



Human Factors in Security Administration: Brainstorming the Research Directions

**Konstantin Beznosov
University of British Columbia**



Outline

- State of the practice
- State of the art
- Research directions ideas

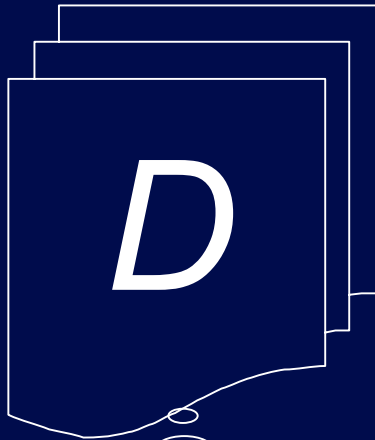


State of the Practice



Classical Access Control Solution

Domains

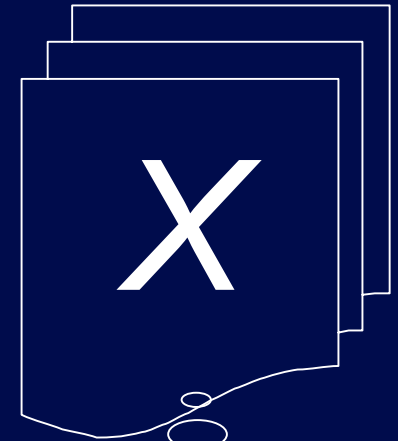


Have access to objects

Access Matrix

	Domain 1	Domain 2	Domain 3	File 1	File 2	Process 1
Domain 1	*owner control	*owner control	*call	*owner *read *write		
Domain 2			can	*read	write	wakeup
Domain 3			owner control	read	*owner	

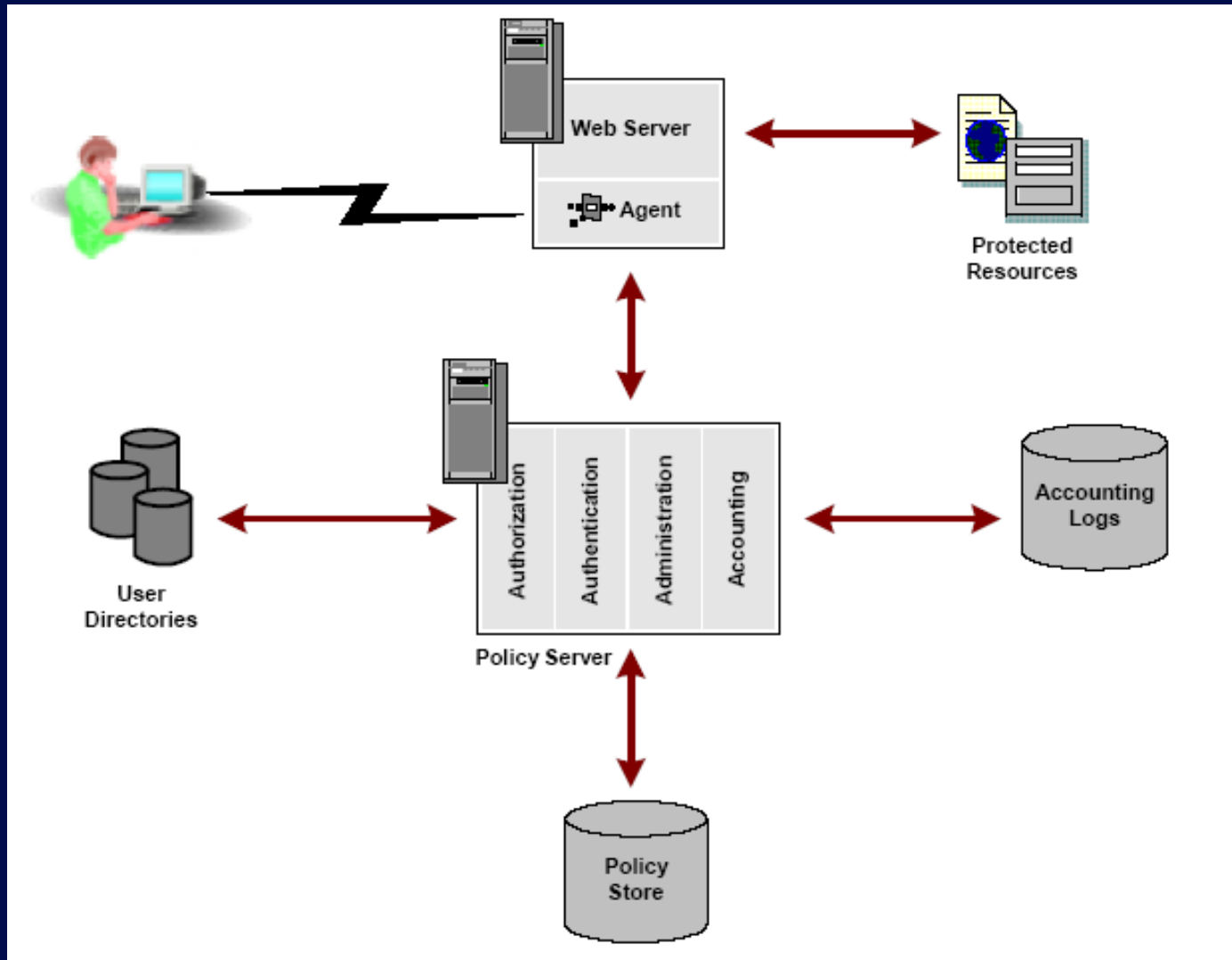
Objects



To be protected



Enterprise-scale authentication/authorization server





Everything starts with simple directory tree like structure

The screenshot shows the SiteMinder Administration console. The title bar reads "SiteMinder Administration" and the menu bar includes "Session", "Edit", "View", "Tools", "Advanced", and "Help". The "System" tab is selected, and the left-hand navigation pane shows a tree structure with "System Configuration" expanded. The main area displays an "Object List" table with two columns: "Name" and "Description". The table is currently empty.

Name	Description
------	-------------

The screenshot shows the "Domains" tab in the SiteMinder Administration console. It displays a hierarchical tree structure of domains and realms. The "Domain1" node is expanded, showing its sub-objects. The "Realms" node is also expanded, showing "Domain1" and "REALM RealmA". The "REALM RealmA" node is further expanded, showing "RADIUS RealmA". Other nodes in the tree include "Rule Groups", "Responses", "Response Groups", and "Policies".

- Policy Domains
 - Domain1
 - Realms
 - Domain1
 - REALM RealmA
 - RADIUS RealmA
 - Rule Groups
 - Responses
 - Response Groups
 - Policies



Then continues with simple forms to fill out ...

The screenshot displays three overlapping SiteMinder configuration windows:

- SiteMinder Active Rule Editor:** The main background window showing the configuration environment.
- SiteMinder Realm Dialog:** A dialog box for configuring a realm. The "Realm Properties" tab is active, showing:
 - *Name: MyRealm
 - Description: (empty)
 - Advanced tab selected.
 - Registration section: "New users access this Realm will be by this registration scheme" with a dropdown menu showing "Registration", "(None)", and "Registration".
 - Events section: "Process Authentication" checked.
 - Bottom: "Realm MyRealm" and "Signed by: Netegrity, Inc."
- SiteMinder Rule Dialog:** A dialog box for configuring a rule. The "Rule Properties" tab is active, showing:
 - *Name: DMS 0 Launch
 - Description: (empty)
 - Realm and Resource section: "Realm: DMS 0 Launch" (dropdown), "Resource: *" (text field), "Effective Resource: [gdemetrick\(192.168.2.164\)/servlet/MSR/Launch/*](#)" (text field).
 - Bottom: "Perform regular expression pattern matching" checked.
- SiteMinder Authentication Scheme Dialog:** A dialog box for configuring an authentication scheme. The "Authentication Scheme Properties" tab is active, showing:
 - *Name: DMS 1 Admin
 - Description: DMS Administration Authentication Scheme
 - Scheme Common Setup section: "Authentication Scheme Type" dropdown set to "HTML Form Template", "Protection Level: 5" (range 1-20), and "Password Policies Enabled for this Authentication Scheme" checked.
 - Scheme Type Setup section: "Server Name: myserver.myorg.orgcom", "Use SSL Connection" checked, "Target: /siteminderagent/forms/login.fcc", and "Allow Form Authentication Scheme to Save Credentials" unchecked.
 - Bottom: "Additional Attribute List" (empty text field).
 - Buttons: OK, Cancel, Apply.
 - Bottom status: "Authentication Scheme DMS 1 Admin"



... or select

Time Dialog [X]

Set Time Restriction [HELP]

Effective Starting Date: <now> [Select...]

Expiration Date: <never> [Select...]

Hourly Restrictions

	A.M.											Noon												P.M.										
	12	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3	4	5	6	7	8	9	10	11										
Sunday	Don't Fire	Don't Fire	Don't Fire	Don't Fire	Don't Fire	Don't Fire	Don't Fire	Don't Fire	Don't Fire	Don't Fire	Don't Fire	Don't Fire	Don't Fire	Don't Fire	Don't Fire	Don't Fire	Don't Fire	Don't Fire	Don't Fire	Don't Fire	Don't Fire	Don't Fire	Don't Fire	Don't Fire										
Monday	Don't Fire	Don't Fire	Don't Fire	Don't Fire	Don't Fire	Don't Fire	Don't Fire	Don't Fire	Don't Fire	Fire	Fire	Fire	Fire	Fire	Fire	Fire	Fire	Fire	Fire	Fire	Fire	Fire	Fire	Fire										
Tuesday	Don't Fire	Don't Fire	Don't Fire	Don't Fire	Don't Fire	Don't Fire	Don't Fire	Don't Fire	Don't Fire	Fire	Fire	Fire	Fire	Fire	Fire	Fire	Fire	Fire	Fire	Fire	Fire	Fire	Fire	Fire										
Wednesday	Don't Fire	Don't Fire	Don't Fire	Don't Fire	Don't Fire	Don't Fire	Don't Fire	Don't Fire	Don't Fire	Fire	Fire	Fire	Fire	Fire	Fire	Fire	Fire	Fire	Fire	Fire	Fire	Fire	Fire	Fire										
Thursday	Don't Fire	Don't Fire	Don't Fire	Don't Fire	Don't Fire	Don't Fire	Don't Fire	Don't Fire	Don't Fire	Fire	Fire	Fire	Fire	Fire	Fire	Fire	Fire	Fire	Fire	Fire	Fire	Fire	Fire	Fire										
Friday	Don't Fire	Don't Fire	Don't Fire	Don't Fire	Don't Fire	Don't Fire	Don't Fire	Don't Fire	Don't Fire	Fire	Fire	Fire	Fire	Fire	Fire	Fire	Fire	Fire	Fire	Fire	Fire	Fire	Fire	Fire										
Saturday	Don't Fire	Don't Fire	Don't Fire	Don't Fire	Don't Fire	Don't Fire	Don't Fire	Don't Fire	Don't Fire	Don't Fire	Don't Fire	Don't Fire	Don't Fire	Don't Fire	Don't Fire	Don't Fire	Don't Fire	Don't Fire	Don't Fire	Don't Fire	Don't Fire	Don't Fire	Don't Fire	Don't Fire										

Always Fire []
Never Fire []

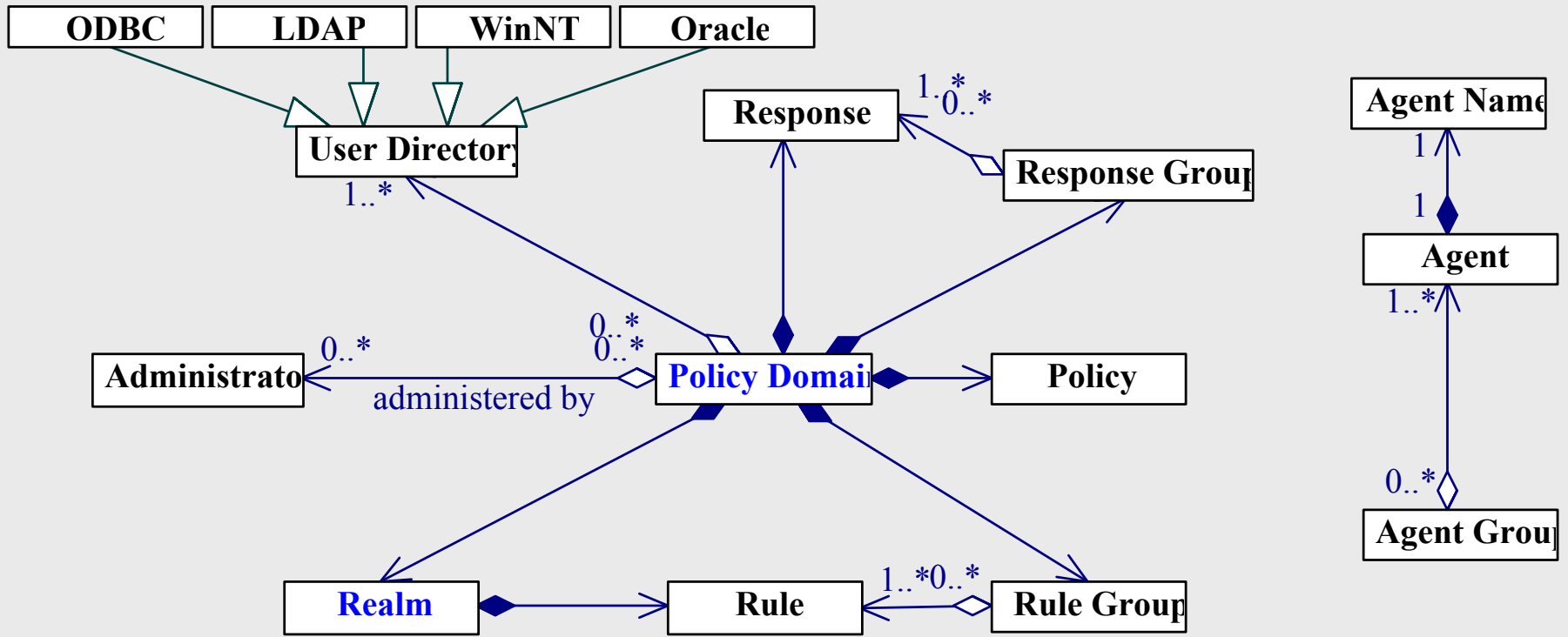
Rule Behavior
Fire []
Don't Fire []

OK Cancel Reset

Unsigned Java Applet Window

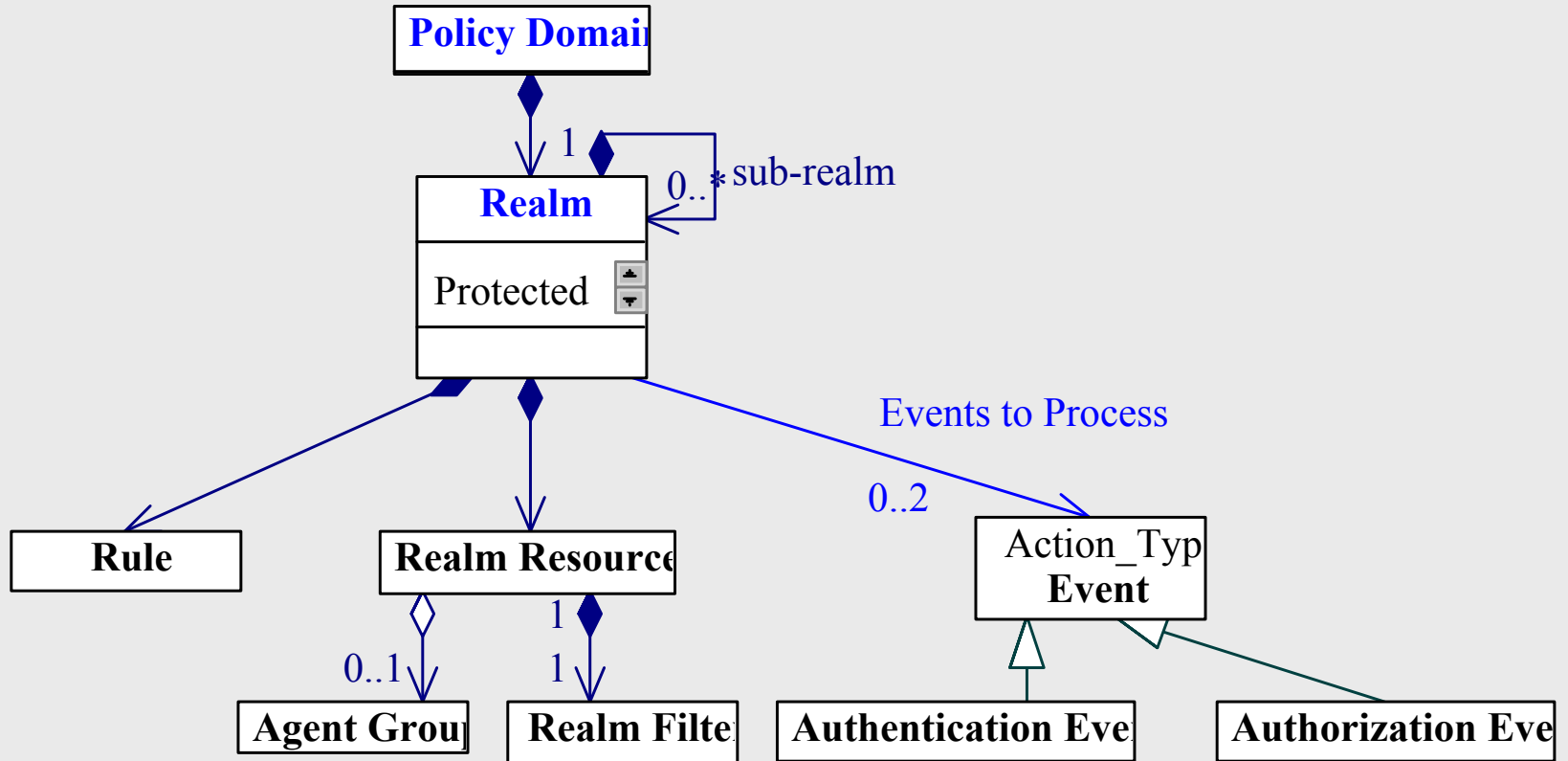


But the mental model is complex ...



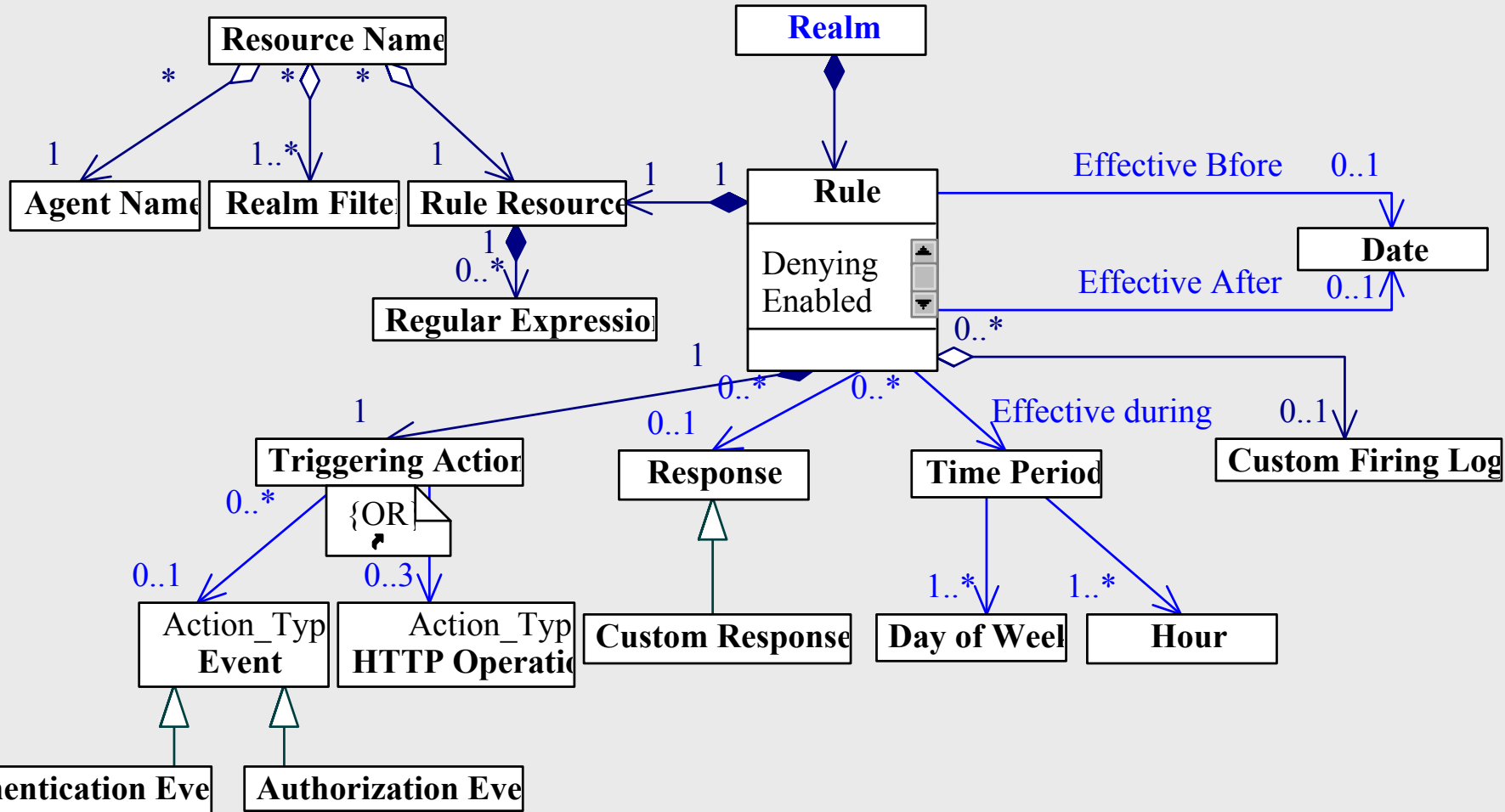


... and even more ...





... complex





So what?

- Steep learning curve
- Hard to fit real world into the model
- Easy to make costly mistakes
 - “friendly” DoS
 - inadvertent hard to catch vulnerabilities
- Hard to test
 - Expensive to test required scenarios
 - No “what if” scenarios to test before changing
 - Hard to perform complete testing
- Lacks domain-specific abstractions



State of the Art (Science, Engineering?)



Security Usability Comes in ...

Waves:

- First, mid 1970s: **acknowledging**
 - [Saltzer and Schroeder, 1975]
- Second, late 1980s: **evaluating**
 - [Karat, 1989], [Mosteller, W. S. and Ballas, 1989]
- Third, late 1990s -- early 2000s: little bit of everything
 - **acknowledging, raising** [Schultz, et al., 2001]
 - **evaluating** [Jendricke and Markotten, 2000], [Whitten, 1999]
 - **suggesting** how to address [Holmstrom, 1999], [Patrick and Kenny, 2003], [Whitten and Tygar, 2003], [Yee, 2002]
 - **building community** (HCI & Security Systems Wrkshp, 2003)

with 14 year period!



What kind of waves?





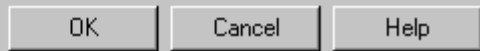
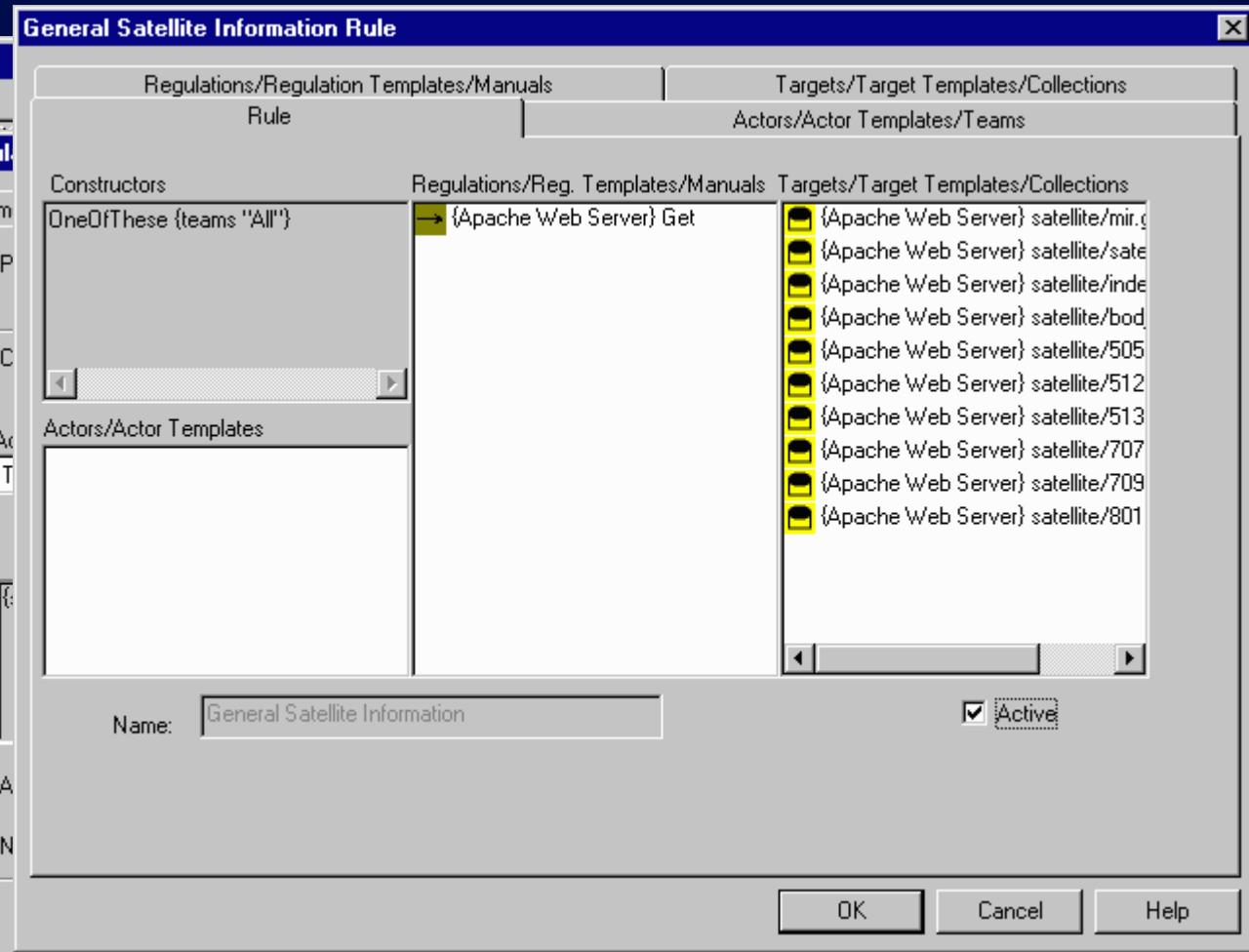
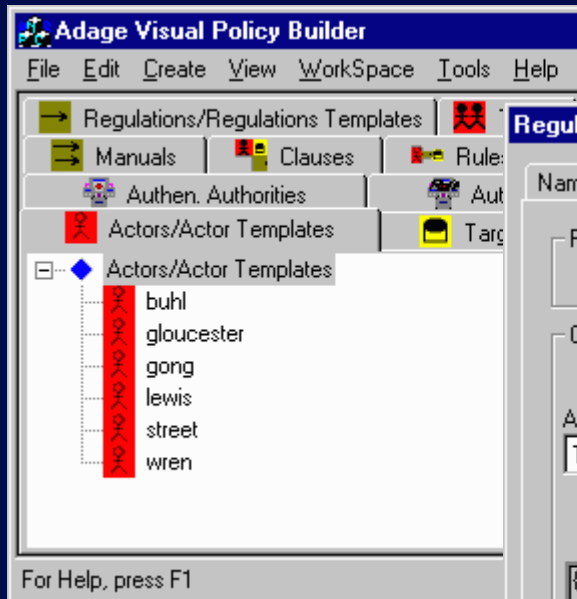
What about security administration and HCI?

One published study [Zurko, Simon, Sanfilippo, 1999], [Zurko, Simon, 1996]

- Experimental authorization system **Adage**
 - trial-and-error approach with feedback gained through:
 - contextual interviews,
 - verbal protocols, and
 - affinity mapping techniques for interpreting and categorizing notes on the subject's actions,
 - discount usability testing
 - lab testing
- Admin UI usability studies
- Results
 - admins are willing to learn the UI
 - difficult to
 - display events such as role conflicts
 - provide good feedback of admin actions' effects
 - to comprehend system model | difficult to visually display the global system overview
 - Would be nice to have
 - "test" mode could be very useful
 - ability to query the state of the system



Adage admin GUI Screenshots





Research Directions Ideas



What can be done at UBC?

1. Develop distributed secure application admin UI
 - A. Improve visualization of security information presented to administrators
 - B. Develop support for opportunistic and incremental task planning
 - C. Employ newly suggested principles
 - designing interactions with security mechanisms [Yee, 2002]
 - Safe Staging (creating learnable security software) [Whitten and Tygar, 2003]
2. Compare usability with existing technologies
 - CORBA, EJB, COM+, (ASP).NET