Making Sound Security Decisions Using Quantitative Security Metrics

Ken Keefe

The Problem: Assessing Security and Resilience

- Systems operate in adversarial environments
  - Adversaries seek to degrade system operation by affecting the confidentiality, integrity, and/or availability of the system information and services
  - "Resilient" systems aim to meet their ongoing operational objectives despite attack attempts by adversaries
- System security is not absolute
  - No real system is perfectly secure
  - Some systems are more secure than others
  - *But which ones are more secure?*
  - *And how much more secure are they*?

Related Work Motivating ADVISE

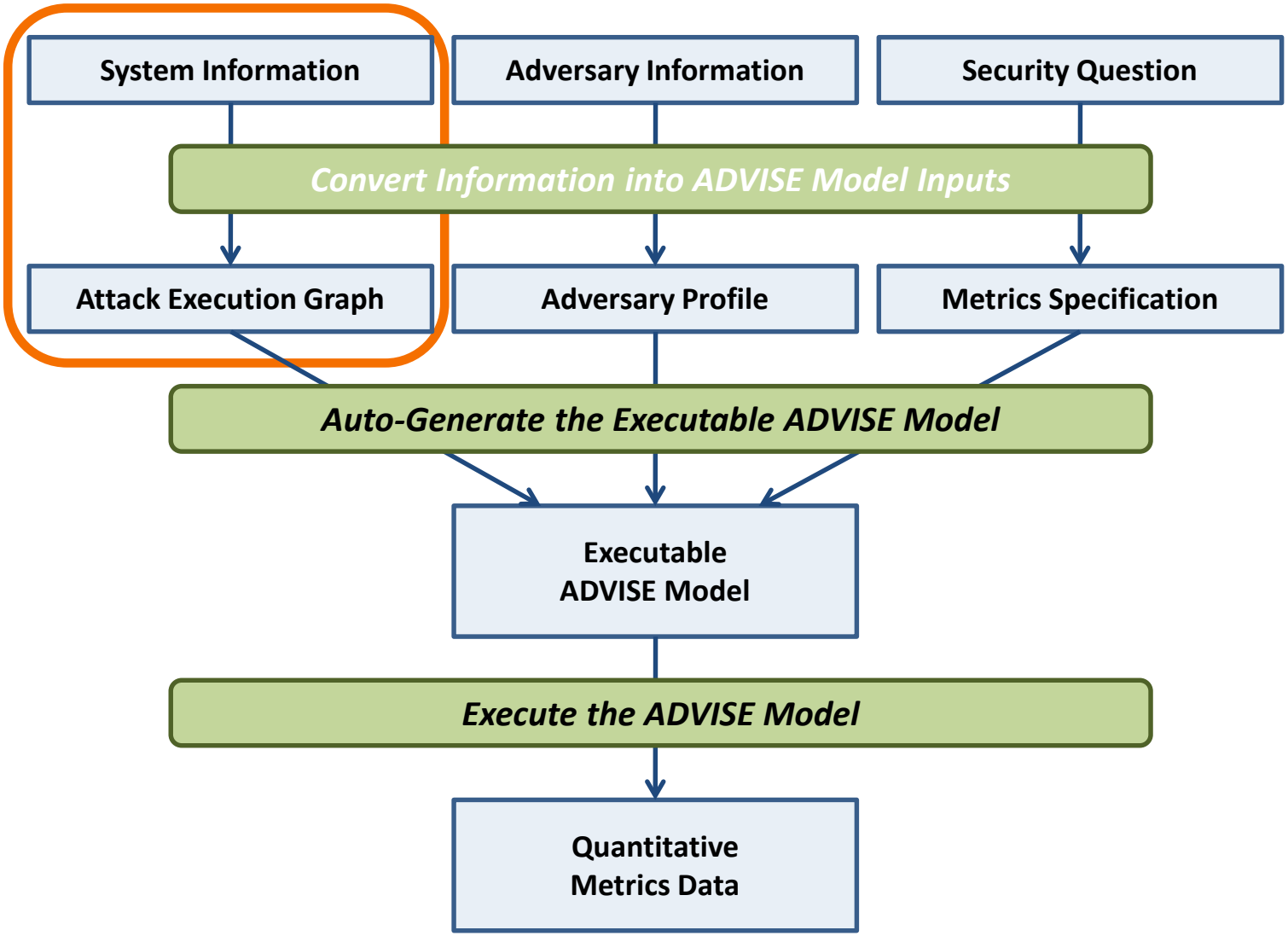
- Model-based security analysis
  - Attack Trees
  - Attack Graphs and Privilege Graphs
- Adversary-based security analysis
  - MORDA (Mission-Oriented Risk and Design Analysis)
  - NRAT (Network Risk Assessment Tool)

ADVISE integrates the benefits of both
model-based and adversary-based security analysis

ADversary VIew Security Evaluation (ADVISE) approach

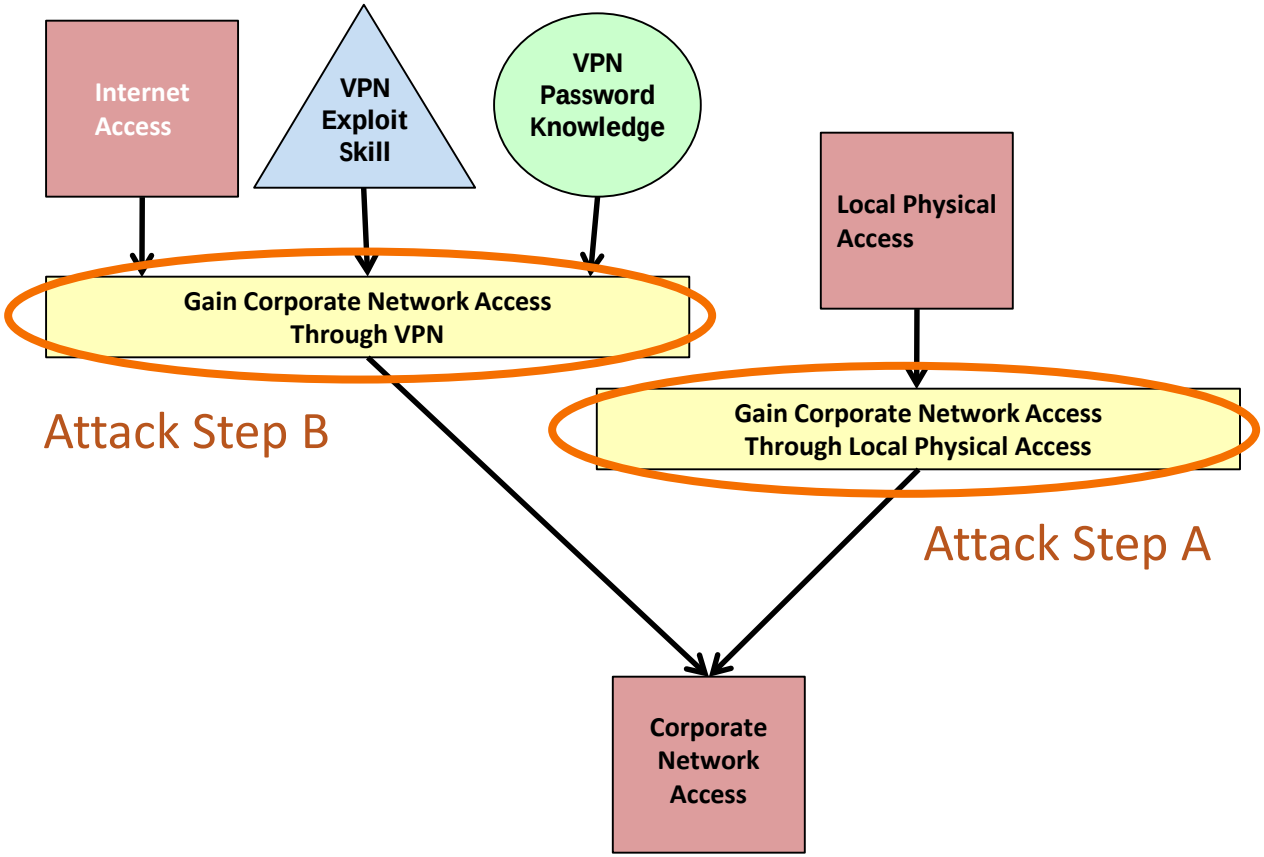
- Adversary-driven analysis
  - Considers characteristics and capabilities of adversaries
- State-based analysis
  - Considers multi-step attacks
- Quantitative metrics
  - Enables trade-off comparisons among alternatives
- Mission-relevant metrics
  - Measures the aspects of security important to owners/operators of the system

ADVISE Method Overview



Representing Attacks Against the System

An "attack execution graph" describes potential attack vectors against the system from an attacker point of view. Attempting an attack step requires certain skills, access, and knowledge about the system. The outcome of an attack can affect the adversary's access and knowledge about the system.



ADVISE System Information: Attack Execution Graph

An attack execution graph is defined by

<A, R, K, S, G>,

where

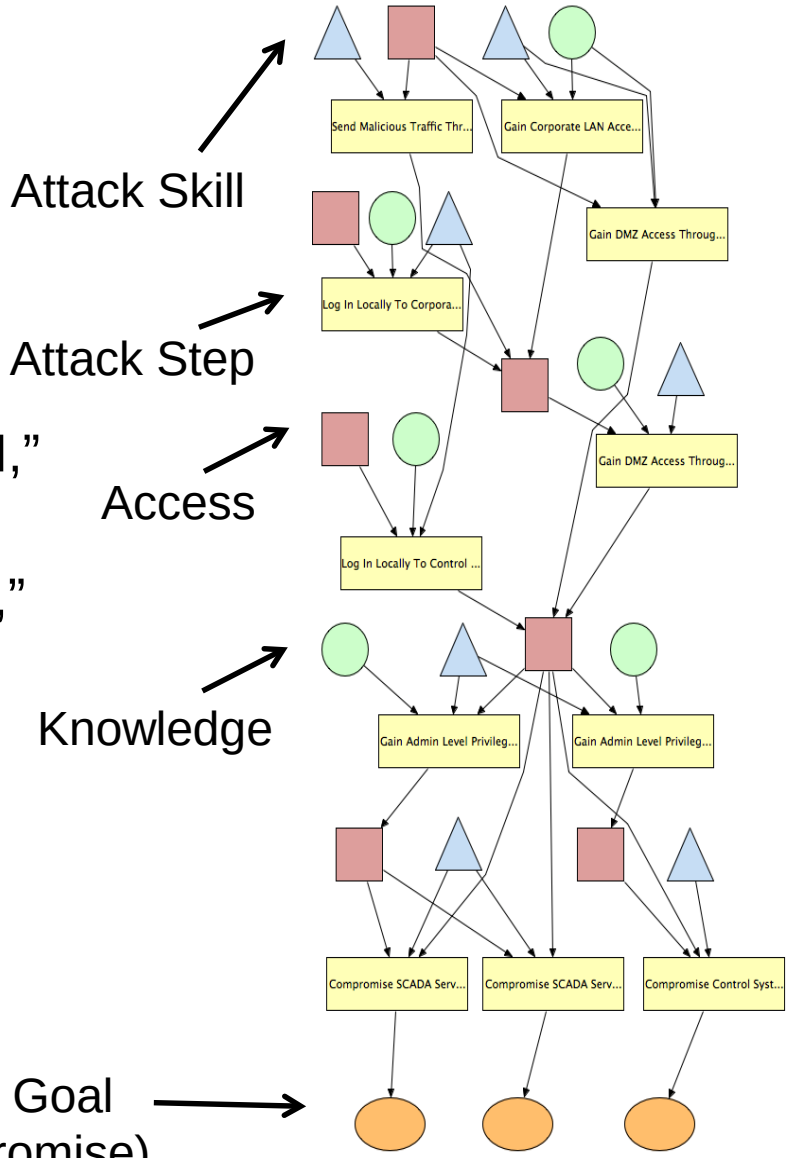
A is the set of attack steps,
e.g., “Access the network using the VPN,”

R is the set of access domains,
e.g., “Internet access,” “Network access,”

K is the set of knowledge items,
e.g., “VPN username and password”

S is the set of adversary attack skills,
e.g., “VPN exploit skill,” and

G is the set of adversary attack goals,
e.g., “View contents of network.”



Attack Step Definition

An attack step ai is a tuple:

ai = <Bi, Ti, Ci, Oi, Pri, Di, Ei>

Note: X is the set of all states in the model.

B_i: $X \rightarrow \{True, False\}$ is a Boolean precondition,
e.g., (Internet Access) AND ((VPN account info) OR (VPN exploit skill)).

T_i: $X \times R^+ \rightarrow [0, 1]$ is the distribution of the time to attempt the attack step,
e.g., normally distributed with mean 5 hours and variance 1 hour.

C_i: $X \rightarrow R^{\geq 0}$ is the cost of attempting the attack step, e.g., $1000.

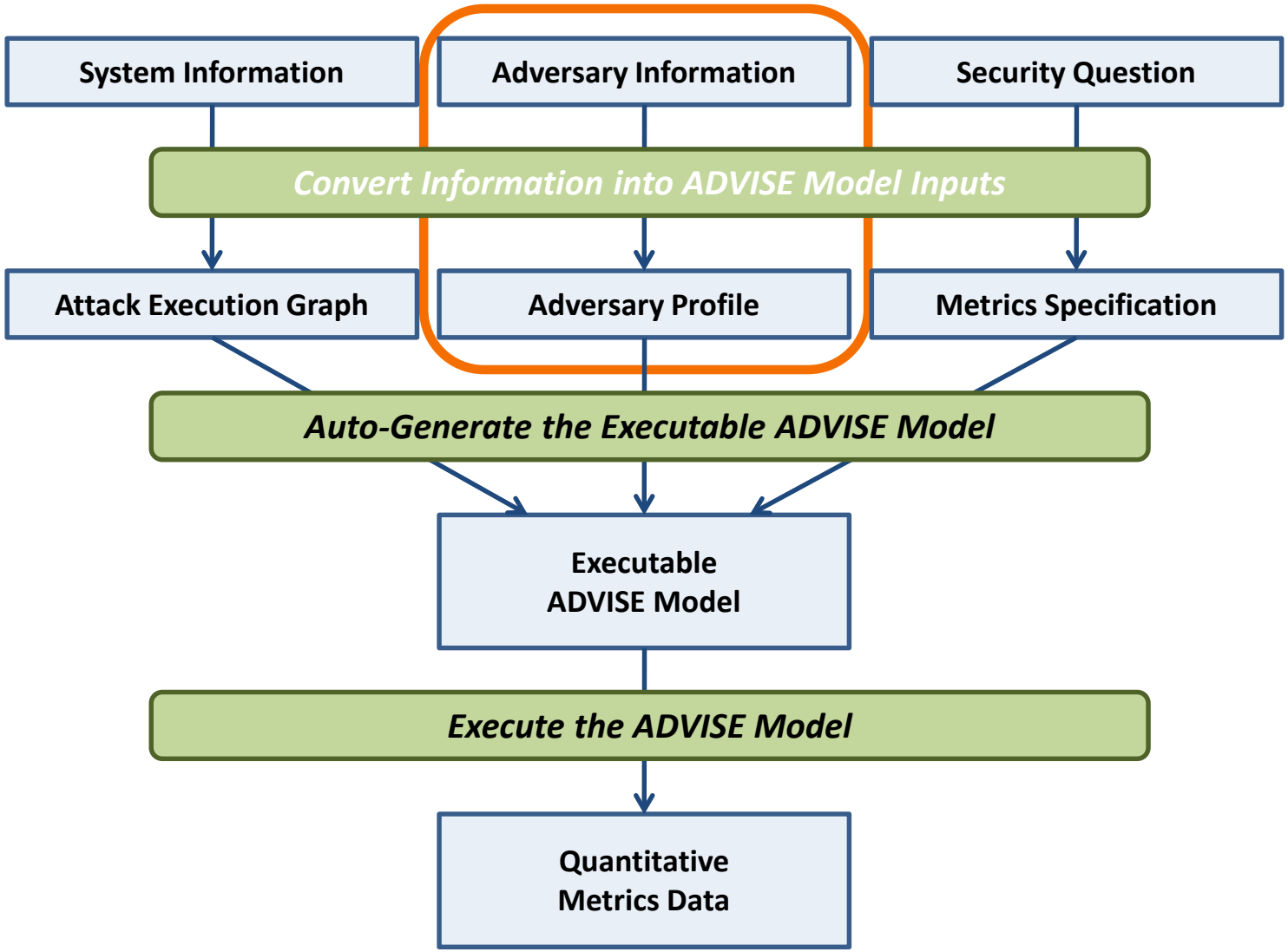
O_i is a finite set of outcomes, e.g., {Success, Failure}.

Pr_i: $X \times Oi \rightarrow [0, 1]$ is the probability of outcome o ∈ Oi occurring,
e.g., if (VPN exploit skill > 0.8) {0.9, 0.1} else {0.5, 0.5}.

D_i: $X \times Oi \rightarrow [0, 1]$ is the probability of the attack being detected when outcome o ∈ Oi occurs, e.g., {0.01, 0.2}.

E_i: $X \times Oi \rightarrow X$ is the next-state that results when outcome o ∈ Oi occurs,
e.g., {gain Network Access, no effect}.

ADVISE Method Overview



ADVISE Adversary Information: Adversary Profile

The adversary profile is defined by the tuple

<s0, L, V, wC, wP, wD, UC, UP, UD, N>,

where

$s_0 \in X$ is the initial model state, e.g., has Internet Access & VPN password,

L is the attack skill level function, e.g. has VPN exploit skill level = 0.3,

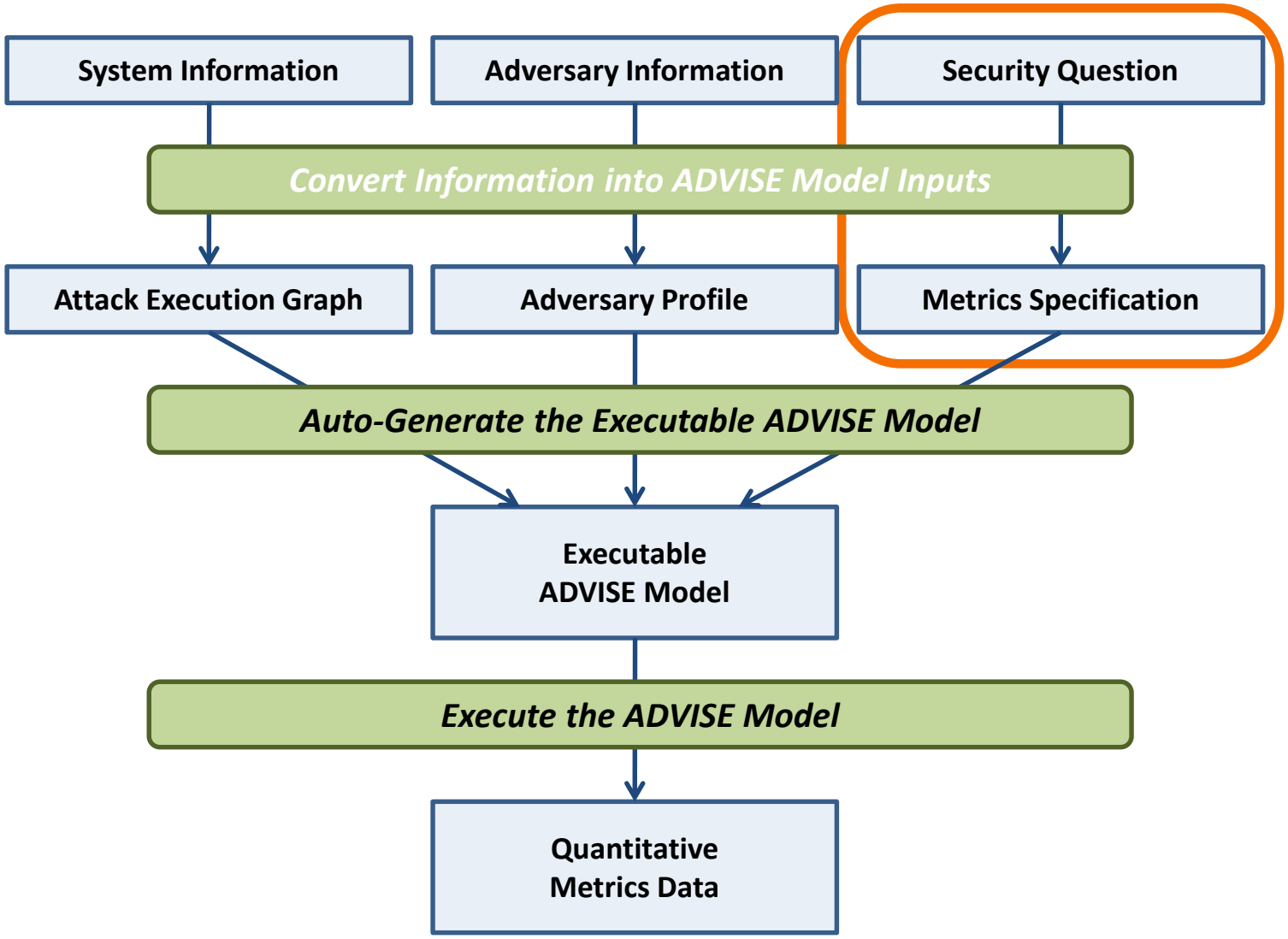
V is the attack goal value function, e.g., values "View contents of network" at $5000,

w_C, w_P, and w_D are the attack preference weights for cost, payoff, and detection probability, e.g., $w_C = 0.7$, $w_P = 0.2$, and $w_D = 0.1$,

U_C, U_P, and U_D are the utility functions for cost, payoff, and detection probability, e.g., $U_C(c)=1 - c/10000$, $U_P(p)=p/10000$, $U_D(d)=1 - d$, and

N is the planning horizon, e.g., N = 4.

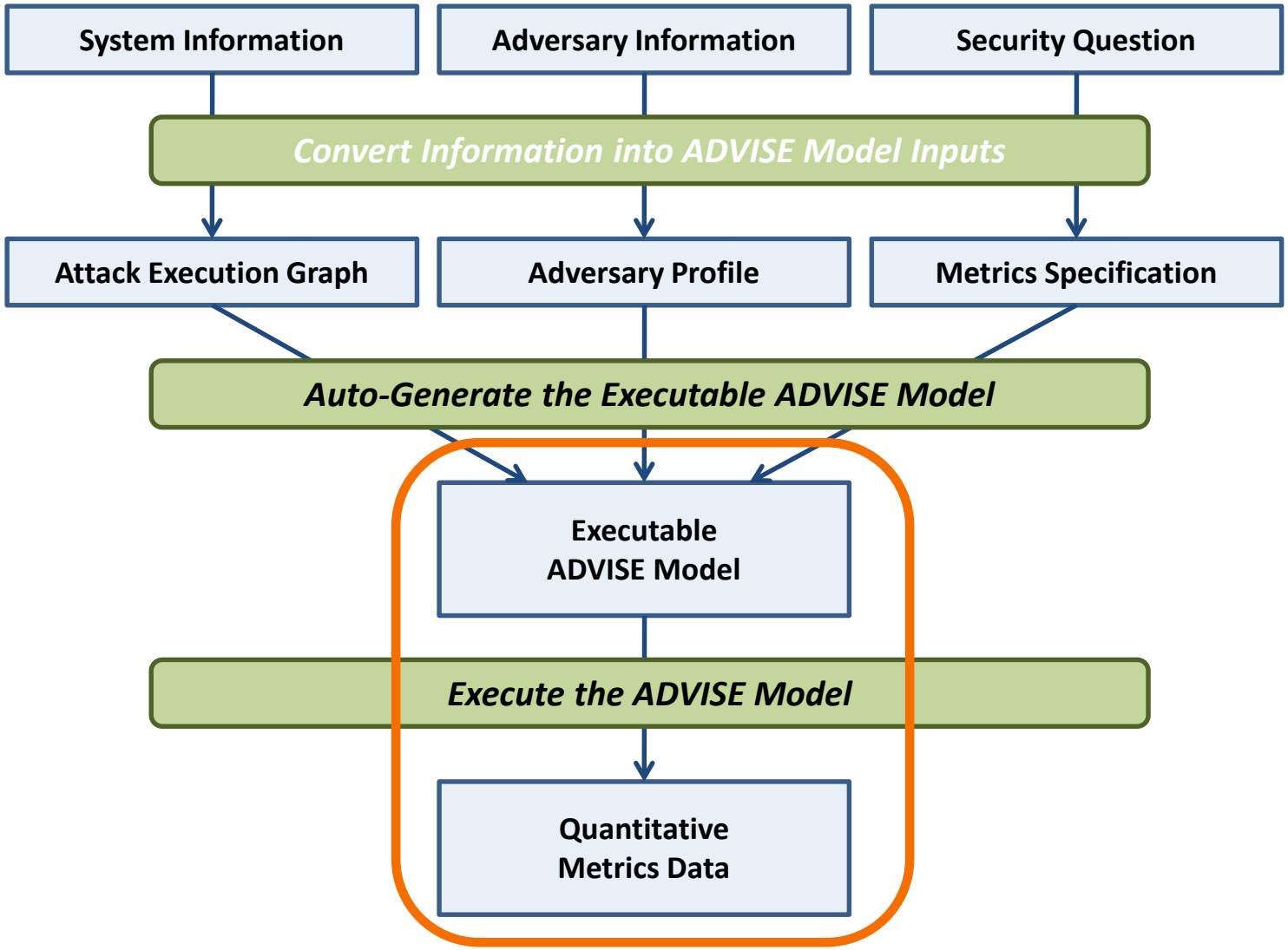
ADVISE Method Overview



ADVISE Security Question: Metrics Specification

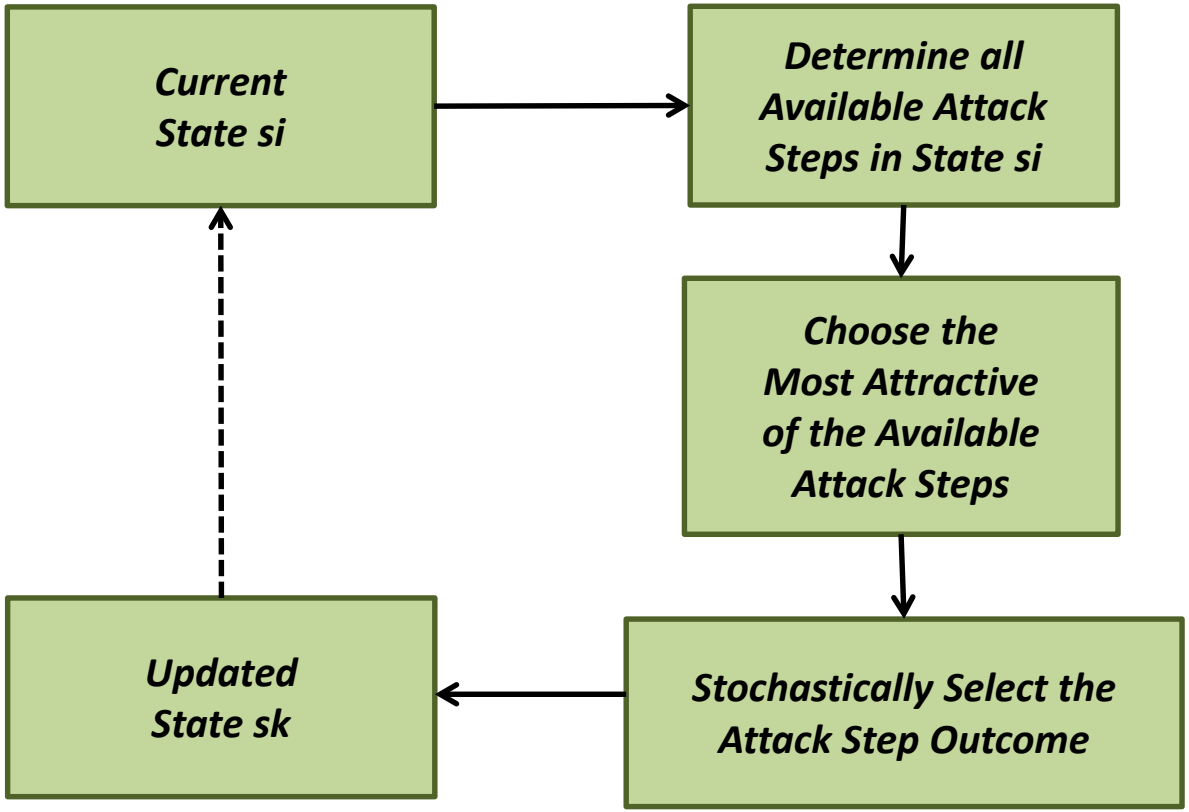
- State metrics analyze the model state
  - State occupancy probability metric (probability that the model is in a certain state at a certain time)
  - Average time metric (average amount of time during the time interval spent in a certain model state)
- Event metrics analyze events (state changes, attack step attempts, and attack step outcomes)
  - Frequency metric (average number of occurrences of an event during the time interval)
  - Probability of occurrence metric (probability that the event occurs at least once during the time interval)

ADVISE Method Overview



Model Execution: the Attack Decision Cycle

- The adversary selects the most attractive available attack step based on his attack preferences.
- State transitions are determined by the outcome of the attack step chosen by the adversary.



ADVISE Model Execution Algorithm

1: Time ← 0

2: State ← s_0

3: **while** Time < EndTime **do**

4: Attack$_i$ ← β^N(State)

5: Outcome ← o, where o ~ Prob$_i$(State)

6: Time ← Time + t, where t ~ T$_i$(State)

7: State ← E$_i$(State, Outcome)

8: **end while**

Simulation time and model state initialization

Adversary attack decision

Stochastic outcome

Time update

State update

$\beta^N(s)$ selects the most attractive available attack step in model state *s* using a planning horizon of *N*

Goal-driven Adversary Decision Function

When the planning horizon N is greater than 1, the
attractiveness of an available next step
is a function of
the payoff in the expected states
N attack steps from the current state

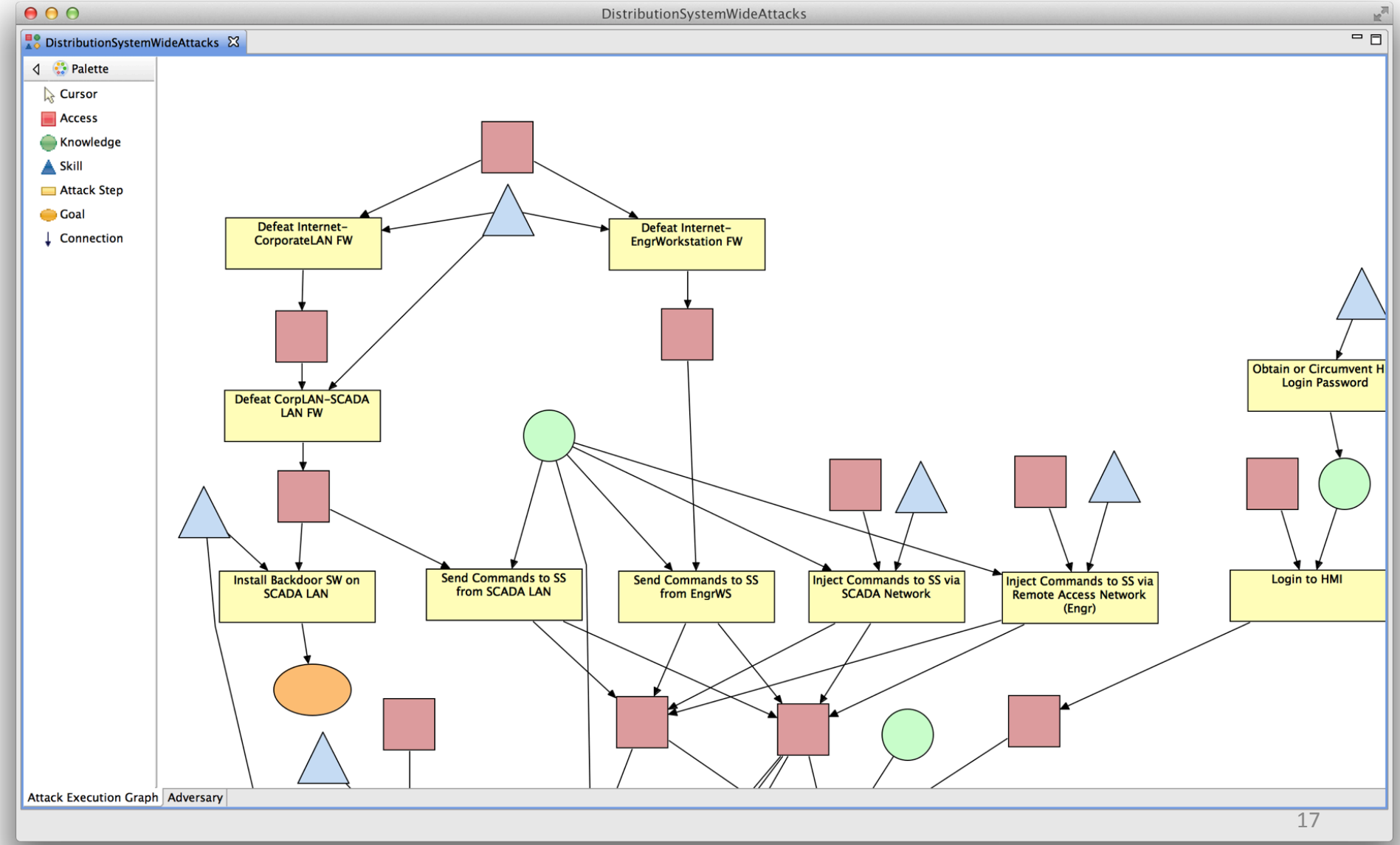
(the **expected horizon payoff**)

and

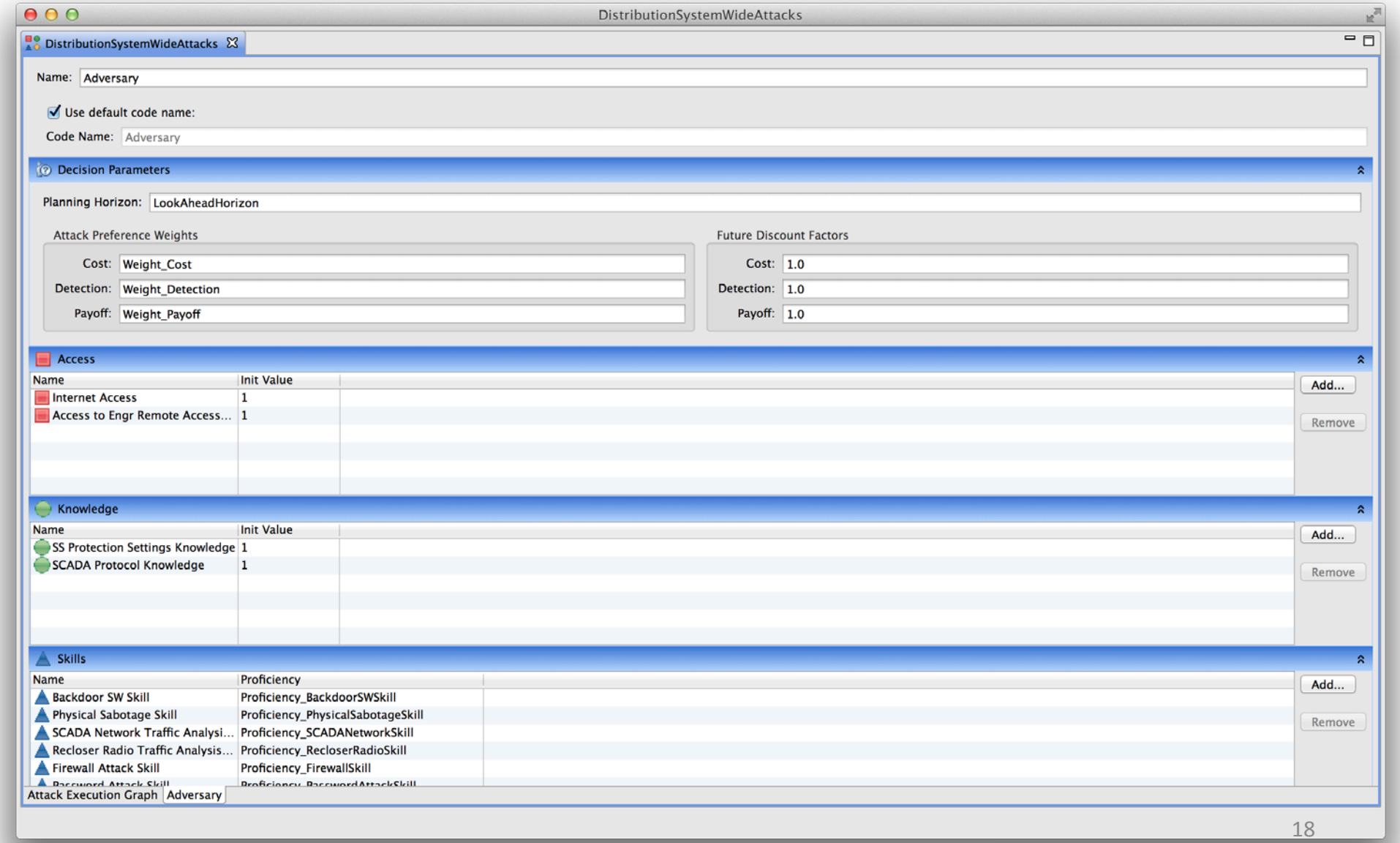
the expected cost and detection probability
of those N attack steps

(the **expected path cost** and **expected path detection**).

ADVISE Attack Execution Graph



ADVISE Adversary Profile

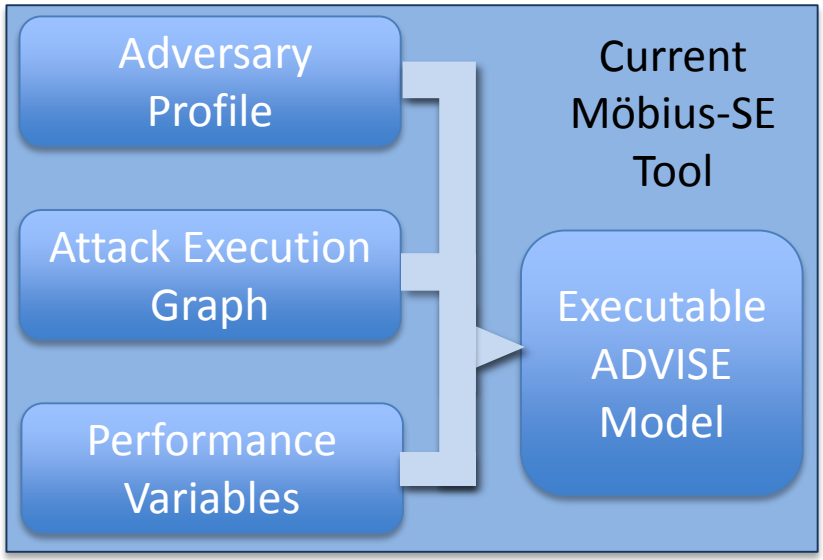


Security Metrics

- Average Number of Attempts
  - Report for each attack step
  - Gives insight on preferred attack path of adversary
- Probability of Attack Goal Achieved at End Time
  - Report for each attack goal
  - Gives insight on what goals the adversary is actively pursuing and reaching
- Average Time-To-Achieve-Goal
  - For attack goals where the above probability metric is 1 (or close to 1)
  - Gives insight on the speed of the adversary's attack

ADversary VIew Security Evaluation (ADVISE) Approach to Cyber Security Metrics

- Adversary-driven analysis
  - Considers characteristics and capabilities of adversaries
- State-based analysis
  - Considers multi-step attacks
- Quantitative metrics
  - Enables trade-off comparisons among alternatives
- Mission-relevant metrics
  - Measures the aspects of security important to owners/operators of the system
- DHS program FA8750-09-C-0039, 2009-2011 achieved design and implementation
  - Leveraged mature Möbius simulation platform

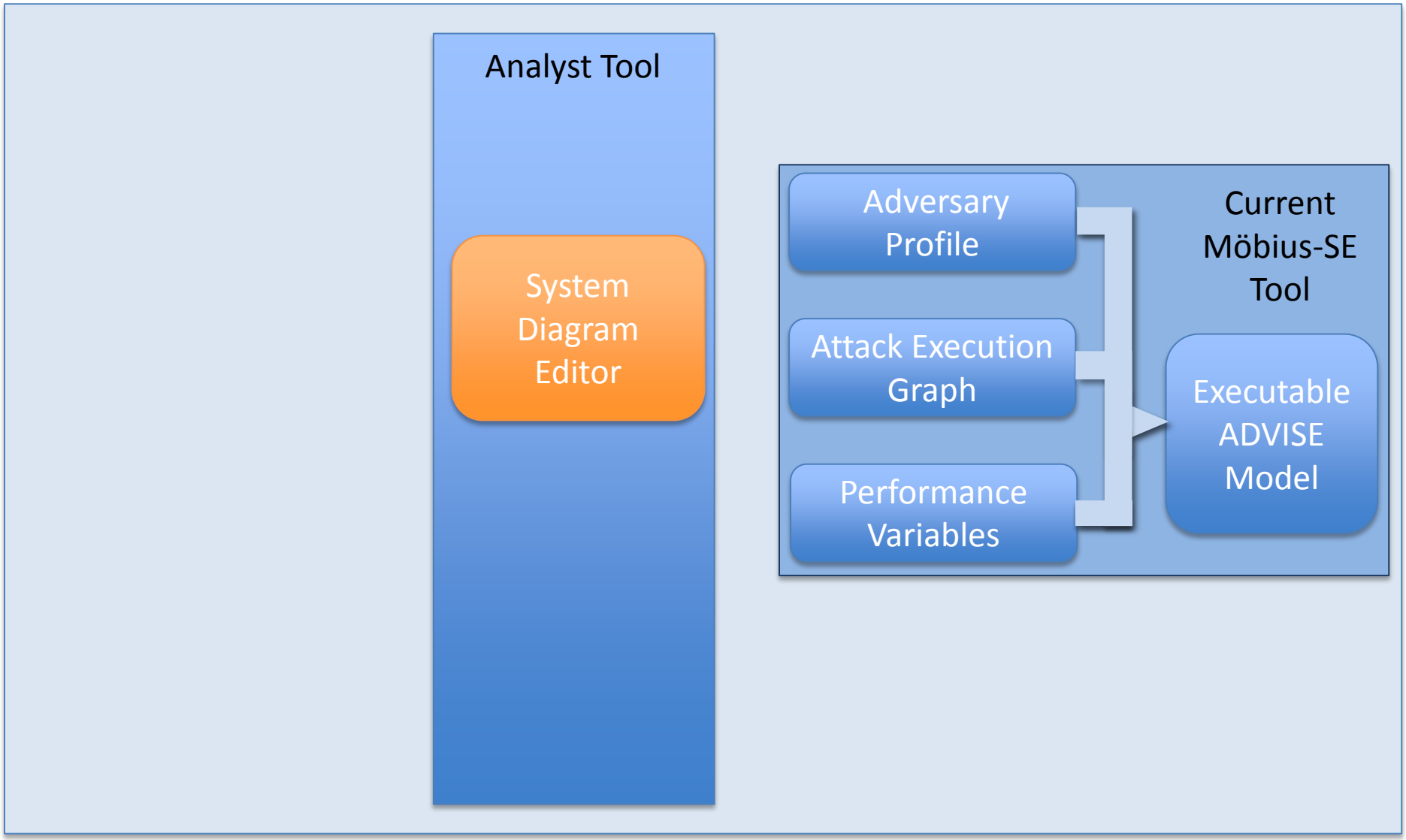


User Roles - System Analyst

- Builds a *System Diagram* of a specific system to be modeled.
  - Includes components and services
    - Components are instances of networked objects in component ontologies (core, community, and internal)
    - Service is any user-defined activity supported or partially supported by system
  - Relationships between these entities are defined (e.g. Network_Connection, Controls, Required_For)
  - Related attributes are defined (e.g. cost of component, capacity, QoS levels)

Program Goals Addressed:
- User defines high-level system model.
- System analyst requires system expertise.
- System and adversary separated.

Enhanced Tool Functional Architecture

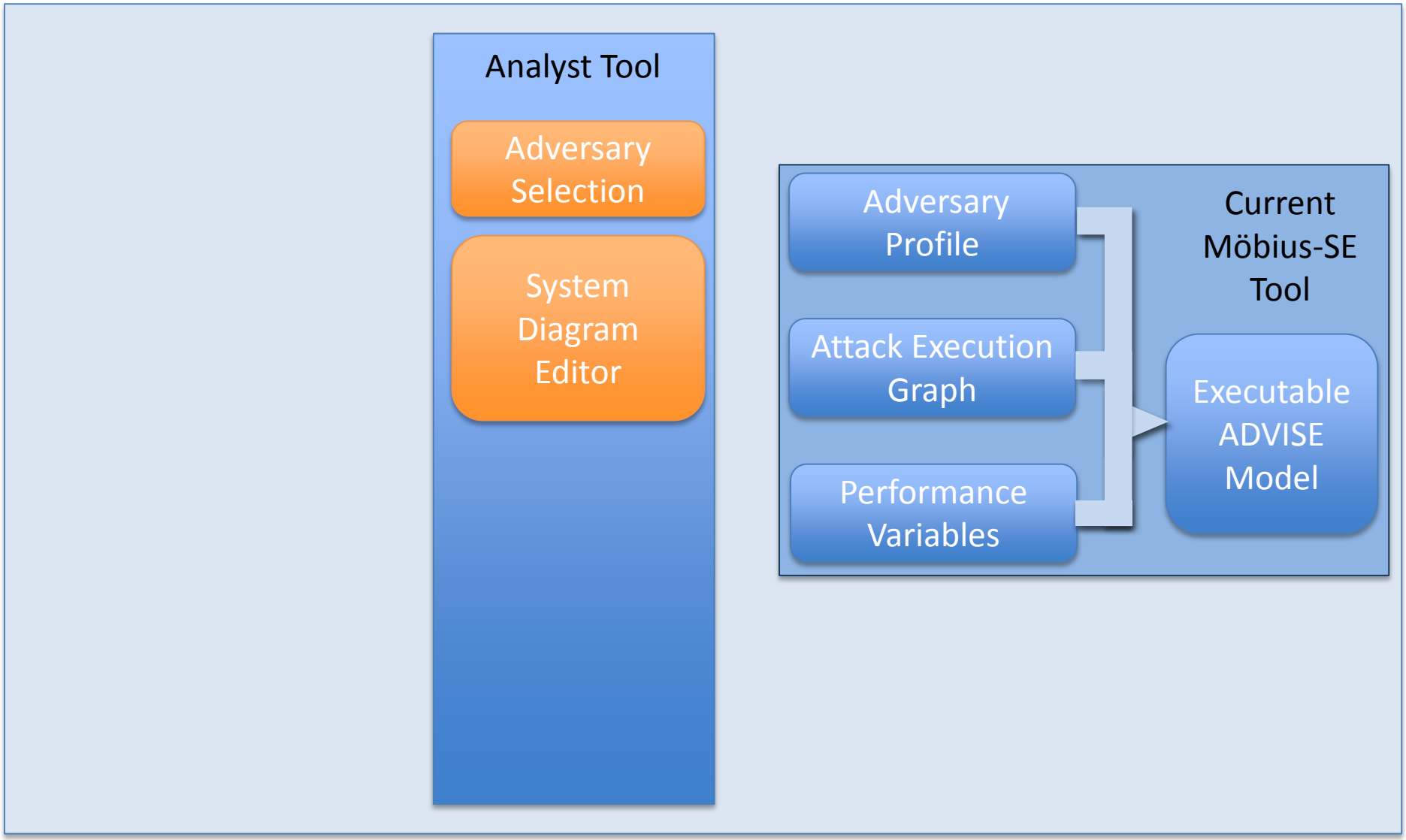


User Roles - System Analyst

- Selects a set of *Adversaries*
  - Adversaries chosen from library based on adversary ontology
  - Preferences and other attributes can be overridden
  - Access, Skill, Knowledge, Goals defined from choices generated from system diagram

Program Goals Addressed:
- System and adversary separated.
- Any adversary can attack any system.

Enhanced Tool Functional Architecture

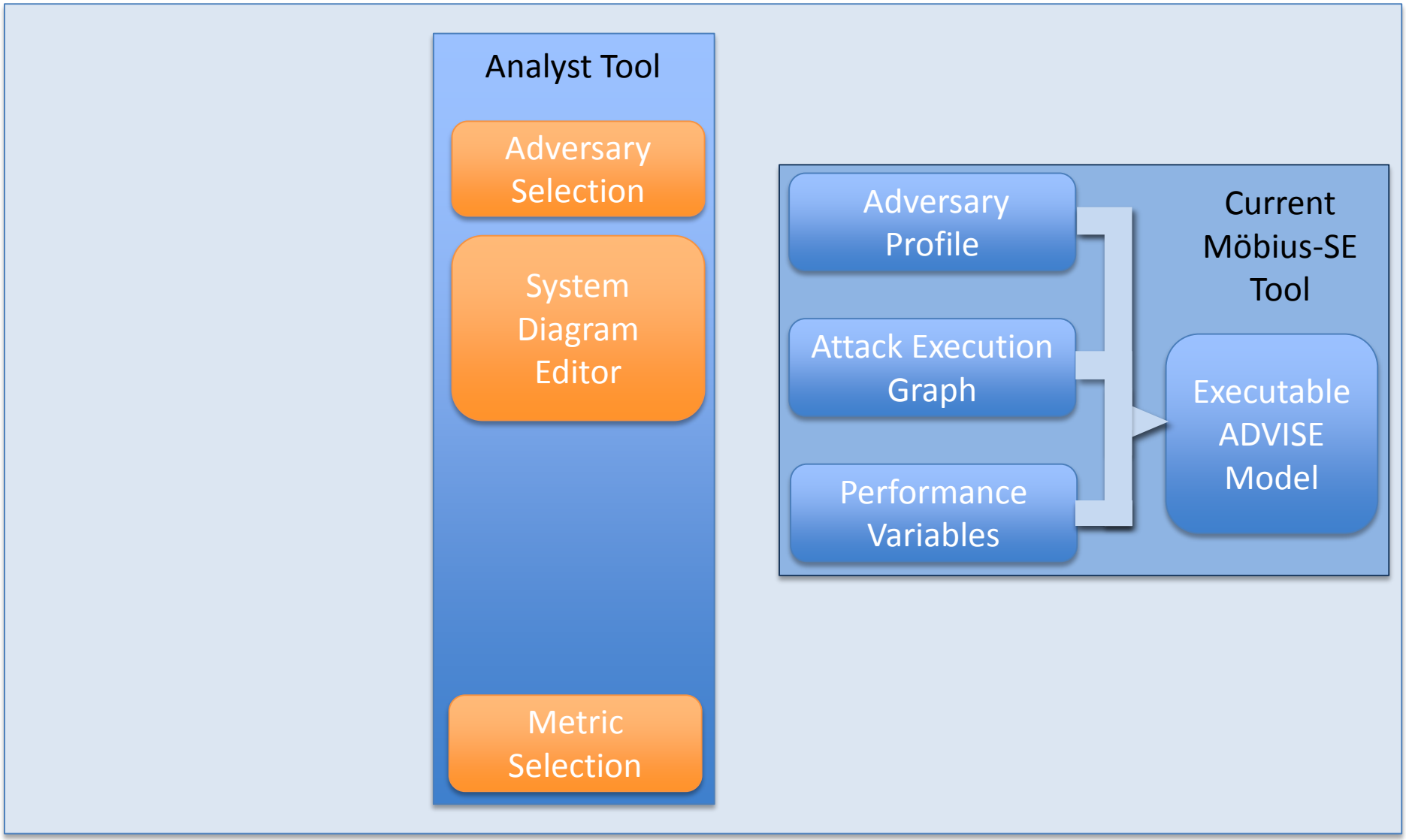


User Roles - System Analyst (cont'd)

- Selects a set of *Metrics*
  - Metrics chosen from library based on metric ontology
  - Metric parameters specified as needed

Program Goals addressed:
- Metrics imported from ontology.

Enhanced Tool Functional Architecture

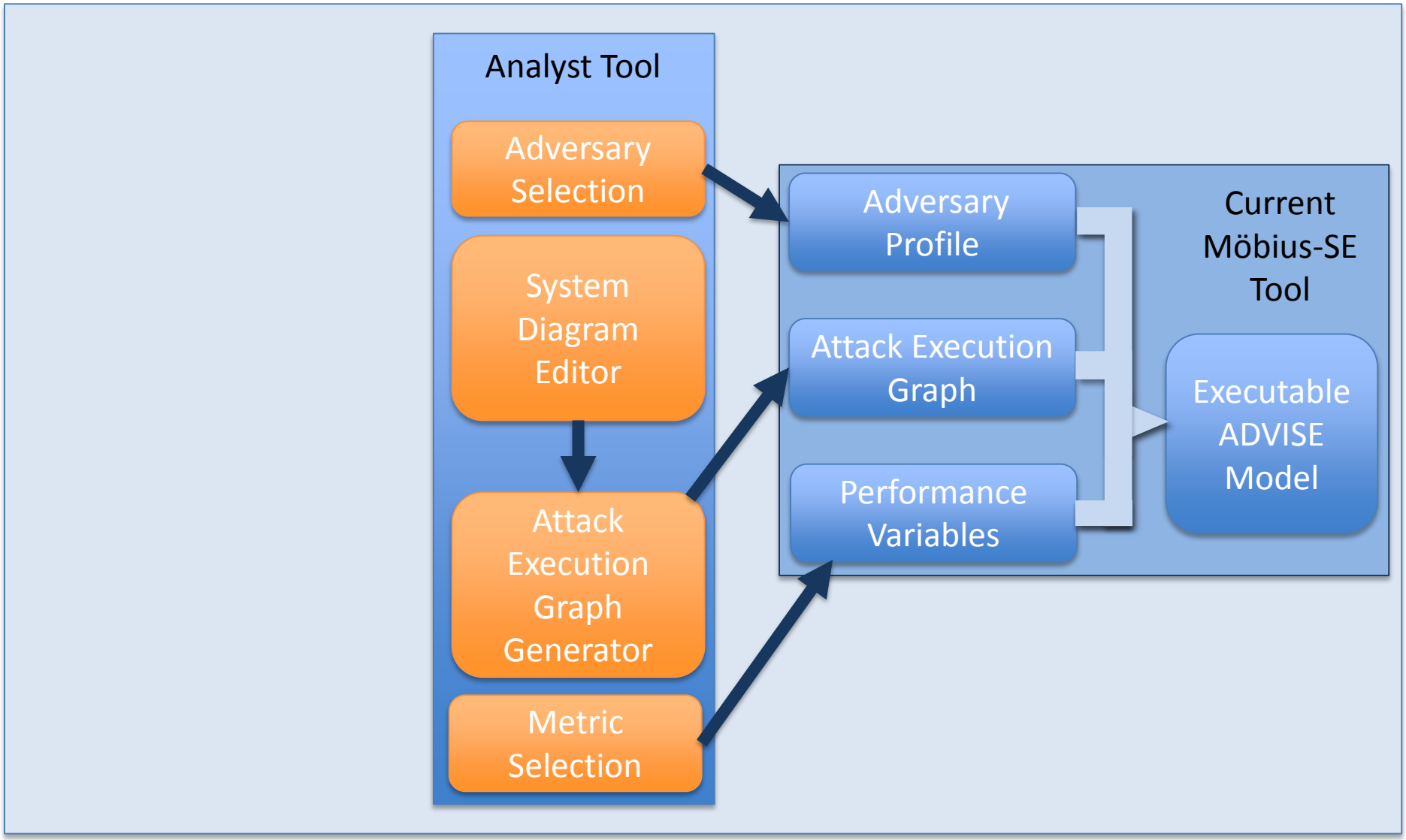


User Roles - System Analyst (cont'd)

- Defines a set of experiment *Configurations*
  - Pairs adversary, subset of selected metrics
  - Additional generation options per configuration
  - Each configuration generates fully executable model
  - May specify values as variables to easily create different configurations
- Generate an ADVISE model from the:
  - System Diagram
  - Adversaries
  - Metrics
  - Configurations

Program Goals Addressed:
- Flexible configurations.
- AEG automatically generated.
- Attack steps derived from component type and relationships.

Enhanced Tool Functional Architecture



Libraries

- Ontologies defined in SADL (Semantic Application Design Language)
  - Adversaries
  - Components
  - Attack Steps
  - Metrics
- Ontologies are flexible and extensible enabling common, shareable foundation across all users
  - Core
  - Vendor
  - Community
  - Internal

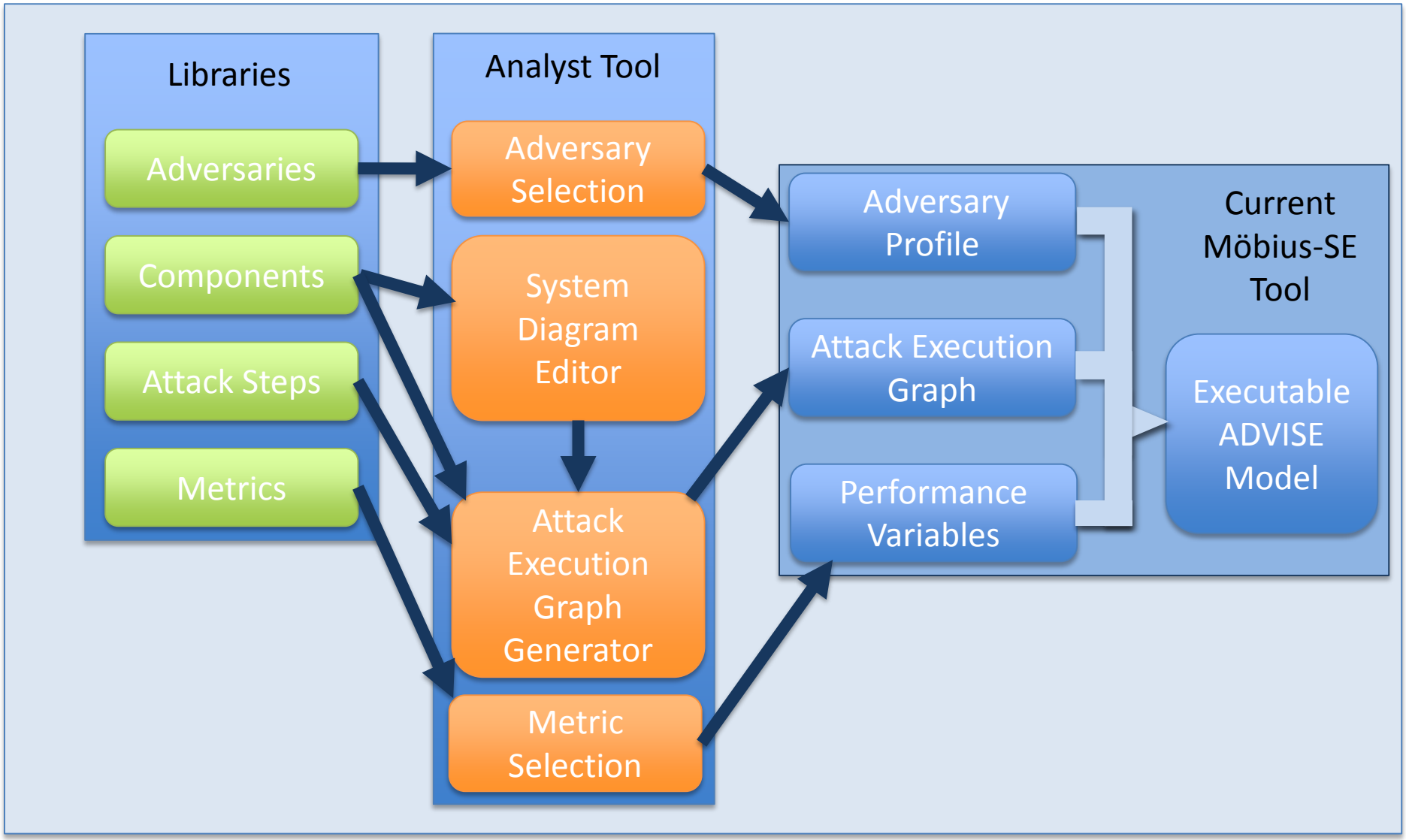
System.sadl

```
22 describedBy is the inverse of about.
23
24 Data is a type of Information.
25 {AdminData, UserData, AppData} are types of Data.
26 {DataAtRest (alias "Stored data"), DataInMotion} are types of Data.
27 ConfigurationData is a type of AdminData.
28
29 Function is a class.
30 {Control, Authentication, Encryption, DataFunction} are types of Function.
31 {View, Read} are types of DataFunction.
32
33 target describes Control with values of type Device.
34 information describes Read with values of type Information.
35
36 Software is a type of Subsystem, described by function with values of type Function.
37 {OperatingSystem (alias "OS"), Application} are types of Software.
38 hardwarePlatform describes Software with values of type Device.
39
40 // TODO: need relationship of Data to Software, e.g., UserData, AdminData to OS, AppData to Application
41 managedby describes Data with values of type Software.
42
43 A Data is an AppData only if managedby has at least one value of type Application. // without OWL entailn
44 Rule AppDataRule:if app is an Application and d is Data and d managedby app then d is AppData. // only ne
45
46 EmailApp is a type of Application.
47 HistorianApp is a type of Application.
```

Program Goals Addressed:

- Ontologies provide well-defined, extensible, and portable definitions for components, adversaries, attack steps, and metrics.

Enhanced Tool Functional Architecture



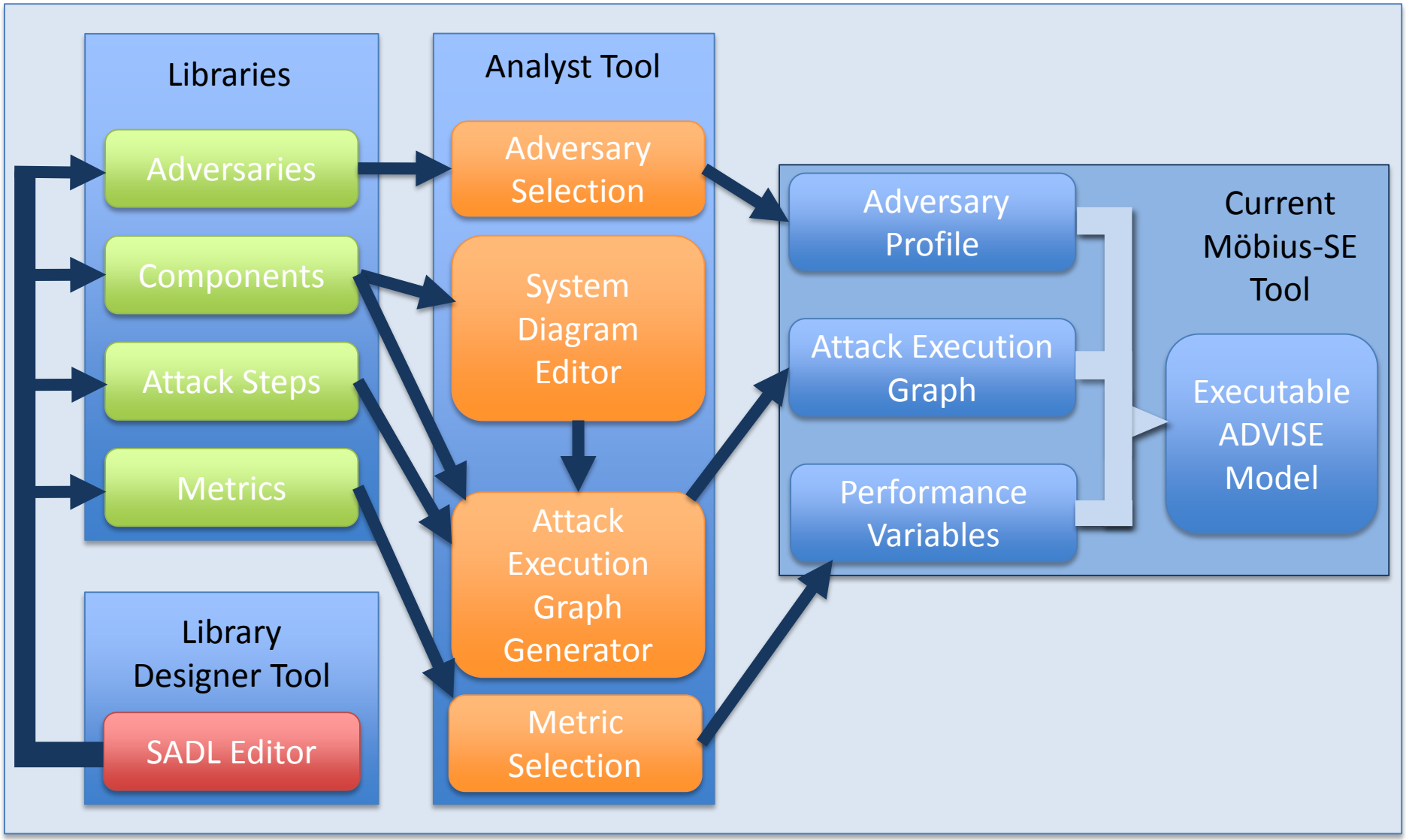
User Roles - Library Designer (optional)

- Leverages core and community ontologies to provide enhanced elements available for the System Analyst
- Ontologies written using SADL (GE)
- Tool provides a text environment to work with SADL code

Program Goals Addressed:

- Library designer requires security expertise.
- Ontologies provide well-defined, extensible, and portable definitions for components, adversaries, attack steps, and metrics.

Enhanced Tool Functional Architecture



Semantic Application Design Language (SADL)

- English like language for semantic models.

Network is a type of System.
{WiredNetwork, WirelessNetwork} are types of Network.

Component is a type of System,
described by connectedTo with values of type Network.
connectedTo of Component has at least one value of type Network.

Device
is a type of {Component and PhysicalThing}.

{Host, Router, Controller, Gateway} are types of Device.

Conclusions

- Since system security cannot be absolute, quantifiable security metrics are needed
- Metrics are useful event if not perfect; e.g., relative metrics can aid in critical design decisions
- The ADVISE formalism, and its implementation in Mobius
  - Is rich enough to adversary, user, and system behavior
  - Natural for security analysts
  - Semantically precise
- ADVISE was included in the recent general release of Mobius

Thank you!

Ken Keefe
www.perform.illinois.edu/~kjkeefe
kjkeefe@illinois.edu