Carnegie Mellon
Software Engineering Institute
Pittsburgh, PA 15213-3890

Foundations for Survivable Systems Engineering

Richard C. Linger
Andrew P. Moore

Sponsored by the U.S. Department of Defense

Agenda

- Survivable System Concepts
- Flow-Service/Quality Engineering
- Intrusion-Aware Design

Survivable System Concepts

Network System Realities

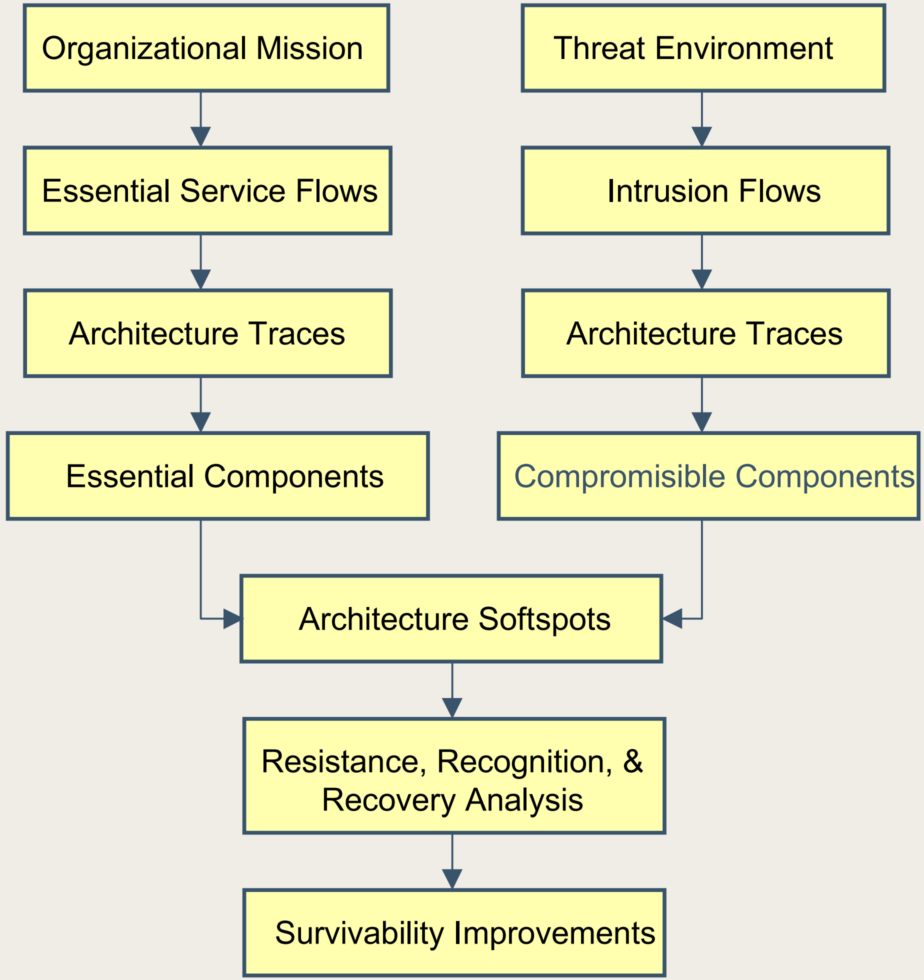
- Ever larger-scale systems
  - Systems-of-systems integration, dependencies
  - Open architectures, increased vulnerabilities
  - Unknown boundaries, untrusted users
  - Lack of central administrative control
  - Escalating threats and consequences
- Security is no longer sufficient

Survivability Defined

Survivability is the capability of a system to fulfill its mission in a timely manner in the presence of attacks, failures, or accidents

- No amount of security can guarantee that systems will not be penetrated
- Survivability analysis
  - Focus on mission
  - Assume imperfect defenses
  - Apply resistance, recognition, recovery strategies

Survivable Systems Analysis (SSA) Method

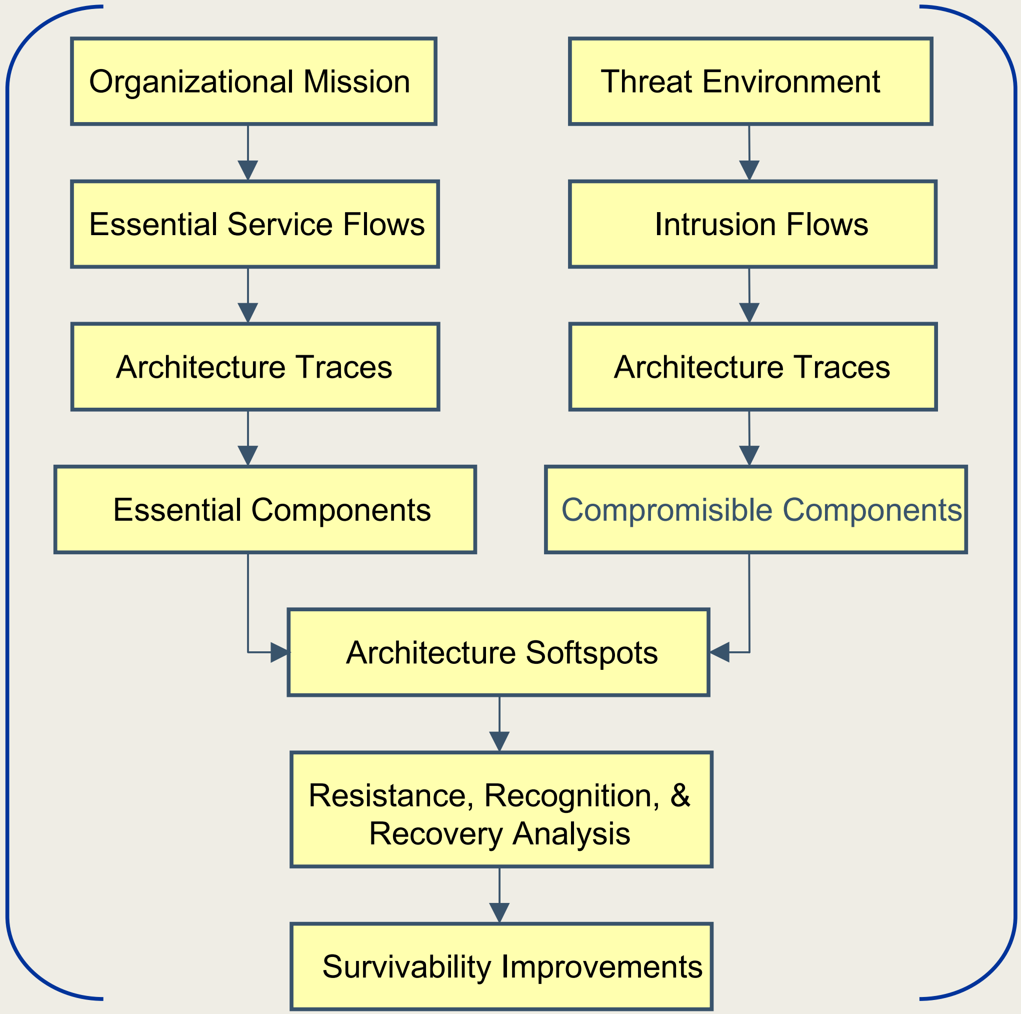


- Structured
- Applied
- Effective
- Documented

NSA-Sponsored Projects

SSA

FS/Q Systems Engineering Project



Intrusion-Aware Design Project

Flow-Service/Quality Engineering: Complexity Reduction and Survivability Analysis in Large-Scale Network Systems

Network System Complexities

- Very large scale, heterogeneous networks
- Unknown boundaries and components
- Uncertain COTS function and quality
- Unforeseen behavior and vulnerabilities
- Unanticipated cascade effects
- Pervasive asynchronous operations
- Survivability an urgent priority

Complexity's burden

- Development of large-scale systems can exceed engineering capabilities
- Difficulty experienced defining systems we have and systems we need
- Intellectual control is lost when complexity exceeds human capabilities

FS/Q Project Objectives

- Complexity reduction requires
  - Maintaining human intellectual control
  - Uniform, scale-free foundations
  - Practical foundations-based engineering

- Survivability improvement requires
  - Knowing usage dependencies in all situations
  - Preparing for compromises in all situations
  - Designing system actions for all situations

Complexity masks and amplifies vulnerabilities and diminishes survivability

Three Key Questions

In a world of large-scale, asynchronous network systems with dynamic function and structure …

- What engineering foundations can reduce complexity in system analysis, specification, and design?
- How should quality attributes such as survivability, reliability, and performance be specified and achieved?
- What architecture frameworks can simplify system development and operation?

Three Engineering Concepts

In a world of large-scale, asynchronous network systems with dynamic functionality and structure …

1. Flow Structures
   User task flows and their architecture flows of service uses are engineering anchors for analysis, specification, and design of functionality and quality attributes

2. Computational Quality Attributes
   Quality attributes can be specified as dynamic functional properties to be computed, not as static, a priori predictions

3. Dynamic Flow Management
   User task flow designs support architecture templates that manage flows and their quality attributes in execution

Flow Structures -- 1

Enterprise mission ⟺ User task flow / User task flow / User task flow ⟺ Architecture flow of service uses / Architecture flow of service uses / Architecture flow of service uses

Enterprise mission is embodied in user task flows of operations and decisions in system usage

Architecture flow refinements of user task flows are conditional compositions of system services that provide functionality and quality attributes

- For complexity reduction:
  - Straightforward flow abstraction, refinement, and verification for human understanding and analysis
  - Flows must exhibit deterministic properties for human use, despite the asynchronous behavior of their shared services

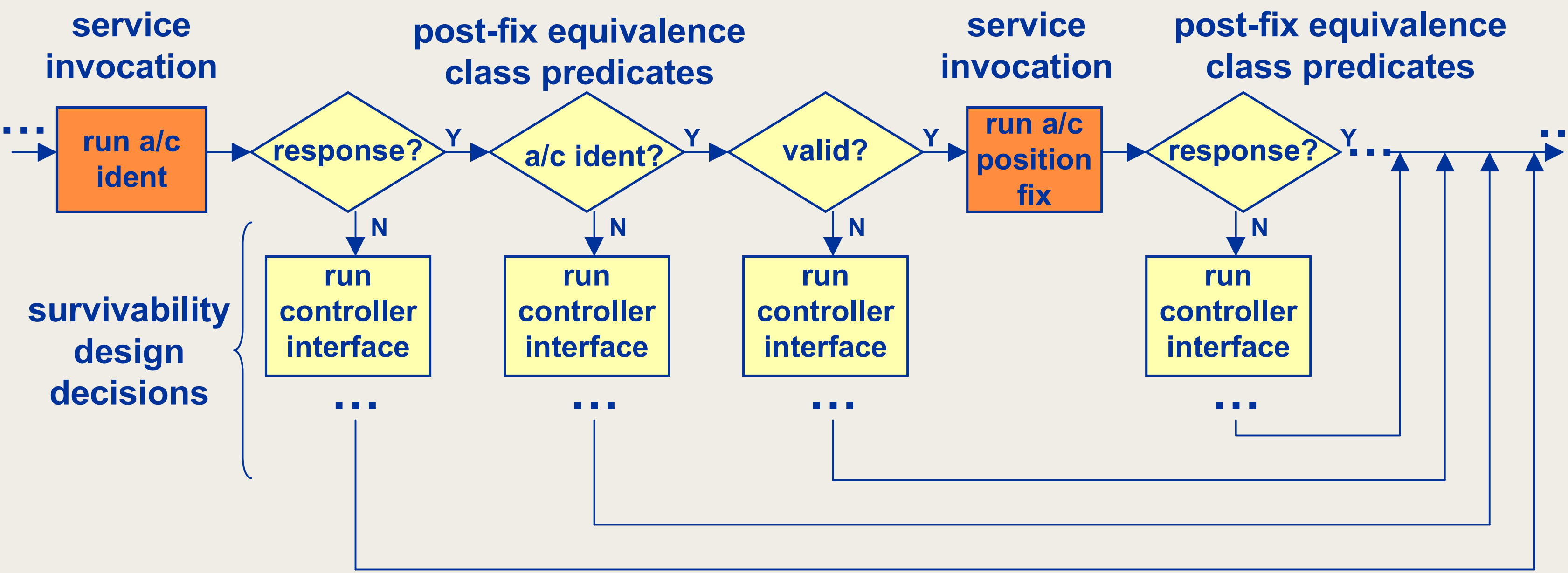
Flow Structures -- 2

- Service invocations in Flow Structures are specified by service response semantics
  - Semantics are response-based, not intention-based – a natural fit with COTS and components
  - Service invocations are composed with post-fix predicates on equivalence classes over all possible responses
- Logic of a flow accounts for all possible outcomes
- Theorems: Flow Structure, Abstraction/Refinement; Verification; Implementation; System Testing

Semantics permit deterministic abstraction, refinement, and verification for human understanding, even though services are engaged in simultaneous asynchronous uses

Flow Structures -- 3

An air traffic control flow fragment:

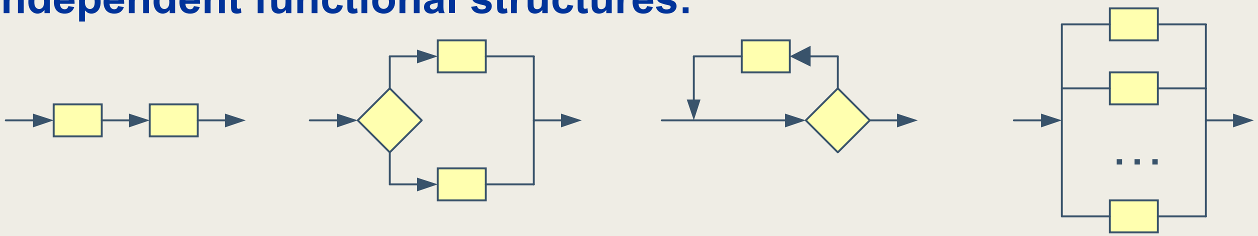


Flow Structures define required behavior for all outcomes

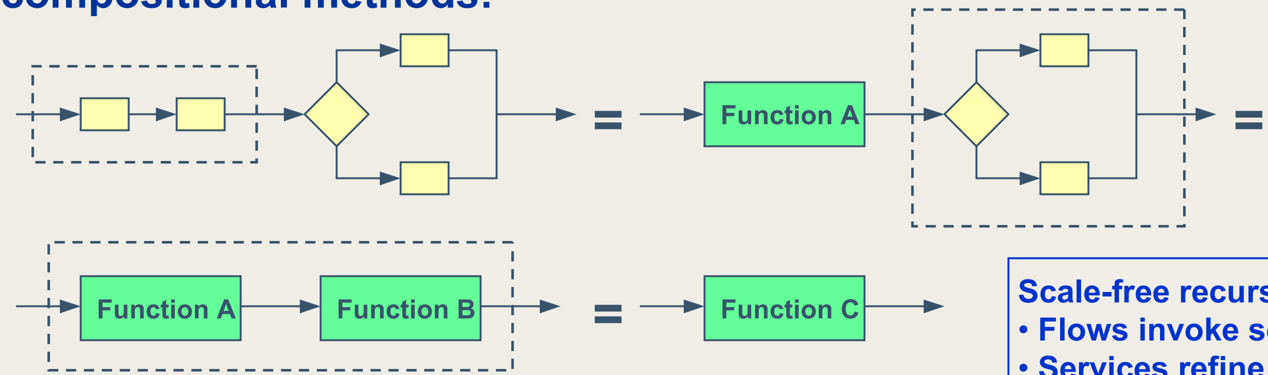
- Risk management requires analysis of all outcomes
- Survivability requires actions for all outcomes
- Design task: Produce intended outcomes in any environment

Flow Structure Algebra

- Semantics permit flows to be expressed in a simple set of language-independent functional structures:



- Flow Structures can be abstracted, refined, and verified through compositional methods:



Scale-free recursive process:
- Flows invoke services
- Services refine into flows

- Stepwise, function-theoretic verification process

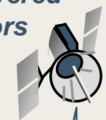
Network-Centric Capability Integration

FlowSets to Manage Complexity in the Future Combat System:

FlowSet:
- Preparation
- Launch
- C3
- Retrieval
- Maintenance
- …

UAVs

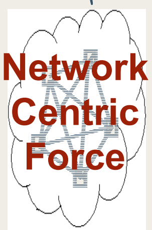
Other Layered Sensors



FlowSet:
- Preparation
- Launch
- C3
- Maintenance
- ..

Robotic Sensors

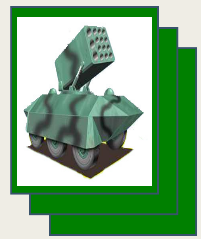
FlowSet:
- Preparation
- Deployment
- C3
- Retrieval
- Safing
- Maintenance
- …



FlowSet:
- Mission Def'n
- Sensor Integration
- C4I
- Fire Integration
- Damage Assmt
- …

FlowSet:
- Preparation
- Deployment
- C3
- Retrieval
- Safing
- Maintenance
- …

Robotic Direct Fire



Robotic NLOS Fire

FlowSet:
- Preparation
- Deployment
- C3
- Retrieval
- Safing
- Maintenance
- …

Manned C2

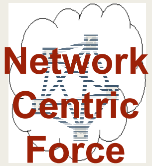
FlowSet:
- Preparation
- Deployment
- C3
- Maintenance
- …

- Distributed platforms
- System-of-systems
- 100s of nodes and users
- Nodes and usage change/evolve

Flow Structures define capabilities and networks, link stovepipes, define compositions of services, support centralized and distributed control

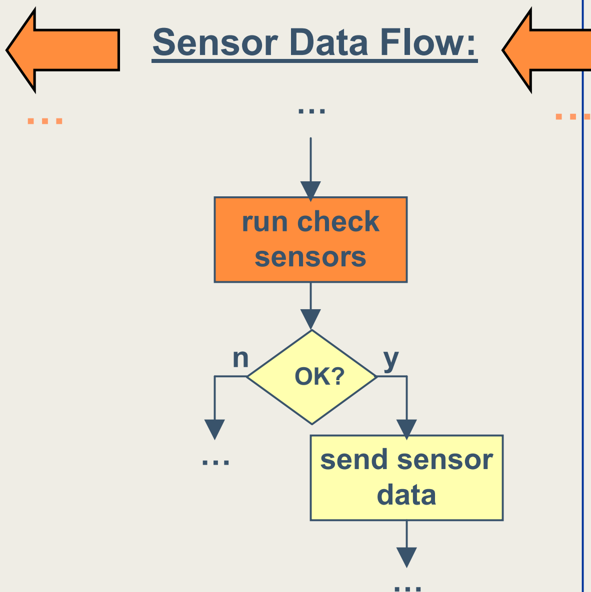
Transitive Dependencies in Flows

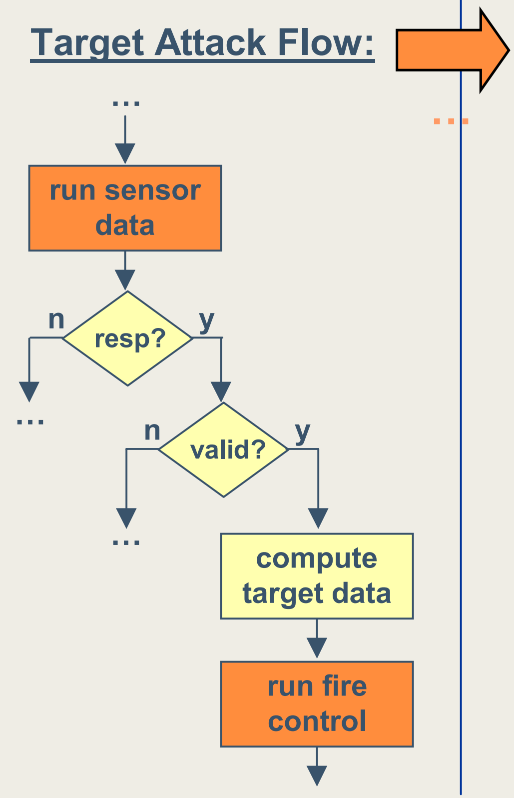
UAV

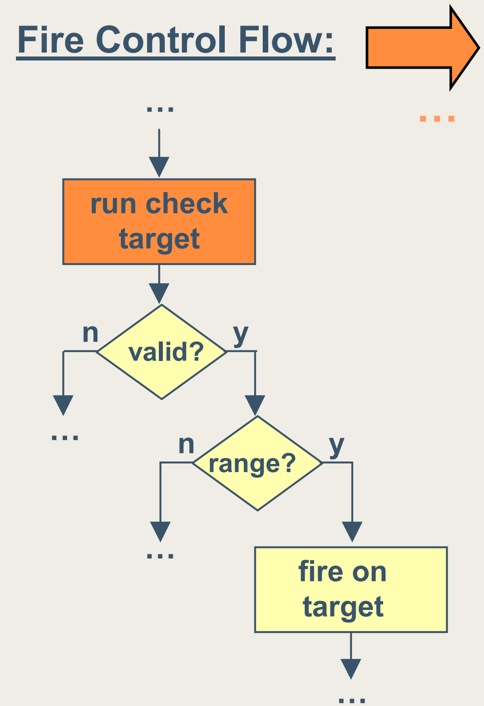


Primary Flow: Mission Control

Robotic Direct Fire







Transitivity analysis reveals dependencies from mission down to code, and defines impact of changes

Flow Structure Application

In survivability analysis

- Extracted mission flows reveal dependencies

In system design

- User task flows
  - Designed and verified at levels of refinement
- Network behavior specification
  - The set of flows of service uses it supports
- Component service specification
  - Defined by all its uses in flows

In management

- Flow-centric from acquisition to operation

In intrusion modeling

- Intruders are users with flows of their own

FS/Q Complexity Reduction

- Flows unify, enable human reasoning in network systems
- Flows are expressed in a few simple structures
- Flows can be abstracted, refined, and verified
- Flows refine missions into architecture services
- Flows are scale-free, define all their required behavior
- Flow transitivity reveals dependencies, impact of changes
- Flows define logical topology and service specifications
- Flows as built can be verified against flows as specified
- Flows prescribe system testing requirements

FS/Q Survivability Analysis

- Flows extracted from existing systems reveal mission survivability dependencies on essential services
- Transitivity analysis of extracted flows reveals cascade service dependencies that impact survivability
- Intrusion flows reveal compromisible services
- Flows require definition of, and actions in, all possible circumstances of use for survivability
- Flow dependencies focus survivability improvements

Project Status

- Progress
  - FS/Q Working Group – three universities
  - Defining FS/Q foundations
  - Two papers published – HICSS, OOPSLA

- Next Steps
  - Document FS/Q foundations
  - Identify case study opportunities

Intrusion-Aware Design (IAD)

IAD Problem Addressed

- Sophisticated intruders can and do
  - Share tools and knowledge to amplify capability
  - Escalate attack with intensity of political conflicts
  - Target people (perceptions), resources, workflows
  - Hide their tracks, fly under the radar of existing IDS
- Engineers not using security failure data
  - Same security mistakes continually repeated
  - Properties must emerge from architectural interaction
  - Survivability considered too late, if at all

Objectives

- Develop cost-effective methods for using
  - Known and hypothesized
    - patterns of attack and
    - strategies for surviving attacks
  - To improve survivability of real-world enterprises.
- Focus on patterns/strategies at architectural level
  - Details of component vulnerabilities overwhelming
  - Assume individual components/connections will fail
  - Architectural focus reduces combinatorial explosion

Relevant Definitions

- Enterprise
  - An information system and its operational environment
  - May include people, technology, work context, procedures
- Enterprise Architecture
  - The structural concept of an enterprise
  - Combination of logical and physical
- Attack Pattern
  - Generic representation of deliberate attack
  - Commonly occurs in specific context (enterprise)
- Survivability Strategy
  - Generic representation of strategy
  - To resist, recognize, recover from attack
  - Commonly useful in specific context (enterprise)

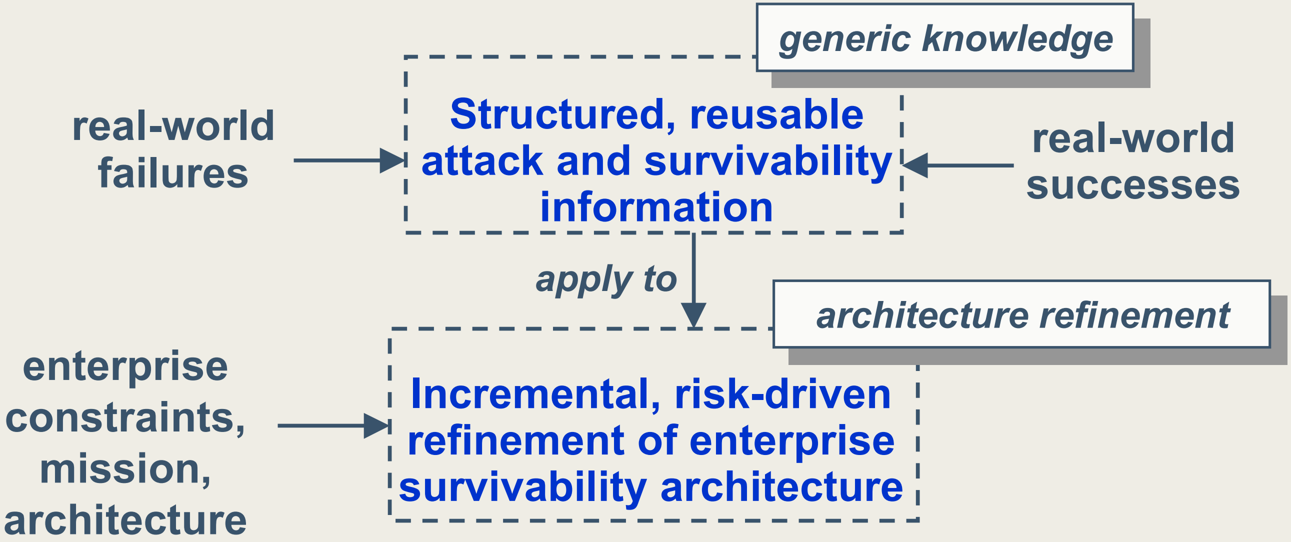
Survivability Strategies

- Redundancy – component, personnel, path, data
- Diversity – functional, design, geographic, personnel
- Separation – physical, logical, cryptographic, temporal
- Deception – hiding, diversion, confusion
- Recognition – patterns, anomalies, virus scanning, integrity checking, surveillance
- Recovery – restoration, apprehension, insurance claim
- Adaptation – adapt intrusion signatures, filtering, logging
- Personnel Management – vetting, training, assessment

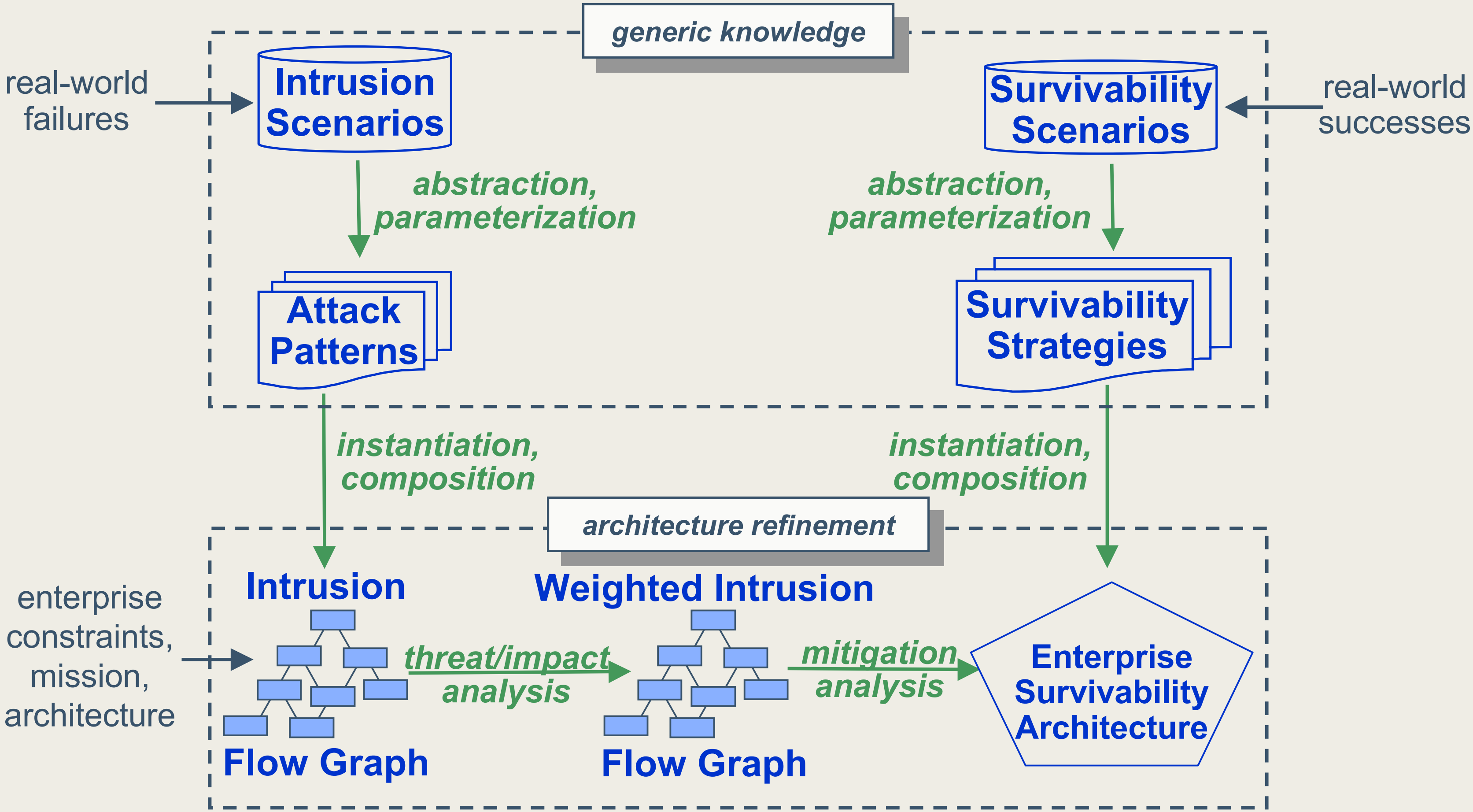
Architectural Responses to Attacks

- Network-based denial of service (possibly distributed)
  - *Focus:* Network architecture, Server redundancy & diversity
  - *Strategies:* Distribute/diversify services, Spare capacity, Intruder traceback, filtering, and apprehension, Insurance claim for lost revenues
- Exploit server vulnerability to gain increased access
  - *Focus:* Host architecture, Layered & diverse defense
  - *Strategies:* DMZ-protected intranet, Proxied web service, Fabricate, mislabel, or crypto-protect files, Monitor file access, Block suspicious activity
- Exploit task flow vulnerability (people, procedure, technology)
  - *Focus:* Application/task flow architecture, Cross-discipline
  - *Strategies:* Virus filtering/scanning, Separation (cryptographic, physical, logical), Periodic personnel training/evaluation

Approach


generic knowledge
real-world failures
Structured, reusable attack and survivability information
real-world successes
apply to
architecture refinement
enterprise constraints, mission, architecture
Incremental, risk-driven refinement of enterprise survivability architecture

Approach (expanded)



Vision of Improved Future

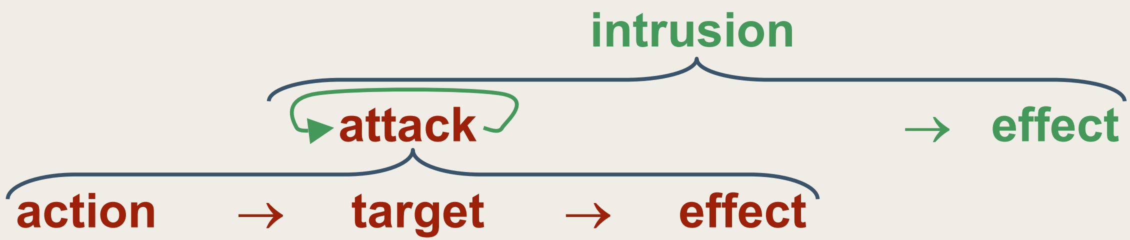
- Rich collection of generic, reusable attack patterns and survivability strategies
- Composition model that enables
  - Quick generation of intrusion flow graphs for particular enterprises
  - Quick identification of survivability strategies to counter likely intrusions
- Improved accuracy and speed of risk analysis and management activities
- Faster, iterated improvement to enterprise architecture and overall survivability

Near-Term Goal

- Explore viability of approach
  - Through its application to improve survivability
  - Of a *particular enterprise architecture*
  - For a *particular class of attacks*
- Viability explored through development of Survivability Decision Model (SDM)
  - Incorporates attack and survivability information into decision model
  - Defines survivability architecture decision criteria
- Initial enterprise architecture of interest:
  - Survivability of eBusiness’s use of online payment system
  - Need to retain paying customers, minimize sales challenged
- Initial attack class of interest:
  - Fraudulent repudiations
  - Disclosure of private customer information

Progress

- Developed initial classification of attacks
  - Target *people*: wants, needs, capabilities, perceptions
  - Target *technology*: computing and networking
  - Target *context*: environment in which people work
- Adopted initial taxonomy for attacks under classification



- Several actual intrusions specified using attack lexicon
  - Mitnick intrusion, cyber-extortion, Trojan horse attack, Emulex hoax
- Initial framework sketched for defining architectural level SDM
  - Demonstration using eBusiness application ongoing

Next Steps

- Document Survivability Decision Model (SDM) framework
- Document attack patterns relevant to eBusiness survivability threats
- Develop SDM for eBusiness example based on attack patterns
- Analyze efficacy of model
- Depending on assessment
  - Make improvements
  - Apply in larger context

Additional Information

- Survivable Systems Analysis
  - General: http://www.cert.org/sna/
  - "The Survivability Imperative: Protecting Critical Systems," CrossTalk, October 2000
- FS/Q Systems Engineering
  - "The Flow-Service-Quality Framework: Unified Engineering for Large-Scale, Adaptive Systems," Proceedings HICSS-35 conference, IEEE Computer Society Press, 2002
- Intrusion-Aware Design
  - Attack pattern spec, reuse, composition:
    - http://www.cert.org/archive/pdf/01tn001.pdf
  - Attack Tree analysis:
    - http://www.cert.org/archive/pdf/intrusion-aware.pdf

Carnegie Mellon
Software Engineering Institute