Carnegie Mellon
Software Engineering Institute

Pittsburgh, PA 15213-3890

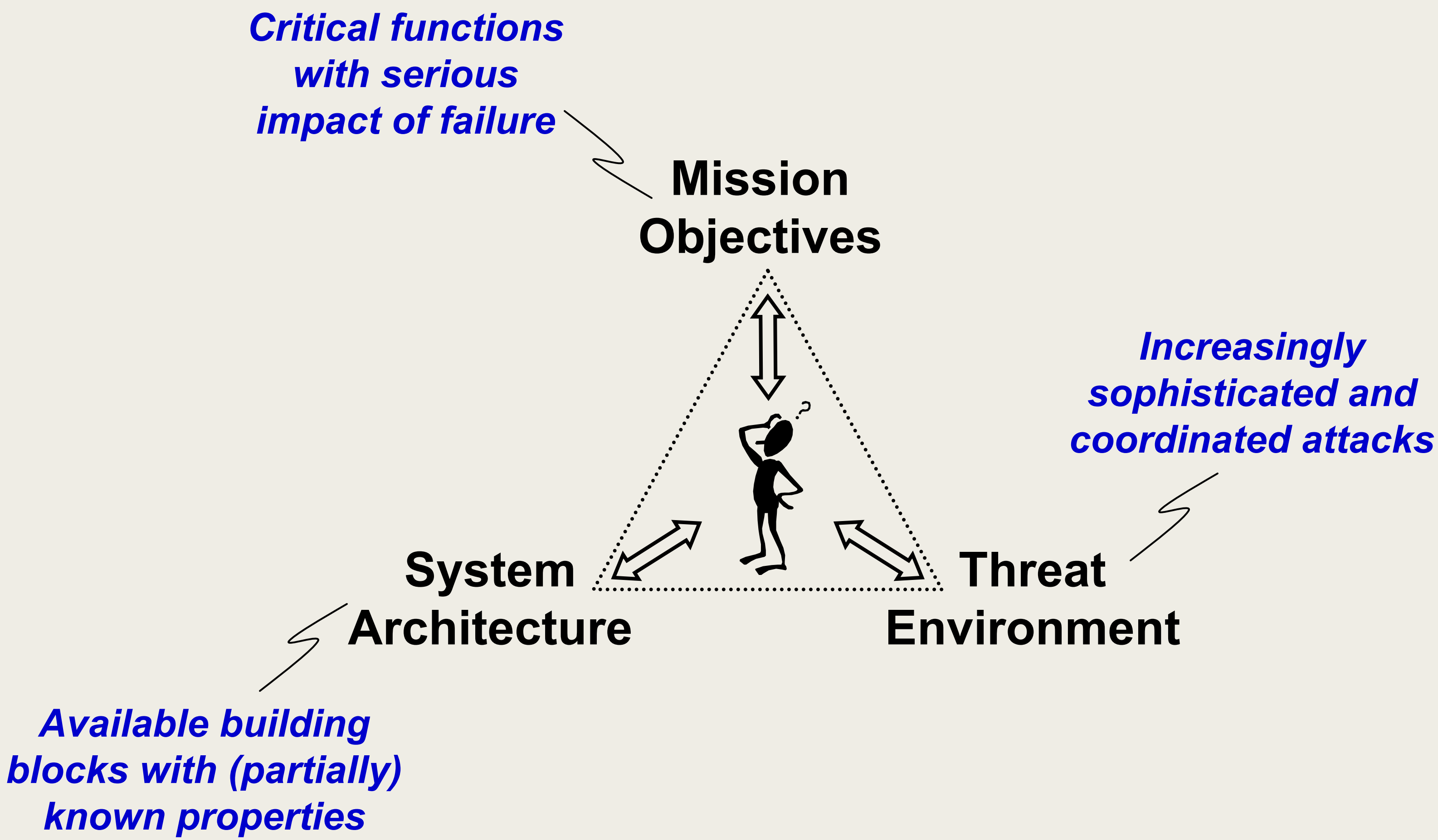
Trustworthy Refinement through Intrusion-Aware Design (TRIAD)

Andy Moore (apm@cert.org; 412-268-5465)
SEI, CERT Research Center
April 2003

based on work with Bob Ellison, CERT/RC

Sponsored by the U.S. Department of Defense
© 2002 by Carnegie Mellon University

System Security Architect's Problem



© 2002 by Carnegie Mellon University

Carnegie Mellon
Software Engineering Institute

"Trust in Cyberspace" Finding

"Experience has taught that systems – and, in particular, complex systems like networked information systems – can be secure, but only up to a point.

There will always be residual vulnerabilities, always a degree of insecurity. …

With this view, the object of security engineering would be to identify insecurity and move them to less exposed and less vulnerable parts of a system … to reposition them in light of the nature of the threat."

[NRC, *Trust in Cyberspace*, Schneider (ed.), 1999]

Objectives and Scope

Support security and survivability architect

- Formal basis for linking three critical aspects
- Rigorous tool support leveraging existing technology

Address system security and survivability

- Malicious threats
  - Failures/accidents different
  - Serious harm possible by even unskilled
- System level
  - Enterprise-level, inter-networked
  - Emergent nature of properties

Architecture is key

- Too late in process, “hard-codes” vulnerability
- Restrict our effort to architecture

© 2002 by Carnegie Mellon University

Progress

Developed intrusion-aware design model (TRIAD)

- Framework for security and survivability architecting
- Technique to analyze threat impact
- Structures to document strategy and rationale
- Technique to assess impact of changes

Applied model in a trial application domain (eBiz)

- Security and survivability architecture for business
- High rate of fraudulent purchases
- Primary tradeoffs explored, active response developed

Refined concepts for TRIAD tool support (Trilogy)

- Leverages existing technology
- Rigorous underlying semantics

© 2002 by Carnegie Mellon University

Overview of Talk

TRIAD Process

TRIAD Artifacts

Trilogy Tool Support

Conclusions

© 2002 by Carnegie Mellon University

TRIAD Process

© 2002 by Carnegie Mellon University

Systems Architecting

"Architectural design processes are inherently eclectic and wide-ranging, going abruptly from the intensely creative and individualistic to the more prescribed and routine.

While the processes may be eclectic, they can be organized.

Of the various organizing concepts, one of the most useful is stepwise progression or 'refinement.'"

[Maier, *The Art of Systems Architecting*, 2000]

© 2002 by Carnegie Mellon University

Impact on TRIAD

Focus on 'R'efinement

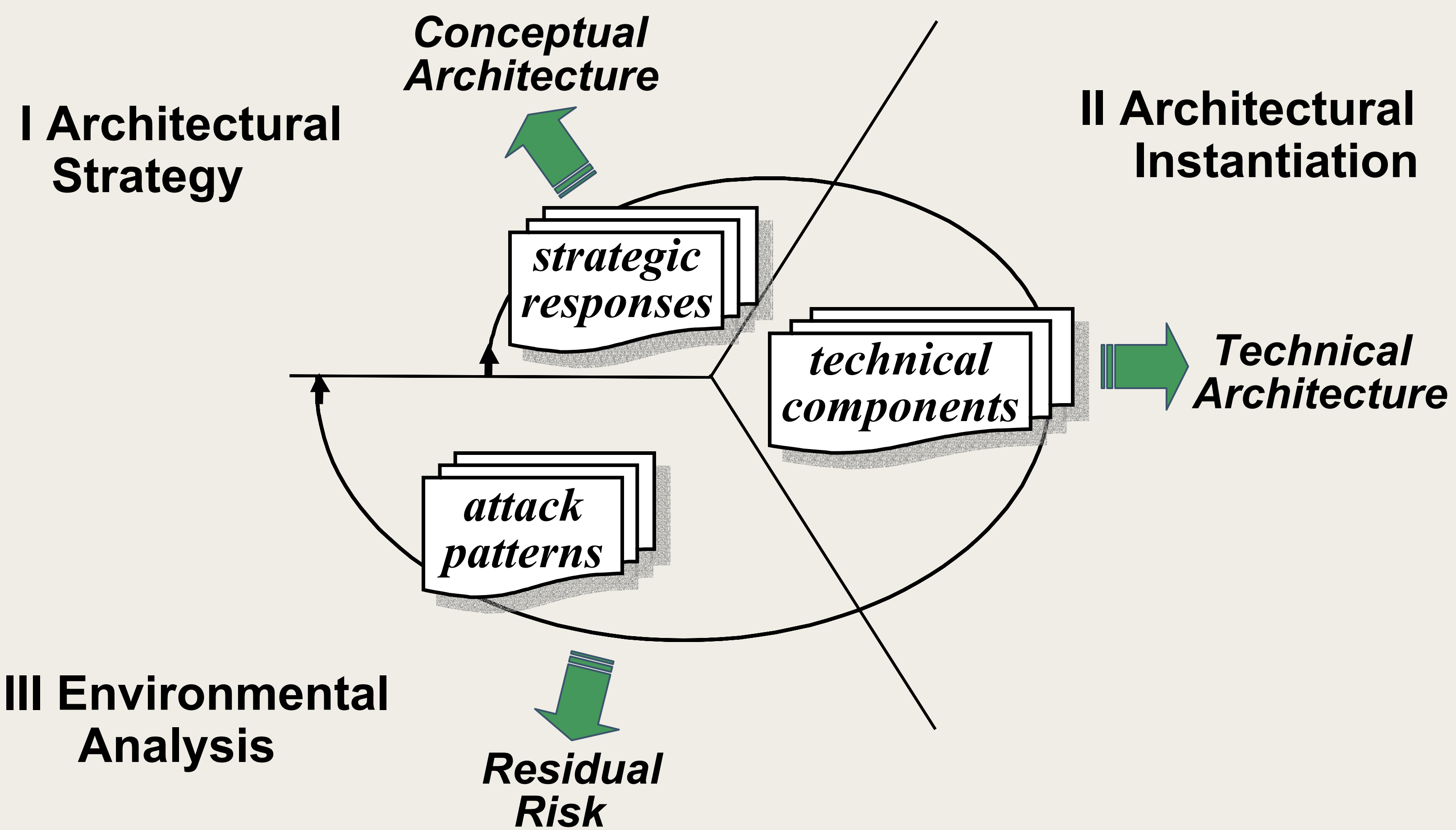
- Secure and survivable systems development is iterative
- Optimal refinement unclear early on
- Incremental experimentation and analysis needed

Spiral model basis

- Intended for software development/maintenance
- Domains where good direction for refinement unclear
- Iteratively refines software development artifacts

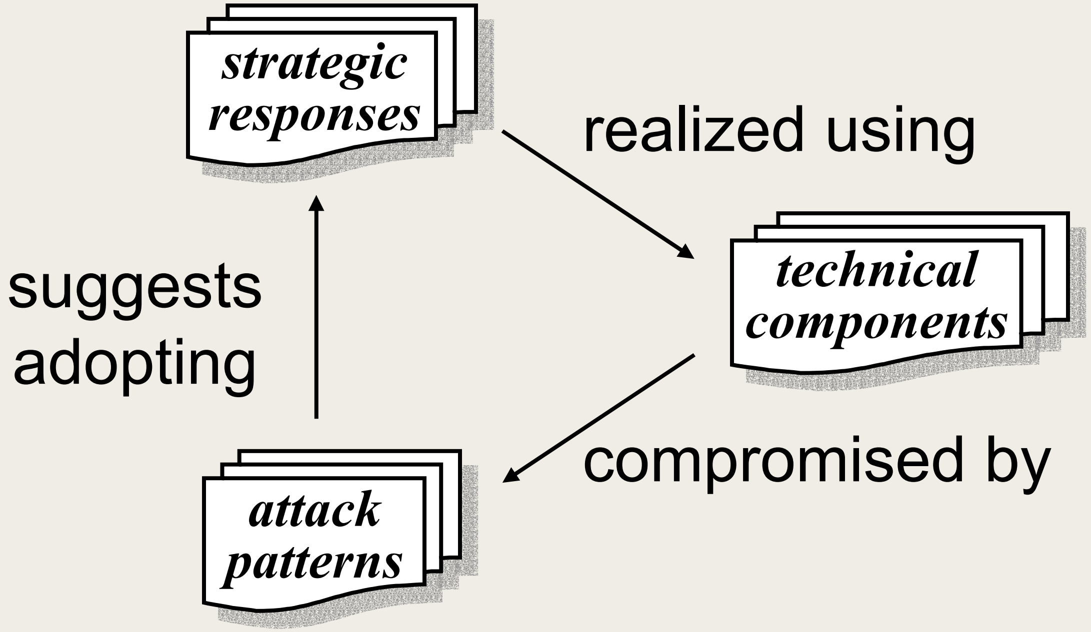
© 2002 by Carnegie Mellon University

TRIAD Overview



© 2002 by Carnegie Mellon University

Structured, Reusable Information



© 2002 by Carnegie Mellon University

Carnegie Mellon
Software Engineering Institute

Example Responses to Attack

Base design decisions on attributes of likely attacks

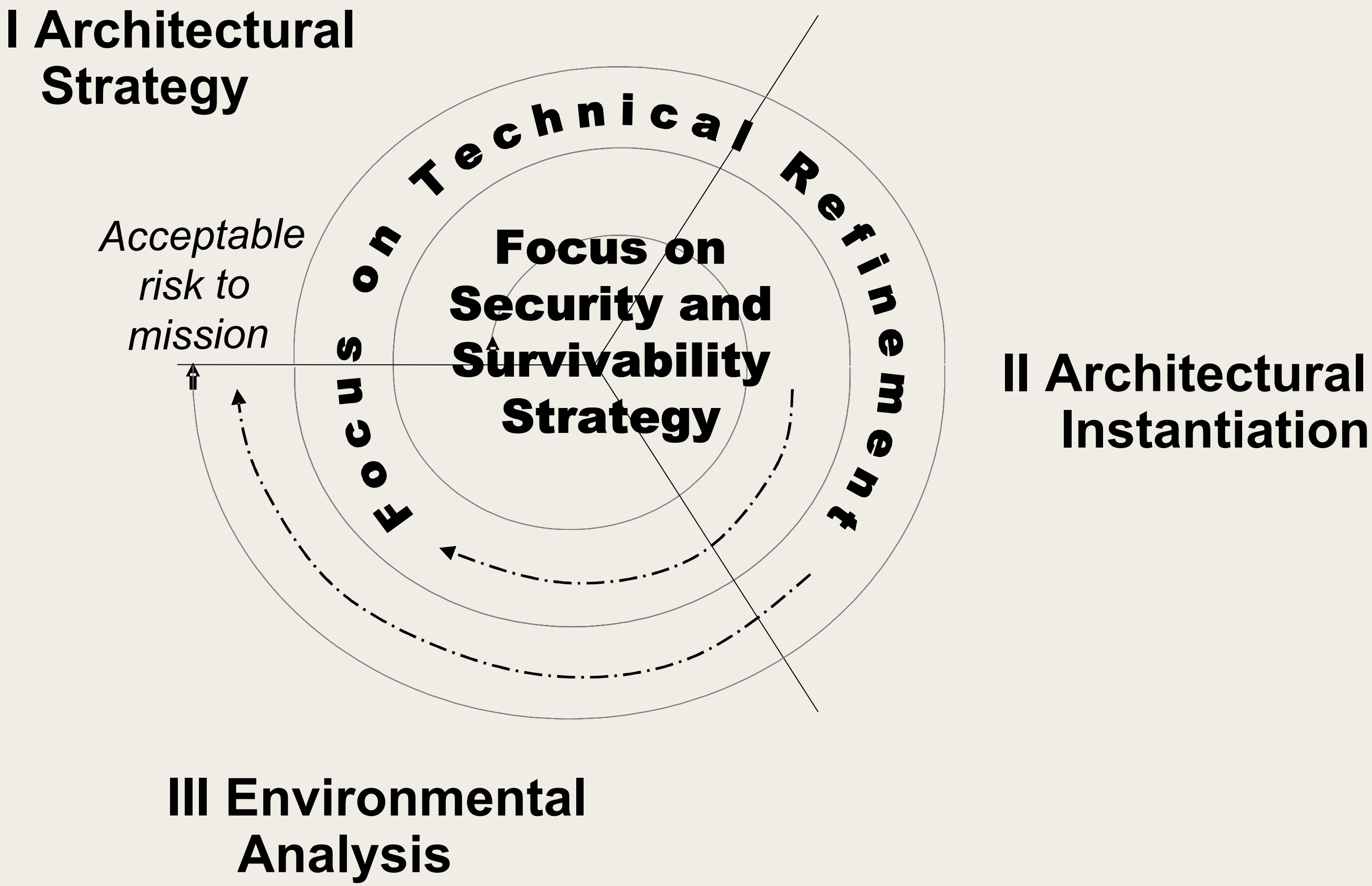
- Attack patterns
  - network-based denial of service (DoS)
  - exploit server vulnerability
  - exploit task flow vulnerability

- Strategic responses for security and survivability
  - High Level: resist, recognize, recover, adapt
  - Mid Level: redundancy, separation, deception, …

- Network DoS attack: focus on network architecture
  - server redundancy & diversity; spare capacity
  - intruder traceback, filtering, apprehension
  - insurance for lost revenue

© 2002 by Carnegie Mellon University

TRIAD Execution



TRIAD Artifacts

© 2002 by Carnegie Mellon University

Primary Artifacts

Mission objectives

Mission threats

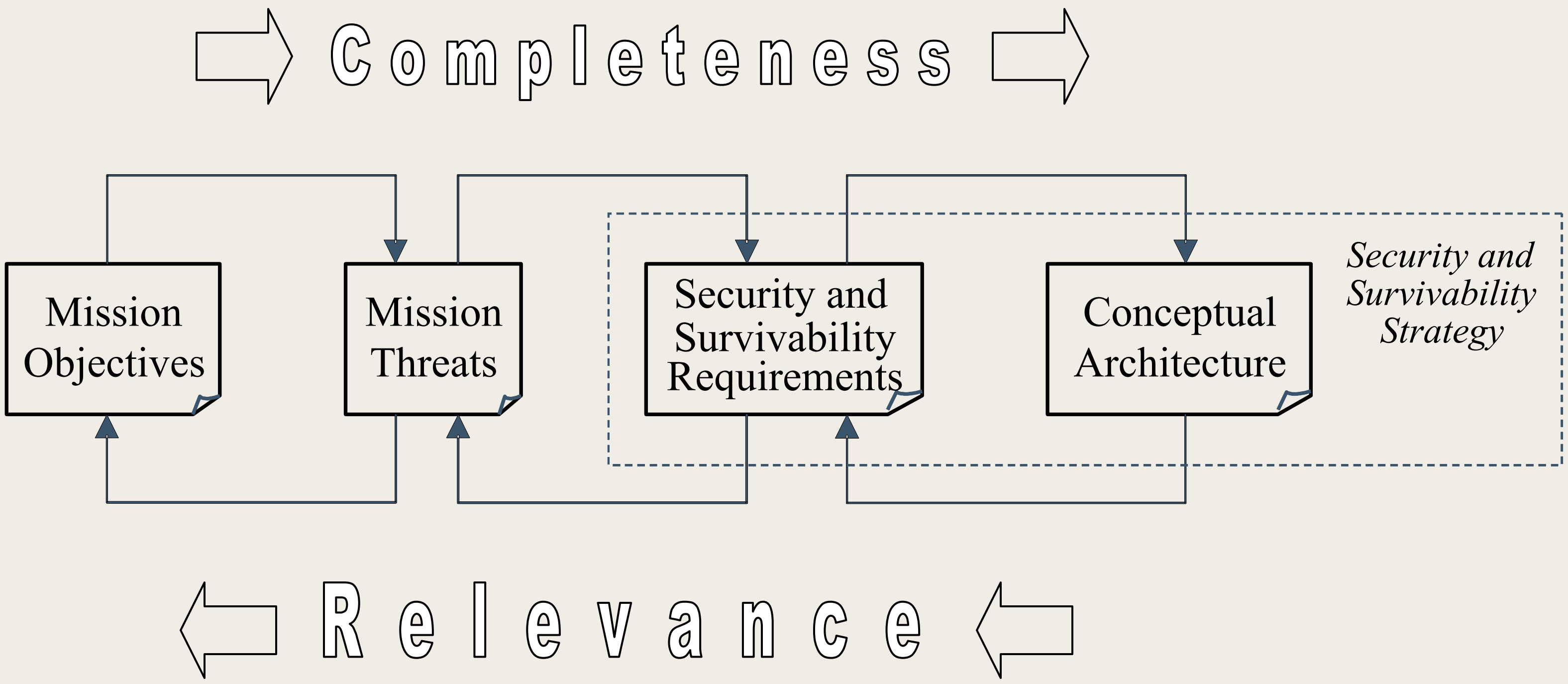
Security and survivability requirements

Conceptual architecture

© 2002 by Carnegie Mellon University

Carnegie Mellon
Software Engineering Institute

Security & Survivability Tracing



Carnegie Mellon
Software Engineering Institute

Tracing Structures: Example Format

Mission Objectives / **Mission Threats**

	O1	O2	O3	O4	O5	O6
T1.1.1.						
T1.1.2.						
T1.1.3.						
T1.2.1.						
T1.2.2.						
T1.2.3.						
T2.1.1.						
T2.1.2.						
T2.1.3.						
T2.2.1.						
T2.2.2.						
T2.2.3.						
T3.1.1.						
T3.1.2.						
T3.2.1.						
T3.2.2.						
T3.2.3.						
T3.2.4.						

Threat-to-Objective Tracing

Mission Threats / **Security & Survivability Reqs**

	T 1. 1. 1.	. . .	1. 2. 1.	. .	2. 1. 1.	. .	2. 2. 1.	. .	3. 1. 1.	. .	T 3. 2. 1.	. . .
R1												
R2												
R3												
R4												
R5												
R6												
R7												
R8												
R9												
R10												
R11												
R12												

Requirement-to-Threat Tracing

© 2002 by Carnegie Mellon University

Requirements: Example Format

<table>
<tr><th colspan="2" rowspan="2">Stimulus</th><th colspan="6">Response</th></tr>
<tr><th colspan="2">Resistance</th><th>Recognition</th><th colspan="2">Recovery</th><th>Adaptation</th></tr>
<tr><td rowspan="2">Primary class of attack</td><td>Subclass #1 of primary attack class</td><td>First technique to resist attacks in subclass #1</td><td>Second technique to resist attacks in subclass #1</td><td rowspan="2">Technique to recognize attacks in both subclass #1 and subclass #2</td><td rowspan="2">Technique to recover from attacks in both subclass #1 and subclass #2</td><td>Additional technique to recover from attacks in subclass #1</td><td>Technique to adapt to attacks in subclass #1</td></tr>
<tr><td>Subclass #2 of primary attack class</td><td colspan="2">Technique to resist attacks in subclass #2</td><td>Additional technique to recover from attacks in subclass #2</td><td>Technique to adapt to attacks in subclass #2</td></tr>
</table>

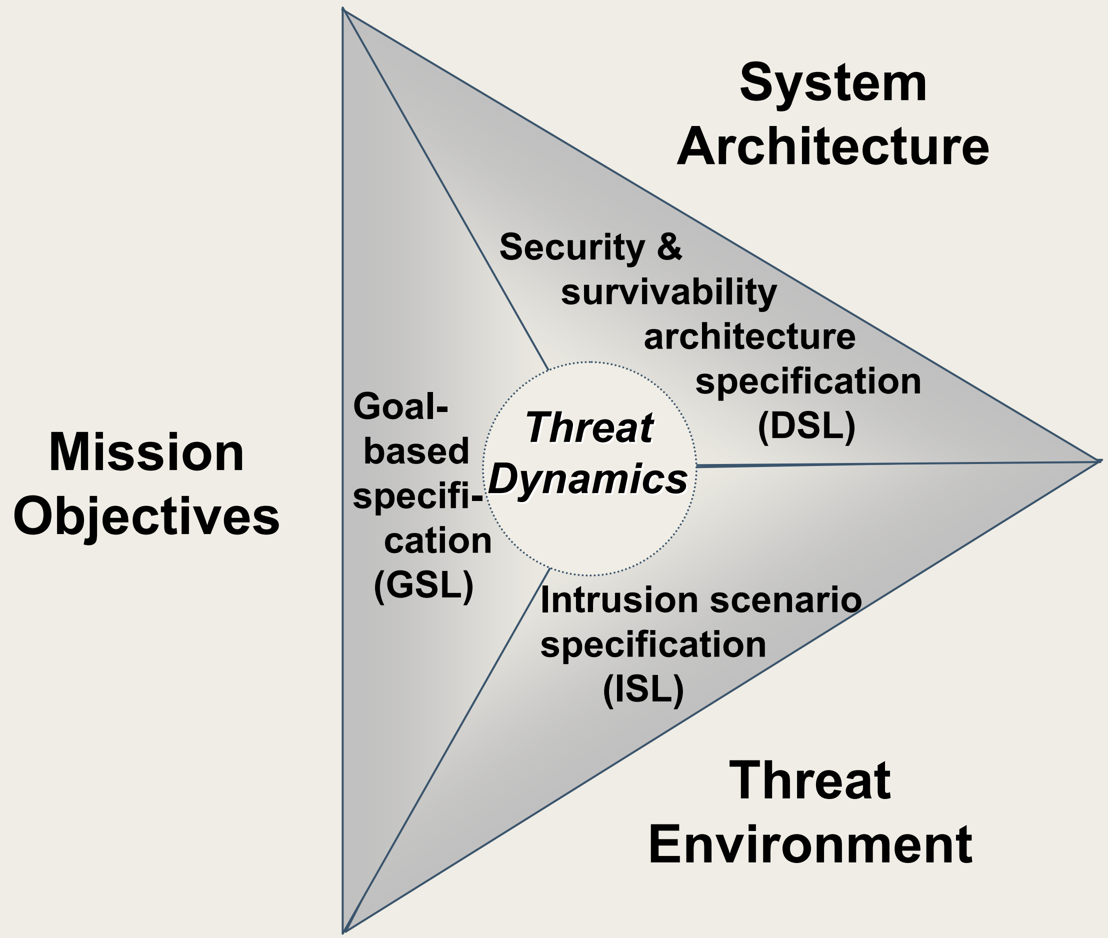
© 2002 by Carnegie Mellon University

TRIAD Tool Support (Trilogy)

© 2002 by Carnegie Mellon University

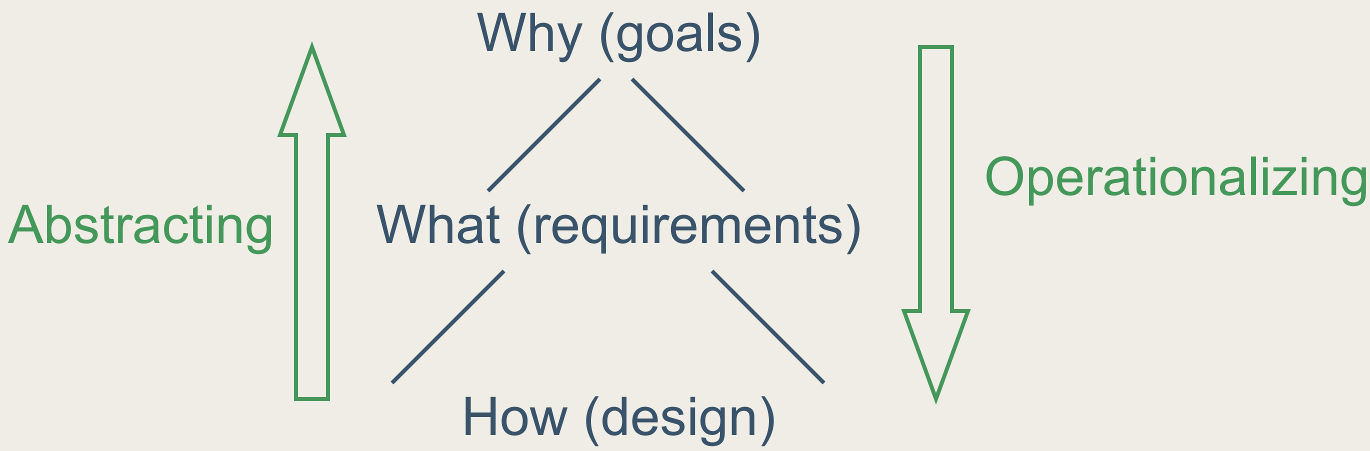
Carnegie Mellon
Software Engineering Institute

Trilogy Overview



© 2002 by Carnegie Mellon University

Goal-based Specification & Reasoning



Goals provide criteria for requirements completeness

Goal structure represented in AND/OR graphs

Formal refinement through satisfaction (KAOS tool)
- Conflicts explicitly represented

Qualitative refinement through satisficing (NFR tool)
- Positive or negative contribution

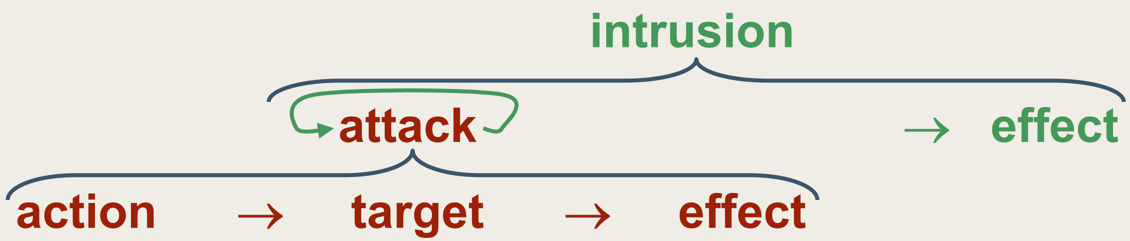
© 2002 by Carnegie Mellon University

Intrusion Scenario Specification

Developed initial classification of attacks

- Target people, technology, context

Adopted initial taxonomy for attacks under classification



Several actual intrusions specified using attack lexicon

- Mitnick intrusion, Trojan horse attack, extortion, hoax

Method defined for organizing scenarios into attack trees

- Allows extending attack trees using attack patterns

© 2002 by Carnegie Mellon University

Security & Survivability Architectures

Specified using domain-specific language

- Programming or executable specification language
- Provides notations and abstractions
- Enhances expressive power in some problem domain

Our usage

- Specification language for system architectures
  - Perspective of security and survivability
  - Enterprise-level, internetworked
- Security and survivability architecture focused domain
  - High level, mid-level, low-level mechanisms

Related to aspect-oriented programming, architecture description languages, domain modeling

© 2002 by Carnegie Mellon University

Threat Dynamics

Based on System Dynamics

- Analysis method for complex, managed systems
  - Design improved feedback structures/control policies
- Interpreted for malicious threats to internetworks
  - Feedback control critical to active defense strategies

Helps deal with dynamic complexity

- Arises from nature of interactions *over time*
- Contrasts with static complexity
- Complicating factors
  - Feedback
  - Uncertainties
  - Changes over time
  - Time delays
  - Non-linearities

© 2002 by Carnegie Mellon University

Carnegie Mellon
Software Engineering Institute

Notation: Influence Diagrams

Influence diagrams: qualitative model of system behavior

- Refined into quantitative (simulation) model

Variables represent system elements

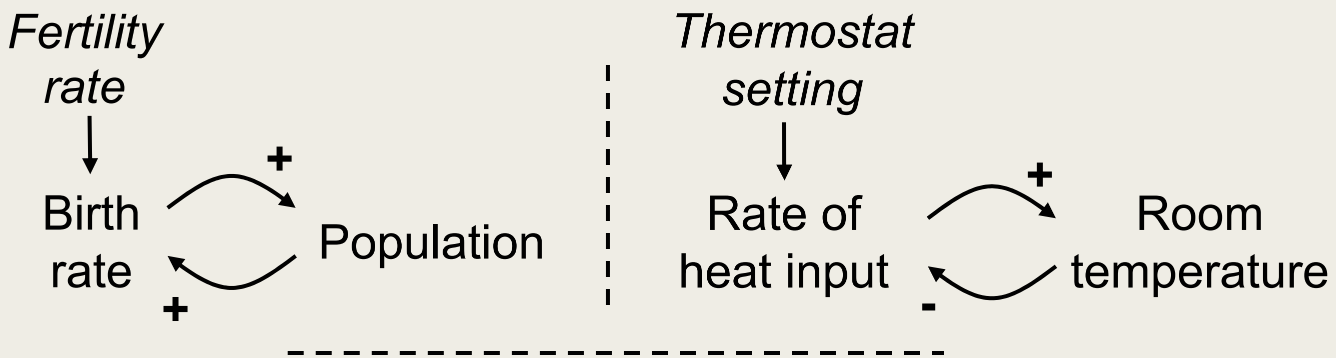
- Elements may be animate/inanimate, tangible/intangible
- Elements in italics represent parameters

Signed arrows represent pairwise causal influence (not correlation)

- +, if source ↑ (↓) then target ↑ (↓) above (below) value o/w
- -, if source ↑ (↓) then target ↓ (↑) below (above) value o/w

© 2002 by Carnegie Mellon University

Key Driver: Feedback Loops



Self-reinforcing (+) loops drive variable values up or down

- Explosive growth or implosive collapse

Self-limiting (-) loops drive variable values to goal state

- Describes aspects that oppose change

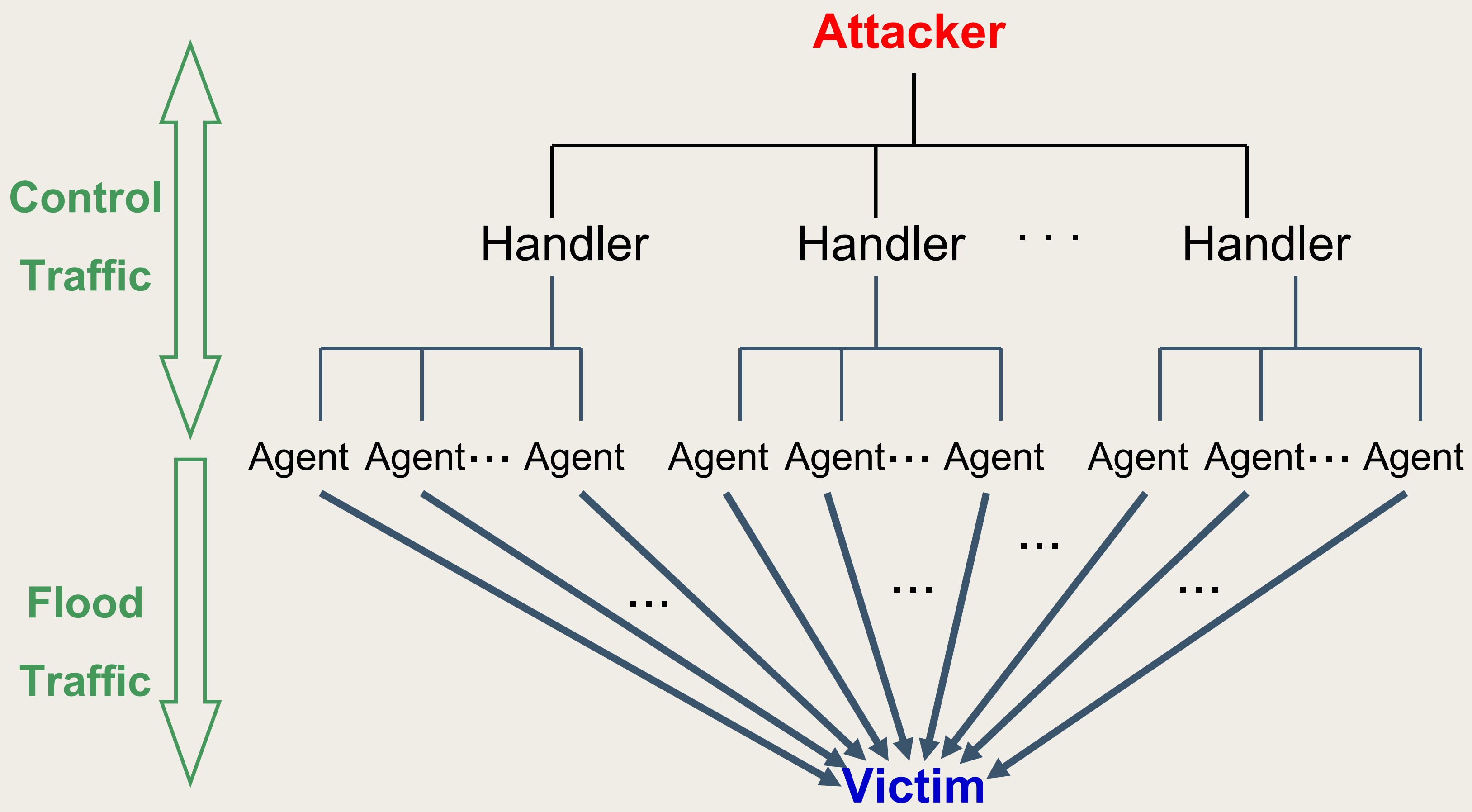
Behavior arises due to interactions of multiple loops

- Limiting loops can moderate influence of reinforcing loops
- Can explain “counter-intuitive” behavior

© 2002 by Carnegie Mellon University

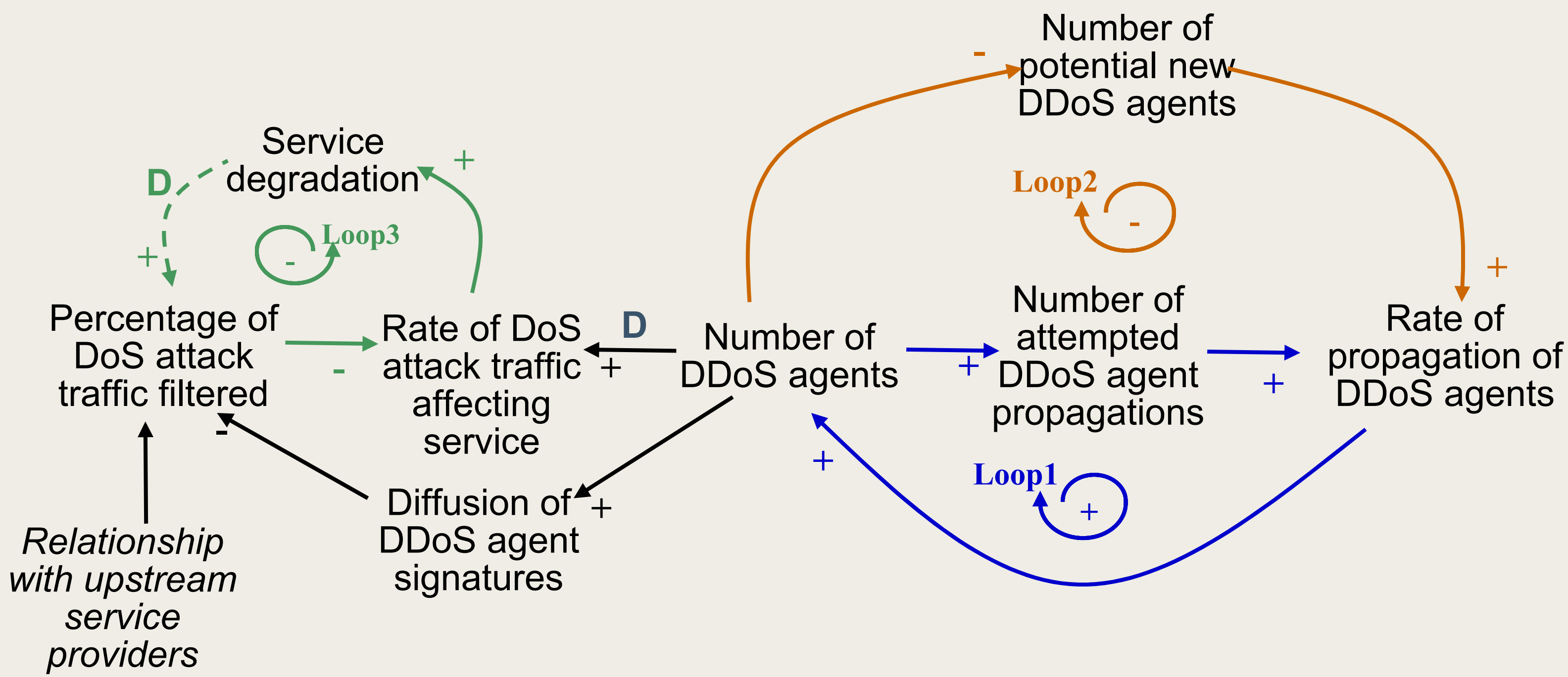
Carnegie Mellon
Software Engineering Institute

Example: Distributed Denial of Service



© 2002 by Carnegie Mellon University

Simple DDoS Influence Diagram



© 2002 by Carnegie Mellon University

+'s and –'s of Influence Diagrams

+

- Model and analyze impact of malicious threats
- Make tradeoffs associated with alternative responses
- Assess proper role of technology
- Evaluate influence of change
- Basis for quantitative analysis

-

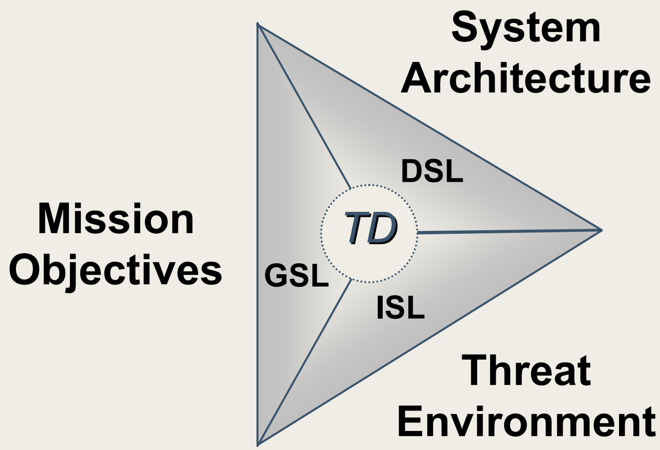
- Misleading if used improperly
- Reusability currently limited
- Correspondence with architecture currently loose

© 2002 by Carnegie Mellon University

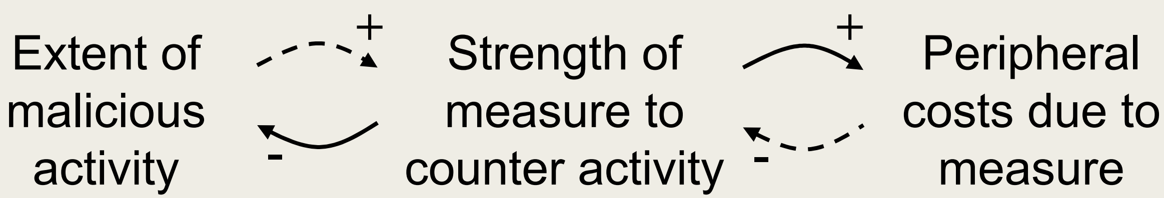
Carnegie Mellon
Software Engineering Institute

Needs

Underlying semantic model for threat dynamics



Threat/response patterns, e.g.,



© 2002 by Carnegie Mellon University

Conclusions

© 2002 by Carnegie Mellon University

Benefits of TRIAD/Trilogy

TRIAD/Trilogy helps

- Construct security and survivability architecture
- Determine mission impact of evolving threat environment
- Formulate strategic response to threats
- Determine how to use technical components to satisfy strategic objectives
- More accurately assess risk of mission failure
- Gain high confidence that mission will succeed

© 2002 by Carnegie Mellon University

Broad Plans

Apply TRIAD in pilot program || **Develop Trilogy tool support**

Expand/refine model/tools

Apply Trilogy in pilot program

Prepare user materials/tutorials

Transition technology to government/industry use

© 2002 by Carnegie Mellon University