

# **CSER 2006 – Conference on Systems Engineering Research**

## **Systems Engineering (SE) Patterns and Pattern Language**

Joseph J. Simpson

Mary J. Simpson

April 8, 2006

# Overview

---

- **Pattern Form**
- **General Systems Pattern**
- **Representative Patterns**
  - **Problem vs. Solution**
  - **Three System Minimum**
  - **FRAT**
- **Behavior Outcomes Matrix**
- **Exploring Relationships**
  - **Laws of Language**
  - **System Language Design Capability**
  - **Sequential Frameworks**
  - **Abstraction Stacks**
  - **Relationship Types**
- **Summary**

# Pattern Form

## Evocative Name

**Problem Statement** that presents difficulty, uncertainty in situation

**Forces** or tensions that influence pattern application

**Context** within which the pattern will be applied

**Related Patterns** may be connected in some way to current pattern

**Solution** that resolves the problem within the given context

# General Systems Pattern

## General Systems (GS)

**Problem:** Entropy and chaos exist for multiple facets of society as a result of increasingly complex activities

**Forces:** Different values in global community create tensions regarding inputs, outputs, boundaries, applicable transforms

**Context:** Complex sociotechnical, social, and industrial activities require shared approach to reduce entropy and chaos

**Related Patterns:** The general systems pattern will relate to all the patterns that are presented in this paper

**Solution:** Apply a systems approach to reduce entropy, complexity, and start to resolve core issues

# Problem vs. Solution Pattern

## Problem vs. Solution

**Problem:** Lack of shared understanding of problem leads to preconceived notions that fail to resolve issues

**Forces:** Differing experiences dictate wide gaps in expectations, assumptions, and needed solutions

**Context:** System users/owners must agree on the problem and consent to a common solution

**Related Patterns:** General Systems Pattern, Three System Minimum Pattern

**Solution:** Address multiple aspects of the problem with participants prior to articulating potential design solutions

# Three System Minimum Pattern

## Three System Minimum

**Problem:** Failure to clearly define 3 systems creates needless complexity, cohesion and binding between problem & solution

**Forces:** Differing points of view & interactions encountered with system user/owners

**Context:** The three system minimum pattern is used in contiguous form throughout system design

**Related Patterns:** General Systems Pattern, Problem vs. Solution Pattern

**Solution:** Define explicit systems to consider the final product, the process used to produce the product, & the environment

# FRAT Pattern

## FRAT

**Problem:** Failure to adequately identify problems and define terms used in solutions at each successive level

**Forces:** Environment provides differing perspectives and values for possible system solutions

**Context:** The FRAT pattern is applied within the constraints of the 3 System Minimum pattern

**Related Patterns:** Problem vs. Solution, General Systems, 3 System Minimum, CCFRAT

**Solution:** Explicitly define problem to be solved and terms used in the solution process

# Warfield – Behavior Outcomes Matrix

		Outcomes		
		<i>Problem System</i>		<i>Solution System</i>
		Description	Diagnosis	Prescription (Design)
				Implementation
Behavior	<b>Process</b>	<ul style="list-style-type: none"> <li>• Limits</li> <li>• Triadic Necessity &amp; Sufficiency</li> <li>• Universal Priors</li> </ul>	<ul style="list-style-type: none"> <li>• Success &amp; Failure</li> <li>• Universal Priors</li> </ul>	<ul style="list-style-type: none"> <li>• Gradation</li> <li>• Validation</li> </ul>
	<b>Individual</b>	<ul style="list-style-type: none"> <li>• Limits</li> <li>• Triadic Compatibility</li> <li>• Small Displays</li> </ul>		<ul style="list-style-type: none"> <li>• Requisite Parsimony</li> <li>• Requisite Saliency</li> </ul>
	<b>Group</b>	<ul style="list-style-type: none"> <li>• Limits</li> <li>• Uncorrelated Extremes</li> </ul>	<ul style="list-style-type: none"> <li>• Inherent Conflict</li> <li>• Structural Underconceptualization</li> <li>• Diverse Beliefs</li> </ul>	<ul style="list-style-type: none"> <li>• Requisite Variety</li> <li>• Induced Groupthink</li> </ul>
	<b>Organizational</b>	<ul style="list-style-type: none"> <li>• Limits</li> <li>• Organizational Linguistics</li> <li>• Vertical Incoherence</li> </ul>	<ul style="list-style-type: none"> <li>• Forced Substitution</li> <li>• Precluded Resolution</li> <li>• Vertical Incoherence</li> </ul>	



# First and Second Laws of Language

---

**1. All communication takes place in shared contextual space**, subject to a fairly complex process of disambiguation, depending on the conditions inherent in the other five Laws. (Six laws total)

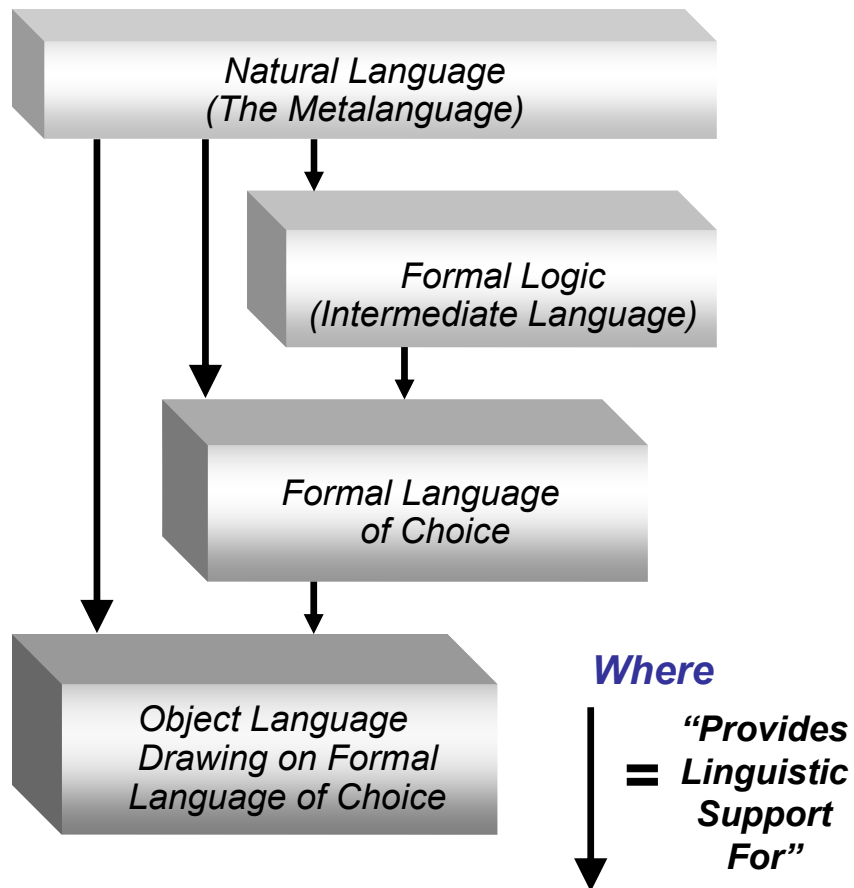
## **2. The Law of Variable Context**

If two people share sufficient context, almost any words, including sheer nonsense--or no words at all--will suffice for them to communicate with each other. If two people do not share sufficient context, then not all the words in the world may be enough for them to grasp each other's meaning. Where intermediate degrees of partial, fragmented, or otherwise limited or "noise-distorted" context are shared, communication will be proportionately difficult and/or unsuccessful.

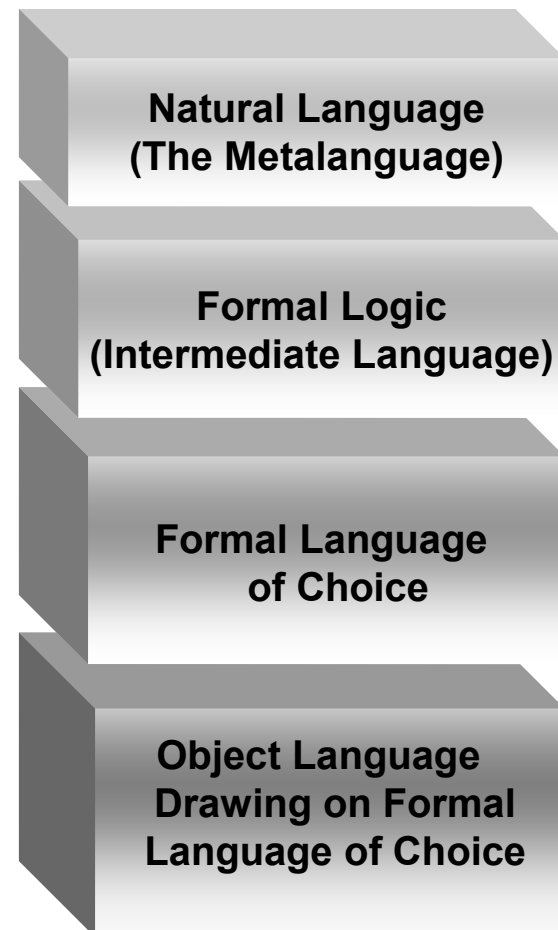
Alexander Gross <http://language.home.sprynet.com/>

# Systems Language Design Capability

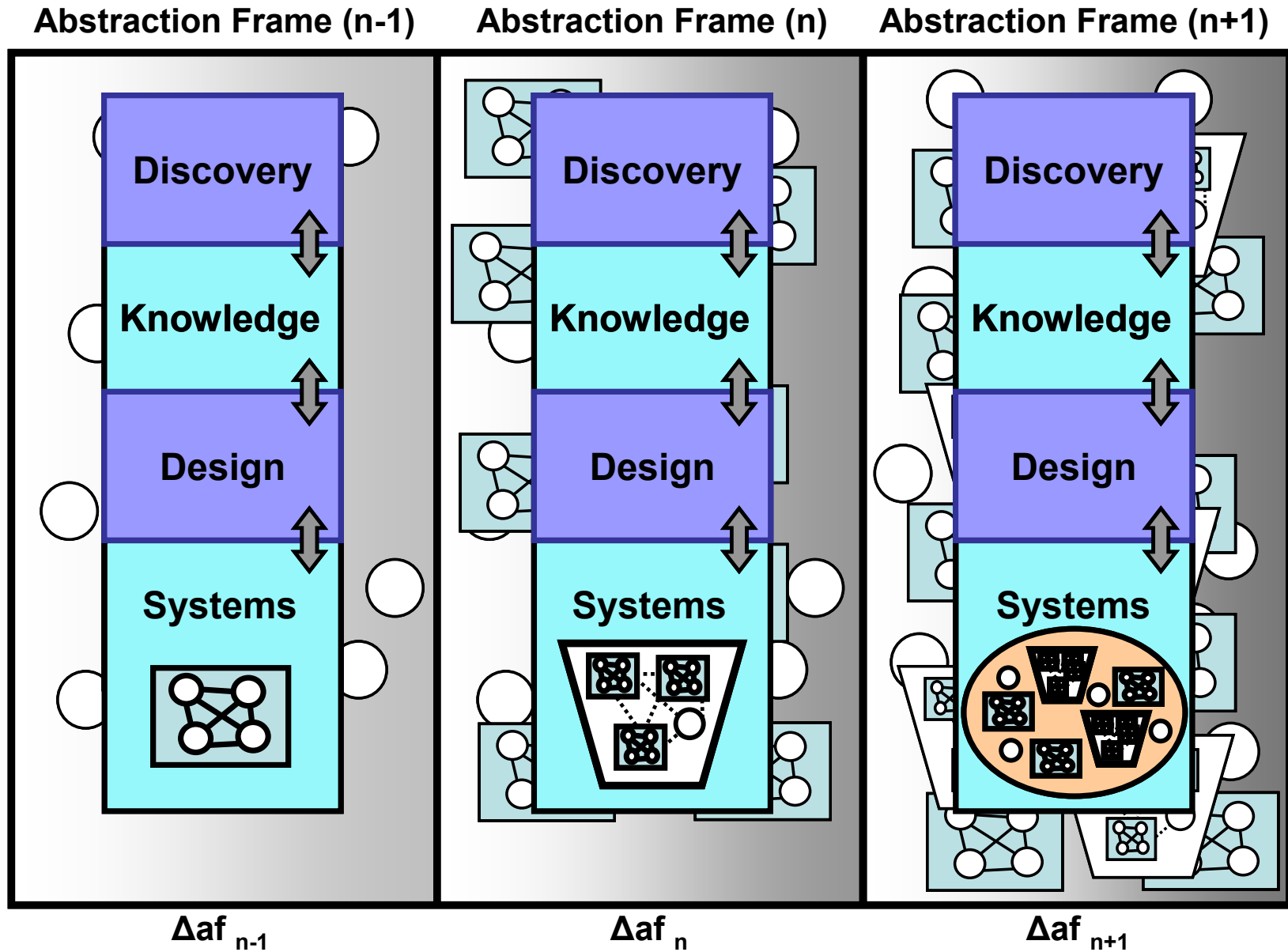
## Initial Inter-Relationships Defined: “Pattern of Infrastructure”



## In Terms of an Abstraction Stack

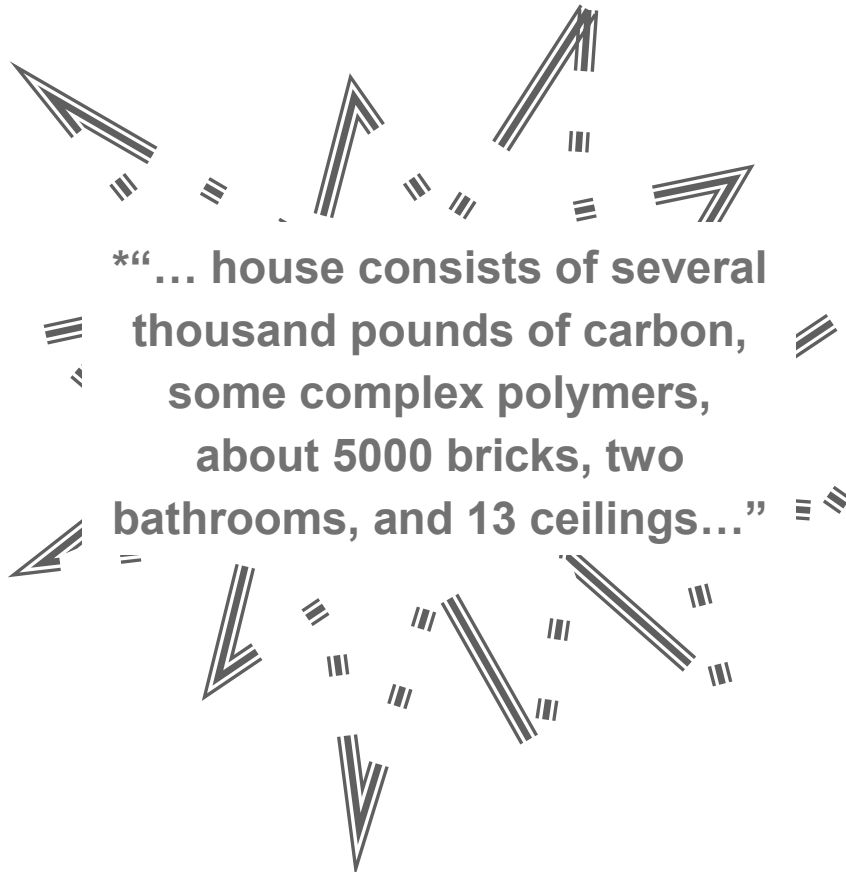


# Sequential Frameworks

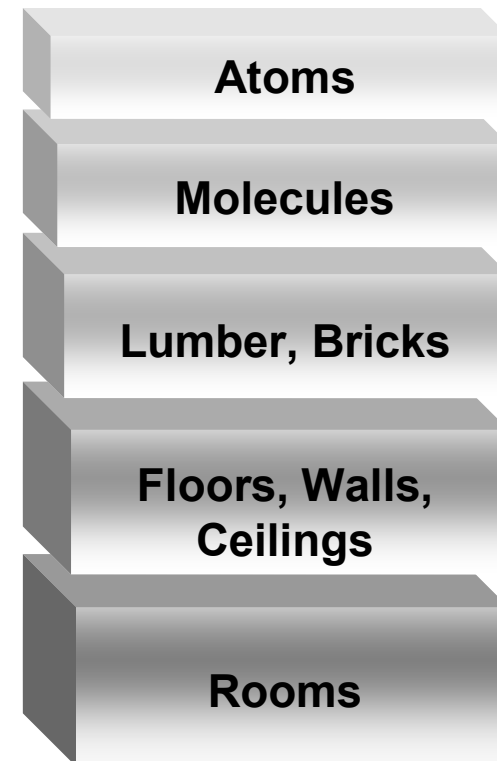


# Abstraction Stacks

A House Consists of:

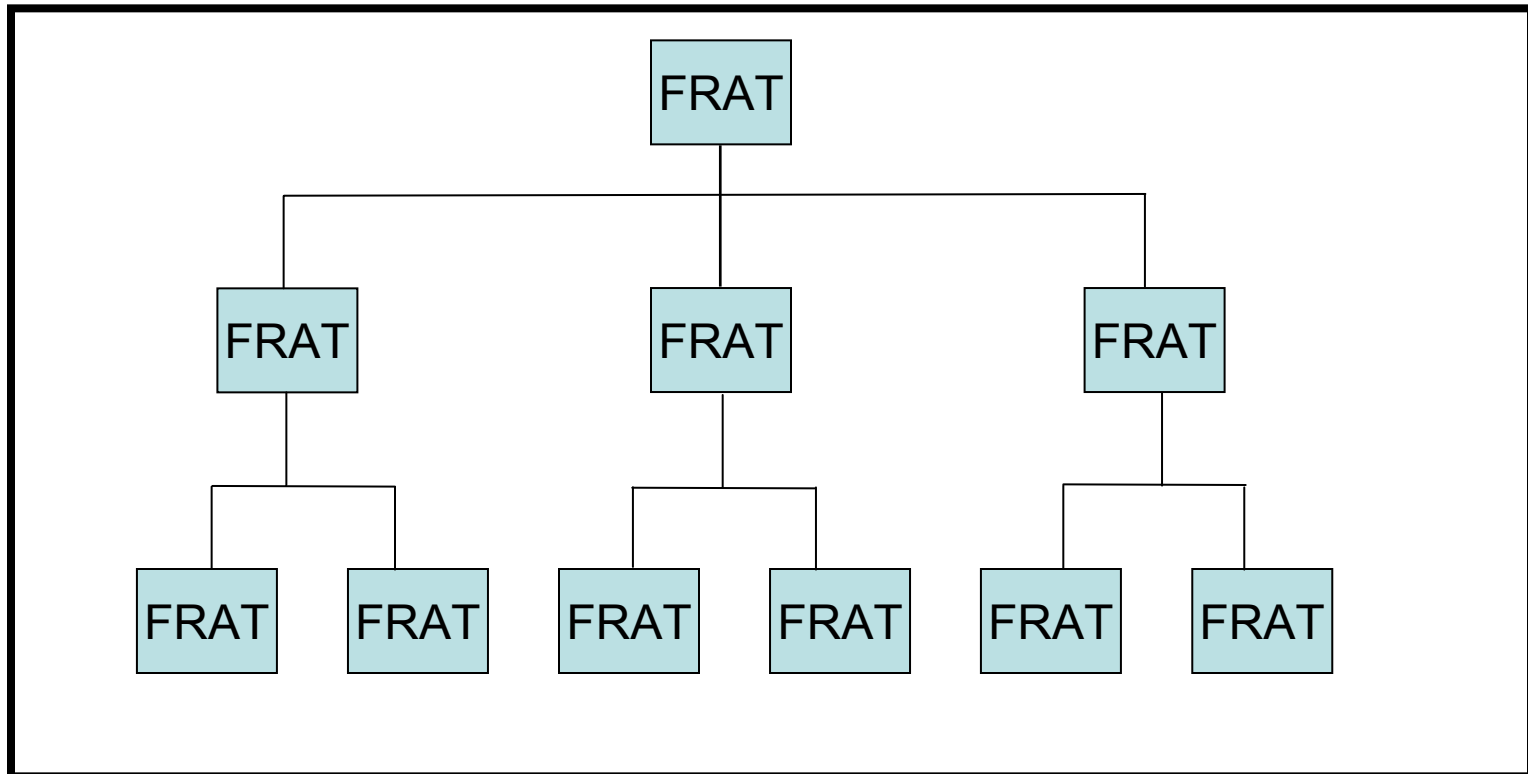


Use of Abstraction ‘Stacks’



# Relationship Types

**Hierarchical Relationship Type:** “..content can be almost completely disassociated from a hierarchy, but there does not seem to be any way to disassociate structure from it and still portray it.” *Warfield 2003.*



# Summary

---

Well established and understood systems engineering patterns provide the foundation for the communication of complex systems concepts. A core set of systems engineering patterns has been presented to stimulate the development and discussion of systems engineering patterns. The 'related patterns' component is a necessary component of the pattern template and provides the basis for the construction of pattern languages using these related patterns.

Systems engineering patterns are one of the first steps in the representation of shared systems engineering context. The greater the span and depth of the shared systems engineering context, the greater the potential for precise systems engineering communication.