ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/266230932

# COMBINATORIAL SCHEMA AND THEIR COHOMOLOGY; APPLICATIONS IN NATURAL AND ORGANIZATIONAL SYSTEMS

**Article**

CITATIONS
0

READS
11

**2 authors**, including:



Andrew Macfarlane
The University of Queensland
**5** PUBLICATIONS **6** CITATIONS

SEE PROFILE

All content following this page was uploaded by Andrew Macfarlane on 23 July 2016.

The user has requested enhancement of the downloaded file.

# COMBINATORIAL SCHEMA AND THEIR COHOMOLOGY; APPLICATIONS IN NATURAL AND ORGANIZATIONAL SYSTEMS.

ANDREW MACFARLANE
IntegratedExpetise.org
3a Jeta Gardens
27 Clarendon Avenue
Bethania
Brisbane
Australia 4205.
combinatorial.schema@gmail.com

February 25, 2011

**Abstract**

This paper defines a new category of combinatorial modules. These are overlapping collections of small categories indexed by a directed set. They are specialized in applications to non-commuting rings acting on overlapping modules. Each combinatorial module is associated with an Abelian category of group valued sheaves which is its combinatorial scheme. An adaptation of Čech cohomology is used to analyze the structure of applications. The paper focuses on the applications of these ideas to supply chains, testing software and ecosystems. As these systems display a variety of phenomena or behaviors the area of interest is in the means of validating scenarios or assertions of capability. Cohomology is used to analyze techniques of validating scenarios in supply chains and assertions in systems. The characterization of scenarios in terms of when they can be satisfied are a start of understanding the many forms of failure (called fragmentation theory) that a supply chain or system can undergo. The intention is to motivate further development of combinatorial schema through applications that that aggregate or combine materials , validated data, concepts or nutrients along a directed graph.

*Keywords*: Combinatorial modules; sheaves; cohomology; supply chains.

2010 Mathematics Subject Classification: 08A99, 18F20, 18G60, 93A30, 92D40.

## 1 Introduction

The development of combinatorial schema was motivated by questions in supply chains of manufacturing organizations. The mathematics was sufficiently abstract that this could be applied to more general systems including education where ideas

are assembled from simpler ideas, taxation where a government accrues revenue from a directed network of different tax types to fulfill a budget. It was clear that it had applications in information systems testing. It also has applications in ecosystem and metabolism. This will be explained once the mathematical machinery is in place. In this paper the applications are limited to supply chains, system testing and ecosystems. This will give the flavor and scope of the concepts.

Combinatorial schema are categories of sheaves on combinatorial modules. An abstract, but illustrative view of a combinatorial module is that of a directed set, $A$, of finitely generated non-commutative rings acting on state spaces. For each $a \in A$, $W_a$ is a ring and $Y_a$ state space with action

$$\alpha_a : W_a \times Y_a \to Y_a.$$

If $b$ is covered by $a$ in $A$ $Y_a \cap Y_b \neq \emptyset$ so the partial order in $A$ controls an exchange of information among the state spaces and produces a form of dynamics. Thus combinatorial modules form an algebraic category analogous to the patching of an atlas of Euclidean spaces to create a manifold. Properties of these systems are framed in terms of sheaves of schedules. A schedule is an analogue of polynomials of actions that continue through subsets of $A$. For a given combinatorial module the combinatorial schema is the Abelian category generated by group valued sheaves of schedules. The structure and the nature of these sheaves are analyzed by an adaptation of Čech cohomology.

# Part I
# Combinatorial Modules and Schema

## 2 Definitions and concepts

Let $Cat$ be a 2-category of small categories with pullbacks.

A combinatorial module is a 2-category $\underline{A} = (A, Tpc : A \to Cat))$ where $A$ is a finite directed graph and so a category. $Tpc(a)$ is the "topic $a$". Each category $Tpc(a)$ will have a set of initial objects that are not the range of any maps in $Tpc(a)$ and a set of final objects that are not the domain of any maps in $Tpc(a)$. We use $e$, $e'$, $e_1$, $e_2$,... , $e_n$ for objects in a category $Tpc(a)$ and $w$, $w'$, $w_1$, $w_2$,.., $w_n$ for maps in $Tpc(a)$.

1. For any pair $a_1$, $a_2 \in A$ there is one or more $a$ that covers them. The set of predecessors of $a$, $\{b \mid b \prec a\}$ is the set $S(a)(= S^1(a))$ and the successors of $a$ are $S^+(a)(= S^{+1}(a))$.

2. $Tpc(\{a_1, a_2\})$ is defined whenever $a_1 \in S(a_2)$. There are monos from $j_i : Tpc(a_i) \rightarrowtail Tpc(\{a_1, a_2\})$ such that the pullback $\tilde{\beta}(a_2, a_1)$ exists.

3. If $e$ is an object in the pullback subcategory $Tpc(a_1) \leftarrow \tilde{\beta}(a_2, a_1) \to Tpc(a_2)$ then $e$ is a final object in $Tpc(a_1)$ or an initial object in $Tpc(a_2)$.

$\tilde{\beta}(a_2, a_1)$ is the overlap of $Tpc(a_1)$ and $Tpc(a_2)$ and is the way the combinatorial module connects to become a single structure. The elements of $A$ are written opposite

to the ordering because in the applications studied below $\tilde{\beta}$ is associated with a "requesting" function $\beta(a_2, a_1) : Obj(Tpc(a_2)) \to Obj(Tpc(a_1))$ which request from $a_1$ "components" for objects in $Tpc(a_2)$.

The following notation is used throughout this paper.

$$S^{n+1}(a) = \{b \mid \exists b' \in S^n(a) \wedge b \in S(b')\}$$

and

$$S^{+(n+1)}(a) = \{b \mid \exists b'' \in S^{+n}(a) \wedge b \in S^+(b'')\}.$$

A subset $U$ of $A$ is combinatorial submodule if $\{a\} \cup S^2(a)$ is in $U$ so is $S(a)$ and $(U, Tpc : U \to Cat)$ is a combinatorial module.

A combinatorial module is called a response system if for all $a$ in $A$ there is a map

$$L : Maps(Tpc(a)) \to \mathbb{N}$$

such that $L(w_2 \circ w_2) = L(w_2) + L(w_1)$. This is a lead-time function. (It can derived from weightings or measures on $Obj(Tpc(a))$. For $a \in A$ "lead-time through $a$" is $max\{L(w) \mid w \in Map(Tpc(a)\}$. A response system in which the objects in $Tpc(a)$ have a state in an ordered set and most $w \in W(a)$ decrease the state in their domain while adding to the state in their range is a *supply chain* if there is a "replenishment logic" that allows each topic to be used indefinitely.

An affine morphism $(A_1, Tpc : A_1 \to Cat)) \to (A_2, Tpc : A_2 \to Cat)$ is a family of functors $f : A_1 \to A_2$ and $Tpc(f) : Cat \to Cat$ such that for each $b \in A_2$ $f^{-1}(b)$ is an combinatorial submodule and the following diagram of functors commutes

$$\begin{array}{ccc} A_1 & \xrightarrow{f} & A_2 \\ \downarrow{\scriptstyle Tpc} & & \downarrow{\scriptstyle Tpc} \\ Cat & \xrightarrow{Tpc(f)} & Cat. \end{array}$$

**Definition.** An affine map $f$ is a logistic map if the following diagram commutes.

$$\begin{array}{ccc} Tpc(f^{-1}(b)) & \xrightarrow{Tpc(f)} & Tpc(f(b)) \\ \downarrow{\scriptstyle L} & & \downarrow{\scriptstyle L} \\ \mathbb{N} & \xrightarrow{\geq} & \mathbb{N}. \end{array}$$

That is lead-time through $b$ is no greater than lead-time through $f^{-1}(b)$.

Clearly combinatorial modules form a category $Comb$. Unless otherwise stated all morphism will be logistic.

If $e$ is an object of $Tpc(a)$ a *stream* to $e$ is a set

$$e_n \xrightarrow{w_n} e_{n-1} \xrightarrow{w_{n-1}} e_{n-2} \xrightarrow{w_{n-2}} ... \xrightarrow{w} e$$

A stream can extend beyond $Tpc(a)$ into $S^n(a)$.

A stream is *full* if none of the included maps is a composition.

Let $T$ be a set of intervals in $\mathbb{N}$. $T$ is a set of time slots. For time slots $t_1$, $t_2$ write $t_1 > t_2$ if the least element of $t_1$ is not less than the greatest element of $t_2$. Let $\xi \subseteq Map(U) = \bigcup_{a \in U} Map(Tpc(a))$ be a set of full streams that can intersect.

**Definition** $\xi$ is a *schedule* if there is a function $s : \xi \to T$ such that

1. If $e'' \xrightarrow{w'} e' \xrightarrow{w} e$ is part of a stream in $\xi$ then $s(w) > s(w')$
2. If $\{a_1, a_2, a_3\} \subseteq U$ and $a_1 \in S(a_2)$ and $a_2 \in S(a_3)$ with $\xi \cap Tpc(a_3) \neq \emptyset$ and $\xi \cap Tpc(a_1) \neq \emptyset$ then $\xi \cap Tpc(a_2) \neq \emptyset$
3. Furthermore for a response system $s(w)$ is an interval $[n, n + L(w)]$

A schedules can be thought of as a network of maps flowing over part of $A$. If $s(w_1) \cap s(w_2)$ has more than two elements then $w_1$ and $w_2$ can be occurring at the same time. Even if $dom(w_2) \cap range(w_1) \neq \emptyset$ they can be in different streams.

A cover $\mathcal{U}$ of $\underline{A}$ is a set of combinatorial submodules in which every element of $A$ is in at least one element of $\mathcal{U}$.

Let $Sch(U)$ be the set of schedules defineable on $U$. If $V \subset U$ there is clearly a restriction of $Sch(U) \to Sch(V)$ so we can define a presheaf $Sch$ of schedules.

**Proposition 2.1.** *Sch is a sheaf.*

**Proof**: Let $\xi_i$ be defined on $U_i$ such that $\xi_i = \xi_j$ on $U_i \cap U_j$ (the 1-cocycle condition) then as each is well defined on $U_i \setminus (U_i \cap U_j)$ we can define a $\xi$ on $U_i \cup U_j$ that restricts to $\xi_i$ and $\xi_j$. In this way a schedule $\xi$ can be defined on $U = \underset{i}{\cup} U_i$ such that $\xi \mid U_i = \xi_i$. □

If $w \in Map(Tpc(a))$ then we represent this as an action on $Obj(Tpc(a))$ by extending to all of $Obj(Tpc(a))$ as the identity map for $e$ not in the domain of $w$. This extension gives a non-commutative monoid [35] $act(a)$. For actions $w_1$ and $w_2$ composition $w_1 \circ w_2$ has its usual meaning. Include a do-nothing action, $id$. The rules for simultaneous actions are incorporated in $act(a)$. Let $w_1 \uplus w_2$ denote actions that can take place at the same time. Then $(w_1 \uplus w_2)$ composed, either before or after, with another action obeys a distributivity law. This provides the ability to express the idea that a repeated action is either idempotent or can produce more output: $w^2 = w$ or $w^2 = w \uplus w$. If $w_1$ and $w_2$ are independent then $w_1 \circ w_2 = w_1 \circ w_2 = w_1 \uplus w_2$. $W(a)$ is identified with the ring $\mathbb{Z}(act(a))$, the $act(a)$ algebra over the ring of integers $\mathbb{Z}$. In $W(a)$ $\uplus$ becomes "+". This is a standard construction [25] [1].

**Proposition 2.2.** *A schedule in U is a polynomial of maps in* $\underset{a \in U}{\cup} W(a)$ *where U is such that if* $b < a$ *are in U then there is a chain* $b \prec a_1 \prec ... \prec a$ *in U.*

**Proof**. Each independent stream $\alpha : e_n \xrightarrow{w_n} e_{n-1} \xrightarrow{w_{n-1}} e_{n-2} \xrightarrow{w_{n-2}} ... \xrightarrow{w} e$ contributes $w \circ ... \circ w_{n-1} \circ w_n$. If $\alpha' : e'_m \xrightarrow{w'_m} ... \xrightarrow{w'} e_j$ is another stream that terminates at $e_j$ in $\alpha$ then we have $w \circ ... \circ w_j(w_{j+1} \circ w_{j+2} \circ ... \circ w_n + w' \circ ... \circ w_m)$. Distributivity will give two streams identified in the time slots. In this way all streams contribute to the polynomial associated with $\xi$. □

Identifying schedules with their associated polynomials allows us to add them.

**Proposition 2.3.** *Sch is a group valued sheaf.*

**Proof**: Given $\xi_1$ and $\xi_2$ in $Sch(a)$ then $\xi_i$, $i = 1, 2$ is a set of elements of $W(a)$ mapped to $t \in T$. Elements $w_i \in \xi_i$ in independent streams give $s(w_1 + w_2) = s(w_1) \cup s(w_2)$ which requires that $T$ allow unions of intervals. Note that if $w_2 \circ w_2$ is defined then $s(w_2) > s(w_1)$. Addition therefore derives from the commutation of

[1] $W$ for workflow; in other contexts $W(a)$ can be interpreted as routings, processes.

$$\begin{array}{ccc} W(a) \times W(a) & \xrightarrow{+} & W(a) \\ {\scriptstyle s\times s}\downarrow & & \downarrow{\scriptstyle s} \\ T \times T & \xrightarrow{+} & T. \end{array}$$

□

**Theorem 2.1.** *Sch is a functor from Comb and logistics morphisms to the category of Abelian groups Ab.*

**Proof:** Writing $Sch(\underline{A})$ for the sheaf of schedules on $\underline{A}$ and $Sch(A)$ for the global sections then we already have defined the functor on objects.

For a logistic morphism $f : \underline{A}_1 \to \underline{A}_2$ and for every $U \in \mathcal{U}$ with $f^{-1}(f(U)) = U$, $\xi \in Sch(\underline{A}_1)(U)$ define $Sch(f)(\xi)$ as

$$\begin{array}{ccc} \xi \subseteq Map(Tpc(U)) & \xrightarrow{f} & Tpc(f)(\xi) \subseteq Map(Tpc(f(U))) \\ {\scriptstyle s}\downarrow & & \downarrow{\scriptstyle s} \\ \mathbb{N} & = & \mathbb{N}. \end{array}$$

wherein $Tpc(f)(\xi) =_{defn} Sch(f)(\xi) \subseteq Map(Tpc(f)(U))$. As the total duration of $\xi$ = range of $s(\xi)$ is the same as $Sch(f)(\xi)$ then the time through $U$ is the same as for $f(U)$. Then $Sch(\underline{A}_1) \xrightarrow{Sch(f)} Sch(\underline{A}_2)$ is a well defined homomorphism. □

Let $\mathcal{P}$ be an arbitrary group valued subsheaf of $Sch$. An adjustment to a schedule in $\mathcal{P}$ is any map that changes a schedule $\xi \in \mathcal{P}$ to another schedule $\xi' \in \mathcal{P}$ in which the start and end objects in the streams of $\xi'$ and the starting time slots of $\xi'$ are the same as for $\xi$. The set of adjustments has a natural groupoid structure: suppose $\mathcal{G}$ is a class of adjustments such as ones that preserve total schedule duration then they can compose and they have an inverse. However adjustments to one schedule need not apply to another. We can defined a groupoid valued sheaf $\mathcal{G}$ that acts linearly on $\mathcal{P}$. The most important application will be the colimit of the diagrams $\mathcal{P} \xrightarrow{g} \mathcal{P}$ where $g \in \mathcal{G}$ is a section of homomorphisms. Call this sheaf $J(\mathcal{G})\mathcal{P}$. An element of $J(\mathcal{G})\mathcal{P}$ is an equivalence class of schedules denoted $\mathcal{G}(\xi)$. There is a natural group valued sheaf homomorphism from $\mathcal{P}$ to $J(\mathcal{G})\mathcal{P}$. This motivates the following definition.

**Definition** The Combinatorial Schema of $\underline{A}$ is the Abelian category generated by the category of group valued sheaves of schedules. The morphisms of this category are constructed from logistic morphisms that induce sheaf homomorphisms over $\underline{A}$. Denote this category as $Comb(\underline{A})$.

This category has direct sums, finite limits and colimits and so contains the sheaf $J(\mathcal{G})\mathcal{P}$.

# 3 Associated model categories.

In this paper the applications of combinatorial modules are first-order predicate calculus systems. What sets of propositions can be proved from the properties of a combinatorial module? Schedules form a class of propositions that expressed much of the capability or behavior of the combinatorial module. What can be proved from knowledge of the schedules is of particular interest.

Let $\pi$ be a class of propositions that can be interpreted in $\underline{A}$. These are assumed to be stated in a first-order language that is meaningful in the relevant $Tpc(a)$ and in $\beta(a, a')$. Interpretation in each $Tpc(a)$ and $\beta(a, a')$, $a, a' \in U$ allows interpretation in $Tpc(U)$. Define $mdl(\pi)$ to be a sheaf of models of $\pi$ defined for all $U$ of the $A$ cover $\mathcal{U}$. This is required to have the properties:

1. $P \subseteq \pi$ has a model $m(P)$ in a subset $U \in \mathcal{U}$ if values in $P$ can be assigned in $Tpc(U)$ so that as a logical model $\forall p \in P\ m(P) \models p$. If $P$ is $\pi$ restricted to $U$ write this as $m(P) \in mdl(\pi)(U)$.

2. $mdl(\pi)$ is a presheaf; $m \in mdl(\pi)(U)$ and $V \subset U$ then there is a well defined model $m \mid V \in mdl(\pi)(V)$.

3. $mdl$ is a sheaf. This constrains the sets of $\pi$ to be consistent if they overlap. That is, if $m_i \in mdl(\pi)(U_i)$ so that $m_i = m_j$ on $U_i \cap U_j$ for every pair $i$, $j$ then there is an overall model $m \in mdl(\pi)(\cup U_i)$ that restricts to the individual $m_i \in mdl(\pi)(U_i)$.

4. $mdl(\pi)(U)$ is a category with morphisms being extensions of models. That is $m_1 \rightarrowtail m_2$ if and only if $m_1 \models p \Rightarrow m_2 \models p$.

There are properties of $Sch$ and various subsheaves $\mathcal{P}$ that can be stated in terms of sets of propositions $\pi$ and their model sheaves. A schedule $\xi$ is a set of propositions of the form $w(e_1, e_2, ..., e_n) = e$ and (by implication) $\beta(a, b)(e) = e'$ hence is a conjunction of basic propositions (that is, equations). This requires a logical relation between schedules and propositions. To this end define a relation $\mathfrak{L} : \mathcal{P} \times mdl(\pi) \rightarrow \{True, False\}$ such that $\mathfrak{L}(U)(\xi, m)$ is true or false depending on whether $\xi(U)$ holds the relationship defined by $\mathfrak{L}$ with $m(U)$. For example whether $m(U) \models \xi$. As $Comb(\underline{A})$ is a category of sheaves $\mathfrak{L}$ is to be a sheaf. Thus it is required to have the properties:

1. $V \subset U \Rightarrow \mathfrak{L}(U)(\xi, m) \Rightarrow \mathfrak{L}(V)(\xi \mid V, m \mid V)$ and

2. $U = \cup U_i$ then provided $\mathfrak{L}(U_i)(\xi_i, m_i)$ remain true on the intersections $\mathfrak{L}(U_i \cap U_j)$ there are extensions $\xi$ and $m$ such that $\mathfrak{L}(U)(\xi, m)$ and $\xi$ and $m$ restrict to the $\xi_i$ and $m_i$ respectively .

3. For any $U$, $(\xi_1 \subseteq \xi_2) \in \mathcal{P}(U) \Rightarrow (\mathfrak{L}(U)(\xi_1, m) \Rightarrow \mathfrak{L}(U)(\xi_2, m))$ and

4. $(m_2 \rightarrowtail m_1) \in mdl(U) \Rightarrow (\mathfrak{L}(U)(\xi, m_1) \Rightarrow \mathfrak{L}(U)(\xi, m_2))$

As $\mathcal{P} \xleftarrow{proj_1} \mathfrak{L} \xrightarrow{proj_2} mdl$ are sheaf maps, maps $\phi : \mathcal{P} \rightarrow \mathfrak{L}$ such that $proj_1 \circ \phi = 1$ are sections as are $\psi : mdl \rightarrow \mathfrak{L}$ with $proj_2 \circ \psi = 1$

For a section $m \in \Gamma(dom(\phi), mdl)$ define $X(\phi, m)$ as the pullback

$$\begin{array}{ccc} X(\phi, m) & \longrightarrow & (dom(\phi) \subseteq A) \\ \downarrow{\scriptstyle incl} & & \downarrow{\scriptstyle m} \\ \mathcal{P} & \xrightarrow{proj_2 \circ \phi} & mdl. \end{array}$$

Each $X(\phi, m)$ is a subsheaf of $\mathcal{P}$. It is closed under addition and therefore can be associated with the least group valued subsheaf of $\mathcal{P}$ containing it. In this way it is associated with an element in $Comb(\underline{A})$. Assuming there are only a finite number of schedules (as will be the case in the selected applications) define

$$X(m) =_{defn} \underset{\phi}{colim}(X(\phi, m) \rightarrowtail \mathcal{P}).$$

**Definition** Let $\mathcal{P}_{\mathfrak{L}}$ be the sheaf defined by $\mathcal{P}_{\mathfrak{L}}(U) =_{defn} \{X(m) \mid m \in mdl(\pi)(U)\}$.

Note that $\mathcal{P}_{\mathfrak{L}}$ has a "null" or zero element corresponding to the $m$ for which there is no $\xi \in \mathcal{P}$ for which $\mathfrak{L}(\xi, m)$ is true.

**Lemma 3.1.** *$\mathcal{P}_{\mathfrak{L}}$ is a sheaf.*

Indeed the sections of this sheaf are defined by a direct correspondence $\mathfrak{X} : mdl \rightarrow \mathcal{P}_{\mathfrak{L}}$ given by $\mathfrak{X}(m) = X(m)$. Where $X(m)$ is not zero the sheaf properties derive from those of $mdl$ and $\mathfrak{L}$.

Swapping the roles of $\mathcal{P}$ and $mdl$ define $Y(\psi, m)$ as the pullback

$$\begin{array}{ccc} Y(\psi,m) & \longrightarrow & (dom(\psi) \subseteq A) \\ \downarrow{\scriptstyle incl} & & \downarrow{\scriptstyle \xi} \\ mdl & \xrightarrow{proj_1 \circ \psi} & \mathcal{P}. \end{array}$$

Define $Y(\xi)$ as the colimit of the subsheaves $Y(\psi, m)$. In this case assume that finite sets of the subsheaves of $mdl$ have colimits then $Y(\xi)$ is all the models for which $\mathfrak{L}(\xi, m)(dom(\xi))$ is true. It is enough that partially ordered finite sets of subsheaves have least upper bounds.

**Definition**. Define $mdl_{\mathfrak{L}}(U) =_{defn} \{Y(\xi) \mid \xi \in \mathcal{P}(U)\}$.

By the same argument as for $\mathcal{P}_{\mathfrak{L}}$, $mdl_{\mathfrak{L}}$ is a sheaf.

**Theorem 3.2.** *If for each $U \in \mathcal{U}$ the partial orders on $\mathcal{P}_{\mathfrak{L}}(U)$ and $mdl_{\mathfrak{L}}$ are complete then there exist maps $\mathfrak{j} : \mathcal{P}_{\mathfrak{L}} \rightarrow mdl_{\mathfrak{L}}$ and $\mathfrak{f} : mdl_{\mathfrak{L}} \rightarrow \mathcal{P}_{\mathfrak{L}}$ such that $\mathfrak{f} \circ \mathfrak{j} = 1_{mdl_{\mathfrak{L}}}$ and $\mathfrak{j} \circ \mathfrak{f} = 1_{\mathcal{P}_{\mathfrak{L}}}$.*

**Proof**.

Define $\mathfrak{j} : \mathcal{P}_{\mathfrak{L}} \rightarrow mdl_{\mathfrak{L}}$ by $\mathfrak{j}(X(m)) = \underset{\xi \in X(m)}{\cap} Y(\xi)$ as this is the most specific class of models associated with $m$. For this to be well defined it should correspond to a unique $Y(\underline{\xi})$. By the complete partial order condition $\underline{\xi}$ exists. For each model $m$ in $Y(\xi)$ $\mathfrak{L}(\underline{\xi}, m)$ is true and is true for all submodels of $m$. To pick out an $m$ that characterizes $\xi$ we want to define $\mathfrak{f} : mdl_{\mathfrak{L}} \rightarrow \mathcal{P}_{\mathfrak{L}}$ by $\mathfrak{f}(Y(\xi)) = \underset{m \in Y(\xi)}{\cap} X(m)$. Again as the order in $\mathcal{P}_{\mathfrak{L}}(U)$ is complete there is a $\bar{m}$ so that $\mathfrak{f}(Y(\xi)) = \underset{m \in Y(\xi)}{\cap} X(m) = X(\bar{m})$. $\bar{m}$ is the largest model for which $\mathfrak{L}(\xi, m)$ is true for the relevant $\xi$.

From these definitions and the properties of $\mathfrak{L}$

1. $m_1 < m_2$ (that is $(m_1 \rightarrowtail m_2)$ implies $X(m_2) \subseteq X(m_1)$ and $\mathfrak{j}(X(m_1)) \subseteq \mathfrak{j}(X(m_2))$.
2. $\xi_1 < \xi_2$ then $Y(\xi_1) \subseteq Y(\xi_2)$ and so $\mathfrak{f}(Y(\xi_2)) \subseteq \mathfrak{f}(Y(\xi_2))$.

Consequently $\mathfrak{j}$ and $\mathfrak{f}$ are order reversing.

Composing these maps, firstly,

$$X(m) \xrightarrow{\mathfrak{j}} Y(\underline{\xi}) \xrightarrow{\mathfrak{f}} \underset{\mathfrak{L}(\underline{\xi}, m')}{\cap} X(m') = X(\bar{m}).$$

If $\xi \in X(m)$ then $\xi$ belongs to all the $Y(\xi)$ determining $\underline{\xi}$ so that $\xi$ is at least as big as $\underline{\xi}$. Hence $\mathfrak{L}(\underline{\xi}, m') \Rightarrow \mathfrak{L}(\xi, m')$ so $\xi$ is in the intersection defining $X(\bar{m})$ so that $X(m) \subseteq X(\bar{m})$. If $\xi' \in X(\bar{m})$ then, as $\mathfrak{L}(\xi', \bar{m})$, $\mathfrak{L}(\xi', m)$ is implied by $\mathfrak{L}(\underline{\xi}, m)$. But $\mathfrak{L}(\underline{\xi}, m)$ so $\mathfrak{L}(\xi', m)$ and $\xi'$ is in $X(m)$. Thus $\mathfrak{f} \circ \mathfrak{j} = 1_{\mathcal{P}_{\mathfrak{L}}}$.

Secondly,

$$Y(\xi) \xrightarrow{\mathfrak{f}} X(\bar{m}) \xrightarrow{\mathfrak{j}} \bigcap_{\mathfrak{L}(\xi', \bar{m})} Y(\xi') = Y(\underline{\xi}).$$

Suppose $m' \in Y(\underline{\xi})$. By definition $\mathfrak{L}(\underline{\xi}, \bar{m})$. As $m' < \bar{m}$ so $\mathfrak{L}(\underline{\xi}, m')$. But $\xi > \underline{\xi}$ which implies $\mathfrak{L}(\xi, m')$ and so $m' \in Y(\xi')$ and therefore $Y(\underline{\xi}) \subseteq Y(\xi)$.

If $m'' \in Y(\xi)$ $m''$ is in all the $Y(\xi')$ whose intersection gives $X(\bar{m})$. In particular $m'' \leq \bar{m}$ and as $\underline{\xi}$ is the least $\xi$ for which $\mathfrak{L}(\xi', \bar{m})$ and so $\mathfrak{L}(\underline{\xi}, m'')$ and $m'' \in Y(\underline{\xi})$ and so $Y(\xi) \subseteq Y(\underline{\xi})$. This gives $Y(\underline{\xi}) = Y(\xi)$ and so $\mathfrak{j} \circ \mathfrak{f} = 1_{mdl_{\mathfrak{L}}}$.

As $\mathfrak{j}$ and $\mathfrak{f}$ are order reversing they give the required duality. □

**Corollary 3.3.** *As set-valued sheafs there is a sheaf map $\mathcal{P}_{\mathfrak{L}}$ to $mdl_{\mathfrak{L}}$ that for each $U$ in the cover is a bijection from $\mathcal{P}_{\mathfrak{L}}(U)$ to $mdl_{\mathfrak{L}}(U)$.*

This theorem is an example of a class of theorems that link objects which we wish to use to describe or specify concepts, describe parts of systems or phenomena with methods by which we verify or learn about them. The sheaf relation $\mathfrak{L}$ is an *epistemological operator* that tells us the nature of the relation between the operational manifestation of a framework, in this case a combinatorial module, and those things that manifest its specifications, its models [2].

# 4 Cohomology

In the following we use the notation $U(i_0, ..., i_n) = U_{i_0} \cap U_{i_1} \cap ... \cap U_{i_n}$ where the $U_{i_k}$ are elements of a cover $\mathcal{U}$ of $A$ [19]. Given a group valued sheaf $\mathcal{P}$ we can define the Čech cohomology on $\mathcal{P}$ giving the groups $\check{H}^n(\mathcal{P})$. As usual denote the cochains by

$$C^n = C^n(\mathcal{P}, \mathcal{U}) = \prod_{i_0, i_1, ..., i_n} \mathcal{P}(U(i_0, ..., i_n)) \tag{1}$$

and define the *coboundary* map $\partial^n : C^n \longrightarrow C^{n+1}$ by setting

$$\partial^n(\underline{\xi})(U(i_0, i_2, ..., i_{n+1})) \tag{2}$$

$$= \sum_{k=0}^{n+1} (-1)^k . \underline{\xi}((i_0, i_1, ..i_{k-1}, i_{k+1}, ..., i_{n+1}) | U(i_0, i_1, ..., i_{n+1})).$$

Define $\check{H}^n(\mathcal{P}) =_{defn} ker(\partial^n)/im(\partial^{n-1} = Z^n/B^n$.

In this paper it is useful to adopt the original geometrical interpretation of cohomology groups. Given $U = \bigcap_{k=0,...,n} U_k$, $\xi_k$ defined on $U_0 \cap U_1 \cap ... \hat{U_k} ... \cap U_n$, $k = 0, 1, ..n$, where $\hat{U}$ denotes omission, then the alternating sum in the definition of a cocycle, whether written as $\xi_0 - \xi_1 + ... + (-1)^n \xi_n = 0$ or $\xi_1 + \xi_3 + ... + \xi_{2k-1} = \xi_0 + \xi_2 + ... + \xi_{2k}$,

[2] A quite separate example of this: a category of spaces can be defined, the ones best defined are those described in terms of sets of roots of polynomials: the varieties and algebraic spaces. The implicit relationship links a class of geometric objects, the analogy of $mdl_{\mathfrak{L}}$ with an effective mode of description, the analogy of $\mathcal{P}_{\mathfrak{L}}$.

is a equation of dependence in the $\mathbb{Z}$ module $\mathcal{P}(U)$. Interpreting $U = U_0 \cap U_1 \cap ... \cap U_n$ as the "interior" of a "space" with "sides" $U_0 \cap U_1 \cap ... \hat{U_k} ... \cap U_n$, a cocyle represents a linear relationship among the $n+1$ schedules. The cocycle generates a cohomology class if it is a relationship on the interior that cannot be rewritten in terms of schedules defined on the sides $U_0 \cap U_1 \cap ... \hat{U_k} ... \cap U_n$. That is, the $\xi_k$ are not coboundaries. If a cocycle can be expressed in terms of differences of other schedules on bigger sets then the cocycle is a set of coboundaries - it does not add to the set of new relationships; the information is already present in fewer intersections. The group $\check{H}^n(\mathcal{P})$ therefore is determined by new, "non derived" $n$-relations that can be located anywhere in $A$. These relationships in schedules can have there origin in equations among the actions and the actions on types. This is an algebraic version of "gluing" in topology. In the case of combinatorial modules "gluing" becomes an algebraic or logical identification of types or actions. Cohomology encodes collections of local effects that can make a difference in the overall phenomena as demonstrated by sheaves of schedules.

Given a logistic morphism $f : \underline{A}_1 \to \underline{A}_2$ and $\mathcal{P}_2$ a sheaf on $\underline{A}_2$ defined for covering $\mathcal{U}_2$, then the pullback of $\mathcal{P}_2$, $f^{-1}(\mathcal{P}_2)$ defined on $f^{-1}(\mathcal{U})$ is a sheaf on $\underline{A}_1$. But it need not be in $Comb(\underline{A})$. For example $U \subset A_1$ is mapped by $f$ to a single element $b \in A_2$. If $\{b\}$ is in $\mathcal{U}$ it is not clear how to interpret a schedule in $\mathcal{P}_2(b)$ to a schedule in $\mathcal{P}_1(U)$. For this reason this paper concentrates on cohomology as a *covariant* functor. In this case $f : \underline{A}_1 \to \underline{A}_2$ is "meet-continuous": for $U_1$, $U_2$,..., $U_n$ all in $\mathcal{U}$, $f(U_1 \cap U_2 \cap ... \cap U_n) = f(U_1) \cap f(U_2) \cap ... \cap f(U_n)$. This ensures that the elements of cochains $C^n(\mathcal{P})$ map appropriately and $Sch(f)$ maps schedules to schedules giving the covariant functor $\check{H}(f) : \check{H}^n(\mathcal{P}_1) \to \check{H}^n(\mathcal{P}_2)$. Requiring $f$ to be meet-continuous is not a serious problem when we come to applications.

The next consideration is the choice of the cover $\mathcal{U}$. This has a significant effect on the cohomology as the following result shows.

**Theorem 4.1.** *Let $\mathcal{U} = \{U_1, U_2, ..., U_N\}$ be a cover for $A$ such that there is number $n_0 < N$ such that any set of $n > n_0$ intersections $U_{i_0} \cap ... \cap U_{i_n}$ is equal to the intersections of a subset of those sets with no more than $n_0$ members. If $\mathcal{P}$ is defined on $\mathcal{U}$ then $H^{n_0+k}(\mathcal{P}) = 0$ for $k \geq 1$.*

**Proof.** This follows as any cocycle on more than $n > n_0 + 1$ intersections is an equation on $n-1$ intersections and hence can be represented as coboundaries. Specifically, let $< \xi_j >$ form a $n$-cocycle. $\xi_j$ is defined on $U_{i_0} \cap ... \hat{U}_{i_j} \cap ... \cap U_{i_n}$. Hence $\Sigma(-1)^j \xi_j = 0$ on $U_{i_0} \cap ... \cap U_{i_n}$. For $n > n_0+1$ we need to show this is a coboundary. If $n > n_0+1$ then for each $j$ there is an $l$ such that $U_{i_0} \cap ... \cap \hat{U}_{i_j} \cap ... \cap U_{i_n} = U_{i_0} \cap ... \cap \hat{U}_{i_j} \cap ... \cap \hat{U}_{i_l} \cap .. \cap U_{i_n}$ so $\mathcal{P}(U_{i_0} \cap ... \cap \hat{U}_{i_j} \cap ... \cap U_{i_n}) = \mathcal{P}(U_{i_0} \cap ... \cap \hat{U}_{i_j} \cap ... \cap \hat{U}_{i_l} \cap ... \cap U_{i_n})$. This means each $\xi_j$ can be written as an alternating sum of schedules defined on fewer intersections of the cover and hence can be expressed as a coboundary. Thus for $n > n_0$ $\check{H}^n(\mathcal{P}) = 0$. □

For this reason we use a "generic" cover: $\mathcal{U}$ is the set of sets $S^{\leq k}(a) = \{a\} \cup \underset{i \leq k}{S}{}^i(a)$ for all $a \in A$ together with their intersections. This is a very redundant cover and many useful covers will have fewer sets.

Under the constraint of $f$ being meet-continuous the usual cohomology properties apply to the Čech cohomology [38, 27]. In particular, for an exact sequence of sheaves the "long exact sequence" holds:

$$... \to \check{H}^n(\mathcal{P}') \to \check{H}^n(\mathcal{P}) \to \check{H}^n(\mathcal{P}'') \to \check{H}^{n+1}(\mathcal{P}') \to ... \tag{3}$$

In the following a combinatorial scheme $(\underline{A}, \mathcal{P})$ is given with a covering $\mathcal{U}$, $U \in \mathcal{U}$ and $\underline{B}$ is a subobject of $\underline{A}$. Define presheaves $\mathcal{P}^B$ and $\mathcal{P}_B$ by

$$\mathcal{P}^B(U) = \mathcal{P}(U) \qquad U \cap B = \emptyset \tag{4}$$
$$\mathcal{P}^B(U) = 0 \qquad U \cap B \neq \emptyset \tag{5}$$

and

$$\mathcal{P}_B(U) = \mathcal{P}(U) \qquad U \cap B \neq \emptyset \tag{6}$$
$$\mathcal{P}_B(U) = 0 \qquad U \cap B = \emptyset \tag{7}$$

If we call the closure of $B$ the set $\underset{b \in B}{\cup}(S(b) \cup \{b\} \cup S^+(b))$ then $\mathcal{P}^B$ zeroes everything inside this closure while $\mathcal{P}^B$ zeroes everything outside the closure.

These definitions allow the definition of a short exact sequence of presheaves defined on $\underline{A}$:

$$0 \to \mathcal{P}^B \to \mathcal{P} \to \mathcal{P}_B \to 0.$$

Making the definitions

$$\check{H}^n(\underline{B}, \mathcal{P}) = \check{H}^n(\underline{A}, \mathcal{P}_B) \tag{8}$$
$$\check{H}^n(\underline{A}, \underline{B}, \mathcal{P}) = \check{H}^n(\underline{A}, \mathcal{P}^B). \tag{9}$$

allows us to apply the long exact sequence to get relative cohomology:

$$... \to \check{H}^{n-1}(\underline{B}) \to \check{H}^n(\underline{A}, \underline{B}) \to \check{H}^n(\underline{A}) \to \check{H}^{n+1}(\underline{A}, \underline{B}) \to ...$$

(Here the sheaves are implicit).

The Excision theorem also has a version here. In topology given $B_1 \subset B \subset A$ the Excision theorem requires that the closure $B_1$ is in the interior of $B$. For combinatorial schema with $B_1 \subset B \subset A$ we require that all $b \in B_1$ $S(b) \cup S^+(b) \subseteq B$ then

$$\check{H}^n(\underline{A} \setminus \underline{B}_1, \underline{B} \setminus \underline{B}_1) = \check{H}^n(\underline{A}, \underline{B}). \tag{10}$$

# Part II
# Applications

The applications that motivated the development of combinatorial modules arose from the indexed set of action algebras as described in the introduction. This corresponds to interpreting $Tpc(a)$ as having objects a set $Typ(a)$ and actions a non-commutative ring of actions $\mathbb{Z}(act(a))$. In this case, if $e \in Typ(a)$ and $w \in act(a)$ then $w : e \longmapsto w(e)$ is a map in $Tpc(a)$. Alternatively we can take the objects of $Tpc(a)$ to be the single object $Typ(a)$ which will usually be a $\mathbb{Z}$ module with the set of morphisms being $W(a)$. This will be explain in detail below.

# 5 Supply Chains.

Supply chains and the integration of expertise were the first motivating examples. Supply chains marshal the material (types) and expertise (actions) required to make something. They can be defined very widely indeed. The material for a computer notebook can include the gold that is used in transistor contacts, silicon, iron, coal, oil or uranium for manufacturing components and generate the electricity for the manufacturing plants. Equally we can have a supply chain of taxation revenue with each $a$ being a business sector and $b \prec a$ indicating economic dependence. Parallel to any supply chain is a response system, the network of expertise required for the supplying and manufacturing industries involved. These supply chains are of paramount importance to modern economies. In this paper a supply chain is a response systems with replenishment (see below) while an expertise chain is a response system.

Supply chains have an extensive history of mathematical modeling. There are many themes or perspectives. Optimization and simulation in particular have an extensive literature and recent surveys give a guide to that literature. The surveys of [6, 36, 16, 37, 31, 23] cover optimization and simulation models. A related perspective considers the design of supply chains as a social structure of obligations and is motivated by Game Theory which studies the effects of contractual arrangements as well as the procurements strategies [8]. Optimal solutions, simulation and contractual games will give structures to aim for. Attaining and keeping the optimal cases can be extremely difficult especially when managing very large numbers of variables such as supplier prices or raw material transport costs all of which are required to hold the optimum. An early concern was over stocking inventory that arises in response to changing demand [15]. Studies have found that information sharing reduces the fluctuations that arise in response to changing demand [22, 9]. This has prompted the evolution from the classical single firm materials-management approach used in manufacturing [29] to the coordination of supply chains of many manufacturers, an approach characterized as the collaborative planning, forecasting and replenishment or "CPFR" [24]. The impetus for the CPFR approach derives from the increased capacity of information systems. Collaboration and information sharing are part and parcel of a well-functioning supply chain but these are neither necessary nor sufficient conditions for good supply chain functioning. It is easy to imagine and to find examples that refute both necessity and sufficiency. The problem is that collaborative or information sharing supply chain are not "behaviorally" different from well-managed ones. Indeed it is fair to say these approaches have yet to develop into a well-based formal theory.

All of these approaches bring out the complexity of descriptions of supply chains. All of them point to deeper levels of structure that have yet to be described. The cohomological approach is new to this area and all that this section can do is hint at what might be developed.

Interpreting a combinatorial module as supply chain is as follows

1. $A$ is the set of organizations, called suppliers, involved in the manufacture or movement of goods to a final destination $a^*$, which is the maximum element of $A$.

2. $Tpc(a)$ is an action $\alpha(a) : W(a) \times Inv(a) \to Inv(a)$ where

   (a) The objects of $Tpc(a)$ in a supply chain are the inventory items of $a$,

denoted as $Inv(a)$; these are the sets of items that are necessary for manufacturing successive assemblies of components for a final product.

(b) $W(a)$ is the ring generated by $act(a)$ of processes required to accomplish the manufacturing. $act(a)$ is therefore the monoid of actions that produce "work in progress" inventory items.

3. $b \prec a$ or $b \in S(a)$ means $b$ is a supplier of $a$ for some of the items in $Inv(a)$.

4. $\beta(a, b)$ with $b$ in $S(a)$ is the "bill of materials" function from $a$ to a supplier of $a$. $\tilde{\beta}(a, b)$ is the set of inventory items that are produced by $b$ at the end of its manufacturing processes that are at the start of $a$'s manufacturing processes.

5. $\beta(a, b) : Inv(a) \to Inv(b)$ with $\beta(a, b)(e)$ being $\Sigma x_i.e_i$ where $e_i \in Inv(b)$. In this case $e$ and the $e_i$ are in $\tilde{\beta}(a, b)$.

6. $L$ is the lead-time between the issuing of an order and the receipt of goods when the supplier has to manufacture the item from starting inventory items.

"Replenishment" means that most actions $w \in W(a)$ deplete $Inv(a)$. If an inventory item $e$ is used up in a manufacturing action then there must be a way of tracking this. The simplest way is that $Inv(a)$ is a $\mathbb{Z}$ module and the states of $Inv(a)$ are inventory states with $\Sigma n_i.e_i$ denoting that the state of the inventory is $n_i$ of $e_i$. The ring $W(a)$ acts on the module $Inv(a)$ changing the quantities $n_i$ as it acts on $e_i$. The replenishment logic is a set of $w_r \in W(a)$ that can act on other actions as a "watcher". $w_r \circ w(e)$ looks at the quantity of $e$ after $w$ has acted on it and then decides to trigger a replenishment action or do nothing. The replenishment action is to send an order $\beta(a, b)(e)$ to one of more suppliers $b \in S(a)$. Replenishment is affected by the way schedules are interpreted in a supply chain and this adds complexity to the logic. When goods are perishable there has to be monitors that delete old items from inventory [3].

Let $\bar{T}$ be an interval of $\mathbb{N}$. In supply chains it is convenient to represent a schedule as a map $\xi : \bar{T} \to W(a)^{\gamma(a)}$ where $\gamma(a)$ is the maximum number of $w \in W(a)$ that can occur at any one time slot. This corresponds to the previous definition by $s^{-1}(t) = (\xi)(t)$.

Reversing time in a supply chain produces a "distribution plan". In this case a sequence of distributions start at $a^*$ and then get either unpacked and distributed further or used at each $a \in A$. Examples include the distribution of oil from a supertanker to the distributions of food and drugs for humanitarian aid.

A combinatorial module that is a supply chain is a first-order predicate system. For applications we need:

**Definition**. A *scenario* is a schedule that fulfills the sequence of requests or orders to a supply chains. It starts with

1. sequence of orders or "requests" $r_1$, $r_2$, ..., $r_n$ issued at time $t_1$, $t_2$, ..., $t_n$

2. a starting state $q(t_0) \in Inv(A)$ with $t_0 \leq t_1$,

3. the requests are to be fulfilled at $t'_1$, $t'_2$, ..., $t'_n$ respectively.

[3] The significance and variety of replenishment methods cannot be underestimated. It cannot be given due treatment here.

and the solution of the scenario is the schedule that can be constructed within the logic of the supply chain to deliver $r_i$ on or before the $t_i$.

The solution to a scenario is a theorem of a supply chain. If $\sigma$ is a scenario and $\xi$ is a schedule that solves it we write $\xi \Rightarrow \sigma$. Not all scenarios can be so solved, some are beyond the supply chains capability. It is the boundary between success and failure of a supply chain that best defines its capacity. The next section applies the results of section 2 on $\mathcal{P}_{\mathfrak{L}}$ and $mdl_{\mathfrak{L}}(\pi)$ where $\pi$ is the class of scenarios. However the relationship "$\Rightarrow$" which corresponds to $\mathfrak{L}$ is not defined directly in terms of sheaves so this has to be done.

## 5.1 Capability: Scenarios and Proofs

$\mathcal{U}$ is the standard cover of $\underline{A}$ and $\mathcal{P}$ a subsheaf of $Sch$. Given a scenario $\sigma$ let $\sigma(a)$ be the bill of materials required at $a$ that is necessary for $\sigma$ to be satisfied. The time that is associated with this has to be no later than the "worst case" lead time for the goods to be incorporated in the requests for $\sigma$. $\sigma(U) = \underset{a\in U}{\cup}\sigma(a)$. Let $Scn(U) =_{defn} \{\sigma(U) \mid \sigma \text{ is a scenario } \}$.

**Definition** The presheaf $\mathcal{S}_0$. For $U \in \mathcal{U}$

1. For $\xi \in \mathcal{P}(U)$, $S(\xi) =_{defn} \{\sigma \in Scn(U) \mid \xi \Rightarrow \sigma\}$
2. $\mathcal{S}_0(U) =_{defn} \{S(\xi) \mid \xi \in \mathcal{P}(U)\}$

Note that if $\xi$ and $\xi'$ imply the same set of local scenarios then $S(\xi) = S(\xi')$.

**Proposition 5.1.** *$\mathcal{S}_0$ is a presheaf.*

**Proof**. $V \subset U$ and $\xi(U) \Rightarrow \sigma(U)$ means for all $a \in U$ $\xi(a) \Rightarrow \sigma(a)$ and so, necessarily, $\xi(V) \Rightarrow \sigma(V)$. Thus $S(\xi)$ in $S_0(U)$ implies that the restriction $S(\xi \mid V) = S(\xi) \mid V$ is well defined and composes with further restrictions.

**Definition** The set-valued sheaf $\mathcal{S}$. This is the associated sheaf of the presheaf $\mathcal{S}_0$.

Using the usual étale construction $\mathcal{S}(U) = \Gamma(U, \underset{a\in U}{\amalg}\mathcal{S}_0(a))$, the set of sections $\phi : U \to \underset{a\in U}{\amalg}\mathcal{S}_0(a)$. A section here is an equivalence class of the direct limit of the $\mathcal{S}_0(U)$ where the $U$ converge to $\{s\}$. Thus $\phi(a) = S(\phi(\xi)(a))$ such that the $\phi(\xi)(a)$ is the value at $a$ of a $\xi \in \mathcal{P}(U)$ for some $U$ containing $a$.

**Definition**. The presheaf $Pv$

1. For a scenario $\sigma$ and $U \subseteq A$, $Z(\sigma) =_{defn} \{\xi \in \mathcal{P}(U) \mid \xi \Rightarrow \sigma(U)\}$.
2. For $U \in \mathcal{U}$, $Pv(U) =_{defn} \{Z(\sigma)(U) \mid \sigma \text{ is a scenario } \}$.

If $V \subseteq U$ then by definition $\xi(U) \Rightarrow \sigma \mid U$ clearly means that $\xi(V) \Rightarrow \sigma \mid V$ and so $Pv$ is a presheaf.

**Definition** The set-valued sheaf $Prv$ is defined to be the associated sheaf of the presheaf $Pv$.

In the same way as was used to define $\mathcal{S}$, the étale construction is used to produce $Prv$. A section $p : U \to \underset{a\in U}{\amalg}Pv(a)$ is given by $p(a) = Z(p(\sigma)(a))$. Thus $p(U)$ corresponds to some $\sigma \in Scn(U)$.

The sheaves $Prv$ and $\mathcal{S}$ are applications of the general case $\mathcal{P}_{\mathfrak{L}}$ and $mdl(\pi)_{\mathfrak{L}}$ where $\mathfrak{L}(\xi, m)$ becomes $\xi \Rightarrow \sigma$. Consequently

**Theorem 5.1.** *If the partial orders given by inclusion on the sets $Z(\sigma)$ and $S(\xi)$ are complete then there are sheaf maps $\mathfrak{j} : Prv \to \mathcal{S}$ and $\mathfrak{f} : \mathcal{S} \to Prv$ that are inverses.*

Note that if $\sigma$ cannot be satisfied by any schedule on any covering set of $\mathcal{P}$ then $Z(\sigma)$ is empty. Furthermore there is no $\xi$ such that $S(\xi)$ contains $\sigma$.

A scenario $\sigma \in Scn(U)$ can be represented as a map $\sigma : \bar{T} \to Inv(U)$ with $Inv(U) = \bigoplus_{a \in U} Inv(a)$ with each $Inv(a)$ a $\mathbb{Z}$ module. The $t$ in $\sigma(t)$ can be chosen as the maximum $t$ for which the goods in $\sigma(t)$ can be produced so that some schedule can deliver the required goods in time [4]

**Definition.** The addition of scenarios. For any $U \in \mathcal{U}$ and $\sigma_1$ and $\sigma_2$ in $Scn(U)$ define $\sigma_1 + \sigma_2$ as the sum of the functions $\sigma_i : T \to Inv(U)$.

**Proposition 5.2.** *The addition of scenarios makes Scn a group valued sheaf.*

The proof is left to the reader.

**Definition.** The sheaves $\widetilde{Prv}$ and $\tilde{\mathcal{S}}$. Define $\widetilde{Prv}$ as the group valued sheaf with underlying sheaf $Prv$ with addition denoted by $\tilde{+}$ and defined at each $U$ by the local relations $Z(\sigma_1)\tilde{+}Z(\sigma_2) =_{defn} Z(\sigma_1+\sigma_2)$. Similarly define $\tilde{\mathcal{S}}$ as the group valued sheaf with underlying sheaf $\mathcal{S}$ with the addition denoted by $\breve{+}$ and defined at each $U$ by the local relations $S(\xi_1)\breve{+}S(\xi_2) =_{defn} S(\xi_1 + \xi_2)$.

As defined these are sheaves as they are defined in terms of schedules and scenarios which are already sheaf sections in existing sheaves. As $S(\xi_1+\xi_2)$ contains both $S(\xi_1)$ and $S(\xi_2)$ while $Z(\sigma_1)$ and $Z(\sigma_2)$ both contain $Z(\sigma_1 + \sigma_2)$ the respective definitions of addition do not indicate set identification. This is added structure.

**Definition**. A supply chain $\underline{A}$ operates on *make to schedule quantity* if orders are only in quantities that fit a range of schedules. This includes "assemble to order". This means that whenever a set of orders (scenarios) $\sigma_i$ with $\sigma_i(a) = \Sigma n_{ij}.e_{ij}$ have make to order schedules $\underline{\xi_i}$ then there are no common components or processes that can be batched or combined. In this case $\Sigma(\sigma_i)(a)$ has a least schedule $\underline{\xi} = \Sigma\underline{\xi_i} = \Sigma\mathfrak{j}(Z(\sigma_i))$ giving $\Sigma\mathfrak{j}(Z(\sigma_i)) = \mathfrak{j}(\Sigma(Z(\sigma_i)))$ [5].

The effect of making to schedule quantity with schedules having minimum orders is illustrated by the following example. Suppose all schedules $\xi$ produce a certain item $e$ in multiples of 5. $\sigma_1$ wants 6 of $e$ and $\sigma_2$ wants 11. $\underline{\xi}_1 + \underline{\xi}_2$ produce 25 between them but $\underline{\xi}$ need only produce 11. If $\underline{A}$ operates on minimum order sizes then $\underline{\xi}_1 + \underline{\xi}_2$ produces the same amount as $\sigma_1 + \sigma_2$. This is equivalent to $\underline{\xi}_i$ producing exactly what $\sigma$ wants and the same for $\underline{\xi}$.

**Proposition 5.3.** *If $\underline{A}$ operates on make to schedule quantity then $\mathfrak{j}$ and $\mathfrak{f}$ are homomorphisms*

**Proof**. Write $\mathfrak{j}(Z(\sigma))$ as $S(\underline{\xi})$ where $\underline{\xi}$ is the least $\xi$ with $\xi \Rightarrow \sigma$. Then $\mathfrak{j}(Z(\sigma_1)\tilde{+}Z(\sigma_2)) = \mathfrak{j}(Z(\sigma_1 + \sigma_2))$ which is to equal $\mathfrak{j}(Z(\sigma_1))\breve{+}\mathfrak{j}(Z(\sigma_2))$.

$$\mathfrak{j}(Z(\sigma_1))\breve{+}\mathfrak{j}(Z(\sigma_2)) = S(\underline{\xi}_1)\breve{+}S(\underline{\xi}_2) = S(\underline{\xi}_1 + \underline{\xi}_2) \text{ (by definition).}$$

$\mathfrak{j}(Z(\sigma_1+\sigma_2)) = S(\underline{\xi})$ where $\underline{\xi}$ is the greatest lower bound with $\underline{\xi} \Rightarrow \sigma_1+\sigma_2$. As $\underline{\xi}_i \Rightarrow \sigma_i$ it is natural to identify $\underline{\xi}_1+\underline{\xi}_2$ with $\underline{\xi}$ in which case $\mathfrak{j}$ would conserve addition. $\underline{A}$ makes

[4] This approximates "just-in-time" operations wherein goods are made and shipped so that the supplier does not store the finished goods and the buyer uses them immediately on arrival.

[5] "Lean manufacturing" has this linearity as a goal. It is the same as "scaling without drama". A similar concept is minimum order quantities to fit jumps in scheduling

to schedule quantities implies item for item $\underline{\xi}$ matches $\underline{\xi}_1 + \underline{\xi}_2$ and so $S(\underline{\xi}) = S(\underline{\xi}_1 + \underline{\xi}_2)$ and $\mathfrak{j}$ preserves the addition. To prove $\mathfrak{j}$ takes 0 to 0 and inverses to inverses is obvious so $\mathfrak{j}$ is a homomorphism.

Again write $\mathfrak{f}(S(\xi)) = Z(\hat{\sigma})$ where $\hat{\sigma}$ is the least upper bound of $\sigma$ with $\xi \Rightarrow \sigma$. We want to show

$$\mathfrak{f}(S(\xi_1)\breve{+}S(\xi_2)) = \mathfrak{f}(S(\xi_1 + \xi_2)) \text{ is equal to } \mathfrak{f}(S(\xi_1))\tilde{+}\mathfrak{f}(S(\xi_2)).$$

But

$$\mathfrak{f}(S(\xi_1))\tilde{+}\mathfrak{f}(S(\xi_2)) = Z(\hat{\sigma}_1)\tilde{+}Z(\hat{\sigma}_1).$$

$\hat{\sigma}_i$ is the least upper bound for $\xi_i$. Hence $\xi_1 + \xi_2 \Rightarrow \hat{\sigma}_1 + \hat{\sigma}_2$. Using the make to schedule quantities requested in scenarios and schedules the least upper bound in both cases is the same and so $\mathfrak{f}(S(\xi_1)\breve{+}S(\xi_2)) = Z(\hat{\sigma}_1)\tilde{+}Z(\hat{\sigma}_1)$. The remaining details to show $\mathfrak{f}$ is a homomorphism are clear. □

The condition "make to schedule quantities" is crucial here as the proof indicates. Its effect is to match schedules for all scenarios that can be accomplished.

**Proposition 5.4.** *If $\underline{A}$ operates on make to schedule quantities the cohomology of the sheaves $\mathcal{P}$, $\widetilde{Prv}$ and $\tilde{\tilde{\mathcal{S}}}$ are identical.*

**Proof.** In $C^n(\widetilde{Prv})$ an $n$-cocycle requires

$$Z(\sigma_0)\tilde{+}Z(\sigma_2)\tilde{+}..\tilde{+}Z(\sigma_m) = Z(\sigma_1)\tilde{+}Z(\sigma_3)\tilde{+}..\tilde{+}Z(\sigma_{m'})$$

where $m$, $m'$ are $n$ and $n-1$ depending on whether $n$ is even or odd. By definition this is $Z(\sigma_0+\sigma_2+..+\sigma_m) = Z(\sigma_1+\sigma_3+..+\sigma_{m'})$ and corresponds to $\sigma_0+\sigma_2+..+\sigma_m$ in $C^n(\tilde{S})$. It also corresponds to

$$S(\underline{\xi}_0)\breve{+}S(\underline{\xi}_2)\breve{+}..\breve{+}S(\underline{\xi}_m) = S(\underline{\xi}_1)\breve{+}S(\underline{\xi}_3)\breve{+}..\breve{+}S(\underline{\xi}_{m'})$$

$$\text{that is } S(\underline{\xi}_0 + \underline{\xi}_2 + .. + \underline{\xi}_m) = S(\underline{\xi}_1 + \underline{\xi}_3 + ... + \underline{\xi}_{m'}).$$

As the schedules effectively match the scenarios we have

$$\underline{\xi}_0 + \underline{\xi}_2 + .. + \underline{\xi}_m = \underline{\xi}_1 + \underline{\xi}_3 + ... + \underline{\xi}_{m'}.$$

which gives a cocycle in $C^n(\mathcal{P})$. □

A similar argument applies to coboundaries. The make to schedule quantities condition is really a statement asserting a close match between scenarios and schedules means that any schedule can appear as a $S(\xi)$ as we simply adjust scenarios to require it. This way the three cohomologies are the same.

The importance of this result is when it fails. The failure of make to schedule quantities or any of its equivalent matching statements is asserted whenever this proposition fails. It is a way of measuring the difference between the operations of $\underline{A}$ and a supply chain that adjusts schedules to orders.

How bad can the difference difference between $\widetilde{Prv}$ and $\tilde{S}$ be? Many factory operations can work on a 8 hour shift, an 8 hour shift with overtime, two shifts, two shifts with overtime and three shifts. Scenarios are then batches of orders added so that they are a collective order load $\sigma = \sum_{i=1}^{n} \sigma_i$ and the scheduling is matched as closely as possible. This is similar to public transport systems that have peak hour and off-peak services which have significant gaps between the total order $\sigma$ for trip

requests and the small number of options for schedules. Scenarios $\sigma_1$, $\sigma_2$... are all satisfied by a single schedule $\xi$ as is their sum so $Z(\sigma_1) = Z(\sigma_2)$ and $\mathfrak{j}(Z(\sigma_1)+Z(\sigma_2)) = \mathfrak{j}(Z(\sigma_1) + \sigma_2)) = S(\xi)$ not $S(2\xi)$. Similarly in the case of the stepped shift pattern $\mathfrak{j}(Z(\sigma)) = \mathfrak{j}(\Sigma Z(\sigma_i)) = S(\xi)$. In fact $Prv$ only contains the set $Z(\sigma)$; the sets $Z(\sigma_i)$ are the same. If $\sigma_1$ and $\sigma_2$ have $Z(\sigma_1) \neq Z(\sigma_2)$ then $\mathfrak{j}(Z(\sigma_k)) = S(\xi_k)$, $k = 1, 2$ and $S(\xi_1) \neq S(\xi_2)$. If $\xi_1$ is the day shift, $\xi_2$ day shift + overtime we can have $\mathfrak{j}(Z(\sigma_1)) + \mathfrak{j}(Z(\sigma_2)) = S(\xi_1) + S(\xi_2)$. But because $\sigma_1$ is so small $\mathfrak{j}(Z(\sigma_1) + Z(\sigma_2)) = S(\xi_2)$.

$Z(\sigma)$ has been defined locally, it is really $Z(\sigma)(U)$ and corresponds to a subsheaf of $Sch$ that illuminates some of the internal dynamics of $\underline{A}$.

## 5.2 Fragmentation Theory

When a supply chain, $\underline{A}$, is faced with a stream of orders that cannot be delivered on time certain parts of $\underline{A}$ will fall below the required capacity while other parts will have sufficient capacity. The study of this phenomena is referred to here as "Fragmentation Theory". If a soft sheaf indicates the greatest level of internal order and coordination then as things fail we can expect the cohomology to become more complicated. In this way there is an analogy with phases in physical media that are characterized by correlation [3] and fragmentation is a theory of the phases of lost correlation as measured by cohomology.

Let $\underline{r}$ be an order to $a^*$ to deliver $r_i$ goods no later than $t_i$ [6] and let $\underline{q} \in Inv(A)$ be the starting inventory given by $\underline{q}(a)$ for each $a \in \underline{A}$. That is, a scenario $\sigma(\underline{r}, \underline{q})$.

**Definition**. Define $Sch(\underline{r}, \underline{q})$ to be the group-valued sheaf generated by schedules that satisfy $\sigma(\underline{r}, \underline{q})$.

Hence $Sch(\sigma(\underline{r}, \underline{q})) \models \sigma(\underline{r}, \underline{q})$. Because the subsheaves $Sch(\underline{r}, \underline{q})$ of $Sch$ are concerned with time constrained schedules they cannot be expected to be soft. Also if $\xi$ is a solution to $\sigma(\underline{r}, \underline{q})$ then any $\xi' \geq \xi$ is a solution to $\sigma(\underline{r}, \underline{q})$.

**Definition** The scenario category. The objects of this category are the sheaves $Sch(\underline{r}, \underline{q})$ and morphisms are embeddings $Sch(\underline{r_1}, \underline{q_1}) \hookrightarrow Sch(\underline{r_2}, \underline{q_2})$.

We shall say that $\underline{A}$ is "normal" if the following is satisfied:

$$\underline{r}_1 \geq \underline{r}_2 \text{ and } \underline{q}_1 \leq \underline{q}_2 \Rightarrow Sch(\underline{r_1}, \underline{q_1}) \hookrightarrow Sch(\underline{r_2}, \underline{q_2}). \tag{11}$$

This need not be the case. There can be products that are easier to make in large amounts by utilizing batch processes that can use economies of scale that are not available for smaller amounts. Small amounts cannot use the batch processes and so take more time to "craft by hand" than manufacture using robotized factories.

**Theorem 5.2.** *(The coboundary influence theorem) If $\mathcal{P}_i$, $i = 1, 2$ are two subsheaves of $Sch$ and $j : \mathcal{P}_1 \hookrightarrow \mathcal{P}_2$ then the induced homomorphism $\check{H}(j) : \check{H}(\mathcal{P}_1) \to \check{H}(\mathcal{P}_2)$ is defined and*

$$ker(\check{H}^k(j)) = (Z^k(\mathcal{P}_1) \cap (C^k(j))^{-1}(B^k(\mathcal{P}_2)))/B^k(\mathcal{P}_1) \tag{12}$$

*where $C^k(j)$ is the induced map $C^k(\mathcal{P}_1) \to C^k(\mathcal{P}_2)$.*

**Proof**: For any set of schedules defined on $\mathcal{P}_1$ they are also in $\mathcal{P}_2$.

This means

[6] interpreted as $Inv(a^*)$ will have $r_i$ goods available at $t_i$.

$$\begin{array}{ccc} C^{k-1}(\mathcal{P}_1) & \xrightarrow{\partial^{k-1}} & C^k(\mathcal{P}_1) \\ \downarrow{\scriptstyle C^{k-1}(j)} & & \downarrow{\scriptstyle C^k(j)} \\ C^{k-1}(\mathcal{P}_2) & \xrightarrow{\partial^{k-1}} & C^k(\mathcal{P}_2) \end{array}$$

commutes. A cocycle in $C^k(\mathcal{P}_1)$ will satisfy the same equation in $C^k(\mathcal{P}_2)$ and so $Z^k(\mathcal{P}_1)$ is a subgroup of $Z^k(\mathcal{P}_2)$ and similarly $B^k(\mathcal{P}_1)$ is a subgroup of $B^k(\mathcal{P}_2)$.

If $z \in \check{H}^k(\mathcal{P}_1)$ and $z \neq 0$ then $z$ represents an element in $Z^k(\mathcal{P}_1)$ that is not in $B^k(\mathcal{P}_1)$. Hence

$$\check{H}^k(j)(z) = 0 \Leftrightarrow z \in B^k(\mathcal{P}_2)$$

so that

$$ker(\check{H}^k(j)) = \check{H}(j)^{-1}(B^k(\mathcal{P}_2)))/B^k(\mathcal{P}_1)$$

and $(\check{H}(j)^{-1}(B^k(\mathcal{P}_2)) = Z^k(\mathcal{P}_1) \cap (C^k(j))^{-1}(B^k(\mathcal{P}_2))$ which gives the result. □

**Definition**. The *frontier* of $\underline{A}$ is the set $(\underline{r}, \underline{q})$ such that for any pair $(\delta r, \delta q) \in Inv(a^*) \times Inv(A)$, $\delta r$ and $\delta q$ positive, $\check{H}^0(Sch(\underline{r}, \underline{q}))$ is non trivial and $\check{H}^0(Sch(\underline{r} + \delta r, \underline{q} - \delta q) = 0$.

The frontier divides the scenario space of elements $(\underline{r}, \underline{q})$ into demand that can be accomplished and demand for which there is no global schedule.

To simplify the discussion set $\underline{q} = 0$ so "initializing" with an empty inventory and hence the scenario is in terms of the longest initial lead-times for $\underline{A}$. Let $\mathcal{P}(\underline{r}) = Sch(\underline{r}, 0))$ and $\check{H}^0(\mathcal{P}(\underline{r})) \neq 0$. If $\mathfrak{P}(r) = \left\{\delta r \mid \check{H}^0(\mathcal{P}(\underline{r} + \delta r)) = 0\right\}$ where $\delta r \in Inv(a^*)$ then the smallest $\delta r \in \mathfrak{P}(r)$ to change the cohomology from soft to have $\check{H}^0(\mathcal{P}(\underline{r} + \delta r)) = 0$ is the *fragmentation gradient*.

At the frontier an order is as big as can be with a given, achievable lead-time. The fragmentation gradient gives the increase in size of the order for which things become unpredictable.

The difference between $dim(\check{H}^n)$ (as a $\mathbb{Z}$ module) for each $n$ and $dim(Z^n)$ for each $n$ give significant information as to the extend that a class (sheaf) of schedules are coordinated.

**Definition**. The *cocycle mark* of a sheaf $\mathcal{P}$ denoted $mk_Z$ is the formal sum

$$mk_Z(\mathcal{P}) = \sum_{n \geq 0} dim(Z^n(\mathcal{P})).t^n \tag{13}$$

and the *class mark* of $\mathcal{P}$ denoted $mk_H$ is the formal sum

$$mk_H(\mathcal{P}) = \sum_{n \geq 0} dim(\check{H}^n(\mathcal{P})).t^n \tag{14}$$

Example: With rather weak constraints on the duration of schedules $Sch$ can be shown to be soft. Any production of any goods can be extended to be an order of something; $mk_Z(Sch) - mk_H(Sch) = mk_Z(Sch) - dim(\check{H}^0(Sch))$.

If $Sch(\underline{r_1}) \hookrightarrow Sch(\underline{r_2})$ (as when $r_1 > r_2$) then the coboundary influence theorem gives a description of the way $Sch(\underline{r_2})$ can grow. If for a given $k$, $B^k(Sch(\underline{r_2}))$ is large then a non-zero kernel is possible giving a measure of the change in cohomologies. For a sufficiently large $\delta r$ eventually $Sch(\underline{r} + \delta r)$ cannot deliver $\underline{r} + \delta r$ in the time required by the scenario $\sigma(\underline{r})$. The extreme case is:

**Definition**. A request $\underline{r} + \delta r$ is on the *disordered boundary* when

$$mk_Z(Sch(\underline{r}+\delta r)) = mk_H(Sch(\underline{r}+\delta r)).$$

This means that whenever schedules coincide locally it happens because of local circumstances that are not derived from any larger structure. What order there is is local. Coming in from the disordered boundary the sheaves get bigger and bigger until there are enough coboundaries to kill off the cocycles. The coboundary influence theorem is applicable here.

**Proposition 5.5.** *If* $(\underline{r}+\delta r)$ *is on the disordered boundary and* $\delta r > \delta r'$ *and* $j : Sch(\underline{r}+\delta r) \hookrightarrow Sch(\underline{r}+\delta r')$ *then*

$$ker\check{H}^k(j) = Z^k(Sch(\underline{r}+\delta r)) \cap B^k(Sch(\underline{r}+\delta r')) \tag{15}$$

.

**Proof**: This is really a corollary of the coboundary influence theorem as $B^k(Sch(\underline{r}+\delta r)) = 0$. The kernel measures how much the cohomology diminishes and more order is created as coboundaries indicate the growth of order. The set of all $\delta r$ between the fragmentation boundary and the disordered boundary can be characterized by the cohomology of the $Sch(\underline{r}+\delta r)$. □

# 6 The Coordination of Change

In this section the sheaf, $J\mathcal{GP}$ (introduced in section 2) is applied to supply chains. This object is a preoccupation of managers of jobbing shops across the world. Not, of course, in this form.

## 6.1 Groupoid Classes and the $J\mathcal{GP}$ sheaf

There are many classes of groupoids that are relevant to changing schedules in an industrial setting.

1. Adjustments for schedules that do not change elements of $w \in W(a)$ that use large amounts of an expensive and not always available resource. Let $u : W(a) \to \mathbb{N}$ be a ranking of factory resources required to accomplish $w$ in a given amount of time. Define the groupoid $G(u^{-1}(k))$ as the set of changes that leave $\{w|u(w) \geq k\}$ unchanged within a schedule. This is clearly a groupoid and corresponds to the practice of changing schedule around the "large" operations such as batch jobs or processes that take up disproportionate amount of factory resources.

2. Adjustments associated with a groupoid $G^+$ which is defined by the property that $\forall g \in G^+$ there is a $t' \in \bar{T}$ such if $t \leq t' < max\{t|t \in dom(\xi)\}$ then $g\xi(t) = \xi(t)$. This allows a schedule to be changed while in process. In this case an adjustment $\xi \xrightarrow{g} \zeta$ might divide into adjustments earlier and later than a given time, such as the time of arrival of newly scheduled goods. Then $g.\xi(t \leq t_a) = \xi_1$ and $g.\xi(t > t_a) = \xi_2$.

3. Schedules that depend on a set of variables or parameters that can be changed are a source of groupoids and their associated equivalences classes. The groupoid acts on schedules through the schedule's dependence on parameters that can be changed. If $\xi = \xi(v_1, v_2, ..., v_n)$ where $v_1, v_2, ..., v_n \in V$ such that if $g$ is an bijection of $V$ then $g$ changes $\xi$ by the action $g\xi(\underline{v}) = \xi(g(\underline{v}))$ for some $\underline{v} \in V$. For this to make sense there must be a $\xi' = \xi(g(\underline{v}))$. Typical parameters include times to reach various production stages, costs of labor, costs of using fast or slow machines, slower or faster processing

time that come from using different input materials which eventually give the same final output. In particular not all parameter changes have a schedule associated with them. Groupoids acting on such parameterize schedules can be decomposed into sub-groupoids that give the independence of subsets of parameters.

**Alternative definition of the group valued sheaf of adjustment classes**. When the $g \in \mathcal{G}$ need not be homomorphisms the colimit definition of $\mathcal{J}(\mathcal{G})\mathcal{P}$ remains the same by interpreting $\mathcal{P}$ as a set-valued sheaf. The structure of a given $\mathcal{G}(\xi)$ can be analyzed in terms of it subgroupoids. If $g : \xi_1 \to \xi_2$ there is an associated set function $\tilde{g} : \mathcal{G}(\xi_2) \to \mathcal{G}(\xi_1)$ given by $\tilde{g}(h) = h \circ g$. It is not a groupoid homomorphism [21] but allows the decomposition of groupoids in a way that reflects the hierarchy of equivalence relations among the schedules connected by adjustments. For example given

$$\xi_1 \xleftarrow{g_1} \xi_0 \xrightarrow{g_2} \xi_2$$

where $g_1$ and $g_2$ are the only adjustments from $\xi_0$ to $\xi_1$ and $\xi_2$ then $G(\xi_0)$ decomposes as $G(\xi_1) \sqcup G(\xi_2)$. Furthermore any equivalence relation between a $\xi'$ connected to $\xi_1$ and $\xi''$ connected to $\xi_2$ must be related through $\xi_0$.

**Definition**. The relationship of $\mathcal{G}(\xi_0)$ and $\mathcal{G}(\xi_1) \sqcup \mathcal{G}(\xi_2)$ is denoted "$\approx$" so $\mathcal{G}(\xi_0) \approx \mathcal{G}(\xi_1) \sqcup \mathcal{G}(\xi_2)$ means any relation that $\xi_0$ holds with any other schedule arises through adjustments through one of $\xi_1$ or $\xi_2$.

**Definition** Define $\tilde{J}(\mathcal{G})\mathcal{P}(U)$ as the free group generated by the elements of the set $\{\mathcal{G}(\xi) \mid \xi \in \mathcal{P}(U)\}$ factored by the subgroup generated by the set of relations

$$\{\mathcal{G}(\xi_1) + \mathcal{G}(\xi_2) - \mathcal{G}(\xi_3) | \mathcal{G}(\xi_3) \approx \mathcal{G}(\xi_1) \sqcup \mathcal{G}(\xi_2)\}.$$

Given $\mathcal{G}(\xi)$ denote its class in $\tilde{J}(G)\mathcal{P}$ as $[\mathcal{G}(\xi)]$.

This is a standard Grothendieck Group construction [25]. The action of the adjustments is fully consistent with sheaf structure in the sheaf of sets. This addition is additional structure that is defined in terms of sections and so it gives a group structure on the sheaf $\tilde{J}(G)\mathcal{P}$. The remainder of this section will concentrate on $\mathcal{J}(\mathcal{G})\mathcal{P}(U)$ although almost all the results can be reinterpreted in terms of $\tilde{J}(G)\mathcal{P}$.

For the rest of this section assume that among the possible groupoids of adjustments, $\mathcal{G}(\xi)$ is the full set of adjustments that are relevant to the $\xi$. The phrase "relevant to" is to imply that a category of groupoids has been chosen to reflect the type of adjustments that are to be studied. This means that $\mathcal{G}$ contains all the decompositions of groupoids. If $\mathcal{G}_1$ and $\mathcal{G}_2$ act on $\xi$ with $g_1 \in \mathcal{G}_1$, $g_2 \in \mathcal{G}_2$, then $g_2 \circ g_1^{-1} \in \mathcal{G}_2$, $g_1 \circ g_2^{-1} \in \mathcal{G}_1$ so that $\mathcal{G}_1 \cup \mathcal{G}_2$ is a subgroupoid of the set of all adjustments on $\xi$.

From now on we drop the mention of $\mathcal{G}$ in $\mathcal{J}(\mathcal{G})\mathcal{P}$ and just write $J\mathcal{P}$. The sheaf $J\mathcal{P}$ represents the methodological or operational freedoms in response systems or, more generally, in a combinatorial module with type weightings. These are features of everyday decisions in countless organizations.

The nature of $\mathcal{G}$ for $\mathcal{P}$ can influence the cohomology of $\mathcal{P}$. Define $\mathcal{G}$ as quasi-linear if for all $U \in \mathcal{U}$ and for all $g \in \mathcal{G}(U)$ there are $g_1$ and $g_2$ such that $g(\xi_1 + \xi_2) = g_1.\xi_1 + g_2.\xi_2$. Let

$$\mathcal{P} \xleftarrow{sr} \mathcal{G} \xrightarrow{tg} \mathcal{P}$$

be the source and target of $g : \xi \mapsto \xi'$. One can then define a fibration in groupoids from the map $\mathcal{G} \xrightarrow{sr,tg} \mathcal{P} \times \mathcal{P}$.

**Definition**. $\mathcal{P}$ is $\mathcal{G}$ constrained at $U$ if there exists $V \subset U$ with

$$< im(\mathcal{P}(U) \xrightarrow{restrict} \mathcal{P}(V)) > \subsetneq \bigcup_{\xi \in sr(\mathcal{G}(V))} \mathcal{G}(\xi).$$

Here the angle brackets are in the group category while the rest is in Sets. This means that the restriction of $\mathcal{P}(U)$ to $\mathcal{P}(V)$ does not generate all the $\xi \in \mathcal{P}(V)$ that can be reached by adjustments. In fact it is more than that. Let $\zeta_k$ be the set of $\mathcal{P}(V)$ generated by the restriction image of $\mathcal{P}(U)$. Then not all the "$\mathcal{G}$ closure" of the image can be generated by the $\zeta_k$.

**Proposition 6.1.** *If $\mathcal{G}$ acts on $\mathcal{P}$ quasi-linearly, $\mathcal{P}$ is $\mathcal{G}$ constrained at $U$ then if there is an $n$ with $dim(Z^n(\mathcal{P})) \neq 0$ then $\check{H}^n(\mathcal{P}) \neq 0$.*

**Proof.** In the case $U = \cup U_i$ and $V = \cap U_i$ even if $\xi \in \mathcal{P}(V)$ has an expression in terms of restrictions of $\eta_i \in \mathcal{P}(U_i)$ there might be $g.\xi$ that cannot be expressed in terms of elements of the form $\eta_i \mid V$.

Suppose we have an $n$ cocycle with support in $V$. Quasi-linear means we can apply adjustments to cocycles and get other cocycles:

$$\sum_{i=2k} \xi_i = \sum_{j=2k+1} \xi_j$$

implies

$$g \sum_{i=2k \leq n} \xi_i = g \sum_{j=2k+1 \leq n} \xi_j$$

hence there exists $g_i$ for $i$ even and $g_j$ for $j$ odd so that

$$\sum_{i=2k \leq n} g_i.\xi_i = \sum_{j=2k+1 \leq n} g_j.\xi_j.$$

Even if the cocycle is a coboundary there are adjustments of the cocycle that cannot be expressed in terms of the original cocycle so we have a new cocycle without a corresponding coboundary - a cohomology class. □

## 6.2 The Coordination of Changes, the Cohomology of $JSch$

The Čech cohomology construction can be applied to any group-valued sub-sheaf $\mathfrak{J} = J\mathcal{P}$ of $JSch$. The dependence on $\mathcal{G}$ is understood.

Starting with $C^n(\mathfrak{J})$ the additive free group ($\mathbb{Z}$ module) is generated by the elements

$$\mathcal{G}(\xi_{i_0,i_1..i_n})(U_{i_0} \cap U_{i_1} \cap ... \cap U_{i_n})$$

over all the non-empty intersections.

$\partial : C^n(\mathfrak{J}) \to C^{n+1}(\mathfrak{J})$ is defined as in section 3.1 with $\xi$ replaced with $\mathcal{G}(\xi)$. The resulting cohomology is related to the cohomology of $\mathcal{P}$ as follows. Given a cocycle in $\mathcal{P}$: $\xi_0 + \xi_2.. + \xi_{2k} = \xi_1 + \xi_3.. + \xi_{2k+1}$ on some $U$ and the groupoid elements act linearly or quasi-linearly then in $\mathfrak{J}$

$$\mathcal{G}(\xi_0 + \xi_2 + ... + \xi_{2k}) = \mathcal{G}(\xi_1 + \xi_3 + ... + \xi_{2k+1}).$$

so that

$$\mathcal{G}(\xi_0) + \mathcal{G}(\xi_2) + ... + \mathcal{G}(\xi_{2k}) = \mathcal{G}(\xi_1) + \mathcal{G}(\xi_3) + ... + \mathcal{G}(\xi_{2k+1})$$

However there is no rule that says these sets are the same sets. Indeed consider the following counter-example: given a 2-cocycle $\xi_0+\xi_2 = \xi_1+\xi_3$ on some $a \in A$ in which $Inv(a)$ has raw materials $\{e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8\}$. $\xi_0$ uses $\{e_1, e_2, e_3, e_4\}$, $\xi_2$ uses $\{e_5, e_6, e_7, e_8\}$, $\xi_1$ uses $\{e_1, e_2, e_3\}$, $\xi_3$ uses $\{e_4, e_5, e_6, e_7, e_8\}$. Adjustments can arise by substitutions in $\{e_4, e_5, e_6\}$. This leaves $\mathcal{G}(\xi_1)$ and $\mathcal{G}(\xi_3)$ distinct sets but $\mathcal{G}(\xi_0)$ and $\mathcal{G}(\xi_2)$ share adjustments by swapping around $e_4$ and $e_5$ and so $\mathcal{G}(\xi_0) \cap \mathcal{G}(\xi_2) \neq \emptyset$. In the case where adjustments act linearly and $\mathcal{K}$ is the map $\mathcal{P} \to Colim(\mathcal{P} \xrightarrow{g} \mathcal{P})$ where the colimit is taken over all $\mathcal{P} \xrightarrow{g} \mathcal{P}$, $g \in \mathcal{G}$, then:

**Proposition 6.2.** *If $\mathcal{G}(\xi_0) \cap \mathcal{G}(\xi_2) \neq \emptyset$ then $(\xi_0 - \xi_2) \in ker(\mathcal{K} : \mathcal{P} \twoheadrightarrow J\mathcal{P})$.*

**Proof**. let $g$ in $\mathcal{G}(\xi_0) \cap \mathcal{G}(\xi_2)$ then the groupoids overlap and there can be found a $g'$ so that $g'\xi_0 = \xi_2$ so $\mathcal{K}(\xi_0) = \mathcal{K}(\xi_0)$. □

The kernel of $\mathcal{K}$ is a measures of the changes in the induced cohomology. Thus the difference in $\check{H}^n(J\mathcal{P})$ and $\check{H}^n(\mathcal{P})$ reflects on the extent that the local variations, as indicated by cocycles that are not coboundaries, are smoothed out by adjustments.

The ability to coordinate change is an important goal for supply chain managers; it is the much desired "integration". If $\mathcal{P} \subset Sch$ and $J\mathcal{P}$ is soft then every change can be accommodated throughout the supply chain. The existence of useful sub-sheaves $\mathcal{P}$ of schedules with desirable properties bears significantly on questions of coordination and integration. Suppose each schedule $\xi$ in $Sch$ can be perturbed or adjusted to one, $\gamma(\xi)$, in $\mathcal{P}$ giving a sheaf map $\gamma : Sch(U) \to \mathcal{P}(U)$ that need not be a group homomorphism. Suppose, also, that $\gamma$ is a retract so $\gamma$ is the identity on $\mathcal{P}$.
**Definition**: $\gamma$ is $\mathcal{G}$ *defined* if $\forall \xi \ \exists g^\gamma$ such that $\gamma(\xi) = g^\gamma(\xi)$ for some $g^\gamma \in \mathcal{G}(\xi)$.

**Theorem 6.1.** *If $\gamma$ is $\mathcal{G}$ defined then for all $U \subseteq \underline{A}$, $JSch(U) = J\mathcal{P}(U)$.*

**Proof**. If $\forall \xi \ \exists g^\gamma$ such that $\gamma(\xi) = g^\gamma$ for some $g^\gamma \in G(\xi)$ then $\mathcal{G}(\gamma(\xi)) = \mathcal{G}(\xi)$ as $(g^\gamma)^{-1}(\gamma(\xi)) = \xi$ so that $JSch = J\mathcal{P}$. □

If $J\mathcal{P}$ is soft $JSch$ itself is soft. The level of integration can be localized so that all the above considerations are only applicable to certain coordinated sub-networks $B \subset \underline{A}$ where $JSch(B) = J\mathcal{P}(B)$.

# 7 System Testing

Let $\mathfrak{S}$ be a software system. It can be thought of as a collection of information "objects" that require data from a variety of sources. In many cases it can be represented as a combinatorial module. Testing $\mathfrak{S}$ mimics its structure in many ways. Define $\underline{A}(\mathfrak{S})$ as the combinatorial module with each $Tpc(a)$ being an action $\alpha(a) : W(a) \times Typ(a) \to Typ(a)$ where $Typ(a)$ is a set of data inputs and $W(a)$ the set of actions that transform or connect them. Each test monitors the inputs to the actions and compares the outputs to the specifications. $a_1 \ ¡ \ a_2$ when the tests of $a_2$ cannot start until $a_1$ completes its tests. For a large $\mathfrak{S}$ possibly containing thousands of programs of all sizes $\underline{A}(\mathfrak{S})$ can be subsystems, algorithms and information subsystems connecting to data outside the system. $\underline{A}(\mathfrak{S})$ is a response system in any disciplined software environment as lead-time is the time allocated to testing.

Testing programs and systems is a strand of computer science research that started with Hoare's initiative for a "science of programming" [11] [28] [32]. The difficulties encountered in this work have developed into work on model-based testing, predicate abstraction [1, 2, 34, 4] and state-space simulation with programs such as Asml [5]

and SPIN [17][7]. As state spaces grow combinatorially it is difficult apply these ideas to large system in any convincing way. To reduce this state-space growth "Hyper-states" were introduced in [14] as equivalence classes that reflect some inherent logical structure. Independently of these streams of research commercial testing developed a process-oriented approach [33]. This process oriented approach has a track record of good results (most of the software we rely on) but very little attention is given to the structure of the system.

The approach here has more in common with the commercial practice. A set of requirements are identified and associated with very precisely written test scenarios that describe how the system must respond to a large range of inputs. That is tests verify that the system satisfies a set of statements that derive from the requirements. These are "local logics" and the tests establish those local logics. This is an application of the sheaf $\mathfrak{L}$ introduced in section 2. and we keeps most of the notation. $\underline{A}(\mathfrak{S})$ has an associated sheaf of models $mdl(\mathfrak{S})$ which we write as $mdl$. If the $l$-length input sequences (of supplier chains) of $a$ are given by

$$S^{\leq l}(a) =_{defn} \{a\} \cup \underset{k\leq l}{\cup} S^k(a)$$

and $U = S^{\leq l}(a) \subseteq A$ then $mdl(U)$ is the logics $W(a') \times Typ(a') \to Typ(a')$ along with $\beta(a', a'')$ for all $a', a'' \in U$ where $a'' \in S(a')$.
**Definition**. $\xi \in \mathcal{P}$ *establishes* $m \in mdl(U)$ if and only if

$$(m \models \varphi \Rightarrow \xi \Rightarrow \varphi) \wedge ((\xi \Rightarrow \neg\varphi) \Rightarrow \neg(m \models \varphi))$$

This relationship is denoted $\mathfrak{E}(\xi, m)$ (and is a specific example of $\mathfrak{L}$)

Applying the concepts of section 2. If $\xi_1$, $\xi_2 : T \to W^{\gamma(a)}$ and $\xi_2$ factors through $\xi_1$ so that $dom(\xi_2) \subseteq dom(\xi_1)$ and $range(\xi_2) \subseteq range(\xi_1)$ then write $\xi_1 > \xi_2$. We make the following assertions to be consistent with the application of $\mathfrak{L}$ even though it imposes some mild conditions of what schedules can be used in the establishment relation. If $\xi_1 \geq \xi_2$ then $\mathfrak{E}(\xi_2, m) \Rightarrow \mathfrak{E}(\xi_1, m)$ thus $\{\xi \in \mathcal{P}(U) \mid \mathfrak{E}(\xi, \mu(U))\}$ is a filter. If $m_1$ is an extension of $m_2$ then $\mathfrak{E}(\xi, m_1) \Rightarrow \mathfrak{E}(\xi, m_2)$.

Note that $\xi = \xi_1 + \xi_2$ implies $\xi \geq \xi_1$ and $\xi \geq \xi_2$.
**Definition.** Define

1. $\mathcal{P}_{\mathfrak{E}}(U) = \{X(\mu(U)) \mid X(\mu(U)) \neq \emptyset \text{ and } \mu \in mdl(U)\}$.
2. $mdl_{\mathfrak{E}}(U) = \{Y(\xi)) \mid Y(\xi(U)) \neq \emptyset \text{ and } \xi \in \mathcal{P}(U)\}$.

Applying theorem 2.2.

**Theorem 7.1.** *If for each $U \in \mathcal{U}$ the partial orders on $\mathcal{P}_{\mathfrak{E}}(U)$ and $mdl_{\mathfrak{E}}$ are complete then there exist maps $\mathfrak{j} : \mathcal{P}_{\mathfrak{E}} \to mdl_{\mathfrak{E}}$ and $\mathfrak{f} : mdl_{\mathfrak{E}} \to \mathcal{P}_{\mathfrak{E}}$ such that $\mathfrak{f} \circ \mathfrak{j} = 1_{mdl_{\mathfrak{E}}}$ and $\mathfrak{j} \circ \mathfrak{f} = 1_{\mathcal{P}_{\mathfrak{E}}}$. That is there is a duality between $\mathcal{P}_{\mathfrak{E}}$ and $mdl_{\mathfrak{E}}$.*

$\mathcal{P}_{\mathfrak{E}}$ contains information about patching together tests that establish models. This is information that testing assumes: that all the different tests applied to overlapping models can add to a test of the whole system. Frequently the system itself has added logic that allows one test to establish partial results for another tests. This adds

[7]Two major contributors Rajeev Alur and David Dill received the 2008 Computer Aided Verification for their work. Notice of American Mathematical Society December 2008, p 1429. The pioneers of models checking received the Turing Award in 2007 [7].

to the possible relations among the actions that make up schedules and so lead to equations among schedules; in some case to cocycles. This gives the motivation to develop the cohomology of $\mathcal{P}_{\mathfrak{E}}$. For this there is addition of models denoted $\oplus$; this has the property $m_1 \oplus m_2 \models \varphi \Leftrightarrow (m_1 \models \varphi) \vee (m_2 \models \varphi)$ and there is no contradiction. (If $\oplus$ is not to be trivial with $m \oplus m = m$, $\oplus$ needs to be like direct sum in groups or vector space or disjoint union in topological spaces.)

**Lemma 7.2.** *Let* $i = 1, 2$ *then* $\mathfrak{E}(\xi_i, \mu_i) \Rightarrow \mathfrak{E}(\xi_1 + \xi_2, \mu_1 \oplus \mu_2)$

**Proof**. For each proposition $\varphi_i$ modeled in $\mu_i$ and with $\xi_i \Rightarrow \varphi$ necessarily $\mu_1 \oplus \mu_2 \models \varphi_i$. Now the $\varphi_i$ for $i = 1, 2$ occur in the polynomial defining $\xi_1 + \xi_2$ which is the sum of the polynomials for $\xi_i$. This implies $\xi_1 + \xi_2 \Rightarrow \varphi_i$. $\mu_1 \oplus \mu_2 \models \varphi \Leftrightarrow (\mu_1 \models \varphi) \vee (\mu_2 \models \varphi)$ hence $(\xi_1 \Rightarrow \varphi \vee \xi_2 \Rightarrow \varphi)$ so $\xi_1 + \xi_2$ establishes $\mu_1 \oplus \mu_2$. □

This allows us to identify $X(m_1) + X(m_2)$ as $X(m_1 \oplus m_2)$ and $Y(\xi_1) + Y(\xi_2)$ as $Y(\xi_1 + \xi_2)$ then, using the usual construction going from a monoid to a group, we have group-valued sheaves.

**Proposition 7.1.** $\mathcal{P}_{\mathfrak{E}}$ *and* $mdl_{\mathfrak{E}}$ *are group-valued sheaves.*

The cohomology sequence can therefore be applied to these sheaves. The relations among test schedules reflect structures in the $\underline{A}(\mathfrak{S})$. If $U = \{a_1, ..., a_n\}$, schedules on $U$ are defined in terms of sets of processes $\xi(a_i)(t)$. If $U = U_0 \cap ... \cap U_n$, $\xi_k$ defined on $U_0 \cap ... \hat{U}_k ... \cap U_n$, $k = 0, 1, ..., n$ then a cocycle $\Sigma(-1)^k \xi_k = 0$ indicates relations among the set of actions $\{w_{kj} \mid w_{kj} \in range(\xi_k(t))\}$ holds in $U$. These relations are linear. Let $E(w_1, w_2, ..., w_m) \equiv_{defn} (w_1 - w_2 + ... + (-1)^{m-1} w_m) = 0$ be such a relation. Then cocycles correspond to relations $E(w_1, w_2, ..., w_m)$ in $\underline{A}(\mathfrak{S})$ and so to similarly defined relations in $\mathfrak{S}$ from which the actions $w_i$ are constructed. Thus the logic of the system $\mathfrak{S}$ is interwoven in $\underline{A}$ and determines properties of the sheaves $\mathcal{P}$ and their classes of establishment sheaves $\mathcal{P}^n_{\mathfrak{E}}$ of tests of limited iterations. This goes beyond the length of this paper and can found in [27]. The concept of adjustments and proposition 5.1 also applies to system testing.

Another perspective of system testing and one with much more in common with the computer science and model checking theme is as follows.

$A$ is the set of subsystems that can be logically described in terms of state-space actions $\alpha(a) : W(a) \times Typ(a) \to Typ(a)$. The members of $Typ(a)$ are variables that have states. These state actions are not necessarily isolated in the sense they can either overlap with other states or interact with other state spaces. $\underline{A}$ is a response system so a lead-time $L : W(a) \to \mathbb{N}$ is defined on each $W(a)$. This system is represented in a "causally reversed" manner: the interpretation of $b \prec a$ is inverted from the usual "$a$ depends on $b$", instead the states of $b$ depend on the states of $a$. Equally one can say $\alpha(b)$ is a recursive function or automaton with the states of $a$ being an oracle [13] for those of $\alpha(b)$. $\beta(a, b)(e)$ is the set of possible states in $Typ(b)$ that propagate from $e$ in $Typ(a)$. That is, a change in $e$ causes a change in members of $\beta(a, b)(e)$. State change propagates outward from $a^*$. The actual reaction depends on the state of the types in $\beta(a, b)(e)$. Define schedules as for supply chains. Let $\gamma(a)$ be the effective parallelism of the actions of $W(a)$: the number of operations that can happen in one time slot. Schedules as polynomials have no intrinsic direction of time. This can easily be reversed so that for $\xi$ defined over $\{a, b\}$ with $b \in S(a)$ the time slots for actions in $W(b)$ are after those in $W(a)$. A covering $\mathcal{U}$ of subsystems $S^{\leq k}(a)$ with there intersections and unions corresponds to subsystems of effects propagating

from $a$ and the intersections of these sets allows the definition of $Sch$ and subsheaves $\mathcal{P}$. All the rest of the "machinery" of the sheaves of schedules $Sch$ or $\mathcal{P}$ and their cohomology remains in place.

With $\underline{A}$ schedules being a propagation of state changes from $a^*$ consider scenarios with $\underline{q}$ as a starting state of the whole system. Indeed the $\underline{q}$ alter the effects of the system and are part of the "indexing" of the schedules. Among these sheaves are the ones that start with $\underline{q}$ or end with $\hat{\underline{q}}$ which might be undesirable states. If the cohomology of such sheaves that end in undesirable states is soft then undesirable states are easily reached. If they have cohomology with $\check{H}^n(\mathcal{P}) \neq 0$ then there are "$n$-dimensional" collections of these undesirable schedules in some local area indicating considerable degrees of freedom or lack of controls preventing such propagation to or from undesirable states.

### 7.1 Mathematical Epistemology.

$\mathfrak{j}$ and $\mathfrak{f}$ join $\underline{A}(\mathfrak{S})$ and its sheaf of models. This is critical for systems as well as epistemology as it converts the system $\mathfrak{S}$ into sets of verifiable statements - the local models. That is the core of any epistemology of systems. Furthermore this verification is localized and so a process of verification can be planned. In science this is a stream of experiments. In engineering it a set of trials that confirm that the design fulfills the requirements (the triumphs of this can be seen in aerospace research and development of a satellite or space probe that you do not get to adjust or fix once it is launched). Finally, cohomology informs as to whether verifications can extend to the whole system or whether there are degrees of freedom so that parts that are verified cannot extend to single schedule of verification for the whole systems. Choices have to be made or no extension is possible. The concerns of this section provide an approach to what should be a fruitful development of mathematical epistemology.

## 8 Ecosystems.

This section explores the application of combinatorial schema to trophic networks as the basis of ecosystem modeling[39]. Ecosystems will be seen to have features of supply chains while having considerable difference from the ones described above. It is not the intention to add to specific ecosystem models but rather position combinatorial schema as a modeling language in the spectrum of perspectives on ecosystems. We use the language of Game Theory but see games as categories. The states $S(G)$ of a $n$-person game become the objects of a category $G$. If the players are numbered 1, 2,.., $n$ and each round of play is parameterized by $t$, the plays $p_1(t)$,.., $p_n(t)$ correspond to maps $< p_1, .., p_n >: S(G) \to S(G)$. Players can be specialists and the aim is to attain a stable value of their coalition as defined below. Not all players survive to the end of the game.

An ecosystem is modeled as a directed set $A$ each element of which is a coalition which can be thought of as cooperating sets of animals such as a mating pair of birds, a clan of Hyenas or perhaps a whole herd of a migrating species. $a_1 \prec a_2$ if $a_2$ eats members of $a_1$ (This requires avoiding such relations as $a_2$ eats any of $a_1$ which eats the immature of $a_2$ when it can. For example, some $a_2$ being a bird of prey which preys on a lizard $a_1$ which eats its eggs). Each coalition $a \in A$ has a set of maturity cohorts (depending on what time intervals we wish to use in the

analysis) with a measure of its long term value to the coalition, that is $Inv(a) = Inv(a)(t) = \{e_1(t), e_2(t), ..., e_{n(a)}(t)\}$. Thus $Inv(a)$ is the set of cohorts over time and each $e(t) \in Inv(a)$ has a population at $t$ denoted by $\| e(t) \|$ and so the population of $a$ at $t$ is

$$\| a(t) \| =_{defn} \Sigma \| e_i(t) \|.$$

This becomes a supply chain with $S(a)$ being all the $b \prec a$ and $\beta(a,b)(e) = \sum_{e_i \in b} x_i.e_i$ with $x_i$ being an expected hunting or gathering amount. $W(a)$ as the maps of $Tpc(a)$ is the plays available to members of $a$. These plays are "generic". They are not specific to any particular animal. They are actions that members of the coalistion have to perform in order to survive and thrive.

There is a special element $\mathfrak{T}$ that represents available territory or range of all species in $A$. This can change with time. The measure of time, the duration of time-slots is decided for all $A$. Time slots need to be smaller than the lifetime of the least population affecting the entire system. Otherwise a population has to be considered a ecosystem parameter unaffected by plays.

$Tpc(a)$ is the set of plays (the struggle for necessities of life) each coalition plays against coalitions with which it interacts as prey, competitor or predator. This set is the *perspective* of a coalition $a$, denoted $pers(a)$ which is shorthand for $S(a) \cup \{a\} \cup S^+(a)$. The game involves actions which involve seeking food, holding territory as necessary and issuing young cohorts to become mature cohorts. A play is an "exchange":

$$p_i(t_n) : Inv(pers^2(a)(t_n)) \to Inv(pers^2(a)(t_{n+1})).$$

This of course includes plays involving only $pers(a)$. The maps are a debit and credit transfer: credit predator, debit prey or credit competing coalitions with territory and debit the losers. This can be in terms of biomass, or numbers of animals of a cohort or an exchange of territory. Ecologically, mating fights are of little interest as long the reproduction rate is maintained. Death provides a natural decline in numbers to a lower value state. Thus coalitions are like perishable inventory. Less successful hunting / foraging (ineffective replenishment) or loss of territory (others have successful plays against the coalition) results in loss of coalition value. The evolution of the system $A$ can be a permanent change of players.

The plays of each $a$ are located in $\mathfrak{T}$. Where this arena of competition is important we write $pers(a)/\mathfrak{T}$. Thus $Tpc(a)$ has objects $Inv(pers(a)(t))/\mathfrak{T}$ with the maps of $Tpc(a)$ being plays.

Assume that for $a$ there is a finite number of plays (or plays of given type) that can occur in any time period. This is $\gamma(a)$ and so for a time period $t$ the plays can be $p_1, p_2, ..., p_m$ where $m \leq \gamma(a)$. As each play sets the scene for succeeding plays they can be composed and in the usual way we define $W(a)$ the ring over the monoid of plays. A schedule is then a map $\xi : T \to W(a)^{\gamma(a)}$. Although we can define schedules for each $a$ they require $pers(a)$ to be consistent; to take into account of both sides of an exchange. As defined $Sch(a)$ forms the usual group.

Schedules are easily seen to form presheaves provided that any exchange resulting from a play involves only two coalitions. If a play $p$ involves a credit to $a_1$ and balancing debits to $a_2$ and $a_3$, a restriction $\{a_1, a_2, a_3\}$ to $\{a_1, a_2\}$ cannot be defined. Thus we do not allow "split exchanges". Let $\mathcal{U}$ be chosen so that $U \in \mathcal{U}$ and $U \cap pers^n(a) \neq \emptyset \Rightarrow U \cap pers^{n-1}(a) \neq \emptyset$. (This implies that the sets in $\mathcal{U}$ are connected

in an obvious sense.) If $\xi_i$ defined on $U_i \in \mathcal{U}$ are equal on any pair of intersections $U_i \cap U_j$ then the plays involving the coalitions in the intersection are the same and the exchanges balance. In $U_i \setminus (U_i \cap U_j)$ the plays in $\xi_i$ do not involve split exchanges. This holds for $\xi_j$ and hence all the plays across $U_i \cup U_j$ are not split exchanges. A schedule defined on $U = \underset{i}{\cup} U_i$ can be constructed. This proves:

**Proposition 8.1.** *Sch is a group valued sheaf.*

The ecosystem model differs from the supply chain in two essential respects. There is rarely a single top element $a^*$ that is connected by iterations of bill of materials to all of $A$ and each $a$ acts like a a business competing for supplies in order to kept its employees busy creating inventory which it accumulates but does not sell. The difference being that each $a$ is a society with its own "economy".

As with the previous applications the description of the dynamics is in the relationship of sub-sheaves $\mathcal{P}$ of $Sch$ and a classes of statements $\pi$ that have well defined models $mdl(\pi)$. Lokta-Volterra models [30, 20] have been the standard "dynamics" of ecosystems. They involve population dynamics arising from estimates of growth and death rates of interacting species. The other end of the scale of modeling resolution is the Individual Based Models or "IBM" [18] [8]. These are simulations of individuals in populations and are very specific. As stressed before, the modeling here is more generic than IBM but more specific than Lokta-Volterra dynamics. In order to use population statistics for species as the basis of a class of statements $\pi$ we need to define the model sheaf. $\underline{A}$ has a natural map that takes each coalition $a$ to its species $a^s \in A^s$. The models will be the set of interaction models (the simplest being the Lokta-Volterra (LV) class of models). An $n$-species model of a interacting species is an $n \times n$ matrix representing interactions among $n$ species. If $\vec{s} =< s_i, ..., s_n >$ is the vector of populations of interacting species for an interaction matrix $M(\vec{s})$ then each term $m(ij)$ of $M$ is a coefficient in a difference equation that is an exchange between the $s_i$ and $s_j$ species. Hence with discrete times being $t_n$, $n \in \mathbb{Z}$.

$$\vec{s}(t_{n+1}) = M.\vec{s}(t_n).$$

There a various ways of passing to a continuous models where time is a real number. In the case of continuous models the $n \times n$ matrix is the differential operator on an associated tangent space [20].

**Definition**. The category of Lokta-Volterra models for $\underline{A}$, $mdl_{LV}$, has objects matrices $M(\vec{s})$ where $\vec{s} =< s_i, ..., s_n >$ is a set in $A^s$. More precisely, objects of $mdl_{LV}$ are functions $M : U^s \times U^s \to \mathbb{R}$. The morphisms in $mdl_{LV}$ are extensions, wherein $i : V \rightarrowtail U \in \mathcal{U}$ and $M_1 : U^s \times U^s \to \mathbb{R}$, $M_2 : V^s \times V^s \to \mathbb{R}$ implies $M_1 \circ (i \times i) = M_2$. Furthermore $M_2$ is a restriction of $M_1$.

The definition of the restriction reflects the intentions of constructing models. Models of ecosystems cannot contain all the species. If a model of a small set of species is to match observations in any way its matrix elements must correspond to what is going on among the species in the model and the rest of the species in the ecosystem. In effect any realistic model with few species implies something about the matrix elements in some larger model. The growth rates in the restricted models usually have the effects of excluded species built into their interactions rates; for example predation by other species of prey is modeled by an adjusted lower kill /per energy expended.

[8] See http://www.red3d.com/cwr/ibm.html for available listings of simulations programs

**Definition**. A Lokta-Volterra valued presheaf is a presheaf from a cover $\mathcal{U}$ of $\underline{A}$ to $mdl_{LV}$.

Why would there not be just one model? In fact there are a large number of contingent conditions including climate, soil conditions, territory size and carrying capacity of the system all of which affect growth rates. Define the category $\mathcal{X}$ to have objects (connected) homotopy zero open sets of $\mathbb{R}^n$ with non-zero measure; the morphism of $\mathcal{X}$ are inclusions. A model $M(U^s)$ assumes that various contingent parameters are in a given range called the support of $M$. This gives a map $spt : mdl_{LV} \to \mathcal{X}$. For an object $\mathcal{Y} \in \mathcal{X}$ $spt^{-1}(\mathcal{Y})$ is the set of models that can be maintained in the conditions $\mathcal{Y}$. For a $mdl_{LV}$ presheaf, $M$, $spt : M \to \mathcal{X}$ can be defined. The objects of $\mathcal{X}$ index $mdl_{LV}$ presheaves. This approach has its origin in "Dynamic structures"[26] but in the context of combinatorial modules, ideally, this should be stack (as well)[12, 10]. The objects of this category are $mdl_{LV}$ valued presheaves, $M$, equipped with maps $spt : M \to \mathcal{X}$ defined for each $U \in \mathcal{U}$ so that $spt(M(U^s)) = \mathcal{Y}$ meaning that for the set of species in $U^s$, $U \in \mathcal{U}$, the model $M(U)$ assumes support parameters are in $\mathcal{Y}$. A morphism $\phi$ from $M_1 \xrightarrow{spt} \mathcal{X}$ to $M_2 \xrightarrow{spt} \mathcal{X}$ is a commuting triangle $spt_2 \circ \phi \subseteq spt_2$ such that in the following diagram $M_1$ is isomorphic to a pullback

$$\begin{array}{ccc} M_1 & \xrightarrow{\phi} & M_2 \\ \downarrow{\scriptstyle spt} & & \downarrow{\scriptstyle spt} \\ \mathcal{Y}_1 & \xrightarrow{\subset} & \mathcal{Y}_2. \end{array}$$

In this diagram if the morphism $\phi$ is a restriction then we would expect the support parameters for more species to be a smaller set than for fewer species as support will be the intersection of the support sets for the individual species. Parameterizing models and model presheaves (and hence sheaves) by support sets $Y \in \mathcal{X}$ exposes the difference among models. Furthermore the stack structure separates models into equivalences classes defined by groupoids of isomorphisms (cf *coarse moduli* and *fine moduli* [12]). Denote the stack of these parameterized models by $LV_{stk}$, the Lokta-Volterra stack. This serves also as the category of "local Lokta-Volterra models" when $(spt : M \to \mathcal{X})$ is localized to $(spt : M(U^s) \mapsto \mathcal{Y})$. In this case write $(spt : M(U^s) \mapsto \mathcal{Y}) \in LV_{stk}(U^s)$.

The class of propositions $\pi$ is that of trajectories of local Lokta-Volterra models for the system in the context of a support parameters.

An element $\xi$ of $Sch(\underline{A})$ defined on $U$ at time $t_n$ involves species $\xi^s(t_n)$. As species can contain many coalitions and cohorts we scale the "sample" $\parallel \xi(t_n) \parallel$ against the measure of biomass or count $\xi^s(t_n)$. Let the number of coalitions for a species $s$ be denoted $\bar{C}(s)$. This is a known quantity used to define $\underline{A}$. In what follows we shall freely identify a vector $\vec{s}$ with its set of component species. For a model $M$ let $a^s(M)(t_n)$ be the value of the species $a^s$ at time $t_n$.

**Definition**: Given a real number $\varepsilon > 0$, let $p(a(M)(t_n))$ be the statement

$$(|\parallel a(t_n) \parallel - a^s(M)(t_n)/\bar{C}(a^s) \mid) < \varepsilon.$$

Define $\mathfrak{L}(\varepsilon)(U) \rightarrowtail \mathcal{P}(U) \times LV_{stk}(U^s)$ by

$$(\xi(U), M(\vec{s})) \in \mathfrak{L}(\varepsilon) \Leftrightarrow (\vec{s} = U^s) \wedge (\forall t_n \in dom(\xi))(\forall a \in range(\xi(t_n))) \underset{a^s \in \vec{s}}{\wedge} p(a(M)(t_n))).$$

That is $\mathfrak{L}(\varepsilon)(U)$ identifies those schedules that are within $\varepsilon$ of the trajectories of $M$ at least for the relevant species.

This amounts to imposing an operator topology on $\mathfrak{L}(\varepsilon)$. The schedule becomes a data-structure, in fact a polynomial of an action algebra, that acts as a test of a dynamic system and $\mathfrak{L}(\varepsilon)$ re-introduces the phase portrait back into the picture. For $\mathfrak{L}$ to be a sheaf it must be a presheaf; there must be a restriction map $\mathfrak{L}(\varepsilon)(U) \to \mathfrak{L}(\varepsilon)(V)$. $\xi \in \mathcal{P}(U)$, $(\xi, M(\vec{s}))$ is in $\mathfrak{L}(\varepsilon)(U)$ by virtue of being within $\varepsilon$ of the population dynamics given by $M$ for each $a \in range(\xi)$ for the requisite time. This remains true for the subset of coalitions of species in $V^s$. For a sets $U_i^s$ such that $(\xi_i, M_i) \in \mathfrak{L}(\varepsilon)(U_i^s)$ are equal on $U_i^s \cap U_j^s$ then on the species in $U_i^s \cap U_j^s$ they have the same phase portrait as $M_i \mid U_i^s \cap U_j^s = M_j \mid U_i^s \cap U_j^s$ and also in $(U_i \cup U_j) \setminus (U_i^s \cap U_j^s)$. These phase portraits are consistent with the schedules $\xi_i$ and $\xi_j$ given an overall consistency for the species in $(U_i \cup U_j)$. This is true for any pair $U_i^s \cap U_j^s$ and so we conclude there is $(\bar{\xi}, \bar{M}) \in \mathfrak{L}(\varepsilon)(\underset{i}{\cup} U_i)$ that restricts to the individual $(\xi_i, M_i) \in \mathfrak{L}(\varepsilon)(U_i)$. This proves:

**Proposition 8.2.** *$\mathfrak{L}(\varepsilon)$ is a sheaf.*

**Definition** For a real number $\varepsilon$ and a species $s$ in a model $M$

1. $X(\varepsilon)(M)(U)$ is the set of $\xi \in \mathcal{P}(U)$ such that for all species in $U$ $(\xi, M(\vec{s})) \in \mathfrak{L}(\varepsilon)(U)$.
2. $\mathcal{P}_{\mathfrak{L}}$ is the presheaf with $\mathcal{P}_{\mathfrak{L}}(U)$ the set of non-empty $X(\varepsilon)(M)(U)$.
3. $Y(\varepsilon)(\xi)(U)$ is the set of $M(\vec{s}) \in LV_{stk}(U)$ such that for all species in $U$ $(\xi, M(\vec{s})) \in \mathfrak{L}(\varepsilon)(U)$.
4. $mdl(\pi)_{\mathfrak{L}}$ is the presheaf with $mdl(\pi)_{\mathfrak{L}}(U)$ the set of non-empty $Y(\varepsilon)(\xi)(U)$.
5. $\mathfrak{j}_\varepsilon : \mathcal{P}_{\mathfrak{L}} \to mdl(\pi)_{\mathfrak{L}}$ is defined as $\xi \mapsto Y(\varepsilon)(\xi)$.
6. $\mathfrak{f}_\varepsilon : mdl(\pi)_{\mathfrak{L}} \to \mathcal{P}_{\mathfrak{L}}$ is defined as $M \mapsto X(\varepsilon)(M)$.

The presheaves $\mathcal{P}_{\mathfrak{L}}$ and $mdl(\pi)_{\mathfrak{L}}$ are sheaves by the construction in section 3. They do not have a natural group structure. Although matrices in $M \in LV_{stk}(U)$ for any $U$ form a group they are not usually useful in models especially when the carrying capacity of the ecosystem is included (for example, $M_1 = M - \delta$ and $M_2 = M + \delta$. $M_1 + M_2 = 2M$. $\vec{s}(t_{n+r}) = (2.M)^r(\vec{s}(t_n))$. Instead, for a given model $M$ we use the set $X(M) \subset Sch$ to define the least group-valued sheaf $\mathcal{P} \subseteq Sch$ containing $X(M)$, denoted $\mathcal{P}/M$, to study the associated cohomology of $M$. $\check{H}^n(\mathcal{P}/M)$ can be regarded as an $M$-adic cohomology of $\mathcal{P}$.

Dynamic models of ecosystems with more than three or four species are hard to study. Especially when chaos appears. The study of sheaves $\mathcal{P}$ or $\mathcal{P}/M$ gives a different conceptual approach based on the behavioral options of species. Concentrating on sheaves that have desirable cohomological properties (not necessarily just Čech cohomology) might allow insights that can capture something of the way dynamic models can be extended to a greater number of species.

What does cohomology tells us? The simplest case is when coalitions do not interact: $U_i = pers(a_i)$ and $a_i$ and $a_j$ do not interact then $U_{ij} = U_i \cap U_j = \emptyset$ and no cohomology classes arise from this. Consider now "weakly interacting" species. The following is a fictitious example. $a_1$ = ferrets, $a_2$ = foxes, $a_3$ = hawks. Ferrets and foxes do not overlap but hawk territory covers both territories. All hunt common

prey so $U_{ij}$ contain hawks and hawk prey and so therefore does $U_{ijk} = U_i \cap U_j \cap U_k$. $U_{ijk}$ contains a subset of hawk prey species for which we suppose $(\xi_{ij} - \xi_{ik} + \xi_{jk}) \mid U_{ijk} = 0 \Rightarrow \xi_{ij} + \xi_{jk} \mid U_{ijk} = \xi_{ik} \mid U_{ijk}$. Hawk and ferret prey hunts + hawk and fox prey hunts = ferret and fox prey hunts suggesting the hawk is the carrion feeder from the ferret and fox prey. In general cocycles can only come from identification of actions. Typically an action by one coalition is a equated with the opposite side of the debit or credit exchange with actions of another coalition. Replace the hawk with a snake which attacks the unprotected nestlings or eggs of birds taken by foxes or ferrets. Each successful hunt by a four legged predator is equivalent to predation of two eggs by a snake. For larger terrestrial, but not marine, ecosystems the iterated perspectives $pers^n(a_i)$ of predators have a common intersections in flora - the food of the prey. Cocycles arise most likely from coordinated actions such as flowering for insect attraction and pollination, in this case showing the long-range effect of many dispersed "suppliers" in a supply chain.

The approach will be to study a system $\underline{A}$ with a suitable subsheaf $\mathcal{P}$ of $Sch$. Assume that the usual cover $\mathcal{U}$ has a subset $\mathcal{U}_0$ such that for $V \in \mathcal{U}_0$ there are creditable dynamic models $LV_{stk}(V)$. Although $\mathcal{U}_0$ is unlikely to be a cover we assume that there are full models for the ecosystem - it is observed to be a functioning ecosystem after all - so an object $\mathfrak{M} \in LV_{stk}$ is defined for all $U \in \mathcal{U}$. The maps $\mathfrak{p}$ and $\mathfrak{q}$ exist in theory so we use the cohomology of $\mathcal{P}$ to extend the models in $\mathfrak{M}(V)$ using the data in $\mathcal{P}$ as a guide and a constraint. The simplest case is a model $M$ so that $\mathcal{P}/M$ is soft so for $n > 0$ $\check{H}^n(\mathcal{P}/M) = 0$ then, even though $M$ is defined only on some $U$, every $\xi \in X(M)$ has the potential to extend to the entire $A$. The existence of global sections in $\mathfrak{M}$ is then equivalent to the existence of $\mathfrak{L}$ and $\mathfrak{p}$ and $\mathfrak{q}$. This is an example of systems dynamics integrating interacting dynamics with $\xi$ being a propagator.

### 8.1 Return to supply chains

The ecosystem interpretation can be reapplied to supply chains. Indeed it is a "scaling up" of the supply chain interpretation. Each coalition can be interpreted as a business $a$ that has to maneuver in $pers(a)$ for its supplies from $S(a)$ while defending its market share - its territory. The elements of $S^+(a)$ are in fact part of that territory. Its plays are against those $a'$ with $S(a) \cap S(a') \neq \emptyset$ and $S^+(a) \cap S^+(a') \neq \emptyset$ that compete for supplies and customers. The ecosystem becomes a set of organizations in a competition for resources and markets. That is, an economy. Further adding support space generalizes this with parameters that are conditions prevailing in the larger economy such as exchange, interest, tax and transport rates.

## 9 Further Concepts

In algebraic geometry a moduli space is a space that indexes all algebraic curves of a selected class in such a way that the moduli space contains all the values a chosen invariant can take [19].

**Definition**. A *moduli sheaf* is a sheaf $\Psi$ defined on $\underline{A}$ for which there is a subcategory $\mathfrak{C}$ of $Comb(\underline{A})$ such that each object $\mathfrak{X}$ in $\mathfrak{C}$ is associated with a section of $\Psi$. That is, there is a map $\psi : Obj(\mathfrak{C}) \to \Psi$ such that $\psi(\mathfrak{X})(U) \in \Gamma(U, \Psi)$.

Observe as a matter of definitions that

**Proposition 9.1.** *1. $Prv$ is the moduli sheaf for the category of sheaves $Sch(\sigma)$.*

*2. $mdl_{\mathfrak{C}}$ is the moduli sheaf for the category of sheaves $\mathcal{P}'$ containing $Y(\xi)$.*

**Proof**.

1. $Sch(\sigma)(U)$ is the set $Z(\sigma)$ in $Prv(U)$.

2. $\mu \in mdl_{\mathfrak{C}}(U)$ is associated with $X(\mu) \in \mathcal{P}_{\mathfrak{C}}(U)$ which can be associated with $\mathcal{P}'$ with $X(\mu) \subseteq \mathcal{P}'(U)$.

Fragmentation can therefore be recast as the "internal dynamics" of $Prv$ which relates to the disappearing sheaves $Sch(\sigma)$ through the epistemological operator $\Rightarrow$ conceived of as a sheaf. Is a moduli sheaf always associated with an epistemological operator? Or is it more general? Is the idea of a moduli sheaf closer to what a epistemological operator *should* be? Answering these questions will require significant exploration of these ideas.

# 10 Conclusion

Response systems and supply chains are usually the subjects of optimization and simulation as are ecosystems. This paper sees them as algebraic structures with sheaves of polynomials providing dynamic information. This provides a useful approach to questions of structure that are not usually formulated in simulation studies. Although testing information systems does look at sets of states it does not usually look at the structure of establishing sets of scenarios. This "epistemological structure" is formulated algebraically giving it a precise, tractable mode of analysis. This papers contribution is to bring an novel algebraic perspective to models of economically significant social structures and activities that stresses the algebra of the phenomena they present.

# References

[1] R. Alur and D.L. Dill. *Automata-theoretic verification of real-time systems.* Trends in Software Series. John Wiley and Sons, 1995.

[2] R. Alur and T.A. Henzinger. Reactive modules. In *Proceedings of the 11th IEEE Symposium on Logic in Computer Science*, pages 207–218, 1996.

[3] D.J. Amit. *Field Theory, The Renormalisation Group and Critical Phenomena.* World Scientific, 1978.

[4] T. Ball. Abstract guided test generation: A case study. Microsoft research technical report MSR-TR-2003-86. 2003.

[5] M. Barnett and W. Schulte. Abc of specification: Asml, behaviour and, components. *Informatica*, 25(4):517–526, 2001.

[6] B.M. Beamon. Supply chain design and analysis: Models and methods. *International Journal of Production Economics*, 55:281–294, 1998.

[7] A. Biere. Tutorial on model checking, modelling and verification in computer science. In M. Rosenkranz K. Horimoto, G. Regensburger and H. Yoshida, editors, *Algebraic Biology Third International Conference, AB 2008*, Lecture Notes in Computer Science Vol 5147, pages 16 – 21. Springer-Verlag, 2008.

[8] G.P. Cachon and S. Netessine. *Game Theory in Supply Chain Analysis*. Kluwer Academic, 2004.

[9] S. Cannellaa and E. Cianciminoa. On the bullwhip avoidance phase: supply chain collaboration and order smoothing. *International Journal of Production Research*, 48(22):6739 – 6776, 2010.

[10] A. Canonaco. *Introduction to algebraic stacks*. http://www.mat.uniroma1.it/ncerca/seminari/geo-alg/dispense/stacks.pdf, 2004.

[11] P. Coursot. Methods and logics for proving programs. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science, Volume B Formal Models and Semantics*, pages 843 – 993. Elsevier, 1990.

[12] D. Edidin. What is a stack. *Notices of the American Mathematical Society*, 50(4):458–459, 2003.

[13] H.B. Enderton. Elements of recursion theory. In Jon Barwise, editor, *Handbook of Mathematical Logic*, pages 527 – 566. North Holland, 1977.

[14] W. Grieskamp et al. Generating finite state machines from abstract state machines. msr-tr-2001-97. *ACMs Digital Library (http://www.acm.org/dl/*, page 19, 2002.

[15] J.W. Forrester. *Industrial Dynamics*. The M.I.T. Press, 1961.

[16] R. Ganeshan and et al. *A Taxonomic Review of Supply Chain Management Research*. Springer, 2000.

[17] J-C. Grgoire, G. Holzmann, and D. Peled. *The Spin Verification System*, volume 32 of *DIMACS, Discrete Mathematics and Theoretical Computer Science*. American Mathematical Society, 1997.

[18] V. Grimm and S.F. Railsback. *Individual-based modelling and ecology*. Princeton Series in Theoretical Biology. Princeton University Press, 2005.

[19] Robin Hartshorne. *Algebraic Geometry*. Springer Verlag, 1977.

[20] M.W. Hirsch and S.Smale. *Differential Equations, Dynamical Systesm, and Linear Algebra*. Academic Press, 1974.

[21] G. Ivan. Special morphism of groupoids. *Novi Sad, J Math*, 31(2):23–36, 2002.

[22] Y.V. Joshi. Information visibility and its effect on supply chain dynamics. *Mechanical Engineering, Boston, MIT*, 67, 2000.

[23] J.P.C. Kleijnen. Supply chain simulation: A survey. *International Journal of Simulation and process Modelling*, 1(1), 2003.

[24] G. Knolmayer, P. Mertens, and et al. *Supply Chain Management Based on the SAP Systems R/3 4.6 APO 3.0 Order Management in Manufacturing Companies*. Springer, 2002.

[25] S. Lang. *Algebra*. Addison Wesley, 1969.

[26] A. Macfarlane. Dynamic structure theory, a structural approach to social and biological systems. *Bulletin of Mathematical Biology*, 43(5):579 – 591, 1981.

[27] A. Macfarlane. *Combinatorial Modules and Schema and their and Cohomology with Applications in Testing software and Supply Chains.* http://IntegratedExpertise.org, 2010.

[28] D. MacKenzie. *Mechanizing Proof, Computing, Risk, and Trust.* MIT Press, 2001.

[29] E.J. Magad and J. Amos. *Total Material Management.* Van Nostrand, 1989.

[30] R.M. May. Models for two interacting populations. In R.M. May, editor, *Theoretical Ecology, Principles and Applications*, pages 49 – 70. Blackwell Scientific, 1976.

[31] H. Min and G. Zhou. Supply chain modelling: past present and future. *Computers and Industrial Engineering*, 43:231–249, 2002.

[32] T. Nipkow. *Hoare Logics in Isabelle/HOL*, volume 62 of *NATO Science Series.* Kluwer, 2002.

[33] W. Perry. *Effective Methods for Software Testing.* John Wiley, 1995.

[34] M R. Alur, McDougall, and Z. Yang. Exploiting behavioural hierarchy for efficient model checking. In E. Brinksma and Guldstrand L. K., editors, *Computer Aided Verification*, volume 2404 of *Lecture Notes in Computer Science*, pages 338 – 342. Springer Verlag, 2002.

[35] A. Salomaa. Formal languages and power series. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science, Volume B Formal Models and Semantics*, pages 105 – 132. Elsevier, 1990.

[36] A.M. Sarmiento and R. Nagi. Recent directions in integrated analysis of manufacturing-distribution systems. *IIE Transactions on Scheduling and Logistics, special issue on Manufacturing Logistics*, 31(11):1061–1074, 1999.

[37] S. R. Smit. Evaluation model for the design of distribution networks. *Beta Working Paper Series*, (68), 2001.

[38] E.H Spanier. *Algebraic Topology.* McGraw Hill, 1966.

[39] R.E. Ulanowicz and J.J. Kay. *NETWRK: A package for the analysis of ecosystem flow networks.* Online, 2003.

View publication stats