112], and autopoiesis [113, 114]. The combined adoption of the mathematical, biological and cybernetic modelling approaches has sought to enhance complexity reduction by negating redundancy.

The research has endeavored to notionally facilitate the generation of timely models, referenced and responded-to by the system. The ultimate aspiration has been to speculatively reflect the real world and autonomic ally address environmental requirements. This would be accomplished via reformulation of the existing recursivity present within the VSM architecture by algebraic set theory.

To-date, a publication portfolio has advocated a context-free design grammar as appearing to offer not only the potential for modelling a dynamic system by providing mechanisms for generation of an internal representation but also a topology imbuing the system with self-awareness [1-4, 24]. The biologically-inspired self/non-self dichotomy [115] principle has thus been one of the core facets to the VCS research.

A 2007 design grammar model relating the VSM systems [1], innovated Viable Computer Systems (VCS). In recognizing that Self-organization is a propriety emerging bottom-up, in principle, a system was modeled without any high-level representation. The VCS was based on a large number of components that interact according to simple and local rules and in which a global organization of the framework can atomically emerge from the resultant local interactions.

This later linked the inter-recursion cohesion, profiling feedback [76, 77], followed by research on ecological dependence highlighting how this assists negation of redundancy and complexity. By referencing Lehman's 'model of a model' [67] notion, a sense of viable self was articulated, prior to publishing a 2009 closed environment case study applied to a previous genetically modified software system [69] that exhibited VCS self-governance [2], within the context of a previously published genetically modified system scenario [23].

More recently [3, 24], von Foerster's:

'order from noise' [80],

paradox was briefly examined inspiring an open environment VCS case study [3, 24] that applies Sommerhoffs *directive correlation* [25], qua Ashby's *goal directedness* [22] notion, with algorithmic hot-swapping to submit a theory of open-bounded homeostasis. Furthermore, this constructed and innovated an architecture that details the application of the extended design grammar, in addition to original, example identities.

These points will each be expanded upon further, throughout this thesis.

3.10 Chapter Summary

The key, underlying research goal has hence been to progress autonomic computing towards the development of Viable Computing Systems, thereby extending the concept of autonomic computing systems to correspond with the way that human autonomic systems are subsumed by cognitive systems. This has been manifest through the construction of the design grammar model. By fusing a, mathematical analogue with the underpinning functionality of Beer's Viable System Model (VSM) , a set-theory blueprint has been developed as the basis of a design grammar model of the VCS.

Chapter 4

VSM Synthesis towards the VCS

Introdection

This chapter provides an informal description of the fundamental requirements of Viable Computer Systems and then goes on to consider how a model can be developed that can be used to represent prototypical model of a Viable Computer System, based on the VSM. In essence, the chapter sets the scene ready for the presentation of the formal design grammar model.

4.1 Requirements

This research view of a VCS is that of a hypothetical formal model, which could serve as the basis for the development of a physical Viable Computer System. Such a VCS may range from the conventional desktop computer systems that is known and use today to an embedded system for which the input is sensed and the output drives actuators. Before development can commence of the VCS model, it is necessary to consider some of the basic requirements that a VCS must satisfy:

Viability

Recursion

Autopoiesis

Internal models

Forecasting

4.2 Use of Set Theory as the VCS Modelling Formalism

Within this framework of ideas, mappings and formulae, this investigation aspired to find a representation of homeostasis [63, 116].

Although other programming languages such as Z or Lisp may have potentially have been chosen, for consistency purposes, this research chose to adopt the Ashbian use of set theory towards this end [22]. Significance lay in the fact that Ashby had, in turn, openly drawn from the Bourbaki stance that all mathematics can be based on set theory [111].

In order to further expand the VCS design grammar model, this investigation deems that the repeated use of algebra empowers the elements in a set to become numbers, or for the functions to be continuous if so required. It is similarly felt that fundamental VCS modelling facets such as atomic recursion, temporality and latterly, the fusion of directive correlation to engender open-bounded homeostasis have been afforded a good fit to the algebraic method.

The mathematical analogy can thus be perceived as solely a medium to articulate research concepts and ideas. Set theory permits the featuring of production rules, a symbol set, and vocabulary including atomic elements of the language.

4.3 Design Grammar Model

The VCS design grammar model is detailed by previous research [1] as a unique, formal, system-wide and context-free algebraic set theory syntax analogue of Beer's VSM . Relationships are formulated from a dual perspective: two sets of rules characterize the relationship between the systems (subscripts) and the recursive levels, (superscripts), based upon a set of production rules and symbols and a vocabulary detailing atomic elements of the language.

In this way, the atomic level, (S_0^0) consists of non decomposable constituents, able to autopoietically generate higher levels. The syntax represents the five main systems as, $S_1 \ldots S_5$, plus a novel system nought (S_0), the new system allowing for dissection of the Beerian S1 . This bifurcation results in the creation of the said S_0 via removal and juxtaposition of the original S2, outside of the metasystem. This process creates a facility for atomic emergence, S_2 acting as both the initiator and terminator of recursion when respectively unioned and omitted from S_0. Atomic breakdown of Beer's model demands separation of his S1 into two parts, the VCS S_2 being isolated from the metasystem, yet remaining juxtaposed to S_3 and enclosed by the boundary of the new S_1.

These unchanged management and operation units are re-classified as S_0: inert as a component until joining with its associated S_2, thereby creating a VCS S_1.

The design grammar echoes the VSM's indefinite recursivity, at a post-atomic level, having no specific starting point or initial conditions. Aspirant to produce a VCS that reduces human agent intervention, it reflects Horn's self-CHOP benchmark [108].

Three levels of recursion are defined; the *atomic* level, termed recursion nought (S^0) with S_0^0 holding constituents able to autopoietically, recurse, promoting emergence with no explicit starting point or initial conditions. All higher levels are generic (S^n), the highest (S^N) necessarily lacking an S_2 to terminate recursion. This recursive syntax promotes stability in the chain of operations, as constant values retain their configuration and efficacy as functions are continually executed upon them. A relationship follows its recursive string, promoting self-stabilization; aspiring a VCS to homeostatically satisfy Horn's self-CHOP benchmark [108]. System-environment dependency sustains emergence and viability [117], exhibited via S (system) integrating its' E (environment) thus: $S \wedge E = S \cup E$. The environments particular to the said systems are specified via the notation E_1 , E_0 etcetera, an exemplification being where S_1 is equal to S_0 in union with S_2, E_1 will equal E_0 in union with E_2.

Application of Sommerhoff's *coenetic* (pronounced *'sennetic'*) *variables* [25] emanating from the Greek meaning for common. These simultaneously delimit variety so that trajectories of the system converge on a subsequent occurrence. Sommerhoff had termed this '*directive correlation*' [25]. In the process of disturbing environmental circumstances, the coenetic variable evokes a response that converges on the adaptive outcome.

The VCS complies with Ashby's Law of Requisite Variety; in order to ensure that each environmental action has an appropriate response.

Infinite recursivity, post-atomic level, is detailed with no explicit starting point or initial conditions. Three levels of recursion are defined: the lowest (atomic) level, named recursion nought, or S^0, with all higher levels to the penultimate infinite recursion, defined as generic, or S^n. The highest level, S^N, exceptionally, is distinct by its lack of an S_2, terminating autopoiesis from this point and spawning of successive recursions. The design grammar model includes syntax representing each of Beer's five main systems, $S_1 \ldots S_5$, plus a further system nought, or S_0.

This dissection of Beer's S1 allows both for a representation of an atomic level within the syntax spectrum of VCS recursions and an interim level dictating the generic states and functionality. Similarly, the highest level is mandatory in order to allow for notional termination of this recursion. Exceptionally, this highest level of recursion, defined as S^N, is distinct by its lack of S_2; terminating autopoiesis and production of further recursions at this point.

Recursion reflects stability in the chain of operations, as constant values maintain their structure or function when operations are repeatedly performed upon them. The identity pursues its indefinite recursive chain, promoting self-stabilisation through homeostasis. As with the generic level, the particular local, future and macro environments, are each incorporated into an identity.

At the atomic point, S_0^0 consists of non-decomposable elements, yet must nonetheless autopoietically spawn higher levels, enabling emergence of a viable system. It is this that drives the bottom-up approach, as opposed to the top-down approach of Beer.

The recursion within the syntax both promotes and exhibits stability in the chain of operations, because constant values, by definition, retain their configuration and efficacy as functions are continually executed upon them.

Where S_1 is equal to S_0 in union with S_2, E_1 will be equal to E_0 in union with E_2. System-environment ecology is vital to sustaining emergence and viability [117] of both design grammar models, the specific environments, being crucial to an identity.

The research has proven VCS concept in both a closed environment [2] and open environment [3, 24] within the context of a previous genetically modified system scenario [69].

Firstly, cybernetic, mathematical and biological metaphors were allied to the human autonomic agent capability of the managerial cybernetics underscoring Beer's VSM. A dual-perspective set theory design grammar model was employed to exhibit relationships between the systems and the recursive levels of the VSM. By incorporating the environment as part of the system, the technique promotes both portability and viability within an initially closed, yet changing, environment. Algorithmic hot swapping was used to provide a repertoire of tailored responses to environmental change within this context. Systemic emergence and viability was thereby promoted, demonstrating proof of the temporal and autonomic properties of the VCS concept.

Striving for uniformity, the open environment VCS case study [3, 24] adopted the same previous genetically modified system scenario [69]. The design grammar model innovated a hybrid VCS architectural representation of the VSM. System One represents a metaphor for homeostasis. The set-theoretical framework defines research specifics, i.e. systems and their environments via algorithmic hot-swapping. Further functions and a set of disturbances are introduced, supplying a potential repertoire of tailored responses to open environmental change.

Fundamental to promoting notional homeostasis and emergent viability is Sommerhoff's concept of directive correlation [25] and Ashby's notion of goal-directedness [22], i.e. the ability to achieve a goal-state under variations in the environment. Example relationships exhibit potential for context-free portability including sets of values of environmental and behavioral variables and a set of outcomes allowing the system to develop an adaptive environmental model of fit responses thereby illustrating temporal and autonomic properties of the VCS concept.

Development of the experience-driven models of the three environmental levels will be continuous, formulating and evaluating systemic capabilities with the lossy data models of scanned environments. It is aspired that S_4 will observe, identify and log systemic hazards and environmental opportunity. In promoting reinforcement learning, this will also allow the VCS to notionally profit from environmental threat and opportunity, resulting in emergence.

As demonstrated in Figure 3.13, a storage facility may be included, to allow containment of data. The multi-level structure potentially including a default hierarchy so that classifiers become more generalized as the top level is scaled. Rules will be reactive to environmental messages, the ideal being minimal rules embracing each permutation within the semi-open environment, suggestive of an aptitude for learning.

4.4 Translation of the VSM into Viable Computing Systems

This research perceives Beers VSM, Figure 3.2, as a blueprint for the VCS framework, primarily as a result of the following attributes:-

The VSM is a five system model with a set of five functions working and similarly recurring at all levels of recursion. The VSM is a triadic alliance of *Management*, *Operation* and the *Environment*. This again is reflected in both the function and form at each and every level of recursion. Cybernetic communication and control manifests between each of the VSM systems and their environments, thus promoting viability and emergence.

The VSM is autopoietic, thereby enabling creation of new S1's when necessary, again promoting emergence and viability. The top-down VSM recursivity facilitates reduction of complexity and allowing systemic properties to be uniformly replicated at each level. Significant to promote systemic emergence and thus viability are the two internal models acting as comparators to the VSM model of the internal systemic capabilities and the perceived world situation.

Managerial cybernetics is unique to the VSM and allows the homeostatic loops to be powered by feedback control. This is assisted by varietal engineering through the properties of attenuation, transduction and amplification. This research required the combination of managerial cybernetics and autonomic computing, in a bid to form the basis of the VCS. In essence, this study attempted to learn from the VSM by trying to replicate key aspects and properties of Beer's model in a set theoretical design grammar for autonomic computing systems.

The set theory method adopted for this research is merely a vehicle to allow manipulation of the VCS architecture. Similarly, formalising it in this way will facilitate any future transposition into software, thus enabling representation of recursivity by way of coding. So in essence this research is attempting to ascertain from the VSM how to mimic homeostasis, by replicating key aspects within a design grammar.

To this end, Beer's approach has been adopted as a blueprint that can be applied in theory, to any system irrespective of its context or nature.

This is due to several VSM attributes that are pertinent and applicable to the VCS goal, namely:

Recursion

Internal models

Autonomy versus governance

Set theory model is a vehicle allowing structural representation

Manipulation

Formalisation

Facilitate the potential transposition into a software demonstrator

4.5 Viable Computing Systems

This research unites autonomic computing principles, with established cybernetic concepts and mathematical set theory. A bi-perspective model is derived from Beer's VSM, resulting in an algebraic design grammar that has been adopted as the core modelling and analysis formalism. The VCS pays homage to the Beerian model by contemporizing its function in line with the requirements of autonomic computing. This is achieved through demonstrating the successful application of a theory of cognition.

Advancing the VSM to a computing context, the operating system undertakes the autonomic role, as recursion is pursued below the level of human autonomic operations; indicative of, and representing the role of those biological agents. The operating system therefore notionally undertakes the autonomic activities of the people within Beer's VSM. The VCS operating system can thus be construed as the next recursive level, possessing a similar, corresponding management structure.

The VCS includes a temporal dimension in the form of three time parameters, which engender a forecasting capability and enable modelling of the environmental situation. The latter also maintains viability in a changing open environment this maintenance of a stable state perpetuating via recognition of environmental changes.

The systems are represented within the syntax through the subscript notation, whereas the recursion levels and any temporal parameters are indicated by the superscripts.

System Four (S_4) is the key complexity sensor, accommodating the context–deviation from human scanning, or inspection of the real world situation present within the VSM. By uniquely linking directly to the open environment, S_4 thereby notionally generates an attenuated decision-model. The temporal syntax models internal capabilities imparting a sense of self and non-self, the VCS S_4 conceptually recognizing and responding to environmental change.

The VCS aims to observe merely the activities of the human agents as executors of this function. Irrelevant human characteristics of those multi-agents will be omitted, this research focusing on determining what the system should sense. This study applied this attribute to enable the context–shift from human scanning, creating potential for the creation of an attenuated decision-model.

The research potential to produce a VCS analogy to the VSM S4 was ascertained, thereby developing a prototypical hybrid system capable of mimicking the timely system plan property of this agent. This was illustrated by replication of the metasystemic location of S_4, in terms of its proximity and powerful position within the $S_{3\text{-}4}$ homeostat. Similarly, it is also balanced by the intervention of S_5, which in turn possesses a plan of the $S_{3\text{-}4}$ homeostat.

Cybernetics itself has evolved since Beer's exploitation of First Principles, with a stronger emphasis being placed on the role of autonomy and the role of the observer in modelling a system. Beerian '*first-order cyberneticists*', will study a system as if it were a passive, objectively given 'thing' that can be freely observed, manipulated, and taken apart.

In researching and understanding this process, the 'cybernetics of cybernetics', or 'meta/second-order' cybernetics movement was encountered, that became active in the early 1970's a movement, spearheaded by von Foerster.

Maruyama and von Foerster [118, 119] have respectively recognized that a second-order cyberneticist working with an organism or system, conversely recognizes that system as an agent in its own right, interacting with another agent - the observer. This commonly became known as second order cybernetics. Recognizing that all knowledge of systems is mediated by a simplified representations, or models, of them, which necessarily ignore those aspects of the system that are irrelevant to the purposes for which the model is constructed.

Resultantly, the properties of the systems themselves must be discerned from those of their models, which depend on any originator system. As Laws has stated:

'…ideally, an optimal regulator will be an isomorphic model of the situation to be controlled...' [23].

He goes on to qualify this by asserting that when this is not feasible, such as in large and very complex systems, then the regulator must contain:

'…a strongly homomorphic model of the situation.' [23].

This makes reference to Lehman's software classification scheme [32] derived from the relationship between the software and the environment in which it is executed. Based on the degree of homomorphism required to obtain a suitable mapping between the problem domain to be addressed and the subsequent software model, the resulting scheme allows software to be categorized into three main classes, namely S, P and E-type software [32].

S-type software engages with those areas where a completely isomorphic mapping between the environment and the resulting software is fully specifiable. The correctness of the software solution obtained is determined exclusively by reference to the specification. The resultant software may not be wholly change free, yet any amendments will generally be restricted to issues of efficiency or correctness.

In the instance where an isomorphic mapping to software cannot be obtained because of physical resource limitations, P-type software results. In order to translate the problem to manageable proportions, approximation and assumption must be applied, necessitating a homomorphic weakening of the mapping between the environment and the software solution. Selection of the particular assumptions and approximations used to achieve such a restricted mapping relies on human judgment and derived solution must reflect that human viewpoint to a degree. This introduces uncertainty in the resulting software by abstraction and assumption.

The correctness of the derived mapping is evaluated by comparing the real environment, with differences then being identified and corrected. Any changes, however, generally reflect a changed perception of the problem domain, not that the problem itself has actually changed.

The last software category, E-type, addresses an application, activity or problem in a real-world domain [32]. The software system is, by its very nature, an incomplete model of the operational domain. Resembling the previous category, the difference lies in the fact that the installed software system becomes part of the world that it models. It thus changes the nature of the problem situation it was developed to address. This incompleteness of the system, leads to the establishment of an intrinsic feedback loop and an inevitable, continuing need for systemic evolution and change.

Lehman asserts that the software system must contain a model of itself and its own operation in the operational domain, the success of initially deploying such a software system in the real world relying upon the validity of the assumptions selected to model that world. For so long as those assumptions hold then the system should operate effectively. Changes in the environment may unexpectedly violate any of those assumptions.

Lehman characterizes this situation via his Uncertainty Principle, namely:

"In the real world, the outcome of software system operation is inherently uncertain with the precise area of uncertainty also not knowable."[32]

Software changes are therefore undertaken to maintain, refine or enhance its' currency as a model of the environment in which it executes.

Conversely, the VCS formalizes the notion of a model that makes valid predictions about the world by applying the notion of homomorphic maps. The VCS research strategy assumed has been to determine the degree of mutability of a hybrid VSM system and/or its parts and agents. The homeostatic loops within Beer's 'black box' cybernetic model, proliferating negative feedback control and autonomy exhibit purposeful behaviour yet this is not strictly influenced by either environmental influences or internal dynamic processes, they are in some senses independent agents with an hypothetical free will. This conduct has been seen as a model behaviour that could be transposed towards the autonomic properties of the VCS.

Laws' et al.'s 2005 state of the art paper explored the possibilities of achieving requisite variety:

"autonomically" [23]

That is, a considered system could enhance its repertoire of available actions by using a genetic algorithm approach. The paper demonstrated the ability of a system to provide tailored responses to environmental change, the product being a model of appropriate, optimized responses, whilst suggesting the latent possibility of applying Holland's associated Learning Classifier Systems (LCS) [120].

This autonomic reduction in variety within a genetic algorithm was a major research impetus. This is reflected in the adoption of a cybernetic stance to amalgamate autonomics, multi-agency and the VSM to engender the VCS.

The case studies [2, 24] demonstrate an exemplar implementation context, whilst manifesting the theory of the VCS specification. Innovation is exhibited by the research deviating from the traditional investigation of the requirements of such a system, by this study proposing a tangible resolution to that objective.

Whilst the VSM is a black box system , these investigations seek to both manifest and exploit its inherent fractal-type recursive geometry [121]. This research depicts the internal processes of each operation, i.e. the relationship between the systems, or subscripts and the recursive levels, or superscripts. The latter makes explicit the potential of this recursivity by replicating feedback control so pivotal to the VSM.

A self-governing context-free system blueprint is designed, pertinent to diverse computing settings. In combining atomic elements, this research transposes Beerian top-down emergence to gain properties specific to meeting the complexity reduction ideals of the industry [6, 8].

Crucially, the VCS provides an analogue to Beer's VSM by demonstrating a theory of how to construct agents to operate according to self-vetoing homeostasis. Each S_1 will determine when its' design conditions have been met, sending a signal to this effect. Should this not be the case, a signal will be sent to request its fellow systems to act accordingly. The metasystem acts as higher level controller to monitor progress, therefore controlling the time it takes for the agents to converge on a solution locally.

Environmental disturbances will be addressed by recourse to a reward and punishment scheme. Initial adaptive attempts may be based on trial and error, those that result in beneficial effects being reinforced by positive feedback, whilst those with detrimental effects being discouraged in a manner similar to that employed by genetic algorithms. This further promotes notional systemic emergence within each S_1, with certain trajectories leading to effective and timely adaptation being reserved for future use, thereby developing a map of effective routes back to viable stability.

The VCS accordingly notionally *learns* to adapt to new or altered circumstances, thus improving its' reaction time to previously occurring disturbances.

Each VCS S_1 possesses a set of local process and knowledge rules. The process rules will hypothetically define how an S_1 will interact with another S_1 under local conditions, in terms of an overall solution strategy.

The rules, however, would allow an S_1 to monitor the time taken so far, computing the remaining time and using this value to change and control then interactive S_1 behaviors. This could possibly be achieved by each S_1 having the capability to apply or adapt one of several solution strategies from a quick with a short solution time, to a fully detailed with a long solution time. Uniting of atomic elements, potentially allows the VCS to adopt germane human agent activities. The aspiration is to automate increasingly complex tasks, facilitating portable systemic self-governance, so addressing the industry's complexity ideals [6, 8]. This research endeavored to subsume human systems by cognitive systems, by electing to utilize and apply the cybernetic and recursive properties of the human-agencied VSM and translate them to a computing context.

The conceptual J-Reference Model was perceived as a stepping-stone from this human framework to a computing environment. The VCS architecture surpasses these by its exposition and application of a mathematical model that not only specifies the fundamental relationships between the recursive levels, but also between the systems that populate those levels. This has uniquely manifested a tangible analogue from which a software system could conceivably be programmed, ergo the first concrete software realization of the autonomic computing ideals.

The design grammar model therefore represents a bi-perspective of the VCS. These relationships can be analogized to production rules, of which there are three, reflecting the bottom, middle and top levels of recursion. The recursion parameter being *N*.

The flexibility of the algebra and the respective detail present within the VCS, essentially dictates that there is more than one means of defining each system.

The VCS reflects the system-environment unity via the design grammar model. This system-environment interplay is crucial to support the model and promote emergence and viability per se.

The utilization of a mathematical analogy is purely a vehicle for expressing research concepts and ideas. In the case of S_1, for example, it can be described as being the union of S_0, with S_2. Taking this further a particular S_1 at the higher, yet generic, recursive level ^{n}, incorporates its respective S_0 at the same position in union with the micro environment pertaining-to that particular S_0, in union with its respective S_2 at that level of recursion, in union with the micro environment of the S_1 in question, in union with the current (or ^{t}, in temporal terms), micro environment of that S_4 associated to a given S_1, unioned with the future (or $^{t+1}$, in temporal terms) micro environment of the S_4 particular to that S_1.

The latter can now be subtracted from the intersection of the current or S_4^t, environment of the generic levels' S_4, in union with that particular S_4's future environment that is specified as S_4^{t+1}.

Within the model, this subtraction of the S_1's current and future environment's intersection from its union will enable systemic emergence, by going some way towards replicating the forecasting capability of Beer's model.

The concatenation could continue by stating that this S_1 is a superset of its' respective S_3 in union with that S_3's environment, in union with its respective systems: S_4 and S_5.

In terms of the highest recursive level, defined as the superscript upper case S^N, this is comprised of a particular management unit, unioned with its particular operation unit, unioned with its particular highest-level environment. These are all unioned with the current, or E^t, macro environment of the highest-level S_4^N in union with that environment's future $E_4^{N(t+1)}$ environment with the intersection of the latter subtracted, so facilitating the emergence capability once again. This in its entirety is defined as a superset of the highest-level $S_{3\text{-}2\text{-}1}^N$ homeostat. This could be further expanded by stating that it is also a superset of the $S_{4—5}^n$ homeostat

This recursive syntax promotes stability in the chain of operations, as constant values retain their configuration and efficacy as functions are continually executed upon them. An identity follows its recursive string, promoting self-stabilization; aspiring a VCS to homeostatically satisfy Horn's self-CHOP benchmark [108]. System-environment dependency sustains emergence and viability [117], exhibited via *S* (system) integrating its' *E* (environment) thus $S \wedge E \rightarrow S \cup E$. This special relationship exhibits the output from one, shaping the input to another. Where S_1 is equal to S_0 in union with S_2, E_1 will equal E_0 in union with E_2. Sommerhoff's coenetic variable principles [25] allow modelling of these, whilst collaboratively fulfilling Ashby's Law of Requisite Variety ; each environmental action having an appropriate response.

The design grammar model is a unique, formal, system-wide, context-free, algebraic set theory representation of Beer's cybernetic VSM. The identities are formulated from dual perspectives: two sets of rules characterizing the relationship between the systems; the subscripts and latterly, novelly, a representation of the relationship between the recursive levels; the superscripts.

The design grammar model includes syntax representing each of Beer's five main systems, S1…S5, plus a further system: Nought, or S_0. Atomic breakdown of Beer's model demands separation of his S1 into two parts, S_2 being isolated from the metasystem, yet remaining juxtaposed to S_3 and enclosed by the boundary of the new VCS S_1. These unchanged management and operation units, sited devoid of that S_2, are re-classified as System Nought (S_0). This new system is inactive as a component, in isolation. When S_0 joins with its associated S_2, this generates an S_1, and thereby the next recursive level. This correspondingly exhibits the VCS capacity for recursion and autopoiesis. The recursion within the syntax both promotes and exhibits stability in the chain of operations, because constant values, by definition, retain their configuration and efficacy as functions are continually executed upon them. An identity will follow its recursive string, promoting self-stabilization, aspirant to developing a VCS satisfying Horn's self-CHOP benchmark [108], via homeostasis.

The identities mirror the system-environment dependency, defining S as a system, whilst the presence of **E** allows integration of the environment of that particular **S**.

System-environment interplay is vital to sustaining emergence and viability [117] of both the sub and superscript design grammar models. Systemic viability depends upon this relationship between the system and its’ open environment, as there must be a suitable mapping between the problem domain to be addressed and the resultant software model.

The specific local, future and global, or macro environments, become vital to an identity. To this end, the research has employed Sommerhoff's coenetic variable principles [25], so enabling modelling of variables to shape the system and its environment, whilst explaining the delimitation of the variety of environmental circumstances, and simultaneously of apparent regulatory responses.

Upcoming research could further evolve the design grammar model holism, enabling a systemic configuration via the deletion and addition of component parts.

Investigations indicate that a future VCS software demonstrator could firstly be executed in a variably open [3, 24] and closed [2] environment. Development of the experience-driven models of the three environmental levels may be continuous, formulating and evaluating systemic capabilities with the lossy data models of scanned environments. S_4 will theoretically observe, identify and log systemic hazards and environmental opportunity. In promoting reinforcement learning, this will also allow the VCS to profit from environmental threat and opportunity, resulting in emergence. As demonstrated in Figure 4.1, a storage facility may be included, to allow containment of data.

The multi-agent system, populated by autonomous learning agents may possibly assume a reward and punishment scheme employing Algedonic regulation accomplished via manipulation of inherent design grammar recursivity and thereby promoting novelty. S_4 will comply with a theory of Ashby’s Law of Requisite Variety , dictating that each environmental action will have an equal and appropriate response and Aulin's Law of Requisite Knowledge [122], dictating that a VCS could know which actions will control perturbations.

The multi-level structure may include a default hierarchy so that classifiers become more generalized as the top level is scaled. Rules will be reactive to environmental messages, the ideal being minimal rules embracing each permutation within the semi-open environment.

The VCS research has interpreted this as suggestive of an aptitude for learning. The completed design grammar model will accordingly automate the design process, generating rules in response to emerging needs.

When applied to the analysis, it will determine legitimacy of the design, whilst appliance to the VCS synthesis will enable fault detection to direct revision.

A research goal is, however, to specify the underlying significance of addition, or union, and any future subtraction, multiplication and division, operators. Discerning concatenations of the elements' structures is, however, context dependant, due to the vast range that could be added or subtracted.

4.6 VSM Topology Post-Application of Design Grammar Model

The design grammar model was applied to the VSM topology, as shown by Figure 4.1. Inherent Beerian characteristics allow a notional context shift from human agency towards an aspiration of VCS software autonomy.

It is shown how the wider, macro environment of the system-in-focus, is an important element of the design grammar and core to the VCS functionality is that $S_4^{[N,n,0]}$ communicates directly with the future, temporal environment. One can also see how embedded, micro environments at the lower recursion are also crucial to the design grammar. The presence of the environment within the design grammar underscores the research credo that it should be viewed as part of the system in order to retain viability and consequently maintain a stable state in a changing environment. Endogenous complexity will therefore reduce via inclusion of a temporal element, denoting the design grammar's aspired forecasting capability.

The self/non-self dichotomy principle [115] will be realized by the retention of the comparator model of the internal capabilities of the system, along with a model of the future environment. These both assist the system to remain viable by possessing a notional sense of self, whilst promoting emergence via response to the requirements of a changing environment.

Rather than deriving an isomorphic [67], i.e. complexity-proliferating mapping of the environment and internal systemic capabilities [67], the lossy data compression approach [123] can be assumed thereby reflecting an homomorphic depiction [67] which will lessen redundancy. Such models are therefore incorporated as representations, within S_4 and S_5 enabling the system to distinguish it's *self* from the environment in which it has been implemented.

In terms of the Beerian S_1's, Generic level System Nought; that is S_0^n or subscripted system nought to the superscripted lower-case n, has now been created within the design grammar. This is achieved by the removal of generic level S_2^n from Beer's generic level S_1^n. The interim or generic level, systems S_n, are demonstrated here by the new S_1^n, a fusion of S_0^n and S_2^{n+1}. This dissection of the Beerian S_1 results in the removal of its respective $S_2^{[n,0]}$ from the metasystem, yet as shown, still juxtaposed to $S_3^{[n,0]}$. For design grammar purposes, the highest level S_2^N has necessarily been removed from the systemic whole at the S_0^N position, therefore there is no S_2^N i.e. highest-level S_2. Failure to omit this would result in a spawning of a higher level system, thereby demoting this level to highest minus 1 and so on and so forth.

Figure 4.1 also reflects the three recursive levels from the design grammar model. It's important to note that the VSM's open-ended, yet predominantly top-down, recursive structure, has been transposed to bottom-up. This is evidently visible by the lack of a higher level operation unit; post the top-level recursive structure of upper case N. This results in a new atomic or lowest level of recursion i.e. a system nought at recursion level nought that cannot be further decomposed. With the S_2, situated at recursion nought: S_2^0 excluded, this is now reclassified as S_0^0.

As mentioned earlier, the VCS S_{3*} at each level is now incorporated into its respective S_3, due to the fact that intermittent auditing is obsolete when undertaken by software agents.

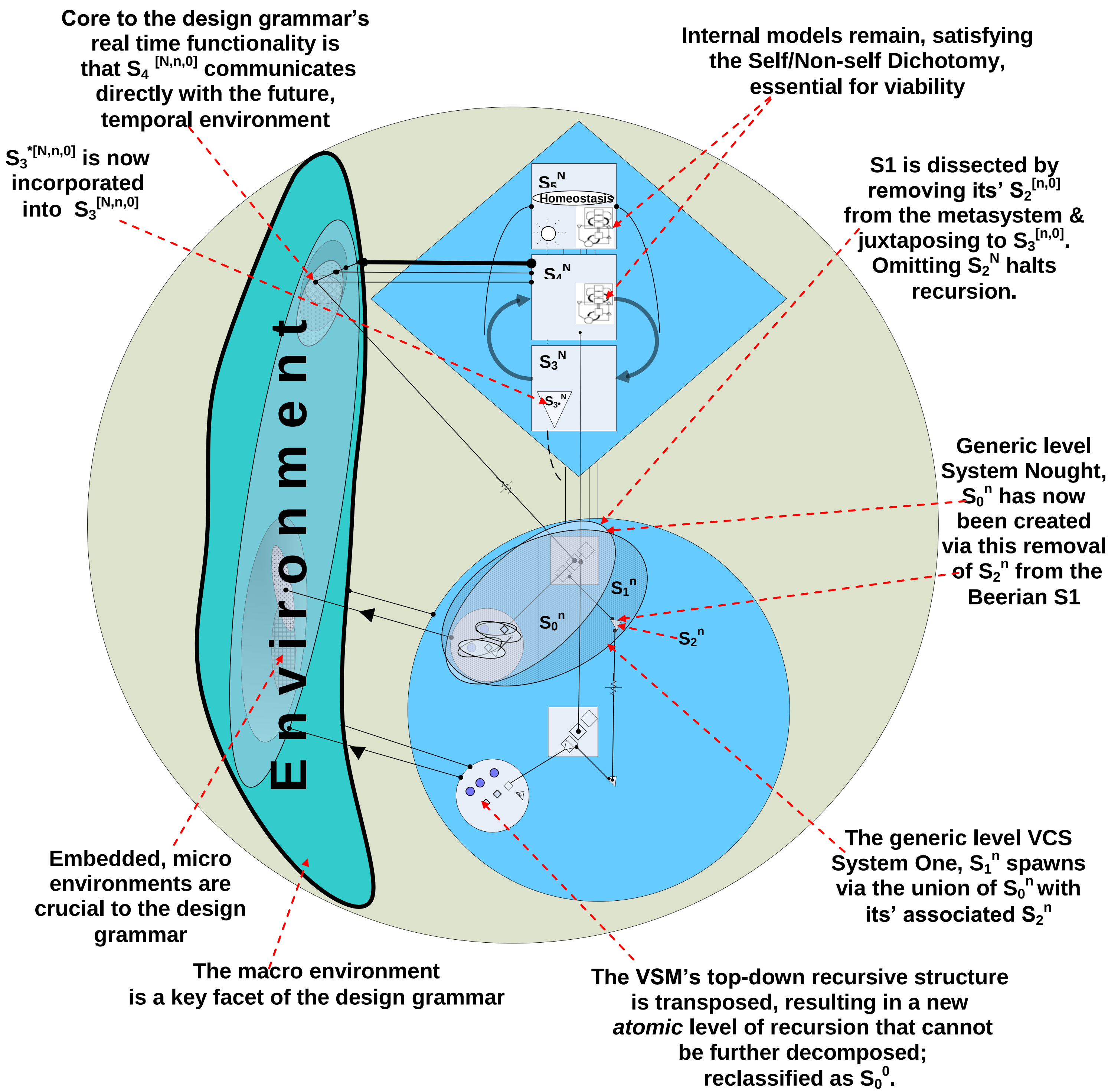


Figure 4.1: VSM Topology Post-Application of Design Grammar Model

4.7 Chapter Summary

The core basis of this research is the manifestation of a bi-perspective set-theory oriented analogue based upon an integrated management method that promotes each sub-system as a whole. This encompasses the development of a first-stage, functional, decomposition of the VSM from the viewpoint of modelling the relationships between the recursive levels and between the systems.

A temporal facet promotes autonomy versus governance by enabling the VCS to model the *future* environment situation, a compartor to its' self model of the *present* internal capabilities, in turn constructed from its' *past* experience-driven modelling.

The context-shift from human to computer agency, has necessitated that the VCS design grammar deviate from the Beerian stance in several respects; Atomic recursion engendered a novel system nought, a dissection of the VSM system one avhieved via removal of the S2. Re-introduction of the VCS S_2 triggers recusion via autopoietically creating a S_1 at the successive level. Rather than adhering to the VSM stipulation that there should be no greater quantity than 7 S1's per recursive level, these systems representing the human agents and/or their self-governing internal biological components or organs, the VCS context-deviation to software agents allows for the definition of an infinite range.

A future research challenge is defining the semantics of addition, or *union,* and any future *subtraction, multiplication* and *division,* operators. Understanding what lies behind the concatenation of compositions of VCS elements would be context dependant, due to the range of different elements able to be added/subtracted.

Chapter 5

Design Grammar Model Identities Syntax

Introduction

The design grammar model is a unique, formal, system-wide, context-free, algebraic set theory representation of Beer's VSM. The identities are formulated from dual perspectives: two sets of rules characterize the relationship between the systems, via subscripts and the recursive levels via superscripts. A mathematical analogy acts as a medium, articulating the research concepts. It features production rules, a symbol set, and vocabulary including atomic elements of the language.

5.1 Overview

Whreas the VSM's infinite recursivity exhibits with no explicit starting point or initial conditions, three recursive levels were defined within the VCS: the lowest, atomic level is recursion *nought*, or $S_{0,}$ higher levels to the penultimate being *generic*, or S^n. The *highest* level S^N exceptionally omits $S_{2,}$ terminating the spawning of successive recursions.

The syntax denotes the five main systems, $S_1 \ldots S_5$, plus the novel system nought, S_0. Atomic breakdown dissects the VSM S_1, isolating S_2 from the metasystem yet still juxtaposed to the S_3, now enclosed by new S_1 boundary. This then becomes S_0, an inert component until joining with S_2 to create S_0 at atomic level, (S_0^0) possesses non decomposable constituents that are conceptually able to autopoietically generate higher levels. The recursion within the syntax both promotes and exhibits stability in the chain of operations, because constant values, by definition, retain their configuration and efficacy as functions are continually executed upon them. An identity will follow its recursive string, promoting self-stabilization, aspirant towards a VCS satisfying Horn's self-CHOP benchmark [108], via homeostasis.

Where S_1 is equal to S_0 in union with S_2, E_1 will be equal to E_0 in union with E_2. System-environment ecology is vital to sustaining emergence and viability [117] of both design grammar models, the specific environments, being crucial to an identity.

5.1.1 Identities Syntax Examples of the Design Grammar Model of Subscript Relationships between the Systems:

<u>Notations:</u>

S : a system$\{_{0,1,2,3,4,5}\}$

S : a system recursion$\{^{0,n,N}\}$

i : a system identifier

A : a system set $\{a_1,a_2,a_3,a_4,\ldots_i\} \mid a_i$ *is atomic*

M : a system Management Unit

O : a system Operation Unit

E : a system Environment

t : CurrentTime parameter$|t^{+1}$*:FutureTime*$|t^{-1}$*: PastTime*

Binary Relations:

$A \Rightarrow B$: If A is true, B is also true

Binary Operators:

$A \cup B$: The set containing all of those elements within A and B

$A \cap B$: The set containing those common elements of A and B

$A - B$: The set that results when B is subtracted from A

$A \not\subset B$: The set A is not a subset of set B

There follow key examples from the subscript design grammar model, of the relationships between the VCS systems, exemplified by the equations 5.1 to 5.50.

The subscripts represent the system numbers i.e. $S_{0\ldots}S_5$, the superscripts S^0,S^n,S^N represent the recursive levels and, where relevant, S^t, S^{t+1}, S^{t-1} the temporal parameters.

System Nought (S_0): *Atomic Catalyst*:

$$S^0_{0,i\leq\infty} \Rightarrow A$$

(5.1)

This equation 5.1, is significant, in that it iterates how the atomic level system nought is the basis of the bottom-up VCS recursion method. At this level, the *atomic catalyst* maps to the atomic set and therefore has no identifiable constituent parts..

The design grammar reflects how atomic decomposition of Beer's model incorporates dissection of the conventional S_1 into two distinct parts. This engenders the novel System Nought (S_0), or *Atomic Catalyst*. Enabling atomic, that is bottom-up, recursion from atomic component parts as in equation 5.1.

S_0 is created when S_2, is removed from the S_1 management unit, or metasystem and re-positioned still adjacent to S_3. This facilitates enclosure by the new S_1's, boundary, this union now becoming the trigger to generate a VCS S_1, and therefore the next recursive level.

$$S_{0,i\leq\infty}^{n} \Rightarrow \begin{pmatrix} M_{0,i\leq\infty}^{n} \cup O_{0,i\leq\infty}^{n} \cup E_{0,i\leq\infty}^{n} \cup \\ \left(E_{4,i\leq\infty}^{n(t)} \cup E_{4,i\leq\infty}^{n(t+1)} \right) - \left(E_{4,i\leq\infty}^{n(t)} \cap E_{4,i\leq\infty}^{n(t+1)} \right) \end{pmatrix}$$

(5.2)

The above equation 5.2, represents system nought at the generic point, presenting the componenets as a management unit, operation and relevant environment. These are unioned with the VCS forescasting function provoked by the intersection of the pertinent S_4's respective links to the *present* and *future* environment. This technique promotes systemic viability and emergence by enabling the creation of a model of the required world situation, a comparator to the model of the internal capabilities.

$$S_{0,i\leq\infty}^{N} \Rightarrow \begin{pmatrix} M_{0,i\leq\infty}^{N} \cup O_{0,i\leq\infty}^{N} \cup E_{0,i\leq\infty}^{N} \cup \\ \left(E_{4,i\leq\infty}^{N(t)} \cup E_{4,i\leq\infty}^{N(t+1)} \right) - \left(E_{4,i\leq\infty}^{N(t)} \cap E_{4,i\leq\infty}^{N(t+1)} \right) \end{pmatrix}$$

(5.3)

Equations 5.2 and 5.3 show similarly, the same management and operation units without that S_2 are re-classified as system nought, S_0. Isolating S_2 from the metasystem enables creation of S_0, an inert component, until joined with its associate S_2, thereby autonomically spawning the next recursive-level S_1.

Equations 5.2 and 5.3 show S_0 as being comprised of the management and operation unit, the latter being the focus of recursivity within the whole. The latter said equations additionally show the VCS theory of temporal forecasting, via the subtraction of the intersection of the particular, current S_4 environment with its respective future environment.

System One (S1): Implementation:

$$S^{0}_{1,i\leq\infty} \Rightarrow A \qquad \textbf{(5.4)}$$

This directional system is a recursive, autonomous homeostat. Recursion is exhibited within equations 5.4, 5.5 and 5.6 by the atomic recursion nought: S^0, generic level represented by S^n, and the top level S^N. What the system *does* is performed by S_1, clearly shown within equations 5.4, 5.5 and 5.6, to interact directly with the environment.

$$S^{n}_{1,i\leq\infty} \Rightarrow \left(\left(\begin{array}{l} \left\{ S^{n}_{0,i\leq\infty} \cup E^{n}_{0,i\leq\infty} \cup S^{n}_{2,i\leq\infty} \cup E^{n}_{2,i\leq\infty} \cup E^{n}_{1,i\leq\infty} \right\} \cup \\ \left\{ \left\{ E^{N(t)}_{4,i\leq\infty} \cup E^{N(t+1)}_{4,i\leq\infty} \right\} - \left\{ E^{N(t)}_{4,i\leq\infty} \cap E^{N(t+1)}_{4,i\leq\infty} \right\} \right\} \end{array} \right) \supset \atop \left\{ S^{n}_{3,i\leq\infty} \wedge E^{n}_{3,i\leq\infty} \wedge S^{n}_{2,i\leq\infty} \wedge E^{n}_{2,i\leq\infty} \wedge S^{n}_{1,i\leq\infty} \wedge E^{n}_{1,i\leq\infty} \right\} \right)$$

(5.5)

Equation 5.5 illustrates how, at the generic recursive level a VCS S_1 is created when an S_0 is unioned with its associated S_2. This allows for the autopoietic spawning of the next recursion and has been adopted as a method to promote atomic recursion

$$
\begin{gathered}
S^{N}_{1,i\leq\infty} \Rightarrow \\
\left(\left(\begin{array}{l} M^{N}_{1,i\leq\infty} \cup O^{N}_{1,i\leq\infty} \cup E^{N}_{1,i\leq\infty} \cup \\ \left(\left(E^{N(t)}_{4,i\leq\infty} \cup E^{N(t+1)}_{4,i\leq\infty} \right) - \left(E^{N(t)}_{4,i\leq\infty} \cap E^{N(t+1)}_{4,I\leq\infty} \right) \right) \end{array} \right) \supset \right. \\
\left. \left(S^{N}_{3,i\leq\infty} \wedge S^{N}_{1,i\leq\infty} \wedge \left(S^{N}_{2,i\leq\infty} \Rightarrow \phi \right) \right) \right)
\end{gathered}
$$

(5.6)

Comprised of a management and operation unit in conjunction with the respective environment, both equation 5.5 and 5.6 articulate how the S1 is nested within a higher parent S_1. Autopoiesis is illustrated via the ability of S_1 to self-produce lower-level recursions. These are shown to be to an indeterminate infinite number that terminate at the top parameter defined within equation 5.6 as level S^N.

Equation 5.6 is notable via the omission of the aforesaid S_2, dictating that recursion is terminated. It is demonstrated via the empty set [124] notation indicating the absence of an S_2 within S_0.

System Two (S2): ***co-ordination:***

$$
S^{0}_{2,i\leq\infty} \Rightarrow A
$$

(5.7)

Equation 5.7 reflects how atomic level S_2^0 maps to the empty set, as it is comprised of components that are only able to autopoietically spawn a successive recursion when unioned with either an atomic or generic S_0.

$$
\begin{array}{l}
S^{n}_{2,i\le\infty} \subseteq \\
\left(\begin{array}{l}
\left(S^{n}_{2,i\le\infty} \cup E^{n}_{2,i\le\infty}\right) \cup \left(\left(E^{N(t)}_{4,i\le\infty} \cup E^{N(t+1)}_{4,i\le\infty}\right) - \left(E^{N(t)}_{4,i\le\infty} \cap E^{N(t+1)}_{4,i\le\infty}\right)\right) \supset \\
\left(\begin{array}{l}
\left(\left(S^{0}_{0,i\le\infty} \subset \left(S^{0}_{0,i\le\infty} \cup S^{0}_{2,i\le\infty}\right) \subset S^{0}_{1,i\le\infty}\right) \bigcup\limits_{1\le k\le\infty} s^{n+1}_{k,i\le\infty}\right) \\
\mathrm{For\ N \ge n > 0}
\end{array}\right) \subset \left(S^{n}_{1,i\le\infty}\right)
\end{array}\right)
\end{array}
$$

(5.8)

As a co-coordinating, anti-oscillatory, local-regulatory element of S1, the syntax within equation 5.8, demonstrates how S_2 unites S_1 with its’ S_0 antecedent, whilst also demonstrating its interaction with the environment.

$$
S^{n}_{2,i\le\infty} \subset S^{N-1}_{2,i\le\infty}
$$

(5.9)

Equation 5.9 reflects how all recurisons post-atomic level, contain an entire VCS; each of the five VCS functions thus working and recurring at each of these levels. This fractal-like recursive geometry reduces systemic redundancy and so complexity .

$$
S^{N}_{2,i\le\infty} \not\subset S^{n}_{2,i\le\infty}
$$

(5.10)

Equation 5.10 reflects how there is no top-level VCS S_2, as the latter acts as both the trigger and terminator of recursion.

$$S_{2,i\leq\infty}^{N} \rightarrow \phi$$

(5.11)

Equation 5.11 therby illustrates how there is thus no requirement for a top recursive level S_2^N, as it would be obsolete at this recursive level. It thus maps to the empty set at equation 5.11.

System Three (S_3): Control:

$$S_{3,i\leq\infty}^{0} \Rightarrow A$$

(5.12)

As the controlling facility within the model, S_3 regulates, optimizes and stabilizes internal activity. Equation 5.12 illustrates how the atomic level maps to the empty set.

$$S_{3,i\leq\infty}^{n} \Rightarrow \left(\left\{ E_{3^*,i\leq\infty}^{n} \cup E_{3,i\leq\infty}^{n} \cup E_{3^*,i\leq\infty}^{n} \right\} \cup \left(\begin{matrix} \left\{ E_{4,i\leq\infty}^{n^{(t)}} \cup E_{4,i\leq\infty}^{n^{(t+1)}} \right\} \\ \left\{ E_{4,i\leq\infty}^{n^{(t)}} \cap E_{4,i\leq\infty}^{n^{(t+1)}} \right\} \end{matrix} \right) \right) \subset \left\{ S_{3,i\leq\infty}^{n} \wedge S_{2,i\leq\infty}^{n} \wedge S_{1,i\leq\infty}^{n} \right\}$$

(5.13)

S_3 is a vital fulcrum that is assisted by $S_{2,}$ as exhibited within equations 5.13 and 5.14. It can be seen how S_3^* has been translated as a constant VCS auditor within the new, computing, context.

$$S^{N}_{3,i\leq\infty} \Rightarrow \left(\left\{ S^{N}_{3^*,j\leq\infty} \cup E^{N}_{3,i\leq\infty} \cup E^{N}_{3^*,j\leq\infty} \right\} \cup \left\{ E^{N^{(t)}}_{4,i\leq\infty} \cup E^{N^{(t+1)}}_{4,i\leq\infty} \right\} - \left\{ E^{N^{(t)}}_{4,i\leq\infty} \cap E^{N^{(t+1)}}_{4,i\leq\infty} \right\} \subset \left\{ S^{N}_{3,i\leq\infty} \wedge S^{N}_{2,i\leq\infty} \wedge S^{N}_{1,i\leq\infty} \right\} \right)$$

(5.14)

It is the case that S_3 provides overall structure, integrating cohesive activities of the S_1's, the syntax within the latter equations reflecting how S_3 is a subset of the influential $S_{3\text{-}2\text{-}1}$ homeostat.

System Three Star (S_{3*}): ***Intermittent Audit:***

$$S^{0}_{3^*,j\leq\infty} \Rightarrow A$$

(5.15)

System Three Star (S_{3*}) is a sporadic auditing system, assimilated into $S_3^{[N,n,0]}$ within both the subscript and superscript design grammar VCS models, thereby facilitating holistic control. It thus acts as a backup inspection facility to both the validity and functionality of S_1 and S_3 respectively. Equation 5.15 shows atomic level S_{3*}^{0} as mapping to the empty set.

$$
S^{n}_{3*,i\le\infty} \Rightarrow
\left(\left(S^{n}_{3*,i\le\infty} \cup E^{n}_{3,i\le\infty} \cup E^{n}_{3*,i\le\infty}\right) \left(\left(E^{n^{(t)}}_{4,i\le\infty} \cup E^{n^{(t+1)}}_{4,i\le\infty}\right) \left(E^{n^{(t)}}_{4,i\le\infty} \cap E^{n^{(t+1)}}_{4,i\le\infty}\right)\right) \subset \left(S^{n}_{3,i\le\infty} \wedge S^{n}_{2,i\le\infty} \wedge S^{n}_{1,i\le\infty}\right) \right)
$$

(5.16)

Equation 5.16 shows how advancing the VSM into a VCS emphasizes the intermittency and thus redundancy, of this element in a computing context. It has the capability to monitor in a constant state in this situation.

$$
S^{N}_{3*,i\le\infty} \Rightarrow
\left(\left(S^{N}_{3*,i\le\infty} \cup E^{N}_{3,i\le\infty} \cup E^{N}_{3*,i\le\infty} \cup \left(\begin{matrix} \left(E^{N^{(t)}}_{4,i\le\infty} \cup E^{N^{(t+1)}}_{4,i\le\infty}\right) \\ \left(E^{N^{(t)}}_{4,i\le\infty} \cap E^{N^{(t+1)}}_{4,i\le\infty}\right) \end{matrix} \right) \right) \subset \left(S^{N}_{3,i\le\infty} \wedge S^{N}_{2,i\le\infty} \wedge S^{N}_{1,i\le\infty}\right) \right)
$$

(5.17)

Equations 5.16 and 5.17 attempt to exemplify how S_{3*} may potentially be effected as a constituent. This is illustrated by the generic and top-level S_{3*}^{n} as a subset of the $S_{3\text{-}2\text{-}1}^{n}$ homeostat.

System Three-Two-One ($S_{3\text{-}2\text{-}1}$) *Metasystem Homeostat*:

$$\left(S^{0}_{3,i\leq\infty} \wedge S^{0}_{2,i\leq\infty} \wedge S^{0}_{1,i\leq\infty} \right) \Rightarrow A$$

(5.18a)

Equation 5.18 and 5.19a show how the $S_{3\text{-}2\text{-}1}$ metasystem homeostat is bonded via the ^ operator, aiming to reflect that the output of each of these systems deteremines the imput to another. This interdependence is important to maintain equilibrium within the VCS metasystem, and respective holism. This trilogy of systems, here situated at the atomic level, maps to the atomic set.

$$\left(S^{n}_{3,i\leq\infty} \wedge S^{n}_{2,i\leq\infty} \wedge S^{n}_{1,i\leq\infty} \right) \Rightarrow \left(\begin{pmatrix} S^{n}_{3^*,i\leq\infty} \cup E^{n}_{3^*,i\leq\infty} \cup S^{n}_{2,i\leq\infty} \cup \\ E^{n}_{2,i\leq\infty} \cup S^{n}_{1,i\leq\infty} \cup E^{n}_{1,i\leq\infty} \end{pmatrix} \cup \begin{pmatrix} \left(E^{N(t)}_{4,i\leq\infty} \cup E^{N(t+1)}_{4,i\leq\infty} \right) - \\ \left(E^{N(t)}_{4,i\leq\infty} \cap E^{N(t+1)}_{4,i\leq\infty} \right) \end{pmatrix} \right) \in$$

(5.19a)

The $S_{3\text{-}2\text{-}1}$ composite stabilizes the inner milieu of the system via cross-recursion co-ordination , reflected within equations: 5.18a to 5.20 inclusively by the appearance of the three recursive parameters of $S^{0,\ n,}$ and S^{N}, with systems $S_{3\text{-}2\text{-}1}$ being identified with the ancient brain or system. This trio recursively control the inner system by direction and co-ordination, equations

Equation 5.19a shows the $S_{3\text{-}2\text{-}1}$ metasystem homeostat at the generic level of recursion, and the VCS incorporation of Beer's human-agencied, intermittent auditing, S_3^{*} system into the advanced, constant auditing S_3.

5.19a to 5.20 show the constituent elements of the generic level. These include the definition of generic level S_3* and its particular environment as being a subset of the $S_{3\text{-}2\text{-}1}{}^{n}$ homeostat, as represented in equation 5.19a. Each of these equations reflects the temporal forecasting capability. This manifests as the conceptual subtraction of each of the particular *current* and *future* S_4 environments, from their respective intersections.

$$\left\{S^{n}_{2,i\leq\infty} \cup E^{n}_{2,i\leq\infty}\right\} \cup \left(\begin{array}{c} \left\{E^{N(t)}_{4,i\leq\infty} \cup E^{N(t+1)}_{4,i\leq\infty}\right\} - \\ \left\{E^{N(t)}_{4,i\leq\infty} \cap E^{N(t+1)}_{4,i\leq\infty}\right\} \end{array} \right) \in$$

(5.19b)

The equation part 5.19b ensues from equation co-part 5.19a; illustrating how the former is an element of the generic S_2 and its respective environment. The subtraction of the top-level current and future environmental levels from its' intersection, conceptualizes the VCS forecasting capability. Equation 5.19b comprises element parts of the successive equation 5.19c.

$$\left\{S^{n}_{0,i\leq\infty} \cup E^{n}_{0,i\leq\infty} \cup S^{n}_{2,i\leq\infty} \cup E^{n}_{2,i\leq\infty} \cup E^{n}_{1,i\leq\infty}\right\} \cup \left(\begin{array}{c} \left\{E^{N(t)}_{4,i\leq\infty} \cup E^{N(t+1)}_{4,i\leq\infty}\right\} - \\ \left\{E^{N(t)}_{4,i\leq\infty} \cap E^{N(t+1)}_{4,i\leq\infty}\right\} \end{array} \right) \in$$

(5.19c)

This equation 5.19c depicts how the preceding recursion contains a generic system nought, or atomic catalyst as it is termed, unioned with its' corresponding environment, the S_1 and recursion trigger of S_2 that will be located at the same recursion and associated range position and its' particualr environment. The forecasting notion is again shown, as explicated in equation 5.19b and the assemblage classified as set parts of the ensuing equation 5.19d.

$$\left(\left(M^{N}_{1,i\le\infty} \cup O^{N}_{1,i\le\infty} \cup E^{N}_{1,i\le\infty} \right) \cup \left(\begin{array}{c} \left(E^{N(t)}_{4,i\le\infty} \cup E^{N(t+1)}_{4,i\le\infty} \right) - \\ \left(E^{N(t)}_{4,i\le\infty} \cap E^{N(t+1)}_{4,i\le\infty} \right) \end{array} \right) \right)$$

(5.19d)

The equation 5.19d elucidates the elements of equation 5.19c as part of this top level $S_{3\text{-}2\text{-}1}$ metasystem homeostat. The uppermost management unit, operation and germane environment, are shown to union with the current and future S_4^N environment, being subtracted from their intersection. This provokes VCS forecasting and emergence so endorsing viability.

$$S^{N}_{3,i\le\infty} \wedge S^{N}_{2,i\le\infty} \wedge S^{N}_{1,i\le\infty} \Rightarrow$$
$$\left(\begin{array}{l} \left(S^{N}_{3^*,i\le\infty} \cup E^{N}_{3,i\le\infty} \right) \cup \left(\begin{array}{c} \left(E^{N(t)}_{4,i\le\infty} \cup E^{N(t+1)}_{4,i\le\infty} \right) - \\ \left(E^{N(t)}_{4,i\le\infty} \cap E^{N(t+1)}_{4,i\le\infty} \right) \end{array} \right) \subset \\ \left(S^{N}_{3,i\le\infty} \cup S^{N}_{1,i\le\infty} \right) \end{array} \right)$$

(5.20)

Equation 5.20 illustrates how top-level $S_{3\text{-}2\text{-}1}^N$ homeostat omits an S_2 to terminate spawning of a successive recursive level. An example of this is the definition of the top level S_3^N, unioned with its S_1^N, as a subset of the constituents.

System Four (S_4): ***Intelligence:***

$$S^{0}_{4,i\leq\infty} \Rightarrow A$$

(5.21)

System Four at the lowest recursive level maps to the atomic set, within the range of 1 to infinity.

$$S^{n}_{4,i\leq\infty} \supset \left(\left\{ E^{N^{(t)}}_{4,i\leq\infty} \cup E^{N^{(t+1)}}_{4,i\leq\infty} \right\} - \left\{ E^{N^{(t)}}_{4,i\leq\infty} \cap E^{N^{(t+1)}}_{4,i\leq\infty} \right\} \subsetneq \left\{ S^{N}_{3,i\leq\infty} \wedge S^{N}_{4,i\leq\infty} \wedge S^{N}_{5,i\leq\infty} \right\} \right)$$

(5.22a)

System Four or S_4 is the key complexity sensor, accommodating the context–deviation from human scanning by uniquely linking directly to the open environment and thereby generating an attenuated decision-model.]

Equation 5.22a reflects how the constituents of generic level S_4 are a subset of the topmost $S_{3\text{-}4\text{-}5}$ metasystem homeostat. The special relation and mutual dependence between this trio is indicated by the ^ operator signifying how their individual outputs define the inputs of their fellow systems.

$$S^{n}_{4,i\leq\infty} \subset S^{N-1}_{4,i\leq\infty}$$

(5.22b)

Equation 5.22b reflects how generic level S_4 is a subset of the topmost *past* S_4., so emphasizing the VCS temporal facet, this system being generated based upon the decision model that compares both the *present* and *future* forecasting VCS models.

$$S_{4,i\leq\infty}^{N^{(t)}} \supset \left(M_{4,i\leq\infty}^{N(t)} \cup O_{4,i\leq\infty}^{N(t)} \cup E_{4,i\leq\infty}^{N(t)} \cup \left(\left(\begin{Bmatrix} E_{4,i\leq\infty}^{N^{(t)}} \cup E_{4,i\leq\infty}^{N^{(t+1)}} \\ E_{4,i\leq\infty}^{N^{(t)}} \cap E_{4,i\leq\infty}^{N^{(t)}} \end{Bmatrix} \right) \supset S_{3,i\leq\infty}^{N^{(t)}} \wedge S_{5,i\leq\infty}^{N^{(t)}} \right) \right) \in$$

(5.23a)

It is shown via equation part 5.23a, how S_4 is the only system with direct connection to all of the wider environments, each recursive S_4 links directly to both its’ parent and subordinate counterparts and thus promotes inter-recursive cohesion. These elements relate to the equation part 5.23b.

$$S_{4,i\leq\infty}^{N^{(t+1)}} \Rightarrow \left(M_{4,i\leq\infty}^{N^{(t+1)}} \cup O_{4,i\leq\infty}^{N^{(t+1)}} \cup E_{4,i\leq\infty}^{N^{(t+1)}} \cup \left(\begin{Bmatrix} E_{4,i\leq\infty}^{N^{(t)}} \cup E_{4,i\leq\infty}^{N^{(t+1)}} \\ E_{4,i\leq\infty}^{N^{(t)}} \cap E_{4,i\leq\infty}^{N^{(t+1)}} \end{Bmatrix} \right) \right) \supset \left\{ S_{3,i\leq\infty}^{N^{(t+1)}} \wedge S_{5,i\leq\infty}^{N^{(t+1)}} \right\} \in$$

(5.23b)

Equation 5.23b shows how the top-level S_4 adopts a temporal capability to facilitate the VCS forecasting attribute. By uniquely linking to the *future* $^{(t+1)}$ environment, the system engenders a capability to predict and acclimatize-to the required world situation. These are elements of subsequent equation part 5.23c.

$$\left(S_{4,i\leq\infty}^{N^{(t+1)}} \supset S_{4,i\leq\infty}^{N^{(t)}} \right)$$

(5.23c)

Equation 5.23c shows how the *future* linking top-level S_4 is a superset of its' *present* counterpart, so again exhibiting the temporal facet to the VCS that allows viability and emergence.

$$\left(S_{4,i\leq\infty}^{N^{(t)}} \subset S_{4,i\leq\infty}^{N^{(t+1)}} \right)$$

(5.23d)

Equation 5.23d illustrates how the *present* linking top-level S_4 is a subset of the *future* counterpart, the latter requirements being compared to the systemic capabilities. The temporal syntax exemplifies how the VCS models internal configuration to notionally impart a sense of self and non-self. The VCS S_4 thus conceptually recognizes and responds to environmental change, aiming to observe merely the activities of the human agents as executors of this function. Irrelevant human characteristics are omitted. this research focuses on determining *what* the system should sense, applying this attribute to enable the context–shift from human scanning, generating potential for the creation of an attenuated decision-model.

System Three-Four ($S_{3\text{-}4}$): *Metasystem Homeostat*:

$$\left(S_{3,i\leq\infty}^{0} \wedge S_{4,i\leq\infty}^{0} \right) \Rightarrow A \qquad \textbf{(5.24)}$$

Equation 5.24 shows how this duo maps to the atomic set. Similarly, it exhibits the special relation between the atomic systems within this homeostat, in that they are mutually dependant and their respecitive outputs dicatate their subsequent inputs.

$$
\begin{aligned}
&\left\{ S^{n}_{3,i\le\infty} \wedge S^{n}_{4,i\le\infty} \right\} \Rightarrow \\
&\left(\left(\begin{matrix} S^{n}_{3,i\le\infty} \cup S^{n}_{3^*,j\le\infty} \cup E^{n}_{3,i\le\infty} \cup \\ E^{n}_{3^*,j\le\infty} \cup S^{n}_{4,i\le\infty} \cup E^{n}_{4,i\le\infty} \end{matrix} \right) \cup \left(\begin{matrix} \left\{ E^{N(t)}_{4,i\le\infty} \cup E^{N(t+1)}_{4,i\le\infty} \right\} \\ - \left\{ E^{N(t)}_{4,i\le\infty} \cap E^{N(t+1)}_{4,i\le\infty} \right\} \end{matrix} \right) \right) \subset \\
&\left\{ S^{n}_{3,i\le\infty} \wedge S^{n}_{2,i\le\infty} \wedge S^{n}_{1,i\le\infty} \right\}
\end{aligned}
$$

(5.25)

The $S_{3\text{-}4}$ metasystem homeostat is one of the most critical homeostatic forces within the system, containing models of both the extra-systemic environment and the internal systemic capabilities. This is reflected within the syntax by appliance and manipulation of temporal elements within equations 5.25 to 5.26b inclusive.

The VCS forescasting capability is expressed by subtraction of the top-level S_4 environments from their intersection. The identity 5.25 exhibits the special relationship within the generic $S_{3\text{-}4}$ homeostat, as identified at the atomic level. Identical properties appear in the $S_{3\text{-}4}$ subsystem of the $S_{3\text{-}2\text{-}1}$ metasystem homeostat. The generic level $S_{3\text{-}4}$ homeostat comprises S_3, its' marsupial-like auditor of S_3^*, the pertinent S_4 and respective environments.

$$
\begin{aligned}
&\left\{ S^{N}_{3,i\le\infty} \wedge S^{N}_{4,i\le\infty} \right\} \Rightarrow \\
&\left(\begin{matrix} \left(\begin{matrix} S^{N}_{3^*,j\le\infty} \cup E^{N}_{3^*,j\le\infty} \cup S^{N}_{3,i\le\infty} \cup E^{N}_{3,i\le\infty} \cup S^{N}_{4,i\le\infty} \cup E^{N}_{4,i\le\infty} \\ \left\{ E^{N(t)}_{4,i\le\infty} \cup E^{N(t+1)}_{4,i\le\infty} \right\} - \left\{ E^{N(t)}_{4,i\le\infty} \cap E^{N(t+1)}_{4,i\le\infty} \right\} \end{matrix} \right) \\ \subset \left\{ S^{N}_{3,i\le\infty} \wedge S^{N}_{2,i\le\infty} \wedge S^{N}_{1,i\le\infty} \right\} \end{matrix} \right)
\end{aligned}
$$

(5.26)

The identity 5.26 exhibits the special relationship of the top level $S_{3\text{-}4}$ homeostat; mutual depenndancey and each systemic output impacting upon the input of its' fellow homeostat system. this relationship is mirrored by its' subordinate of the top-level $S_{3\text{-}2\text{-}1}$ metasystem homeostat The highest level VCS $S_{3\text{-}4}$ homeostat incorporates not only the uppermost systems S_3 and S_4, but also the incorporated S_3* and respectfive environments. The temporal, forecasting capability engendered by top-level E_4's, is shown akin to equation 5.25.

$$S_{4,i\leq\infty}^{N^{(t)}} \supset \left\{ \left(E_{4,i\leq\infty}^{N^{(t)}} \cup E_{4,i\leq\infty}^{N^{(t+1)}} \right) - \left(E_{4,i\leq\infty}^{N^{(t)}} \cap E_{4,i\leq\infty}^{N^{(t)}} \right) \right\} \supset \left\{ S_{3,i\leq\infty}^{N} \wedge S_{4,i\leq\infty}^{N} \wedge S_{5,i\leq\infty}^{N} \right\}$$

(5.26a)

The top-level, *present*-linking VCS S_4 , shown in equation 5.26a, is a superset of the union of its' uppermost *present* and *future* environments, when subtracted from its intersection. This, in turn, is a superset of the uppermost $S_{3\text{-}4\text{-}5}$ homoestat.

$$S_{4,i\leq\infty}^{N^{(t+1)}} \Rightarrow \left(\left\{ \left(E_{4,i\leq\infty}^{N^{(t)}} \cup E_{4,i\leq\infty}^{N^{(t+1)}} \right) - \left(E_{4,i\leq\infty}^{N^{(t)}} \cap E_{4,i\leq\infty}^{N^{(t+1)}} \right) \right\} \supset \left\{ S_{3,i\leq\infty}^{N} \wedge S_{4,i\leq\infty}^{N} \wedge S_{5,i\leq\infty}^{N} \right\} \right)$$

(5.26b)

The top-level, *future*-linking VCS S_4 , shown in equation 5.26b, is a superset of the union of its' uppermost *present* and *future* environments, when subtracted from its intersection. This, in turn, is a superset of the uppermost $S_{3\text{-}4\text{-}5}$ homoestat. Whilst exhibiting its S_3, S_4 and S_5 constituents within the syntax, the VCS design grammar model seeks to depict the environment of the system-in-focus as the highest recursive level of the metasystem. The set theory thus reflects the fundamental incorporation of the environment into the system whole, thereby potentializing both a theoretical sense of self and the maintenance of viability in an open-bounded environment. Within the $S_{3\text{-}4}$ metasystem homeostat, this research has ascertained the possibility to produce a VCS analogy to the VSM S4, thereby developing a prototypical hybrid system capable of mimicking the timely system plan property of this agent.

System Five (S_5): *Policy:*

$$S^{0}_{5,i\leq\infty} \Rightarrow A$$

(5.27)

The identity 5.27 shows how the lowest, atomic level, system five, maps to the atomic set.

$$S^{n}_{5,i\leq\infty} \supset \left(\begin{array}{l} M^{n}_{5,i\leq\infty} \cup O^{n}_{5,i\leq\infty} E^{n}_{5,i\leq\infty} \cup \left(\begin{array}{c} \left[E^{N(t)}_{4,i\leq\infty} \cup E^{N(t+1)}_{4,i\leq\infty} \right] - \\ \left[E^{N(t)}_{4,i\leq\infty} \cap E^{N(t+1)}_{4,i\leq\infty} \right] \end{array} \right) \supset \\ \left[S^{n}_{3,i\leq\infty} \wedge S^{n}_{4,i\leq\infty} \right] \end{array} \right)$$

(5.28)

The identity 5.28 illustrates generic S_5 as being comprised of its' particular management unit, operation and environment, unioned with the forescasting capability, enabled by the subtraction of the current and future environment. It is also defined as being a superset of the generic $S_{3\text{-}4}$ metasystem homeostat. The syntax also exhibits the notional future forecasting VCS capability, achieved through the union of the current, relevant S_4 environments with its future counterpart. This is then subtracted from its intersection; this being a subset of the holism policy maker.

$$S^{N}_{5,i\leq\infty} \supset \left(\begin{array}{l} \left\{ M^{N}_{5,i\leq 7} \cup O^{N}_{5,i\leq\infty} \cup E^{N}_{5,i\leq\infty} \right\} \cup \left(\begin{array}{l} \left\{ E_{4}^{N(t)} \cup E_{4}^{N(t+1)} \right\} - \\ \left\{ E^{N(t)}_{4,i\leq\infty} \cap E^{N(t+1)}_{4,i\leq\infty} \right\} \end{array} \right) \supset \\ \left\{ S^{N}_{3,i\leq\infty} \wedge S^{N}_{4,i\leq\infty} \right\} \end{array} \right)$$

(5.29)

The equations 5.28 and 5.29 exhibit S_5 as a superset of top level S_3 and $S_{4,}$ it having ultimate authority over its counterparts within the systemic federation, whilst simultaneously monitoring the $S_{3\text{-}4}$ homeostat.

The VCS research ideal is for S_5 to contain a plan of the $S_{3\text{-}4}$ homeostat. The sysntax within 5.29 also exhibits the notional future forecasting VCS capability, achieved through the union of the current, relevant S_4 environments with its future counterpart. This is then subtracted from its intersection; this being a subset of the holism policy maker; S_5 attaining normative planning, partly illustrated by both this and equation 5.28 relating temporal environments.

System Three-Four-Five ($S_{3\text{-}4\text{-}5}$) ***Metasystem Homeostat:***

$$\left(S^{0}_{3,i\le 7} \wedge S^{0}_{4,i\le 7} \wedge S^{0}_{5,i\le 7}\right) \Rightarrow A$$

(5.30)

The combined $S_{3\text{-}4\text{-}5}$ system alliance is atomic at the lowest recursion, nought. The inter and mutual-dependence between the respective systems within, denotes a stronger connection than a union operator would imply.

$$\left(\begin{array}{l} \left(S^{n}_{3,i\le\infty} \wedge S^{n}_{4,i\le\infty} \wedge S^{n}_{5,i\le\infty}\right) \Rightarrow \\ \left(\begin{array}{l} \left(S^{n}_{3^*,i\le\infty} \cup E^{n}_{3,i\le\infty} E^{n}_{4,i\le\infty} E^{n}_{3^*,i\le\infty} E^{n}_{5,i\le\infty}\right) \cup \\ \left(\begin{array}{l} \left(E^{N(t)}_{4,i\le\infty} \cup E^{N(t+1)}_{4,i\le\infty}\right) - \\ \left(E^{N(t)}_{4,i\le\infty} \cap E^{N(t+1)}_{4,i\le\infty}\right) \end{array}\right) \subset \left(S^{N}_{3,i\le\infty} \wedge S^{N}_{2,i\le\infty} \wedge S^{N}_{1,i\le\infty}\right) \end{array}\right) \end{array}\right) \subset$$

(5.31a)

Equation 5.31a illustrates the compostion of the VCS $S_{3\text{-}4\text{-}5}$ metasystem homeostat at the generic level of recursion. As in 5.30, the operator ^ reflects the special relationship between these mutually-dependant systems, in that the output from each influences the input to their fellow systems. 5.31a is shown to be a subset of the ensuing part 5.31b.

The metasystem assumes the role of the composite management vortex that is masterminded by $S_{5,}$ and presides over and beyond the $S_{3\text{-}2\text{-}1}$ homeostat. Although of lower logical order, S_5 is not necessarily of higher authority.Supported by S_{3*} and S_2, it bridges the distinction between the intra and extra, systemic requirements ipso facto promoting viability. It is the critical homeostatic force within the system, containing models of both the extra-systemic environment and the internal systemic capabilities. This is reflected within the syntax by appliance and manipulation of the S_4 and E_4 temporal dimensions, whilst exhibiting its S_3, S_4 and S_5 constituents. Nevertheless, $S_{3\text{-}4\text{-}5}$ is not, necessarily of higher authority.

$$S_{4,i\leq\infty}^{n} \supset \left(\left(\begin{array}{l} M_{4,i\leq\infty}^{n} \cup O_{4,i\leq\infty}^{n} \cup E_{4,i\leq\infty}^{n} \cup \\ \left(\begin{array}{l} \left\{ E_{4,i\leq\infty}^{N(t)} \cup E_{4,i\leq\infty}^{N(t+1)} \right\} - \\ \left\{ E_{4,i\leq\infty}^{N(t)} \cap E_{4,i\leq\infty}^{N(t+1)} \right\} \end{array} \right) \end{array} \right) \subset \left\{ S_{3,i\leq\infty}^{N} \wedge S_{4,i\leq\infty}^{N} \wedge S_{5,i\leq\infty}^{N} \right\} \right) \subset$$

(5.31b)

Equation part 5.31b exhibits how the S_4 at generic level of recursion is a composed of its' mangemnt unit, operation and pertinent environment. This is unioned with the VCS forecasting capability achieved through the union of the current, relevant S_4 environments with its future counterpart.

This is then subtracted from its intersection; this being a subset of the uppermost $S_{3\text{-}4\text{-}5}$ metasystem homeostat.

This equation part is collectively defined as a subset of the following equation 5.31c.

$$S^{n}_{5,i\leq\infty} \supset \left(\begin{array}{l} M^{n}_{5,i\leq\infty} \cup O^{n}_{5,i\leq\infty} \cup E^{n}_{5,i\leq\infty} \cup \\ \left(\left\{ E^{N^{(t)}}_{4,i\leq\infty} \cup E^{N^{(t+1)}}_{4,i\leq\infty} \right\} - \left\{ E^{N^{(t)}}_{4,i\leq\infty} \cap E^{N^{(t+1)}}_{4,i\leq\infty} \right\} \right) \subseteq \left\{ S^{N}_{3,i\leq\infty} \wedge S^{N}_{4,i\leq\infty} \wedge S^{N}_{5,i\leq\infty} \right\} \end{array} \right)$$

(5.31c)

Equation part 5.31c exhibits how the VCS S_5 at generic level of recursion is comprised of its' particular management unit, operation and associated environment. These are necessarily unioned with the VCS forecasting concept, manifest via the union of the *present*, relevant S_4 environments with its *future* counterpart. This is then subtracted from its intersection; this being a subset of the uppermost $S_{3\text{-}4\text{-}5}$ homeostat.

As a member of the $S_{3\text{-}4\text{-}5}$ metasystem homeostat and the holism policy maker, S_5 attains normative planning. This significance of S_5, is that it has ultimate authority over its counterparts within the systemic federation, whilst monitoring the $S_{3\text{-}4}$ homeostat.

$$\left\{ S^{N}_{3,i\leq\infty} \wedge S^{N}_{4,i\leq\infty} \wedge S^{N}_{5,i\leq\infty} \right\} \Rightarrow$$
$$\left(\begin{array}{l} S^{N}_{3,i\leq\infty} \cup E^{N}_{3,i\leq\infty} \cup S^{N}_{4,i\leq\infty} \cup E^{N}_{4,i\leq\infty} \cup \\ S^{N}_{5,i\leq\infty} \cup E^{N}_{5,i\leq\infty} \cup S^{N}_{3^*,i\leq\infty} \cup E^{N}_{3^*,i\leq\infty} \end{array} \right) \cup \left(\begin{array}{l} \left\{ E^{N^{(t)}}_{4,i\leq\infty} \cup E^{N^{(t+1)}}_{4,i\leq\infty} \right\} - \\ \left\{ E^{N^{(t)}}_{4,i\leq\infty} \cap E^{N^{(t+1)}}_{4,i\leq\infty} \right\} \end{array} \right) \supset$$

(5.32a)

The composition of the upper level $S_{3\text{-}4\text{-}5}$ trilogy is illustrated in equation 5.32a. It is an incorporation of not only those particular systems and environments, but also that of the associated S_{3^*}, at an equivalent recursion level and range position. Similarly, this example indicates how the temporal forecasting is hypothesized by subtraction of the intersection of the *present* and *future,* S_4 environments from its' union. This equation part is shown to be a collective superset of the ensuing equation part 5.32b.

$$\left(S_{4,i\le\infty}^{N(t)} \supset \left(M_{4,i\le\infty}^{N(t)} \cup O_{4,i\le\infty}^{N(t)} \cup E_{4,i\le\infty}^{N(t)} \cup \left(E_{4,i\le\infty}^{N(t)} \cup E_{4,i\le\infty}^{N(t+1)} \right) - \left(E_{4,i\le\infty}^{N(t)} \cap E_{4,i\le\infty}^{N(t+1)} \right) \right) \right) \subset$$

(5.32b)

This identity part 5.32b, shows the cmpostion of the present top-level S_4 as being a management, operation and environment, in addition to the forecasting method illustrated in 5.32a. indicates how temporal VCS forecasting is hypothesized. These elements are collectively classified as being a subset of the ensuing equation part 5.32c.

$$\left(S_{4,i\le\infty}^{N(t+1)} \supset \left(\left(M_{4,i\le\infty}^{N(t+1)} \cup O_{4,i\le\infty}^{N(t+1)} \cup E_{4,i\le\infty}^{N(t+1)} \cup \right) \cup \left(\left(E_{4,i\le\infty}^{N(t)} \cup E_{4,i\le\infty}^{N(t+1)} \right) - \left(E_{4,i\le\infty}^{N(t)} \cap E_{4,i\le\infty}^{N(t+1)} \right) \right) \right) \right) \subset$$

(5.32c)

The identity part 5.32c, depicts the elements of the future top-level S4 as being a management, operation and environment, in addition to the forecasting method elucidated in 5.32a.

These elements are collectively classified as being a subset of the equation part, 5.32d

$$
S_{5,i\leq\infty}^{N} \supset
\left(\left(\left\{ M_{5,i\leq 7}^{N} \cup O_{5,i\leq\infty}^{N} \cup E_{5,i\leq\infty}^{N} \right\} \cup \left(\begin{matrix} \left\{ E_{4}^{N(t)} \cup E_{4}^{N(t+1)} \right\} \\ \left\{ E_{4,i\leq\infty}^{N(t)} \cap E_{4,i\leq\infty}^{N(t+1)} \right\} \end{matrix} \right) \right) \supset \left\{ S_{3,i\leq\infty}^{N} \wedge S_{4,i\leq\infty}^{N} \right\} \right)
$$

(5.32d)

Equation part 5.32d, illustrates the topmost compositon of S5, as its' particular mangment unit, operation and environment. It also exhibits the proximity of S_3 and S_4 within not only the metasystemic VCS S_{3-4} homeostat, but also reflects how it is moderated by the intervention of S_5 as a superset, which in turn possesses a notional plan of the S3-4 homeostat.

Conversely, the S_{4-5} coupling compares the real time *self* model of the internal systemic capabilities and the *non-self* analogue of its' embedded environment. Viability is thereby promoted by S_{4-5}.

5.1.2 Identities Syntax Examples of the Design Grammar Model of Superscript Relationships between the Recursions:

There follow key examples from the superscript model, illustrating the relationships between recursive levels.

Notations:

S : *a system*$\{_{0,1,2,3,4,5}\}$

S : *a system recursion*$\{^{0,n,N}\}$

i : *a system identifier*

A : *a system set* $\{a_1,a_2,a_3,a_4,\ldots_i\}|\ a_i$ *is atomic*

M : *a system Management Unit*

O : *a system Operation Unit*

E : *a system Environment*

t : *CurrentTime parameter*$|t^{+1}$:*FutureTime*$|t^{-1}$: *PastTime*

Binary Relations:

$A \Rightarrow B$: If A is true, B is also true

Binary Operators:

$A \cup B$: The set containing all of those elements within A and B

$A \cap B$: The set containing those common elements of A and B

$A - B$: The set that results when B is subtracted from A

$A \not\subset B$: The set A is not a subset of set B

System Nought (S_0): ***Atomic Catalyst:***

$$S^0_{0,i\leq\infty} \Rightarrow A \tag{5.33}$$

The atomic level system nought, or *atomic catalyst* as it has been termed, is represented syntactically by equation 5.33 at recursion nought. It is shown to mirror its' VCS counterparts, by mapping to the notional empty set. Atomic level identities between both subscript and superscript models are isomorphic, as the VCS will, innovatively, atomically emerge and so recurse upwards from this point.

$$\left(\left(S^{0}_{0,i\leq\infty} \subset \bigcup_{1\leq k\leq\infty} S^{n+1}_{k,i\leq\infty}\right) \text{For } \mathrm{N} \geq \mathrm{n} > 0\right)$$

(5.32)

In equation 5.32, the set theory modelling method has deviated from showing the inner composition of a management and operation unit, to identifying the fact that each and every S_0 at recursion nought, is a subset of all systems at the higher levels. This attempts to exhibit, in a short-form approach, the fractal-like recursive VCS nature that is so crucial to reducing redundancy ergo complexity.

This atomic recursion technique exclusively spawns subsequent levels when S_0 unites with an S_2. An S_0 cannot alone yield higher level recursions. S_0 being originated from the dissection of the Beerian S1, its' S_2 removed yet still juxtaposed to the metasystem outside the novel S_0 boundary.

System One (S_1): *Implementation:*

$$S^{0}_{1,i\leq\infty} \Rightarrow A$$

(5.33)

The S_1 syntax in equations 5.33 and 5.34, depicts the relationship between the recursive levels, or recursions. Syntax within equation 5.33, depicts how the atomic level, S_1^0 of the Superscript design grammar Model isomorphically maps to the respective S_1^0 within the Subscript Model [1]. Isomorphic, atomic S_1^0's [1] show bottom-up emergence.

$$\left(\left(S_{1,i\le\infty}^{n} \supset \bigcup_{1\le k\le\infty} S_{k,i\le\infty}^{n-1}\right)\text{For N} \ge \text{n} > 0\right)$$

(5.34)

Equation 5.34 declares S_1 at a particular position is superset of all of the S_1's nested at lower levels of recursion at the equivalent range-position, therefore indicating how all systems at S^n and S^N levels contain an entire self-governing VCS model at a lower recursive level. Both the primacy of S_1 and its' autopoietic capability,is illustrated by the syntax spawning successive recursive levels, within no stipulated, terminable range. The set theory Whilst the Beerian human-agencied stance was that each S1 at a particular position within the range of 1 to 7, is a superset of all of the S1's nested at lower levels of recursion at the equivalent range-position, VCS research digresses from this, having determined that an infinite number of S_1's may occur per recursion, to reflect the context shift towards computing.

System Two (S_2): *Co-ordination:*

$$S_{2,i\le\infty}^{0} \Rightarrow A$$

(5.35)

The S_2 VCS constituent is an anti-oscillatory and local-regulatory S_1 element, defined in equation 5.35 as a superset of each of it's particualr S_1 at the generic level of recursion.

$$\left(\left(S_{2,i\le\infty}^{n}\supset\bigcup_{1\le k\le\infty}S_{k,i\le\infty}^{n-1}\right)\text{For}\,\mathrm{n}\ge 0\right)$$

(5.36)

A standardizing body, S_2 assists S_3 towards integrative function as the locus of homeostasis. Identity 5.36 reflects the power of generic S_2 as superset of the S_1'S at that and the atomic levels of recursion.

$$\left(\left(\left(S_{2,i\le\infty}^{n-1}\subset S_{2,i\le\infty}^{n}\right)\bigcup_{1\le k\le\infty}S_{k,i\le\infty}^{n-1}\right)\text{For}\quad \mathrm{n}<\mathrm{N}\right)$$

(5.37)

Equation 5.37 depicts each generic S_1^n to be a subset of its sucessive generic S_1^n, thereby highlighting the bottom-up recursive structure.

This is realized via a generic level S_2^n ‘s unity with an atomic and generic-level $S_0^{0,n}$, thus initiating atomic recursion.

$$S_{2,i\le\infty}^{N}\Rightarrow\phi$$

(5.38)

Equations 5.38 and 5.39 ilustrate how top level identities omit $S_2^{[N]}$, so terminating the spawning of further VCS recursions. The former identity specifies the uppermost S_2 to map to the empty set

$$S_{2,i\leq\infty}^{N-1} \not\subset S_{2,i\leq\infty}^{N}$$

(5.39)

Equation 5.39 depicts an alternative notation to show this, by stating that those S_2's at a lower level of recursion than the top, are not a subset of a notional, top-level S_2, as this does not exist.

System Three (S_3): *Control:*

$$S_{3,i\leq\infty}^{0} \Rightarrow A$$

(5.40)

Equation 5.40 shows the VCS S3 at the lowest level of recursion, to map to the atomic set.

$$\left(\left(S_{3,i\leq\infty}^{n} \subset \bigcup_{1\leq k\leq\infty} S_{k,i\leq\infty}^{n-3.2,1}\right) \mathrm{For\ N} \geq \mathrm{n}\right)$$

(5.41)

The S_3 constituent regulates, optimizes and cohesively stabilizes internal systemic activity. Equation 5.41 shows S_3 as a subset of the higher, autonomic fellow metasystemic S_{3-2-1} union.

Whereas S_1 is aided by S_2, S_3 gives strategic, overall structure, planning and integrating unified activities of the S_1's. A vital fulcrum, assisted by S_2, S_3 provides overall structure, integrating cohesive activities of the S_1's and shown to be a subset of the trio. Equation 5.41 thereby illustrates the theory of how it stabilizes the internal milieu of the system via cross-recursive co-ordination [27] .

System Three Star (S_{3*}): *Audit:*

$$S^{0}_{3*,i\leq\infty} \Rightarrow A$$

(5.42)

Equation 5.42 illustrates how the notional VCS system three star, may be mapped to the atomic set, should it be instituted as a constituent. Advancing the VSM to the VCS emphasizes the intermittency and thus redundancy, of this element in a computing context, which has the capability to monitor in a constant state. S_{3*} may, however, potentially be effected as a constituent, as illustrated by the identities within equations 5.42 and 5.43.

$$\left(\left(S^{n}_{3*,i\leq\infty} \subset S^{n}_{3,i\leq\infty} \bigcup_{1\leq k\leq\infty} S^{n-1}_{k,i\leq\infty}\right) \text{For } \mathrm{N} \geq \mathrm{n} > 0\right)$$

(5.43)

Equation 5.43 demonstrates how the S_{3*} VCS component, as a sporadic auditing system is assimilated into $S_3^{[0,n,N]}$ within both the subscript and superscript design grammar models. It reflects how S_{3*} is a subset of the S_3 at a particular range position, within the equivalent recursive level. S_{3*} is a backup inspection facility to the validity and functionality of S_1 and S_3 respectively. It regulates and optimizes the system as a whole, cohesively stabilizing internal systemic activity by its monitoring capabilities.

System Four (S_4): *Intelligence:*

$$S^{0}_{4,i\leq\infty} \Rightarrow A$$

(5.44a)

S_4 at the lowest level of recursion is shown to map to the atomic set via equation 5.44a.

$$\left(S_{4,i\leq\infty}^{0^{(t+1)}} \supset \bigcup_{1\leq k\leq\infty} S_{k,i\leq\infty}^{0^{(t)}} \right)$$

(5.44b)

Equation 5.44b shows how the *future*-linking S_4 at the atomic level of recursion, is a superset of the *present*-linking, S_4 counterparts.

S_4 is unique amongst its fellow systems, in that it communicates directly with each of the local, future and global environments. The design grammar model exhibits these facets defined as the parameters time t^{n-1}, t^n and t^{n+1} within the system-in-focus.

$$\left(\left(S_{4,i\leq\infty}^{n} \supset \bigcup_{1\leq k\leq\infty} S_{k,i\leq\infty}^{n-1} \right) \text{For } \mathrm{N} \geq \mathrm{n} > 0 \right) \in$$

(5.45a)

The above equation part 5.45a, illustrates how each generic-level S_4, is a superset of all lower-level S_1's, at a corresponding range position at all three specified levels of recursion. These elements are associated with the following equation part of 5.45b.

$$\left(S_{4,i\leq\infty}^{n^{(t+1)}} \supset \bigcup_{1\leq k\leq\infty} S_{k,i\leq\infty}^{n^{(t)}} \right) \in$$

(5.45b)

Equation part 5.45b explicates how the S_4 VCS recursions are atomically-spawned. The *future*-linked S_4 at a particular recursion and range position is classified as a superset of those corresponding, yet linking to the *present*. This ensemble is an element of 5.45c. System four enables self-reference and planning, embedding an internal model, assisted by the temporal elements within the syntax.

$$\left(S_{4,i\leq\infty}^{n-1^{(t)}} \supset \bigcup_{1\leq k\leq\infty} S_{k,i\leq\infty}^{n-1^{(t+1)}} \right) \in$$

(5.45c)

Identity part 5.45c shows how the *present* S_4 at the generic level of recursion is a superset of its' lower level counterpart linking to the *future* environment. This demonstrates the temporal forecasting and recursive aspect to the VCS design grammar model. These elements relate to 5.45d.

$$\left(S_{4,i\leq\infty}^{n-1^{(t+1)}} \subset \bigcup_{1\leq k\leq\infty} S_{k,i\leq\infty}^{n-1^{(t)}} \right) \in$$

(5.45d)

Equation part 5.45d illustrates how the *future* S_4 at a lower, generic level of recursion is a subset of it's' counterpart linking to the *present* environment. These are also defined as elements of 5.45e.

$$\left(S_{4,i\leq\infty}^{N^{(t)}} \subset \bigcup_{1\leq k\leq\infty} S_{k,i\leq\infty}^{N^{(t+1)}} \right) \in$$

(5.45e)

The identity section 5.45e reflects the *present*-linking S_4 at the topmost level of recursion. This is defined as a subset of it's' counterpart inking to the *future* environment, whilst also stipulated to be elements of the ensuing 5.45f equation part.

$$\left(S_{4,i\leq\infty}^{N-1^{(t)}} \subset \bigcup_{1\leq k\leq\infty} S_{k,i\leq\infty}^{N-1^{(t+1)}} \right)$$

(5.45f)

The equation part 5.45f depicts how the topmost *present*-linking S_4 at a lower level, is a subset of its' *future*-linking counterpart.

It links directly to its parent and subordinates, to exhibit inter-recursive cohesion; thereby conceptually lessening VCS redundancy and so complexity.

System Five (S_5): *Policy*:

$$\left(S_{5,i\leq\infty}^{0} \Rightarrow A \right)$$

(5.46)

Equation 5.46 illustrates how the VCS S_5 at the atomic level, maps to the atomic set.

$$\left(\left(S_{5,i\leq\infty}^{n} \supset \bigcup_{1\leq k\leq\infty} S_{k,i\leq\infty}^{n-1} \right) \text{For } \mathrm{N} \geq \mathrm{n} > 0 \right)$$

(5.47)

Equation 5.47 reflects how S_5 IS a subset of higher recursions. This demonstrates the bottom-up, or atomic, VCS recursion method adopted for the design grammar model.

System Three-Four-Five ($S_{3\text{-}4\text{-}5}$): Metasystem Homeostat:

$$\left(S^0_{3,i\leq\infty} \wedge S^0_{4,i\leq\infty} \wedge S^0_{5,i\leq\infty}\right) \Rightarrow A$$

(5.48)

The metaphorical head of the system [27], the Metasystem Homeostat is comprised of S_3, S_4 and S_5. The ^ operator depicts the special. Mutual dependence between these systems, in that an output will influence the input to its' fellow system. The lowest level $S_{3\text{-}4\text{-}5}{}^0$ maps to the atomic set, as shown within equation 5.48.

$$\left(\left(S^n_{3-4-5,i\leq\infty} \supset \bigcup_{1\leq k\leq\infty} S^{n-1}_{k,i\leq\infty}\right) \text{For N} \geq \text{n} > 0\right)$$

(5.49)

Equation 5.49 reflects how the metasystem homeostat is balanced by the intervention of S_5, the $S_{3\text{-}4}$ loop thereby promotes control and strategic planning within the VCS model.

The generic level $S_{3\text{-}4\text{-}5}$ metasystem homeostat, is shown to be a superset of it's' counterpart at the lower levels of recursion.

5.2 Chapter Summary

The VCS design grammar model dually illustrates the systemic configuration and inter-recursion cohesion. The set theory thus reflects the fundamental incorporation of the environment into the system whole, thereby allowing recognition of the requirements of the the VSM context shift from human multi-agent, world situation of the environment of the system in focus. The environment is depicted as the highest recursive level of the metasystem. The autopoeitic properties are illustrated by the syntax spawning successive recursive level, with the environment being shown as the highest recursive level of the metasystem. The equations additionally exhibit a temporal facet in the presentation of a notional *future* forecasting VCS capability, achieved through the union of the *present*-linking, relevant S_4 environments with its *future* counterpart. This is then subtracted from its intersection. The syntax can accordingly be said to unify the environment into the system whole, thereby articulating a sense of self and therefore the preservation of viability in a changing environment.

Chapter 6

Case Studies

Introduction

This chapter presents two case studies to validate the proposed design grammar in a notional real world context of a sorting software system; S_1 represents a metaphor for homeostasis. Both a closed [2] and open [3, 24] environment VCS case study are presented in the context of previous genetically modified system scenario [69], focussing on demonstrating the validity and pertinence of the design grammar model. By employing the case studies, the VCS research goal is validated. The viability of the VCS, that is, its ability to exist in a changing environment will be exhibited by adopting a previously published system as a vehicle to demonstrate both self-organization and emergence, so reducing redundancy and thus complexity.

Equipping the VCS to conform to the Law of Requisite Variety led to the application of an experiment in algorithmic "Hot Swapping" as the case study scenarios, by first defining an environmental scenario to which the system must respond. To facilitate the system in its task, a means is provided that allows the VCS to determine the efficiency of the responses at its disposal. By considering algorithmic hot swapping in the context of research surpassing autonomic computing, towards Viable Computing Systems, cybernetic, mathematical and biological metaphors are allied to the human autonomic agent capability of the Managerial Cybernetics underscoring Beer's Viable System Model.

A bi-perspective set theory design grammar model is employed exhibiting relationships between the systems and the recursive levels of the VSM. In this context, the VCS S_1 was analysed as a metaphor S_1 represents a metaphor for homeostasis within the design grammar model, the syntax reflects the software system state i.e. a set of variables indicate the current system in focus and the environment in which it needs to retain viability.

By incorporating the environment as part of the system, the technique promotes both portability and viability within an initially closed, yet changing, environment [2], followed by a conceptual open environment [3, 24]. Algorithmic hot swapping has been used to provide a repertoire of tailored responses to environmental change within this context. Systemic emergence and viability is thereby promoted, whilst an associated Learning Classifier System (LCS) is suggested to allow the system to develop an adaptive environmental model of appropriate, optimized responses, similarly demonstrating proof of the temporal and autonomic properties of the VCS concept.

Progressed VCS architectural representations are depicted in Figures: 6.1 and 6.3, showing recursivity with example identities exhibiting the context-free attribute. Further functions and a set of disturbances are introduced, supplying a potential repertoire of tailored responses to open environmental change. Fundamental to promoting emergence, thus viability is Sommerhoff's concept of directive correlation [25] and Ashby's notion of goal-directedness [22], i.e. the ability to achieve a goal-state under variations in the environment. The VCS temporal parameters were superimposed, enabling response to environmental stimulus post time t^{n-1}, enabling intrinsic reaction whilst forecasting. This research applied homeostatic [112] and autopoietic [113, 114] approaches to generate a referential self-model of the internal systemic capabilities t^n and a model reflecting the required world situation t^{n+1}, so autonomically addressing environmental factors via feedback control. Example identities exhibit potential for context-free portability including sets of values of environmental and behavioural variables and a set of outcomes allowing the system to develop an adaptive environmental model of fit VCS responses.

6.1 VCS Case Study; a Closed Environment Genetically Modified System Scenario

This initial VCS case study [2], is based on the hot swapping of sort algorithms [69] developed earlier, that seemed to possess the initial characteristics required, whilst offering scope for the further development of more complex trials. This VCS case study demonstrates response within a closed environmental stimulus post time t^{n-1}, via inherently reacting and forecasting. Both homeostatic [112], and autopoietic [113, 114] methods are related to generate a self-model of the inner systemic capabilities t^n and a model of the environment t^{n+1}.

Significantly, this facilitates portability to differing scenarios. By embracing cybernetic First Principles at the design stage, the design grammar model is endorsed by autonomically addressing environmental requirements, characterizing these temporal events by adapting the VSM's sensor/effector principles; feedback control is therefore integral to the design. An earlier algebraic set theory paradigm articulated the relationships between the VSM [1] systems (subscripts), depicting the architectural recursivity (superscripts). [67]. A design grammar Model that atomically recurses and emerges has thus been produced.

The findings suggest that the power lies in the syntax reflecting a VSM holism within each implementation system, or S_1. The context-free quality offers potential to model dynamic systems via provision of an internal model and an architecture imparting self-awareness [115]. The research objective was to subsume autonomic computing initiatives into cybernetically intelligent Viable Computing Systems [1]; akin to the subsumation of human autonomic systems by cognitive systems. This original case study [2], derived from an experimental, environmentally adaptive system, has hence proven the VCS research concept to prototype level. This case study of the VCS, was applied to a previous genetically modified system scenario [69]. These investigations have concluded that in a viable system all systems are mutually-dependant, yet if any has a special primacy, it is S_1. This is because it consists itself of viable systems. To cite Beer:

> *' The purpose of a system is what it does … and what the Viable System does is done by System One'* [20]

This purpose of a system is what it does, or POSIWID aphorism is one of Beer's most famous. When complex loops within a system that maintain the status quo are understood, investigations are better equipped to make positive changes towards retention of that systems' viability. The POSIWID principle applies a kind of reverse logic to systems thinking, in that it proposes analysis from effects to causes, rather than vice-versa. If a complex system produces a given outcome, or if a given outcome emerges from a complex system, then one may assume some purpose linked to this outcome.

This is a useful guide for investigation and interpretation. The spirit of POSIWID is that Beer felt one should ignore the official purpose of the system, ignore what the designers and custodians of a system say, and, rather, concentrate on its actual behaviour. Although the term was coined by Stafford Beer, it was picked up and developed further in a trio of books written in the 1980s by engineer Bill Livingston.

This research initiated a VCS case study attempting to address Ashby's Law of Requisite Variety in the context of previous system scenario [69]. The analysis of S_1 as a metaphor from the design grammar model led to the syntax reflecting a set of variables indicating the system state and the environment.

A novel architectural VCS representation is shown in Figure 6.1, whereby it is illustrated how the environment presents the system with 5-element arrays for sorting: Bubble sort, Shell sort, Quick sort, Insertion sort, Shaker sort and Merge sort, although Quick, Shaker and Merge are never selected in this experiment.

An environmental scenario was defined to which the VCS system must respond, defined here as providing a supply of n = 5 element arrays of integers for sorting. This technique provided a closed, highly controllable environment potentially ranging from smooth, i.e. relatively small changes between arrays to a highly discontinuous environment where subsequent arrays may vary between almost sorted to entire transposition.

A finite set of responses to environmental change was supplied to the VCS, which assumed the task of matching the most efficient response to the current environmental position at runtime. The response set was represented by a library of sorting algorithms, each capable of sorting any array received from the environment, although not necessarily optimally.

To facilitate the system in its task, a means is provided that allows the system to determine the efficiency of the responses at its disposal; the system experimented to determine the most efficient response to the current environmental stance. Assuming that the environment was changing relatively slowly, the selected algorithm was used for subsequent arrays delivered by the environment.

While algorithmic hot swapping provides the VCS with a limited degree of adaptive capability, the finite set of responses does constrain the potential for optimisation, this approach being analogous to a programmer predicting the circumstances a system may encounter and providing a response to each event.

Environmental change is represented by degrees of unsortedness in the arrays. The system being provided with a set of responses, must determine the optimal response, with example identities exhibiting the context-free attribute.

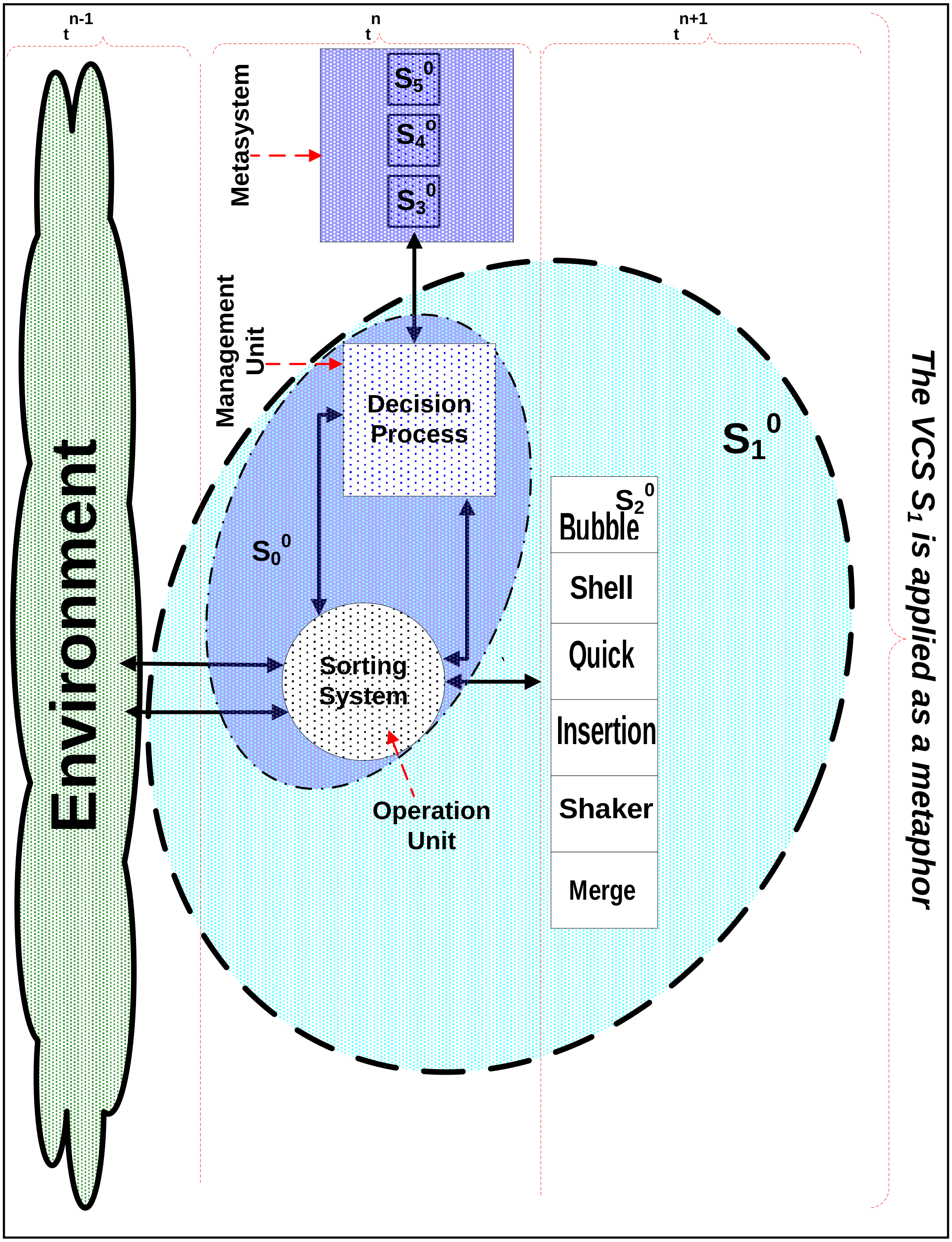


Figure 6.1: VCS Case Study: a Closed Environment Genetically Modified System Scenario

6.1.1 System One (S_1) Case Study VCS Model of the Relationship between the Systems (Subscripts) of the Closed Environment System:

Notations:

S : *a system*$\{_{0,1,2,3,4,5}\}$

S : *a system recursion*$\{^{0,n,N}\}$

i : *a system identifier*

A : *a system set* $\{a_1,a_2,a_3,a_4,\ldots_i\}|\ a_i$ *is atomic*

M : *a system Management Unit*

O : *a system Operation Unit*

E : *a system Environment*

t : *CurrentTime parameter*$|t^{+1}$*:FutureTime*$|t^{-1}$*: PastTime*

Binary Relations:

$A \Rightarrow B$: If A is true, B is also true

Binary Operators:

$A \cup B$: The set containing all of those elements within A and B

$A \cap B$: The set containing those common elements of A and B

$A - B$: The set that results when B is subtracted from A

$A \not\subset B$: The set A is not a subset of set B

$$S^{0}_{1,i<\infty} \Rightarrow A \qquad \textbf{(6.1)}$$

This identity 6.1, represents a piece of software, applying the VCS S_1 as a metaphor to represent homeostasis. The system sets are situated at the lowest, or atomic recursion *nought*.

Specifically, S_1 at recursion nought within a respective location, is defined by the identifier to be within an integer range of 1 to infinity.

$$S_{1,i<\infty}^{n} \Rightarrow \left(\begin{array}{l} S_{0,i<\infty}^{n} \cup E_{0,i<\infty}^{n} \cup S_{2,i<\infty}^{n} \cup E_{2,i<\infty}^{n} \cup \\ \left(E_{4,i<\infty}^{n^{(t)}} \cup E_{4,i<\infty}^{n^{(t+1)}} \right) - \left(E_{4,i<\infty}^{n^{(t)}} \cap E_{4,i<\infty}^{n^{(t+1)}} \right) \end{array} \right) \text{For } \mathrm{N} \geq \mathrm{n} > 0 \qquad (6.2)$$

This identity 6.2 depicts the VCS S_1 at the higher, *generic*, level of recursion. The set theory elucidates the composition of this system in the context of the closed environment case study, as illustrated in Figure 6.1. The associated S_0, and its particular environment are each unioned with the corresponding S_2 *sorting* system. The latter exhibits intrinsic control through the notion of algorithmic *hot-swapping* between these differing sorts.

It is thereby demonstrated how S_1^n makes use of the sorting system, a part of it's' functionality. This can be analogized to the human-agencied VSM making use of the human autonomic system, situated below the lowest level of each recursion.

$$S_{1,i<\infty}^{N} \Rightarrow \left(\begin{array}{l} M_{1,i<\infty}^{N} \cup O_{1,i<\infty}^{N} \cup E_{1,i<\infty}^{N} \cup \\ \left(E_{4,i<\infty}^{N^{(t)}} \cup E_{4,i<\infty}^{N^{(t+1)}} \right) - \left(E_{4,i<\infty}^{N^{(t)}} \cap E_{4,i<\infty}^{N^{(t+1)}} \right) \end{array} \right) \text{For } \mathrm{N} \geq \mathrm{n} > 0 \qquad (6.3)$$

This identity 6.3 reflects the configuration of VCS S_1 at the top level of recursion. The set theory omits the S_2 trigger that initiates a further level of recursion. Likewise this similarly dictates that the sorting algorithm is terminated here.

The system structure is defined as being composed of the relevant management unit, operation and the environment of the top level S_1. This is unioned with the forecasting capability, conceptualized by the subtraction of the current and future environments from the union of the said.

In essence, therefore, the equation 6.1 reflects that the Atomic level S_1 maps to the empty set. Generic level S_1 within equation 6.2, has no constraint on the number per recursion, S_1 being comprised of S_0 in union with its environment, unioned with the S_2 and its' respective environment. The process autopoietically spawns the successive S_1, demonstrating emergence and recursion.

Incorporation of the environment within equations 6.2 and 6.3, also promotes viability via operating upon the temporal elements in order to generate a forecasting capability and a model of the required world situation; a comparator to the internal model of the systemic capabilities.

A conceptual sense of VCS self is promoted via the sort algorithm and application of Ashby's' Requisite Variety ideal that purports every good regulator of a system must be a model of that system, internal variety matching the system. This demonstrates viability and homeostatic-like behaviour by the VCS, in the closed environment genetically-modified system scenario.

6.1.2 System One (S_1) Case Study VCS Model of the Relationship between the Recursive Levels (Superscripts) of the Closed Environment System:

<u>Notations:</u>

S : a system$\{_{0,1,2,3,4,5}\}$

S : a system recursion$\{^{0,n,N}\}$

i : a system identifier

A : a system set $\{a_1,a_2,a_3,a_4,\ldots_i\} \mid a_i$ *is atomic*

M : a system Management Unit

O : a system Operation Unit

E : a system Environment

t : CurrentTime parameter$|t^{+1}$*:FutureTime*$|t^{-1}$*: PastTime*

Binary Relations:

$A \Rightarrow B$: If A is true, B is also true

Binary Operators:

$A \cup B$: The set containing all of those elements within A and B

$A \cap B$: The set containing those common elements of A and B

$A - B$: The set that results when B is subtracted from A

$A \not\subset B$: The set A is not a subset of set B

$$S^{0}_{1,i<\infty} \Rightarrow A \tag{6.4}$$

This identity 6.4, isomorphically maps to its atomic counterpart within the perspective of the design grammar model that relates the systems, in that it similarly represents a piece of software,

In applying the VCS S_1 as a metaphor to represent homeostasis, the system sets are situated at the lowest, atomic, recursion nought.

$$\left(\left(S_{1,i<\infty}^{n} \supset \bigcup_{1\le k<\infty} S_{k,i<\infty}^{n-1}\right) \text{For N} \ge \text{n} > 0\right) \quad \textbf{(6.5)}$$

The algebraic model within equation 6.5, illustrates how atomic level S_1^0 isomorphically maps to its counterpart, so promoting coherence and atomic recursion. This identity also illustrates the atomic recursion capabilities that are core to the closed environment case study, in order to not only engender bottom-up emergence but also reduce redundancy and so complexity.

It is shown within equation 6.5, that this particular system has no limits upon the number of recursive levels, whilst the next, generic level is a superset of all lower levels. This also applies to the top level.

When fused with the sort algorithm that is situated at each S_2 within the respective levels, the system meets the requirement that it must demonstrate adaptive behaviour by the management of requisite variety, which states that for each and every environmental action, there is an equal and opposite response. Through influencing or causing change in other elements that make up the environment, in essence the case study proving the capability of the VCS to exert a degree of control over the environment. This is partially achieved by influencing the environment, through application of Ashby's Law of Requisite Variety.

Conclusion

This case study underscored the VCS research via fusing a priori cybernetic, biological and mathematical principles, explicitly Beer's VSM. The closed environment, system scenario was a good fit for the application of the algebraic method, allowing temporal and portable modelling. In exploiting the power of Beer's fractal-like architectural recursion, fused with the biological concepts of autopoiesis and homeostasis, the research goal of a Viable Computer System, surpasses the autonomic computing ideal.

Development of the VCS set theory syntax design grammar model has similarly been furthered; example identities are also presented, exhibiting the context-free attribute. Figure 6.1 is introduced, that shows an architectural representation of the VCS in this context. It is proposed that the adaptive capabilities of the VSM, normally executed by human agency, can be realized by applying the design grammar model VCS blueprint. Towards this end, it is shown how algorithmic hot swapping can generate a repertoire of tailored responses to environmental change.

This preliminary demonstration shows how viability can be maintained via interaction with an, initially closed, environment comprised of 5-element arrays for sorting. Environmental perturbations are represented by degrees of unsortedness in those arrays, the system being provided with a set of responses from which to determine the optimal response. It is concluded that this had validated the incorporation of the environment as part of the system, whilst demonstrating the VCS homeostatic-like i.e. self-regulatory capabilities.

The system "experimented" to determine the most efficient response to the environmental situation, therefore demonstrating a temporal, forecasting capability, mirrored within the design grammar syntax, via the application of the temporal parameters.

The sort algorithm assumes the role of S_2; Bubble sort, Shell sort, Quick and Merge sort, although Quick, Shaker and Merge are never selected for this case study.

Although algorithmic “hot swapping” provides the system with a limited degree of adaptive capability, the finite nature of the pre-determined set of responses within the closed environment, constrains the degree of optimization at this point. This can be likened to a human agent predicting and programming the permutation of environmental circumstances which the system-in-focus may encounter and presenting a response to each of these. Future research aspired to demonstrate that the system can provide its own set of responses.

The case study conceptually proved the validity of applying the autopoeitic and recursive properties of the VCS formalism. By algorithmic real-time modelling the VCS has captured those mobile elements necessary to satisfy not only the homeostatic requirements of the autonomic computing genre, yet surpassed these by manifesting a cybernetically inspired multi-agent intelligent system able to retain viability and demonstrate portable emergence. This investigation underscores the research seeking to progress Viable Computing Systems through the fusion of a priori cybernetic, biological and mathematical principles, specifically Beer's recursive VSM architecture.

The genetically modified, closed environment [2], system scenario provides a good fit for the application of the adopted algebraic method, allowing temporal and portable modelling. In exploiting the power of Beer's fractal-like architectural recursion, fused with the biological concepts of autopoiesis and homeostasis, the research goal of a Viable Computer System surpasses the autonomic computing ideal.

The intention was to further the ongoing evolution of the design grammar model, therefore expanding its portability to differing computing scenarios. Immediate refinement sought to apply the VCS in an open environment [3, 24], thereby demonstrating proof of the concept.

6.2 Open Environment VCS Case Study of a Previous Genetically Modified System Scenario

The research evolved to initiate an open-bounded case study [3, 5, 24] examining the relevance of the VCS model in an open environmental context of algorithmic hot swapping, towards a previous genetically modified software system [31, 69-71].

Work innovates a hybrid VCS architectural representation of the VSM S_1, which represents a metaphor for homeostasis as illustrated in Figure 6.3. By considering Sommerhoff's concept of '*directive correlation'* [25] and subsequent Ashbian deductions , particularly his notion of *'goal directedness'* [22] , that is, the ability to achieve a goal-state under variations in the environment. In Figure 6.2, equilibrium between the split parts of environment and system are considered. The findings herein, uncovered a special relation between these, within this context, by applying the earlier sort algorithm as a test bed.

Further functions and a set of disturbances are introduced, supplying a potential repertoire of tailored responses to open environmental change. The novel relations are clarified by Figure 6.3, an evolution of the VCS architecture post-application of the open-bounded design grammar model. It demonstrates the theory of how systemic disturbances may be homeostatically managed by the fusion of directive correlation [25] with the sort algorithm. Example identities exhibit potential for context-free portability including sets of values of environmental and behavioral variables and a set of outcomes allowing the system to develop an adaptive environmental model of fit responses illustrating temporal and autonomic properties of the VCS concept.

Fundamental to promoting emergence, thus viability is Sommerhoff’s concept of directive correlation [25] and Ashby's notion of goal-directedness [22], i.e. the ability to achieve a goal-state under variations in the environment.

Example identities exhibit potential for context-free portability including sets of values of environmental and behavioral variables and a set of outcomes allowing the system to develop an adaptive environmental model of fit responses illustrating temporal and autonomic properties of the VCS concept.

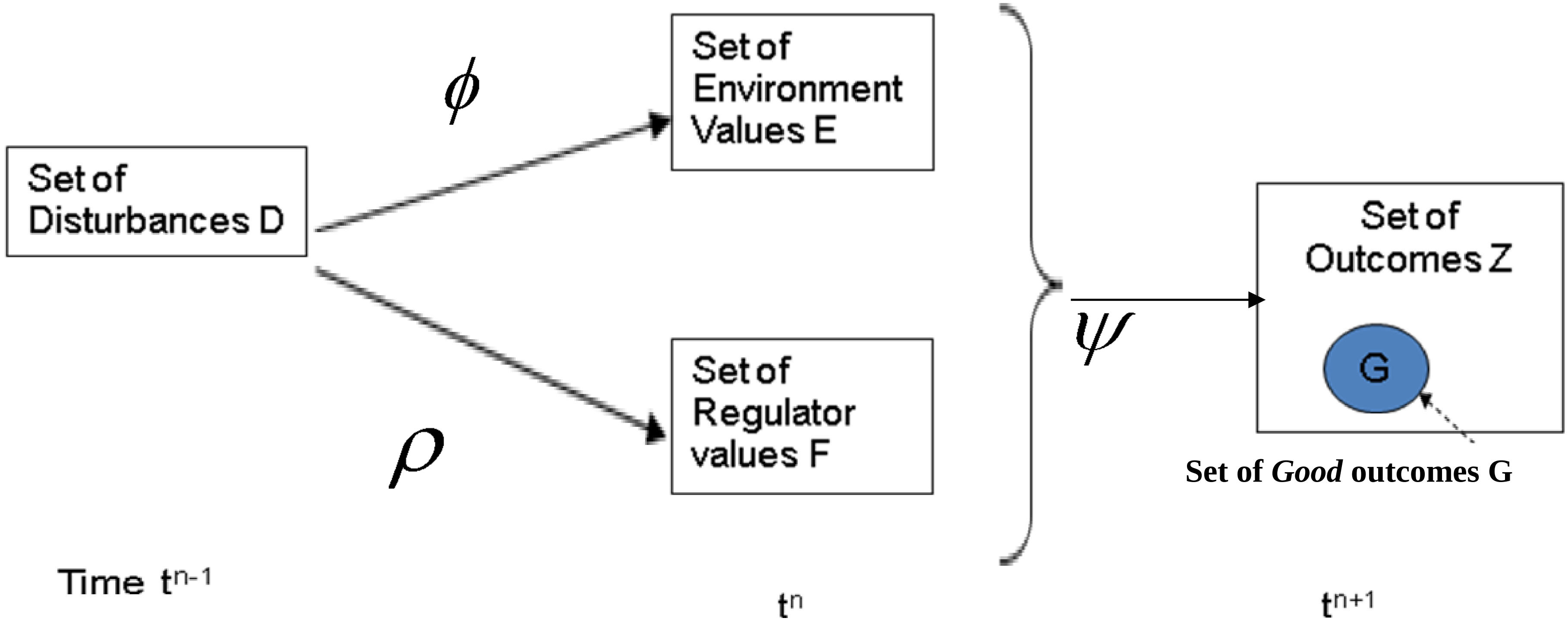


Figure 6.2: Ashby's Set Theory Illustration of Systemic Disturbances via Sommerhoff's Directive Correlation Tenet

The same system scenario was adopted, based upon an earlier closed environment case study [2] pertaining-to the hot swapping of sort algorithms [69]. Open-bounded self-organization and VCS emergence [3, 24] is suggested via the uniform genetically modified system scenario context. The earlier algebraic set theory identities articulated the relationships between the VSM systems (subscripts) [1], whilst depicting atomic recursion by (superscripts) to reflect a VSM holism within each implementation system, or S1.

Potential exists to create an internal model to impart context-free self-awareness [115] and so viability. This work hopes to subsume autonomic computing initiatives towards the VCS [1]; akin to human autonomic systems by cognitive systems, this research conceptualizing open-bounded viability [3, 24]. The investigations led to the modification of temporal parameters adopted by Ashby [22] in his citing of Sommerhoff [25], environmental disturbances occurring at time t^{n-1}, with sorting, directive correlation [25] and algorithmic hot-swapping, at times t^{n} and t^{n+1} respectively. A novel superimposition of these mappings and functions, both syntactically and architecturally, presented a surprisingly good fit. Homeostatic [63], and autopoietic [113, 114] methods allied, so enabling an aptitude to self-model systemic capabilities at time t^{n-1}, in addition to an environmental model at time t^{n}.

Notably endorsing portability, the cybernetic modelling technique proposes autonomic management of environmental factors, feedback control being core to the process. This extension to the VCS design grammar model and topology, theoretically addresses the research objective, whilst also suggesting capacity for further maturity.

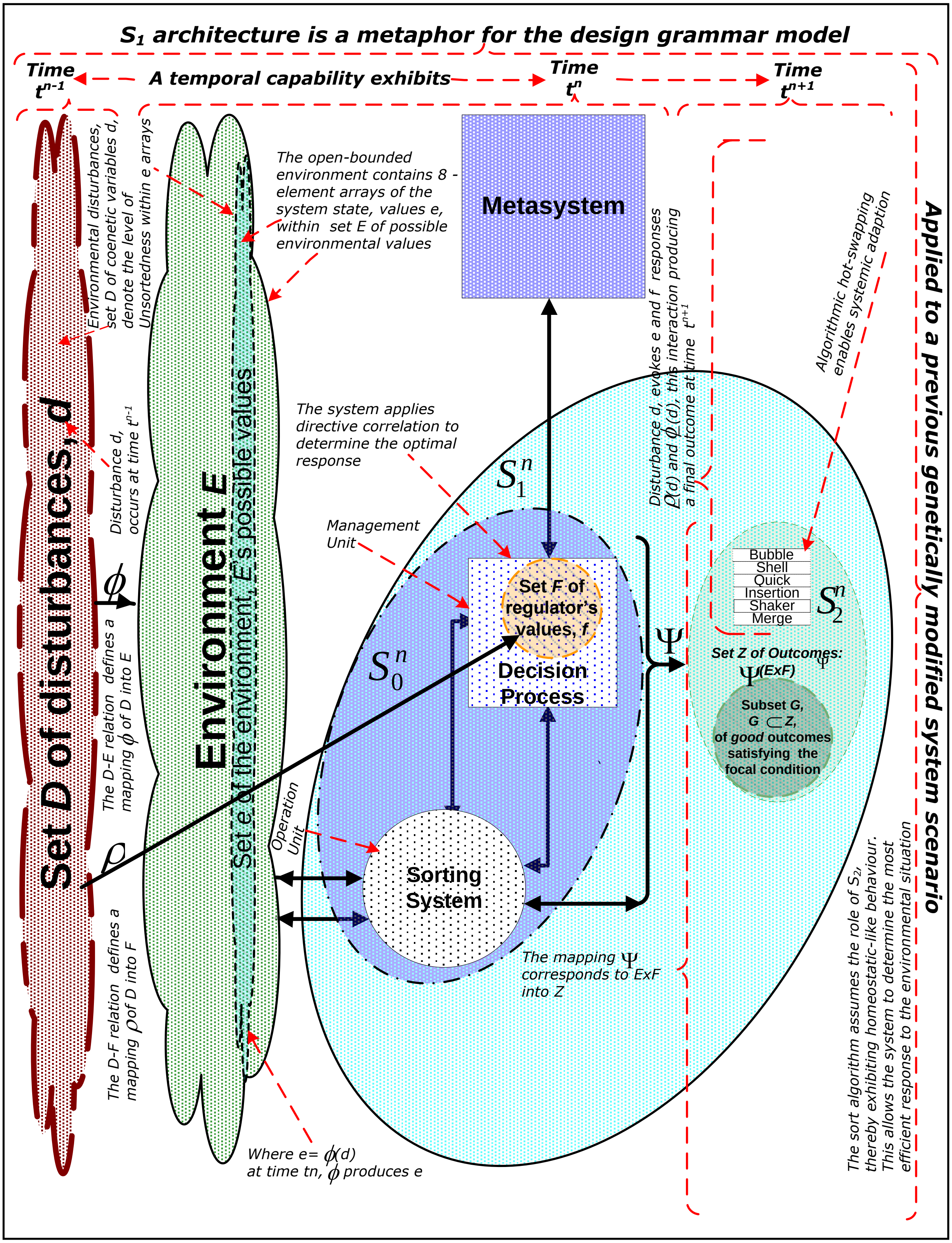


Figure 6.3: VCS Case Study: an Open Environment Genetically Modified System Scenario

6.2.1 Directive Correlation applied to VCS Design Grammar Model and Architecture:

Having developed a bi-perspective design grammar model, the study acknowledges Beers' conclusion that the management of any viable system poses the problem of managing complexity itself, since it is complexity that can threaten to overwhelm the regulators of that system.[125].

Whilst Ashby had proposed that a precise measure of systemic complexity, that is, the number of distinguishable systemic states could be termed *variety*, it was concluded that as all distinguishable states are not equally likely within an open-bounded system, a means is necessary to control this complexity. This led to the analysis of the retrospective work of Ashby in his own quest for a set theory of homeostasis. He drew from Sommerhoff's *coenetic variables* [25] that stem from the Greek meaning for common. This enables concurrent restriction of variety, dictating that those trajectories of the system converge on an ensuing occurrence. Sommerhoff had termed this '*directive correlation*' [25]. In the process of disturbing environmental circumstances, the coenetic variable evokes a response that converges on the adaptive outcome. These notions provide a rigorous mathematical formulation of equilibrium and coordination.

Figure 6.3 illustrates that within the context of the previous genetically modified system scenario, relating to the open-bounded VCS [3, 24]: *E* represents the set of *Environment* points within an open boundary. This contains 8-element arrays reflecting the system state, defined as values *e* a subset of the superset *E* of the possible environmental values.

It can thus be stated that e is a subset of E, that is, a set of possible environment points at a particular time. Within the superset *D* the set of environmental disturbances, is contained the subset *d* of disturbances, representing the set of values of coenetic variables. They cause in the environment, values *e* of the environment's *E* set of possible values, dictating that the effect of *D* on *E* is everywhere defined due to the environment being omnipresent and thus always active. The effect is single valued, because Sommerhoff [25] dictated that the environment could not undertake two actions at once. Disturbance *d* occurs at time t^{n-1} and denotes the level of unsortedness within *e* arrays. In this context, it can be stated that the *D*-*E* relation defines a mapping φ of *D* into *E* and that therefore subset $e=\varphi$ (*d*), at time t^n dictating that φ will produce *e.*

The set *F*, of elements *f*, is located topologically within the VCS S1 management unit and contains regulator values. The VCS operation unit is representative of the sorting system. The system will apply directive correlation [25] to determine the optimal response, thus dictating that $f= \rho$ *(d)*. The *D*-*F* relation consequently defines a mapping ρ of *D* into *F*. The disturbance variable *d* thus evokes *e* and *f* responses φ *(d)* and ρ *(d),* this interaction produces a final outcome at time t^{n+1}. This mapping corresponds to $\psi(E\text{x}F)$ into *Z*, whereby the sort algorithm assumes the role of S_2, to exhibit homeostatic-like behaviour via enabling the VCS to decide the most efficient and viable response to the open-bounded environmental situation. This is achieved by algorithmic hot-swapping permitting systemic adaption.

The superset *Z* contains the set of outcomes ψ *(ExF),* with the subset *G* encompassing those *good* outcomes in *Z* that satisfy the focal condition via appliance of the sort algorithm. There follow key examples from the subscript design grammar model, of the relationships between the VCS systems: from the subscript design grammar model, of the relationships between the VCS systems:

6.2.2 System One (S_1) Case Study VCS Model of the Relationship between the Systems (Subscripts) of the Open Environment System:

Notations:

S : a system$\{_{0,1,2,3,4,5}\}$

S : a system recursion$\{^{0,n,N}\}$

i : a system identifier

A : a system set $\{a_1, a_2, a_3, a_4, \ldots_i\} \mid a_i$ *is atomic*

M : a system Management Unit

O : a system Operation Unit

E : a system Environment

t : CurrentTime parameter$|t^{+1}$*:FutureTime*$|t^{-1}$*: PastTime*

Binary Relations:

$A \Rightarrow B$: If A is true, B is also true

Binary Operators:

$A \cup B$: The set containing all of those elements within A and B

$A \cap B$: The set containing those common elements of A and B

$A - B$: The set that results when B is subtracted from A

$A \not\subset B$: The set A is not a subset of set B

Within the VCS design grammar model, Figure 6.3 exhibits the application of directive correlation [25], now defined as being shown by ρ in respect of D, φ, ψ, and G. The VCS research incorporated a set D of disturbances d, the set of values of the coenetic variables that affect the values e of the set E of possible environmental values. The sort algorithm assumes the role of S_2: Bubble sort, Shell sort, Quick sort and Merge sort. It is shown how Algorithmic hot swapping, when embedded with an extended VCS design grammar model can facilitate theoretical adaptive capability. The regulation can accommodate variety via disturbance D, yet noting that

Hilgartner and Randolph [126] stated that D should be manageable for the agent, the coenetic variable dictates values and environmental conditions, D being able to represent both as the two do not vary independently.

The VCS infrastructure thus exhibits abstract co-ordination, integration, and regulation by a relation between this coenetic variable and the response, such that the outcome of the two is the achievement of a focal condition dictated by the sort algorithm. This work concludes that without this regulatory response the values at the focal condition would be more widely scattered. The open environment [3, 24] is continuously acting, as does the sort algorithm to allow self-organization. The effect of D on E is therefore necessarily ubiquitously defined within the design grammar syntax, thus a D-E relation defines a mapping, say φ of D into E. Figure 6.3 shows disturbance d occurring at time t^{n-1}, and φ produces e, where e = φ (d) at time t^{n}. The system is specified by a set F of elements f with its behaviour and response to d specifying a mapping ρ of D into F. For directive correlation [25] to articulate homeostasis, a mapping ρ, or how the system reacts through genetic modification, must bear some special relation to φ. This is established when disturbance d has evoked the e and f responses φ (d) and ρ (d) respectively. The two values interact, giving a final outcome at time t^{n+1}. It is depicted how its' interface must correspond to a mapping ψ of ExF into Z, where Z is the set of possible outcomes when E and F range uncorrelated over all their values. Algorithmic hot-swapping dictates that Z must be ψ (ExF) and within Z is the subset, G, of good outcomes satisfying the focal condition. The discernible time parameters engender a sense of self, ergo viability by facilitating modelling of internal systemic capabilities, complying with Ashby's requisite variety concept.

$$S^{0}_{1,i<\infty} \Rightarrow A \qquad \textbf{(6.6)}$$

Equation 6.6 shows how the atomic level VCS S_1 maps to the atomic set.

The VCS design grammar model here represents S_1 as a metaphor, with Figure 6.3 illustrating the set theory application of directive correlation [25], now defined as being shown by ρ in respect of D, φ, ψ, and G.

$$S^{n}_{1,i<\infty} \Rightarrow$$
$$\left(\left(\left(\forall d \in D^{n}\right)^{t^{n-1}}\right) : \psi\left(\left\langle \phi\left(d^{n-1}_{1,i<\infty}\right), \rho\left(d^{n-1}_{1,i<\infty}\right)\right\rangle^{t^{n}}\right)\right) \in \left(G^{n}\right)^{t^{n+1}} \qquad \textbf{(6.7)}$$

Equations 6.7 and 6.8 depict the notion of how, when unioned with its associated S_2, S_1 is created; autopoietically spawning a recursion.

$$S^{N}_{1,i<\infty} \Rightarrow$$
$$\left(\left(\left(\forall d^{N-1}_{1,i<\infty} \in D^{N-1}_{1,i<\infty}\right)^{t^{n-1}}\right) : \psi\left(\left\langle\phi(d^{N-1}_{1,i<\infty}), \rho(d^{N-1}_{1,i<\infty})\right\rangle^{t^{n}}\right)\right) \in$$
$$\left(G^{N}_{1,i<\infty}\right)^{t^{n+1}} \Rightarrow$$
$$\left(\left(M^{N}_{1,i<\infty} \cup O^{N}_{1,i<\infty} \cup E^{N}_{1,i<\infty} \cup \left(\left(E^{N(t)}_{4,i<\infty} \cup E^{N(t+1)}_{4,i<\infty}\right) - \left(E^{N(t)}_{4,i<\infty} \cap E^{N(t+1)}_{4,i<\infty}\right)\right)\right) \supset \left(S^{N}_{3,i<\infty} \cup S^{N}_{2,i<\infty} \cup S^{N}_{1,i<\infty}\right)\right) \qquad \textbf{(6.8)}$$

The latter equations show how the VCS research incorporated a set D of disturbances d, the set of values of the coenetic variables that affect the values e of the set E of possible environmental values.

In the open environment case study [3, 24], the sort algorithm assumes the role of S_2: Bubble sort, Shell sort, Quick sort and Merge sort, demonstrating how Algorithmic hot swapping, when embedded with an extended VCS design grammar model can facilitate theoretical adaptive capability. The regulation can accommodate variety via disturbance D, the coenetic variable dictating its values and the environmental conditions. D can represent both, as the two do not vary independently. The VCS architecture thus exhibits co-ordination, integration, and regulation in abstract form, by a relation between this coenetic variable and the response, such that the outcome of the two is the achievement of a focal condition dictated by the sort algorithm. It is concluded that without this regulatory response the values at the focal condition would be more widely scattered. As the open environment is continuously acting, so does the sort algorithm, enabling self-organization. The effect of D on E is therefore necessarily ubiquitously defined within the design grammar syntax, thus the D-E relation defines a mapping, say φ of D into E.

Disturbance d occurs at time t^{n-1}, and φ produces e, where e = φ (d) at time t^n. The system is specified by a set F of elements f with its behaviour and response to d, specifying a mapping ρ of D into F. For directive correlation [25] promoting the homeostatic concept to be shown, the mapping ρ, or how the system reacts through genetic modification, must bear some special relation to φ. This is established when disturbance d has evoked the e and f responses φ (d) and ρ (d) respectively, these two values interact to give some final outcome at time t^{n+1}. The study shoes how its' interaction must correspond to a mapping ψ, of ExF into Z, where Z is the set of possible outcomes when E and F range uncorrelated over all their values. Algorithmic hot-swapping dictates that Z must be ψ (ExF) and within Z is the subset, G, of good outcomes satisfying the focal condition.

The discernible time parameters engender a notional sense of self, ergo viability by facilitating modelling of internal systemic capabilities. This complies with the Ashbian requisite variety concept that every good regulator of a system must be a model of that system its internal variety aspiring to match that of its' environment.

6.2.3 System One (S_1) Case Study VCS Model of the Relationship between the Recursive Levels (Superscripts) of the Open Environment System:

<u>Notations:</u>

S : a system$\{_{0,1,2,3,4,5}\}$

S : a system recursion$\{^{0,n,N}\}$

i : a system identifier

A : a system set $\{a_1,a_2,a_3,a_4,\ldots_i\}\mid a_i$ *is atomic*

M : a system Management Unit

O : a system Operation Unit

E : a system Environment

t : CurrentTime parameter$|t^{+1}$*:FutureTime*$|t^{-1}$*: PastTime*

Binary Relations:

$A \Rightarrow B$: If A is true, B is also true

Binary Operators:

$A \cup B$: The set containing all of those elements within A and B

$A \cap B$: The set containing those common elements of A and B

$A - B$: The set that results when B is subtracted from A

$A \not\subset B$: The set A is not a subset of set B

$$\left\{ S^{0}_{1,i<\infty} \Rightarrow A \right\} \Rightarrow$$

(6.9)

The equation 6.9 reflects the isomorphism apparent within the VCS, at the atomic level of recursion. S_1 at this point maps to the atomic set.

$$\left(\begin{array}{l} \left(S^{n}_{1,i<\infty} \supset \bigcup_{1\le k<\infty} S^{n-1}_{k,i<\infty} \right) \Rightarrow \\ \left(\left(\left(\left\{ \forall d^{n-1}_{1,i<\infty} \in D^{n-1}_{1,i<\infty} \right\}^{t^{n-1}} \right) : \left\{ \gamma \langle \phi(d^{n-1}_{1,i<\infty}), \rho(d^{n-1}_{1,i<\infty}) \rangle^{t^{n}} \right\} \right) \\ \in \left\{ G^{n}_{1,i<\infty} \right\}^{t^{n+1}} \right) \\ \mathrm{For\ N} \ge \mathrm{n} > 0 \end{array} \right)$$

(6.10)

The equation 6.10 depicts in a short form syntax, that all recursive levels of S_1 will be subject to the directive correlation [25] formulae, thus promoting multi-agent coherence and atomic recursion, so lessening redundancy.

This interrelation of the environmental and system elements appears conducive to homeostatic-like sorting by the genetic algorithm. A set of outcomes, including a good subset satisfying the focal condition could then be autonomically mapped.

Innovatively, this implies that the system in this context has the aptitude to self-organize and crucially, derive an appropriate response to an open-bounded environmental region whilst continuing to search for more optimized responses.

Hence theoretically, no explicit human agent intervention is necessary in order to develop a correct sorting algorithm. This offers the capacity for the VCS to autonomically derive a portfolio of algorithms, optimized for a respective environmental area, so enhancing the sorting performance of a universal algorithm. The repertoire of responses could potentially be tailored to adhere to Aulin's Law of Requisite Knowledge [122] i.e. used appropriately within the system, and feasibly adopt Holland's Learning Classifier System (LCS) [120] approach to develop such directed knowledge.

Conclusion

This case study [3, 24] presents a conceptual, open-bounded hybrid Viable Computer System (VCS) architecture based upon Beer's cybernetic VSM and a previous genetically modified system [69]. The design grammar model innovates a hybrid VCS architectural representation of the VSM. The set-theoretical framework defines research specifics, i.e. systems and their environments via algorithmic hot-swapping. Crucially, S_1 represents a metaphor for homeostasis within the design grammar Model. The syntax reflects a set of variables indicating the system and environmental state. The integration of a hot-swapping sort algorithm into the topology exhibits the concept via modification of key elements of Sommerhoff's directive correlation [25] tenet and Ashby's goal directedness [22] and Requisite

Variety principle. It provided an unexpectedly good fit to the previous research method.

These concepts, fused with the algebraic design grammar model offer opportunity for temporal and portable modelling. Scope lies in the complexity-reducing inherent recursion and the self-organizing autopoiesis and homeostasis properties implied by extending the set theory syntax and the adapted VCS topology. The system may theoretically generate its own set of responses via algorithmic hot-swapping. This arguably negates the need for human agent intervention, so endorsing the VCS concept.

The VCS temporal parameters were superimposed, enabling response to environmental stimulus post time t^{n-1}, enabling intrinsic reaction whilst forecasting. Homeostatic [112] and autopoietic [113, 114] approaches were applied to generate a referential self-model of the internal systemic capabilities t^{n} and a model reflecting the required world situation t^{n+1}, so autonomically addressing environmental factors via feedback control. The VCS closed environment case study and conclusions, are followed by a detailing of the open environment case study with example, novel identities. The adapted open-bounded VCS architecture illustrates how Sommerhoff's:

'directive correlation' [25]

qua Ashby's:

'goal directedness' [22]

notion has been grafted onto the existing design grammar syntax, in the context of a previous genetically modified system scenario. A précis ensues, of the conclusions from the VCS case study research. The power of Beer's

VSM has been exploited to extend the autonomic computing ideal by drawing from the following properties:-

Fractal-like inter-recursive recursion

Autopoiesis

Homeostasis

Incorporation of the environment into the system

Internal models

Applied within the context of a previous genetically modified system. This results in an extensible structural design grammar as a basis for viable autonomic software systems. The integration of a hot-swapping sort algorithm into the topology exhibits the concept via modification of key elements of Sommerhoff's directive correlation [25] tenet and Ashby's goal directedness [22] and Requisite Variety principles. The technique provided a surprisingly good fit to the previous research method, resulting in these concepts, fused with the algebraic design grammar model offering opportunity for temporal and portable modelling. Scope lies in the complexity-reducing inherent recursion and the self-organizing autopoiesis and homeostasis properties, implied by extending the set theory syntax and the adapted VCS topology.

This open-bounded case study exhibits the wider potential, and applicability of, the VCS design grammar model in the context of promoting a hybrid theory of Beers VSM to reduce software system complexity, demonstrating a framework for VCS viability, i.e. its ability to exist in a conceptually open environment. Akin to Wright's fitness landscape [127], and von Foerster's Order from Noise [128] concepts, not all attractors are equal, certain possessing higher or better fitness, i.e. more stable and a greater potential for growth. Research aimed to counter the fact that these dynamics do not probabilistically lead to the overall fittest VCS state as the system must necessarily follow the path of steepest descent thus giving a local minimum of the potential, i.e. that of the local environment, rather than in the global minimum which would represent that of the wider, open-bounded environment.

This study thus advanced the VCS fitness function by examining Sommerhoff's directive correlation [25] and Ashby's goal directedness [22] principle, thereby evolving the design grammar model syntax. It was determined that systemic emergence and viability could be made mutually dependant by incorporating a degree of indeterminism to the existing dynamics via enabling the VCS to make transitions to states other than the locally most fit one reflected within the previous closed [2] environment case study. It was hoped that this injection of noise or random perturbations into the system would force it deviate from its preferred trajectory, ergo promoting viability.

This was achieved by opening-up the previously closed environment, so permitting outside perturbations to be introduced and thus allowing unknown factors that have not previously been incorporated into the state description i.e. intrinsic indeterminacy to influence the system state. As with the closed environment case study [2], this research has been applied to a previous genetically modified system scenario with the S_1 architecture representing a metaphor for the design grammar model The open-bounded environment contains 8 - element arrays of the system state, values e, within set E of possible environmental values Environmental disturbances, set D of coenetic variables d, denote the level of unsortedness within e arrays and the wider environment noise, or Disturbance d, occurs at time t^{n-1}.

The system applies directive correlation [25] to determine the optimal response and Algorithmic hot-swapping enables systemic adaption with the sort algorithm assuming the role of S_2, thereby exhibiting homeostatic-like behaviour. This allowed the system to determine the most efficient response to an environmental situation. The perturbations apparently pushed the VCS upwards towards a higher potential, sufficiently increasing the probability that the VCS could escape from a local minimum and then descend towards a possibly deeper valley. The deeper the valley, the more difficult it will be for a perturbation to make the system leave that valley. It was found that noise paradoxically enables the system to progress from the shallower into the deeper valleys, increasing VCS fitness.

6.3 Chapter Summary

This chapter presents a portfolio of two VCS case studies, relating to a respective closed and open environment system context. The design grammar model innovates a hybrid VCS architectural representation of Beer’s VSM. When applied to a previous genetically-modified system scenario, System One represents a metaphor for homeostasis. The set-theoretical framework defines research specifics, i.e. systems and their environments via algorithmic hot-swapping. Further functions and a set of disturbances are introduced, supplying a potential repertoire of tailored responses to open environmental change. Fundamental to promoting emergence, thus viability is Sommerhoff’s concept of *directive correlation* and Ashby's notion of goal-directedness, i.e. the ability to achieve a goal-state under variations in the environment. Example identities exhibit potential for context-free portability including sets of values of environmental and behavioural variables and a set of outcomes allowing the system to develop an adaptive environmental model of fit responses illustrating temporal and autonomic properties of the VCS concept.

The VCS case studies discuss the state of the art of cybernetics relative to autonomic computing followed by respective reviews of Beer's Viable System Model and a context translation towards that of the VCS. This was followed by examples from the design grammar model.

Both the closed [2] and open environment [3, 24] VCS case study were applied in the context of previous genetically modified system scenario [69]. The design grammar model innovates a hybrid VCS architectural representation of Beer's cybernetic Viable System Model (VSM). The set-theoretical framework defines research specifics, i.e. systems and their environments via algorithmic hot-swapping. S_1 represents a metaphor for homeostasis within the design grammar Model. The syntax reflects a set of variables indicating the system and environmental state. Example identities exhibit potential for context-free portability including sets of values of environmental and behavioral variables and a set of outcomes allowing the system to develop an adaptive environmental model of fit VCS responses.

Progressed VCS architectural representations are depicted in Figures: 6.1 and 6.3, showing recursivity with example identities exhibiting the context-free attribute. Further functions and a set of disturbances are introduced, supplying a potential repertoire of tailored responses to open environmental change. Fundamental to promoting emergence, thus viability is Sommerhoff's concept of directive correlation [25] and Ashby's notion of goal-directedness [22], i.e. the ability to achieve a goal-state under variations in the environment. The system has thereby been shown to potentially generate its own set of responses via algorithmic hot-swapping, arguably negating the need for human agent intervention, endorsing the VCS concept.

Chapter 7

Conclusions and Future Work

Introduction

This VCS research has sought to advance autonomic computing by exploiting its compatibility with established cybernetic bases of enquiry, chiefly Beer's self-governing, recursive VSM topology . Adopting an algebraic approach to exploit the power of this fractal-like recursion, autopoiesis, homeostasis and the fundamental ecological dependence , the research goal of a Viable Computer System, is furthered by progression of the design grammar model [1].

The earlier algebraic set theory identities articulated the relationships between the VSM systems (subscripts) [1], whilst atomic recursion (superscripts) was subsequently depicted by reflecting a VSM holism within each implementation system, or S_1. An internal modelling capability imparts context-free self-awareness [115] and so viability.

7.1 Evaluation

Presented is a conceptual, open-bounded hybrid Viable Computer System (VCS) architecture based upon Beer's VSM and a previous genetically modified system [5, 69]. The integration of a hot-swapping sort algorithm into the topology exhibits the concept via modification of key elements of Sommerhoff's directive correlation [25] tenet and Ashby's [22] directedness and requisite variety principles [14]. The superimposition of these mappings and functions, both syntactically and architecturally, presented a surprisingly good fit. Homeostatic [63], and autopoietic [113, 114] methods allied, so enabling an aptitude to self-model systemic capabilities at time t^{n-1}, in addition to an environmental model at time t^{n}. Notably endorsing portability, the cybernetic modelling technique proposes autonomic management of environmental factors, feedback control being core to the process.

The temporal facet can be more easily understood as part of an encompassing scheme describing the interaction between the VCS or self and the environment in which it is situated. In cybernetics an autonomous system or agent is conceptualized as a control system that seeks to achieve its goals or ideals by initiating the right actions that compensate for the disturbances produced by that environment.

The research determined that the VCS therefore needs to perceive or obtain information about the effects of its actions and the effects of the events happening in the environment. It thus needs to understand how past events t^{n-1} cause other future events t^{n+1}, consequently requiring a current model that allows it to explain and anticipate events t^{n}. The open environment case study attempts to reflect the conceptual components necessary for this in order to accommodate all the fundamental aspects of this control scheme.

Scope lies in the complexity-reducing inherent recursion and the self-organizing autopoiesis and homeostasis properties, implied by extending this set theory syntax and the adapted VCS topology.

The system has the potential to generate its own set of responses via algorithmic hot-swapping, arguably negating the requirement for human agent intervention, thereby endorsing the VCS concept. This extension to the

VCS design grammar model and topology, theoretically addresses the research objective, suggesting capacity for further maturity.

The primary research goal is to understand and define the functions and processes of computer systems, enabling them to sense a desired goal, and act upon this.

The studies to-date foster a means for examining the design and function of such a system for the purpose of making them more efficient and effective, thereby demonstrating proof of the VCS research objective.

7.2 Future Aims and Objectives

This VCS research has the potential to be used to analyze, and ultimately design autonomic systems. This could be realized by expanding the design grammar model to incorporate the functionality of the operators, for example addition and subtraction. This progression would then dictate that it could be used as part of the software development process?

Although not the objective of this thesis, one could argue that theoretically, the VCS model could be related back to the original VSM systems, in order that it may be applied to a context other than computing.

Future research hopes to further allude to Wright's fitness landscape [127], and von Foerster's Order from Noise [128] concepts, underscoring that not all attractors are equal, certain possessing higher or better fitness, i.e. more stable and a greater potential for growth. The stronger the noise the more likely the VCS will be able to escape the relatively shallow valleys, and thus reach a potentially deeper valley.

The current ideal of future research would be to apply the design grammar to discover new theories about the VCS attributes, homeostasis, recursion and other cybernetic properties.

7.3 Contribution of the thesis

The thesis exhibits ongoing research, seeking to progress the state of the art of Viable Computing Systems (VCS) [1], which present a conceptual model characterizing homeostatic self-governance, this is argued here to advance the widely accepted autonomic computing reference models.

The design grammar has been applied as a formal representation, characterizing a set of rules dictating how the VCS elements may be put together. The design grammar model will be capable of automating the design process and rules generated by emerging requirements.

It can be applied to both the analysis and synthesis of the VCS design. The former is useful for determining the design legality, the latter facilitates discovery of faults, indicating reformulation.

Having formerly united biological [63, 116] and cybernetic research, this research formulated a mathematical model [1] of the functionality of Beer's Viable System Model (VSM) . This study progressed to reference Hofstadter [129] and Mandelbrot's recursion concepts [121], and to uniquely model the link between the VSM recursions via a set-theory blueprint, emphasizing the importance of ecological VCS dependence .This provided the basis of an evolving design grammar model, embodied as a VCS.

A closed environment case study ensued [2], relating to algorithmic hot-swapping in a previous genetically modified software system [69]. That case study exhibits the wider potential of the VCS design grammar model in the context of promoting a hybrid theory of Beers VSM to reduce software system complexity.

This research has proposed a mathematical framework for VCS viability, i.e. its ability to exist in a conceptually open environment, realized by adding to fundamental codices of Sommerhoff [25] and Ashby [14, 22].

The same system scenario was adopted, based upon the earlier closed environment case study [2] pertaining-to the hot swapping of sort algorithms [31, 69-71] for the open-bounded case study.

Therein, self-organization and VCS emergence is suggested via the uniform research method. The earlier algebraic set theory identities articulate the relationships between the VSM [1] systems (subscripts), whilst depicting atomic recursion by (superscripts) to reflect a VSM holism within each implementation system, or S1. Potential exists to create an internal model to impart context-free self-awareness [115] and so viability. hope to subsume autonomic computing initiatives towards the VCS [1]; akin to human autonomic systems by cognitive systems, this research conceptualizing open-bounded viability.

Temporal parameters have been modified that were previously adopted by Ashby [22] in his citing of Sommerhoff [25], environmental disturbances occurring at time t^{n-1}, with sorting, directive correlation [25] and algorithmic hot-swapping, at times t^n and t^{n+1} respectively. Superimposition of these mappings and functions, both syntactically and architecturally, presented a surprisingly good fit.

Homeostatic [63], and autopoietic [113, 114] methods allied, so enabling an aptitude to self-model systemic capabilities at time t^{n-1}, in addition to an environmental model at time t^n. Notably endorsing portability, the cybernetic modelling technique proposes autonomic management of environmental factors, feedback control being core to the process. This extension to the VCS design grammar model and resultant topology, theoretically addresses the research objective, suggesting capacity for further maturity.

An open-bounded case study of the VCS has been introduced, the design grammar model innovating a hybrid VCS architectural representation of the VSM System One (S1). When applied to a previous genetically-modified system scenario, System One (S1) which represents a metaphor for homeostasis, with the set-theoretical framework defining research specifics, i.e. systems and their environments via algorithmic hot-swapping. Further functions and a set of disturbances are introduced, supplying a potential repertoire of tailored responses to open environmental change.

Essential to promoting emergence, thus viability is Sommerhoff's concept of directive correlation [25] and Ashby's notion of goal-directedness [22], that is, the ability to achieve a goal-state under variations in the environment. Example identities exhibit potential for context-free portability including sets of values of environmental and behavioural variables and a set of outcomes allowing the system to develop an adaptive environmental model of fit responses illustrating temporal and autonomic properties of the VCS concept.

It is believed that the VCS algebraic and architectural representation of a theory of homeostasis is both novel and progressive within the state of the art. The essence of this research is has been the modelling the following key features of Beer's VSM towards progressing the state of the art of autonomic computing, toward the novel VCS. The design grammar model represents a bi-perspective of the VCS, formulated via set theory syntax. Specifically, the VSM is a multi-agent system constituting a quintuple hierarchy of systems numbered 1 to 5, with one intermittent auditing sub-system of S3*.

Topologically, the VSM is largely comprised of interacting homeostatic loops, exhibiting communication and control of organizational operations and their respective environments. This method allows viability to become an emergent property of the system and crucial to the research is that this integrated management of the parts promotes each S1 sub-system as an autonomic whole within a closed metaboundary.

By adopting an algebraic approach, the set theory model can be easily applied to architectural illustrations and facilitate definition of not only novel topologies, but also uniform extension to the syntax as demonstrated within the case studies. The research approach also enables conceptual portability to differing computing scenarios.

7.4 Future Work

Potential undoubtedly exists to further refine the VCS, and to greater expand upon the existing design grammar. The current utilization of a mathematical analogy is purely a vehicle for expressing research concepts and ideas. Potential exists to narrow the research foci towards S_4 of the topology and further delineate the system-environment ecology by increased application of Sommerhoff's coenetic variable [25] principles.

The notion of a ‘coenetic variable’ explains the delimitation of the variety of environmental circumstances and of apparently regulatory responses. Coenetic variables, simultaneously delimit variety so that trajectories of the system converge on to a subsequent occurrence. Sommerhoff’s identification of this ‘directive correlation’ [25], could thus permit advanced modelling of variables to affect both the system-in-focus and it’s’ environment.

Longer-Term: it may be necessary to expand the design grammar, allowing systemic configuration of the architecture that is Deletion, Insertion and Addition of elements. This would require research into what lies behind the concatenation of elements that is uncovering how operators could be represented by the syntax. This would require accommodating the range of different elements that need to be added and their differing contexts.

The definition of addition, subtraction, multiplication and replication may be necessary to fully enable autopoiesis. This self-generation of higher levels and/or parts will arguably not necessitate division operations.

Future scope thus lies in attempting to comprehend how this interconnection of two separate things is effectively achieved to become one and thus allow potential to expand the grammar to allow systemic configuration via deletion and insertion of topological parts. Narrowing the focus by delineating the models' system-environment ecology, the design grammar applies Sommerhoff's coenetic variable principles [25], permitting notional modelling of variables to affect both the system-in-focus and its environment. The longer-term objective is to continue towards development of an implementable VCS.

7.5 Chapter Summary

Essentially and uniquely, this research has led to the complete construction of a first-stage, yet system-wide context-free design grammar model of the innovative VCS.

Fundamentally, it is a formal algebraic representation incorporating production rules and a set of symbols encompassing a vocabulary comprised of atomic elements of the language.

A research objective has been to attempt to learn from Beer's cybernetic Viable System Model (VSM) and in so doing replicate key aspects within the design grammar model, such as the complexity-reducing property of fractal-like recursion, the existence of two internal models; promoting a sense of *self* , whilst allowing for emergence and retention of viability via engendering a temporal facet. Correspondingly, autonomy versus governance is facilitated by the existence of the models.

Whereas managerial cybernetics concerns human systems, this research advances into computer systems via the VCS. Development of an algebraic set theory model, informed by Beer's VSM, reflects Mandelbrotian, fractal-type recursive [29] geometry inherent within the topology. Modelling of the system is endeavoured by addressing its' architectural specification.

Appearing ripe to support a potentially context-free design-grammar, germane to diverse computing scenarios, the notional internal representation imbues the system with aspects of self-awareness via a self/non-self distinction.

The algebraic design grammar is the main modelling formalism intended to realize the VCS specification articulating a focus on addressing how to achieve the requirements, rather than what they are. Importantly, reflecting the context-shift from human to software systems, this research has extended the number of VCS S_1's per recursive level to potentially infinite. This is because Beer stipulated that within the human organization, seven is the greatest number of people who could effectively work together. This concept is obviously obsolete when transposed to software agency.

The set theory model pushes forwards the boundaries of the state of the art by embodying a structural representation of conceptual homeostasis via production of a formalism that allows possible future transposition into the software demonstrator. The design grammar has thereby been applied as a prescribed representation, characterizing a set of rules dictating how the VCS elements may be put together. The design grammar model will be capable of theoretically automating the design process and rules generated by emerging requirements.

The concept can be applied to both the analysis and synthesis of the VCS design; the former is useful for determining the design legality, whereas the latter could facilitate discovery of faults to indicate reformulation.

This research has innovatively progressed the state of the art via the creation of a first-stage, yet system-wide context-free design grammar analogue that models the Viable Computer System. The formal algebraic representation incorporates production rules and a set of symbols encompassing a vocabulary comprised of atomic elements of the language. The VCS design grammar model echoes the VSM's indefinite recursivity, at a post-atomic level, having no specific starting point or initial conditions.

The VCS research ideal is to diminish the patina of legacy system syndrome through reducing the inherent complexity of traditional software design methods via negating the requirement for human agent intervention. The onus is thus placed upon the design grammar apparatus and associated novel topologies, for assuming the role of quasi-human agency. The VCS hence perpetuates an inherent, recursive, self-referential, modelling method, engendering the working and recurring of functions at each systemic level.

The promotion of emergence and viability of the system in-focus is achieved via the fundamental system-environment interplay. This intra and extra-systemic communication, in conjunction with acknowledgment of the environmental requirements, fosters strong homomorphism. This devolvement of autonomy to localized sub-systems endorses meritocratic autonomy, qua homeostasis; versus archaic centralized governance.

The VCS thereby exhibits and surpasses Horn's self-CHOP constituent elements [108], so demonstrating proof of research concept.

References

1. Thompson, R.J., Laws, A.G., Reilly, D.J., Taleb-Bendiab, A., Llewellyn-Jones, D. *Viable Computer Systems: A Cybernetic Approach to Autonomic Computing*. in *IEEE SMC UK&RI 6th Conference on Cybernetic Systems*. 2007. University College Dublin Republic of Ireland: IEEE.

2. Thompson, R.J., Laws A.G, Reilly, D.J, Taleb-Bendiab, A., and D. Llewellyn-Jones. *An Algorithmic Cybernetic Case Study of Viable Computing Systems Within a Genetically Modified Software Environment*. in *8th IEEE International Conference on Cybernetic Intelligent Systems*. 2009. University of Birmingham, United Kingdom: IEEE.

3. Thompson, R.J., Laws A.G, Reilly D.J, Taleb-Bendiab, A. and Llewellyn-Jones, D, *A Set Theory Analysis of Ecological Dependence amid an Agent Infrastructure in Beer's Viable System Model through Viable Computer Systems.* Journal of Emerging Technologies in Web Intelligence (JETWI) In Press, 2011. **3**(3).

4. Thompson, R.J., Laws A.G,Reilly, D.J,Taleb-Bendiab A and Llewellyn-Jones, D. *An Innovative Set Theory Interpretation of Beer's Cybernetic Viable System Model: Aspirant of Viable Computer Systems*. in *The International ACM Conference on Management of Emergent Digital Ecosystems (MEDES 2009)*. 2009. INSA de Lyon, Lyon, France.

5. Thompson, R.J., et al. *An Open-Bounded Cybernetic Case Study of Viable Computing Systems: Applying Directive Correlation to an Algorithmic Hot-Swapping Scenario*. in *9th IEEE International Conference on Cybernetic Intelligent Systems*. 2010. University of Reading, Reading, UK: United Kingdom and Republic of Ireland Chapter of the IEEE Systems, Man and Cybernetics Society.

6. Bennett, K., *Legacy Systems, Coping With Success.* IEEE Software, 1995. **12**(1): p. 19-23.

7. Horn IBM, P., in *Address to the National Academy of Engineers*. 2001, IBM.

8. Horn, P. *Autonomic Computing: IBM's Perspective on the State of Information Technology*. March 2001.

9. Cannon, W.B., *Wisdom of the Body*. 1932, New York: Norton.

10. Ashby, W.R., *Design for a Brain*. 2 ed. 1954: Wiley, J.

11. Ashby, W.R., *An Introduction to Cybernetics*. 1956, London: Chapman and Hall.

12. Ashby, W.R., *An Introduction to Cybernetics*. 2 ed. 1957, London: Chapman and Hall.

13. Ashby, W.R., *Design for a Brain*. Second ed. 1960: Chapman and Hall.

14. Ashby, W.R., *Homeostasis*. in Encyclopedia of Biological Sciences, ed. P. Gray. 1961: Reinhold Publishing Corp..: New York, NY.

15. Ashby, W.R., *An Introduction to Cybernetics.* 4 ed, ed. U. Paperbacks. Vol. 4. 1964, London: Methuen & Co. Ltd. 295.

16. Beer, S., *Decision And Control :The Meaning Of Operational Research And Management Cybernetics*. 1 ed. 1966: Wiley, John & Sons.

17. Conant, R.C. and W.R. Ashby, *Conant-Ashby Theorem.* International Journal of System Science, 1970. **1**(2): p. 89-97.

18. Beer, S., *Brain of the Firm*, ed. J.S. Wiley. 1972, Bath: Pitman Press.

19. Beer, S., *Brain Of The Firm*, ed. J.W.a. Sons. 1981, Bath: Pitman Press.

20. Beer, S., *Diagnosing the System for Organizations*. 1985.

21. Bernard, C., *An Introduction to the Study of Experimental Medicine*. 2 ed. 1865, New York: Courier Dover Publications.

22. Ashby, W.R., *The Set Theory of Mechanism and Homeostasis.* Biological Computer Laboratory, Technical Report, 1962. **4.7**: p. 44.

23. Laws, A., A. Taleb-Bendiab, and S. Wade, *Genetically Modified Software: Realizing Viable Autonomic Agency*, in *Innovative Concepts for Autonomic and Agent-Based Systems*. 2006. p. 184-196.

24. Thompson, R.J., Laws A.G., Reilly D.J.,Taleb-Bendiab A., Llewellyn-Jones, D. *An Open-Bounded Cybernetic Case Study of Viable Computing Systems: Applying Directive Correlation to an Algorithmic Hot-Swapping Scenario*. in *9th IEEE International Conference on Cybernetic Intelligent Systems*. 2010. University of Reading, United Kingdom: IEEE.

25. Sommerhoff, G., *Analytical Biology*. 1950: Oxford University Press.

26. Moore, G.E., *Cramming more components onto integrated circuits.* Electronics, 1965. **38**(8,).

27. Moore, G.E., *Cramming More Components Onto Integrated Circuits.* Proceedings of the IEEE, Jan 1998. **86**(1): p. 82 - 85.

28. Marai, V., *The Build Master: Microsoft's Software Configuration Management Best Practices*. ISBN:10: 0321332059. 2006: Addison-Wesley Microsoft Technology.

29. *The National Programme for IT in the NHS: Progress since 2006*, Published on 27 January 2009 by authority of the House of Commons Public Accounts Committee London U.K. p. HC 153 [Incorporating HC 737-i, Session 2007–08].

30. Ewusi-Mensah, K., *Software Development Failures, Anatomy of Abandoned Projects*. 2003, ISBN:9780262050722.: MIT Press.

31. Laws, A.G., Taleb-Bendiab, A., Wade, S. J., *Towards a Viable Reference Architecture for Multi-Agent Supported Holonic Manufacturing Systems.* Journal of Applied Systems Studies, 2001. **2**(1).

32. Lehman, M.M., *Programs, Lifecycles, and Laws of Software Evolution.* Proceedings of the IEEE, 1980. **8**(69).

33. Horn, P., *Autonomic Computing: IBM's Perspective on the State of Information Technology*. March 2001.

34. IBM, *eLiza: self-manging servers*. 2001: http://www-.ibmcom/servers/eserver/introducing/eliza/.

35. *Workshop on Self-healing systems archive*. 2002. Charleston, South Carolina: Publisher ACM New York, NY, USA.

36. *1st international workshop on autonomic computing systems*. 2003. Prague, Czech Republic: Proceedings of IEEE DEXA 2003 workshops.

37. Wiener, N., *Cybernetics or Control and Communication in the Animal and the Machine*. 1948, New York: Wiley, John and Sons.

38. McCulloch, W. and W. Pitts, *A Logical Calculus of the Ideas Immanent in Nervous Activity.* Bulletin of Mathematical Biophysics, 1943. **5**: p. 115-133.

39. Beer, S., *On the Nature of Models:Let Us Now Praise Famous Men and Women, Too (from Warren McCulloch to Candace Pert).* Informing Science, 1999. **2**(3): p. 69-82.

40. Pask, G., *An Approach to Cybernetics*. 1961, New York: Harper and Brothers.

41. Maturana, H.R., *Autopoiesis*. Autopoiesis: A Theory of the Living, ed. N. Holland. 1981, New York: M. Zeleny.

42. Rosen, R., *Some realizations of (M,R) systems and their interpretation.* J. Math. Biophys, 1971. **33**: p. 303-319.

43. Rosen, R., *Life Itself*. 1991, New York.: Columbia University Press.

44. Varela, F., *Principles of Biological Autonomy*, ed. N. Holland. 1979, New york.

45. Mingers, J., *Self-Producing Systems*. 1995, New York.: Plenum Press.

46. Pask, G., *Consciousness.* Journal of Cybernetics, 1979. **9**(.): p. 211-258.

47. Cariani, P., *Adaptivity and emergence in organisms and devices.* World Futures, 1991. **32**: p. 111-132.

48. Pattee, H.H., *Simulations, realizations, and theories of life*. Studies in the Science of Complexity ed. Artificial Life, ed. C.G. Langton. Vol. VI. 1989, Santa Fe Institute: Addison-Wesley.

49. Pattee, H.H., *Cell psychology: an evolutionary view of the symbol-matter problem.* Cognition and Brain Theory, 1982. **5**: p. 325-341.

50. Maturana, H., *The biology of cognition*. Autopoiesis and Cognition, ed. H.M.a.F. Varela. 1970, Dordrecht, Holland: D. Reidel.

51. Maturana, H. and F. Varela, *Autopoiesis and Cognition: the Realization of the Living.* Boston Studies in the Philosophy of Science, ed. R.S. Cohen and M.W. Wartofsky. Vol. 42. 1973: Dordecht: D. Reidel Publishing Co.

52. Kampis, G., *Self-Modifying Systems in Biology and Cognitive Science*, ed. P. Press. 1991, Oxford.

53. Rosen, R., *Anticipatory Systems*, ed. P. Press. 1985, Oxford.

54. Turing, A., *On computable numbers, with an application to the Entscheidungsproblem.* Proceedings of the London Mathematical Society, 1936. **2**(42): p. 230–65.

55. von Neumann, J., *Re-evaluation of the problems of complicated automata -- problems of hierarchy and evolution.* Reprinted in Papers of John von Neumann on Computing and Computer Theory, ed. W. Aspray and A. Burks. 1948, Cambridge, Massachusetts: MIT Press. 477-490.

56. Jones, R.V., *The Wizard War: British Scientific Intelligence 1939–1945*, ed. M.G. Coward. 1978, New York, NY, USA.

57. Kirby, M. and R. Capey, *The Air Defence of Great Britain 1920–1940: An Operational Research Perspective.* Journal of the Operational Research Society, 1997. **48**: p. 555–568.

58. Kirby, M.W., *Operational Research in War and Peace: The British Experience from the 1930s to 1970*. 1954 ed, ed. J.F.a.T. McCloskey, F.N. 2003, London, UK: Imperial College Press.

59. von Neumann, J., *Letter to Norbert Wiener from John von Neumann.* Proceedings of the Norbert Wiener Centenary Congress, 1994, eds. V. Mandrekar and P. R. Masani, Proceedings of Symposia in Applied Mathematics, 1997. **52**: p. 506–512.

60. Neumann, J.v., *First Draft of a Report on the EDVAC 1945.* IEEE Annals of the History of Computing, 1993. **15**(4): p. 27-75,.

61. Heims, S.J., *Constructing a Social Science for Postwar America: The Cybernetics Group, 1946 - 1953*. 1993: MIT Press.

62. Ashby, W.R., *Design for a Brain*. 1952.

63. Cannon, W.B., *The Wisdom of the Body*. 1932, London.

64. Laws, A.G., Wade, S. J., *Towards A Model Of Viable Software Maintenance Management - VSMM.* 2000.

65. Laddaga, R., *DARPA Broad Agency Announcement on Self-Adaptive Software,*. 1997, http://www.darpa.mil/ito/Solicitations/PIP_9812.html.

66. Bratman, M.E., D.J. Israel, and e. al., *Plans and Resource-Bounded Practical Reasoning.* Computational Intelligence, 1988. **4**(4): p. 349-355.

67. Lehman, M.M., *Programs, Life Cycles, and Laws of Software Evolution.* Proceedings of the IEEE, 1980. **68**(9): p. 1060-1076.

68. Espejo, R. and R. Harnden, *The Viable System Model Interpretations and Applications of Stafford Beer's VSM*. 1989: John Wiley & Sons.

69. Laws, A.G., Taleb-Bendiab, A., Wade, S. J., *Genetically Modified Software: Realizing Viable Autonomic Agency.* Innovative Concepts for Autonomic and Agent-Based Systems, 2005: p. 184-196.

70. Laws A.G., et al., *From Wetware to Software.* Springer Verlag, 2003.

71. Laws, A.G., A. Taleb-Bendiab, and S.J. Wade. *Bridging The Gap: The Cybernetics of Auto-Adaptive Software Systems*. in *Software Evolution and Evolutionary Computation Symposium Abstracts*. 2002. University of Hertfordshire, Hatfield, U.K.: EPSRC Network on Evolvability in Biology & software Systems.

72. Kephart, J.O. and D.M. Chess, *The Vision of Autonomic Computing.* Computer, 2003. **36**(1).

73. Herring, C. and S. Kaplan. *The Viable System Model for Software*. in *4th World Multiconference on Systemics, Cybernetics and Informatics (SCI'2000)*. 2000. Orlando, Florida.

74. Stoyanov, E.A., M.A. Wischy, and D. Roller. *Cybernetics and General Systems Theory (GST) Principles for Autonomic Computing Design*. in *Second International Conference On Autonomic Computing*. 2005: IEEE.

75. Bustard, D.W., et al., *Autonomic system design based on the integrated use of SSM and VSM.* Artificial Intelligence Review, 2006. **Volume 25, Number 4**(0269-2821 (Print) 1573-7462 (Online)): p. 313-327.

76. Lehman, M.M., *The Programming Process*. 1969, IBM: Yorktown Heights, New York, U.S.A. p. 10594.

77. Lehman, M.M. *Feedback in the Software Evolution Process*. in *Keynote Address, CSR Eleventh Annual Workshop on Software Evolution: Models and Metrics*. 1994. Dublin.

78. Shannon, C.E., *"A Mathematical Theory of Communication".* Bell System Technical Journal, 1948. **27**: p. 379-423, 623-656, July, October,.

79. Shannon, C.E. and W. Weaver, *The Mathematical Theory of Communication*. 1963: University of Illinois Press.

80. Von Foerster, H., *Theorem 2, 251; Order from Noise*, in *in Der Wissensnavigator. Das Lexikon der Zukunft*, *Arthur P. Schmidt (Hg.)*, D. Verlagsanstalt., Editor. 1999, Kybernetik: Stuttgart.

81. Lehman, M.M. and L. Belady, *IBM Research Report RC2722*. 1969, IBM: New York.

82. Belady LA and Lehman MM, *An Introduction to Program Growth Dynamics*, in *Statistical Computer Performance Evaluation*, W. Freiburger, Editor. 1972, Academic Press: New York. p. 503-511.

83. Lehman, M.M. *Process models, process programs, programming support*. in *Proceedings of the 9th international conference on Software Engineering*. 1987. Monterey, California, United States: ISBN:0-89791-216-0.

84. Lehman, M.M., et al., *Metrics and Laws of Software Evolution - The Nineties View*. 1997.

85. Lehman, M.M. *Quantitative Studies in Software Evolution - From OS/360 to FEAST Position Paper*. in *Workshop on Process Modelling and Empirical Studies of Software Evolution*. 1997. Boston, MA.

86. Turski, W., M, *Reference Model for Smooth Growth of Software Systems*, University of Warsaw: Warsaw.

87. Lehman, M.M., *Laws of Software Evolution Revisited. Laws of Software Evolution Revisited.* Proc. EWSPT96, 1996: p. 108-124.

88. Nuseibeh, B. and S. Easterbrook, *' Requirements Engineering: A Roadmap'*, in *The Future of Software Engineering*, A. Finkelstein, Editor. 2000, ACM Press.

89. Lehman, M., Stenning V, and Turski WM, *Another Look at Software Design Methodology*. 1984, Dept. of Comp., Imp. Col., London: Rep. 83/13.

90. Naur, P., Randell, B. *Software Engineering*. 1969. Garmisch: Scientific Affairs Division, NATO (North Atlantic Treaty Organization).

91. Beer, S., *The Viable System Model: Its Provenance, Development, Methodology and Pathology*. Journal of the Operational Research Society, 1984. **35**(1): p. 7-25.

92. Beer, S., *The Heart of Enterprise: the Managerial Cybernetics of Organization*. 1 ed. 1979: Wiley.

93. Harnden, R., *Technology for Enabling: The Implications for Management Science of Hermeneutics of Distinction.*, ed. U.o. Aston. 1989, Birmingham.

94. Maturana, H. and F. Varela, *The Tree of Knowledge: The Biological Roots of Human Understanding*. 1987, Boston, MA: Shambhala Press.

95. Bratman, M.E., D.J. Israel, and M. Pollock, E, *Plans and Resource-Bounded Practical Reasoning.* Computational Intelligence, 1988. **4**(4): p. 349-355.

96. von Bertalanffy, L., *A quantitative theory of organic growth.* Human Biology, 1938. **10**: p. 181-213.

97. von Bertalanffy, L., *An outline of General Systems Theory*. British Journal for the Philosophy of Science, 1950. **1**: p. 139-164.

98. Ainsworth, P., *The Millenium Bug*. JULY 1996, IEE REVIEW.

99. Taleb-Bendiab, A., Bustard, D. W., Sterritt, R., Laws, A. G., Keenan, F., *Model-based self-managing systems engineering.* Sixteenth International Workshop on Database and Expert Systems Applications, Proceedings, 2005: p. 155-159.

100. Randles M., et al., *Adjustable Deliberation of Self-managing Systems.* ECBS, 2005. **12th**(IEEE International Conference and Workshops on the Engineering of Computer-Based Systems): p. 449-456.

101. Miseldine, P., *Cloud Architecture Schematic*. 2004, http://www.cms.livjm.ac.uk/2nrich/deliverables.asp.

102. Miseldine, P., *JBel (Neptune) Application Development Guide*. 2004, http://www.cms.livjm.ac.uk/2nrich/deliverables.asp.

103. Karsai, G. and J. Sztipanovits, *A model-based approach to self-adaptive software.* Intelligent Systems and their Applications, IEEE, 1999. **14**(3): p. 46-53.

104. Lehman, M.M.M., *Re: 'Programs, Life Cycles, and Laws of Software Evolution' - Your 1980 Paper*, R.J. Thompson, Editor. 2004: Liverpool, United Kingdom.

105. Laws, A.G., et al., *From Wetware to Software: A Cybernetic Perspective of Self-adaptive Software*, in *Self-Adaptive Software: Applications*. 2003. p. 341-357.

106. Lehman, M.M. and L.A. Belady, *Program evolution : processes of software change*, ed. A. Press. 1985, San Diego, California.

107. Cook, S., et al., *Evolution in software systems: foundations of the SPE classification scheme.* Journal of Software Maintenance and Evolution-Research and Practice, 2006. **18**(1): p. 1-35.

108. Horn, P., *Autonomic Computing: IBM's Perspective on the State of Information Technology*. 2001.

109. Bratman, M.E., D.J. Israel, and M.E. Pollack, *Plans and resource-bounded practical reasoning.* Computational Intelligence, 1988. **4**(4): p. 349-355.

110. Stoyanov, E.A., et al. *Distributed Software Maintenance Using an Autonomic System Management Approach based on the Viable System Model*. in *20th International Conference on Advanced Information Networking and Applications (AINA 2006)*. 2006. Silicon Valley, California,USA: IEEE Computer Society.

111. Bourbaki, N., *Theorie des Ensembles*. Fascicule de Resultats. Vol. 1141. 1951: Hermann et Cie, Paris.

112. Horn, P. and F. West, *Introduction.* IBM Systems Journal, 2005. **44**(3).

113. Varela, F., H. Maturana, and R. Uribe, *Autopoiesis: the Organization of Living Systems, its Characterization and a Model.* Biosystems. **5**: p. 187–196.

114. Maturana, H.R., F.J. Varela, and S. Beer, *Autopoietic systems a characterization of the living organization*. Expanded ed. ed. 1975: Biological Computer Laboratory, Univ. of Illinois.

115. Macfarlane Burnet, F. and F. Fenner, *The Production of Antibodies*. 2 ed. 1949, Melbourne: Macmillan and Co.

116. Bernard, C., of the Académie Française, *Introduction à l'étude de la médecine expérimentale.* 1865, Paris.

117. Stenning, V. *On the Role of an Environment*. in *Proc. Int. Conf. on Software Eng.* 1987. Monterey CA: ACM Press.

118. Maruyama, M., *The Second Cybernetics Deviation-Amplifying Mutual Causal Processes.* American Scientist, 1963. **51**: p. 164-179.

119. von Foerster, H., *Cybernetics of Cybernetics*. 1974, University of Illinois, Urbana,: Illinois.

120. Holland, J.H., *Adaptation in Natural and Artificial Systems*. 1992, Cambridge, Mass: The MIT Press.

121. Mandelbrot, B., *Stochastic models for the Earth's relief, the shape and the fractal dimension of the coastlines, and the number-area rule for islands.* Proceedings of the National Academy of Sciences of the United States of America, 1975, October. **72**(10): p. 3825-8.

122. Aulin, A., *The Cybernetic Laws of Social Progress: Towards a Critical Social Philosophy and a Criticism of Marxism*. 1982, Oxford: Pergamon Press.

123. Ziv, J. and A. Lempel, *A Universal Algorithm for Sequential Data Compression.* IEEE Transactions on Information Theory, 1977. **23**(3): p. 337-343.

124. Weil, A., *L'integration dans les groupes topologiques et ses applications.* Actual. Sci. Ind., 1940. **869**(Hermann et Cie, Paris).

125. Beer, S., The Journal of the Operational Research Society, 1984. **35**(1): p. pp. 7-25.

126. Hilgartner, C.A. and J.F. Randolph, *Psycho-logics : An axiomatic system describing human behavior.* Journal of Theoretical Biology. **23**(2): p. 285-338.

127. Wright, S. *The Roles of Mutation, Inbreeding, Crossbreeding and Selection in Evolution*. in *VI International Congress of Genetics*. 1932. Ithaca, New York.

128. Von Foerster, H., *On Self-Organizing Systems and their Environments*, in *Self-Organizing Systems*. 1960, Pergamon Press. p. 31-50.

129. Hofstadter, D.R., 1945-, *Gödel, Escher, Bach: An Eternal Golden Braid*. 1979, New York: Basic Books.