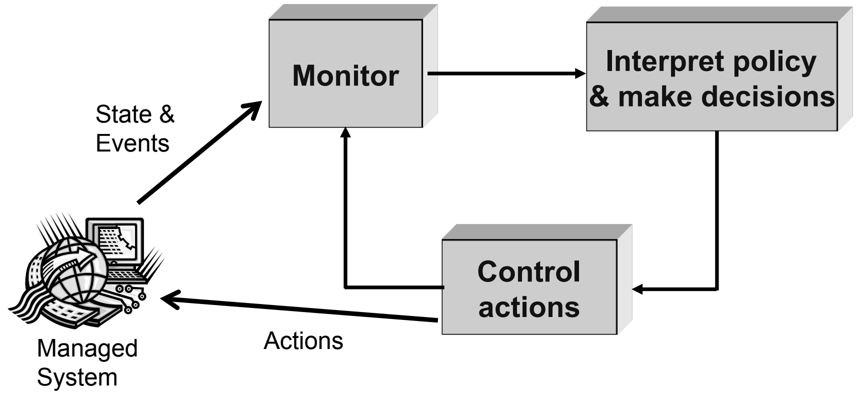
Distributed Systems Management

- What is management
- Management functions
- SNMP & MIBs
- Domains
- Management and security policy specification

Management

- Management of a system is required in order to supply and maintain the services required by users. It is responsible for long term planning, organising, controlling and supervising the components which constitute the distributed system as well as controlling adaptation to changes in context, failures or attacks
- Management policies derived from the goals of the users or organisation define the strategy for dynamically modifying the behaviour of the system.
  - Management will be distributed to reflect the distribution of the system it manages
  - Management is often implemented using the same tools and techniques for implementing a distributed or pervasive system.

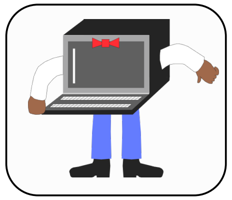
Management Control Loop

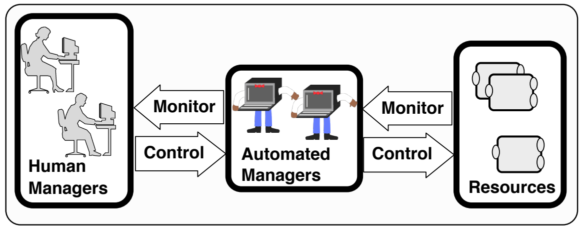


Managers

People

Automated manager agents





Motivation

- Networking and distributed systems are becoming strategic to achieving business objectives. Many organisations are becoming dependent on distributed computing systems e.g. banks, financial dealing rooms.
- Some applications require very high availability
  ➔ require management to make sure they continue working
- Users are non-technical and cannot perform management operations such as fault handling on the systems they use.
- There is a shortage of skilled operators, maintenance engineers etc.
  ➔ automation of management functions needed
- Large scale with millions of heterogeneous objects to be managed
  ➔ Deal with groups of objects rather than individual objects
  Delegate responsibility to automated distributed managers
- Systems will evolve in response to changing requirements & new technology
  ➔ Dynamic change of management system
  Separate policy interpreted by managers, not coded into them
- Need to structure management to:
  - Partition and demarcate responsibility
  - Reflect physical or organisational structure

Configuration Management

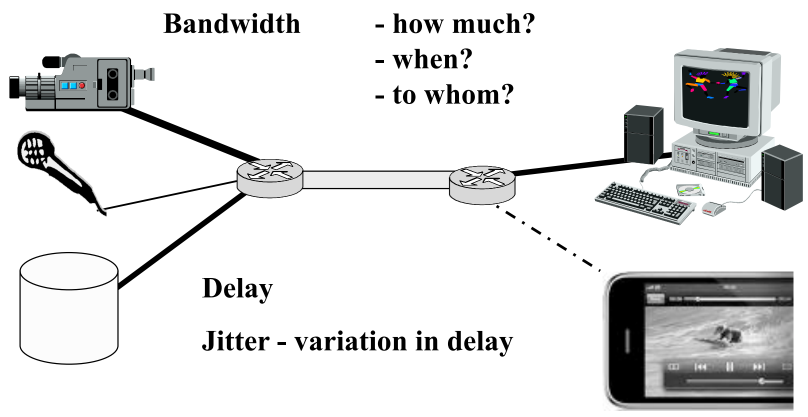
Building an initial system and subsequently changing it (Not version control)

- **Entails:**
  - Specification of components of a service type
  - Specification of instances
  - Allocation of software components to hardware
  - Binding interfaces or interconnecting components
  - Defining initialisation parameters
  - Reconfiguration to cope with failures, dimensioning
    ➔ *operational changes*
  - Reconfiguration to cope with new functionality
    ➔ *evolutionary changes*
- **When ?**
  - Statically – building initial system
  - Dynamically – operational & evolutionary changes

Monitoring

- Obtains information to make decisions and perform control actions
- Required for all aspects of management
- Generation of monitoring information
  - Status reporting by objects
  - Event generation eg failures, thresholds exceeded, security violations
- Processing of monitoring information
  - Merging and multiple trace generation
  - Validation, filtering & analysis
  - Combining events
- Dissemination of monitoring information
  - To operator displays, logs and manager agents
  - Registration of subscribers to dissemination service
  - Specification of information selection criteria

Quality of Service Management



QOS Management

- **QOS Specification**
  - Performance – response time, throughput, transit delay, jitter (variation in latency), error rates, failure probabilities, availability
  - Sometimes include cost and security attributes
  - Priority – criticality of an activity
  - ➔ Service contract between client & service provider
- **Negotiation**

  Provider must allocate resources to provide required QOS
  - To agree QOS between service provider & client.
  - Initially or during a session if quality cannot be maintained.
- **Admission Control**
  - Grant access to service only if sufficient resources available to meet QoS requirements

QOS Management

- **QOS monitoring & maintenance**
  - By service provider to make sure agreed quality is maintained
  - Take recovery action if QOS not within agreed limits.
  - Performance optimisation
- **Fault handling**
  - Detection
  - Diagnosis and analysis
  - Reporting
  - Recovery

Notification

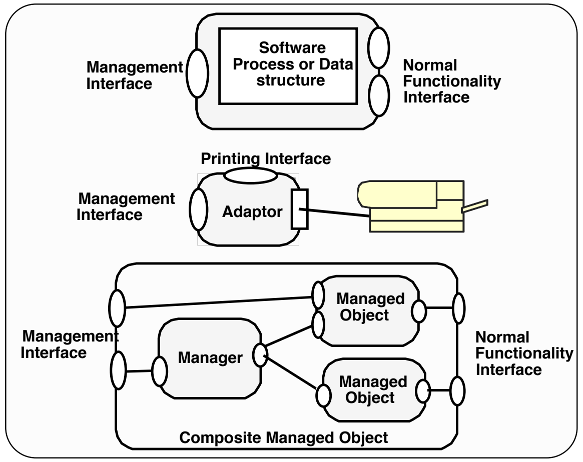
of Change of QOS to client

Accounting & Billing

- Monitoring and recording usage of resources and services by individual users.
- **Service Provider's View:**
  - Needed to provide fair access to shared resources
  - Setting tariffs and billing of users
  - Sub-services – increase complexity
  - Consolidated or itemised billing
  - Monitoring and metering usage → logging → billing
- **User's View:**
  - Provide resource usage statistics
  - Permits users to control usage of allocated resources
  - Registering subscribers for a service
  - Providing service contracts
  - Arranging payment details
  - Providing help service

Managed Objects

Managed Object = an Object with a management interface



Management Interface

Attributes

- Object's state information made visible at interface eg. object id, error counters,
- Single valued, set valued
- Read only or read/write

⇒"Syntactic Sugar" – no need to define simple read and write operations

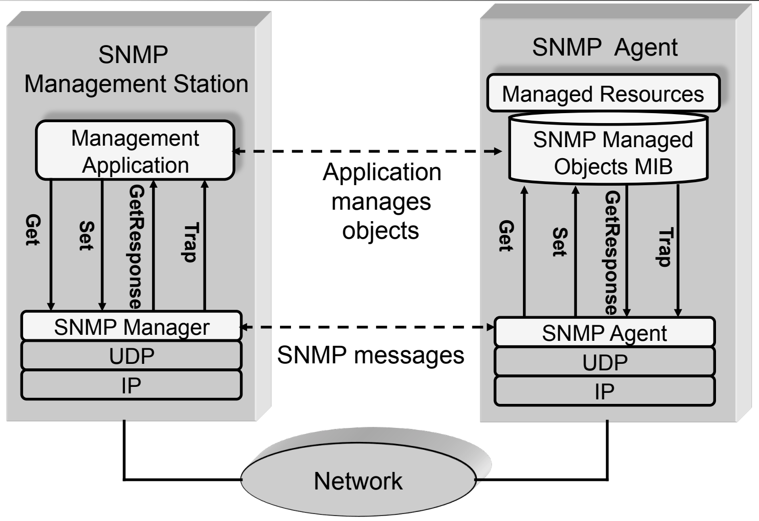
Operations

- Control operation to change state – start/stop, enable/disable etc. Operations involving multiple attributes

Notifications

- Events generated to indicate faults, object creation/deletion etc.

SNMP Model



MIB: Management Information Base

Counter	Value
IPin	30

Name	string
hostname	"skid"

Table

Dest	Link	Next addr	Hops

- Objects defined using ASN.1
- Integers, counters, gauges, string addresses, tables of values
- Not really object-oriented
- MIB consists of set of objects – each with a type and value
- Specific MIB objects for:
  - Protocols eg TCP, IP, Ethernet
  - Systems eg workstation, web server, file server
  - Network devices eg switches, routers, hubs
  - Other devices eg printers

MIB Object Specification

- MIB Breakdown…

- OBJECT-TYPE
  - String that describes the MIB object.
  - Object IDentifier (OID).
- SYNTAX
  - Defines what kind of info is stored in the MIB object.
- ACCESS
  - READ-ONLY, READ-WRITE.
- STATUS
  - State of object in regards the SNMP community.
- DESCRIPTION
  - Reason why the MIB object exists.

<u>Standard MIB Object:</u>

sysUpTime **OBJECT-TYPE**
SYNTAX Time-Ticks
ACCESS read-only
STATUS mandatory
DESCRIPTION
"Time since the network management portion of the system was last re-initialised.
::= {system 3}

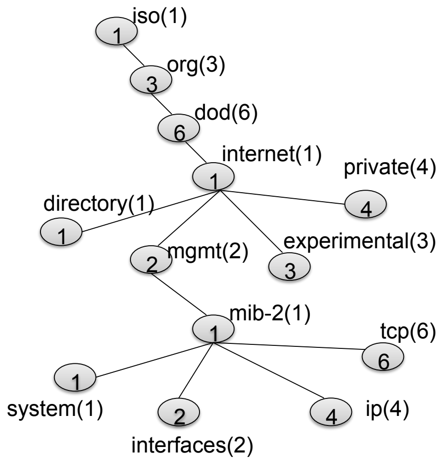
Name used to access object via SNMP

MIB Object Naming

- Object IDentifier (OID)

- Example .1.3.6.1.2.1.1

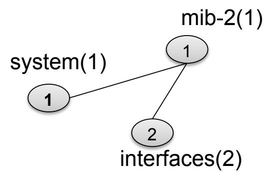
- iso(1) org(3) dod(6) internet(1)
  mgmt(2)
  mib-2(1)
  system(1)



MIB Group

- system(1) group

- Contains objects that describe some basic information relating to an entity.
- An entity can be the agent itself or the network object that the agent is on.



- system(1) group objects

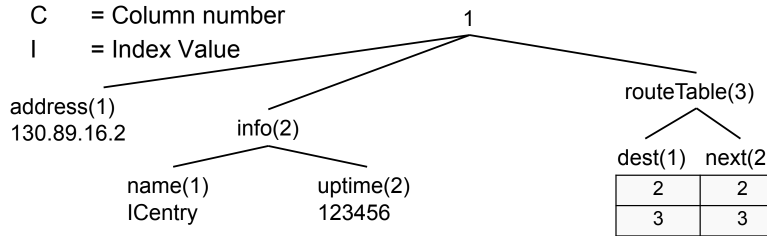
- **sysDescr(1)**	→	Description of the entity.
- **sysObjectID(2)**	→	Vendor defined OID string.
- **sysUpTime(3)**	→	Time since net-mgt was last re-initialised.
- **sysContact(4)**	→	Name of person responsible for the entity.

Naming Table Entries

Use index column accessed by entry value, not row number

X.C.I

X	= OID of Table
C	= Column number
I	= Index Value



dest(1)	next(2)
2	2
3	3
5	2
7	2
8	3
9	3

OID of Table = 1.3
1.3.1.5 => *5*, *1.3.2.5 => 2*
1.3.1.9 => 9, *1.3.2.9 => 3*
1.3.2.7 => 2
1.3.1.1 => entry does not exist
1.3.2.1 => entry does not exist

Table entries can be strings, IP addresses etc.

SNMP Messages

Get	Read a values of one or more named items
GetNext	Read next element from MIB permits manager to traverse MIB without knowing contents
GetBulk	Read large data items = multiple GetNexts
Set	Write values to one or more items
Response	Agent response to manager, after Get, GetNext or Set
Trap	Notification from agent to manager e.g. threshold value exceeded No acknowledgement
Inform	Notification from agent to manager or manager to manager Acknowledged

SNMP

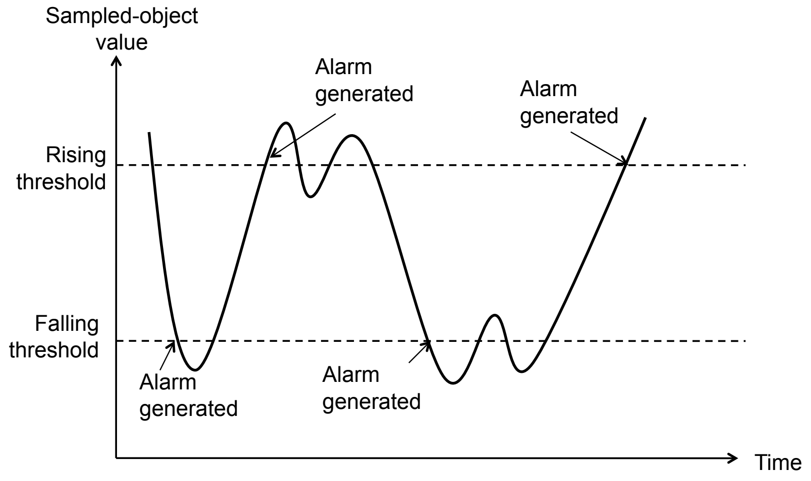
Operations performed by writing and reading values
- Low level protocol based on datagrams (UDP)
- Variable oriented approach c.f. debug or assembly level
- Require information and control variables.
- Very simple and efficient to implement
- No object orientation, very simple data types

- **Computing Industry use SNMP management**

RMON Remote Monitoring MIB

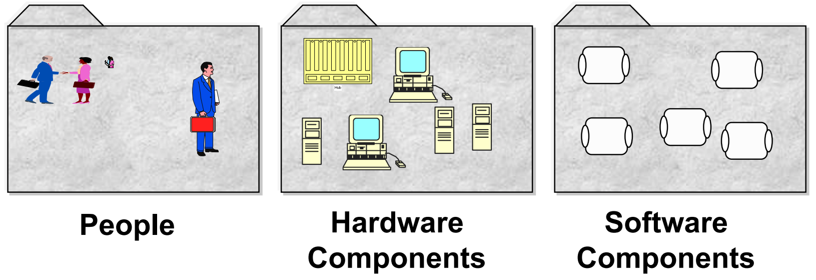
1. statistics: maintains low-level utilization & error statistics for each subnet monitored by agent
2. history: records periodic statistical samples e.g. lost packets
3. alarm: defines sampling intervals & alarm thresholds for counters or integers recorded by RMON agent
4. host: defines counters for various type of traffic to and from hosts attached to a subnet
5. hostTopN: sorted host statistics that top a list based on a parameter in host table e.g. host with top in or out packets
6. matrix: shows error & utilization info in matrix form to enable retrieval of information on any pair of network addresses
7. filter: defines a packet filter for capturing packets or maintaining statistics
8. packet capture: how data is sent to the manager
9. event: table of events generated by RMON agent

Alarm Event Generation

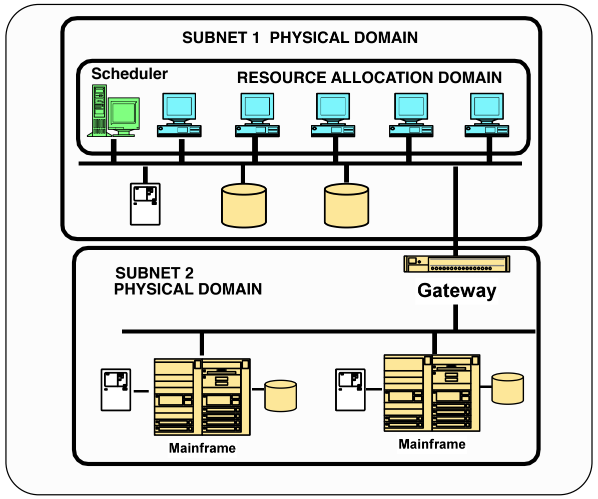


Domains ➔ Grouping

- A **domain** is a collection of objects which have been explicitly grouped together for management purposes e.g. to apply a common policy or partition management responsibility
- Corresponds to Directory

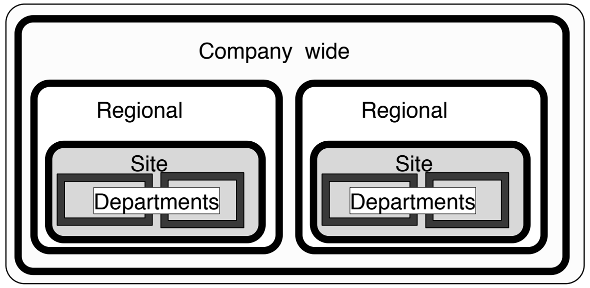


Typical Domains



Typical Domains

ORGANISATIONAL



Common security policy

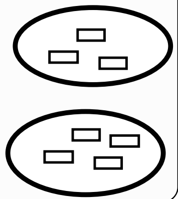
Encryption

- Key distribution list
- Common algorithm

Access Control

- Set of target objects
- Set of users (subjects)

Single security administrator?



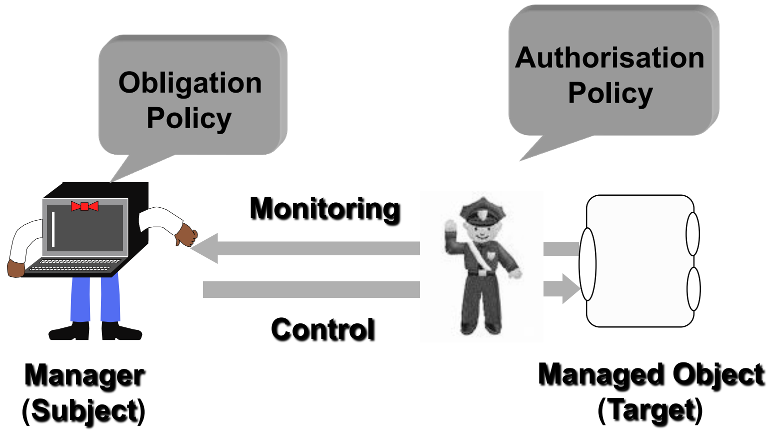
Security Domains

Policy Definition

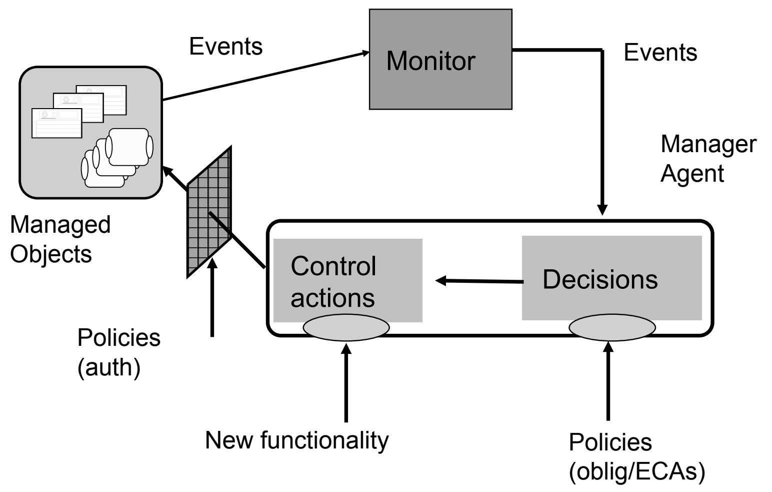
Rule governing choices in behaviour of the system

- ➢ Specify adaptation strategy
- ➢ Derived from enterprise goals and service level agreements
- ➢ Need to specify and modify policies without coding into automated agents
- ➢ Policies are **persistent**
- ➢ But can be dynamically modified
- ➔ Change system behaviour without modifying implementation – **not new functionality**

Management Policy



Policy-based Adaptation



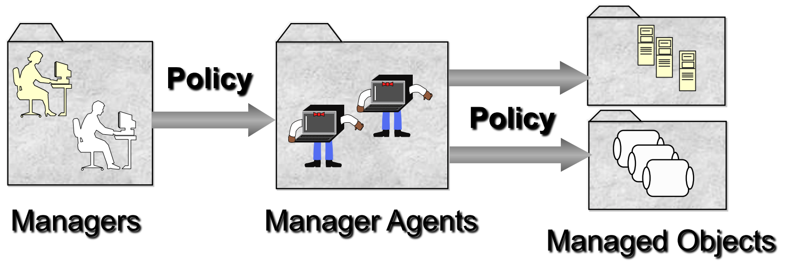
Security Specification

- E-commerce, healthcare – multiple organisations
- Complex security policies with many constraints and exceptions
- Common security policy specification which can map onto heterogeneous implementation mechanisms for OS, firewalls, databases …..
- Need to specify security policy for groups and roles (organisational positions)
- Need to manage security – what actions to perform when a violation detected?

Example Policies

- Who is permitted to access a service, what operations they can they perform, and when. E.g. Research staff can set up video conferences between UK and USA only between 16:00 and 19:00, Monday to Wednesday.
- What resources a mobile user can access when visiting a remote location
- What information transformations and UI adaptations should take place when a user is mobile.
- What actions should be performed when a login violation is detected.
- What diagnostic tests should be performed when an error count is exceeded in a network component.
- Allocate 10% of available bandwidth to voice over IP

Domains and Policies

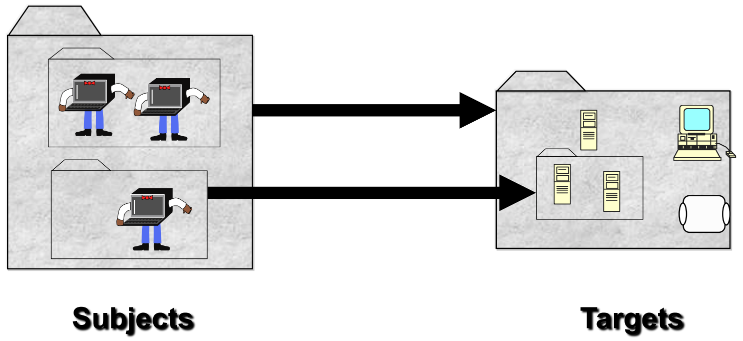


- Impractical to specify policy for individual objects in large systems with many objects

➔ specify policy for domains

- Can change domain membership without changing policy

Policy Propagation



Authorisation Policy

- Defines what a subject is permitted or not permitted (prohibited) to do to a target
  - Permitted operations
- Protect target objects from unauthorised management actions
  - ➔Target based interpretation and enforcement

auth+ /family/kids → /internet/iplayer.start
if time.between (14.00, 19.00)

auth+ /family/adults → /internet/iplayer.start
if time.between (14.00, 24.00)

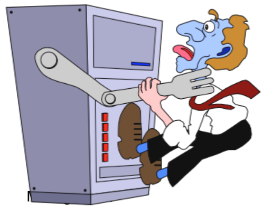
Negative Authorisation

- Used for revocation of access rights

auth**-** /users/JoeBloggs →
/resources/database.any
if time.date > 30:12:2012

- Reflect organisational policies and laws

auth- /doc/staff → /doc/students.strangle



Default Authorisation

- **Default Negative**
  Everything forbidden unless explicitly authorised

auth- /users/sysAdmin → /routers/gateways.{load, enable, disable}
if /users/sysAdmin.location ≠ ComputerRoom;

⇩

auth+ /users/sysAdmin → /routers/gateways.{load, enable, disable}
if /users/sysAdmin.location = ComputerRoom;

- **Default Positive**
  Anything permitted unless explicitly forbidden

Obligation Policy

- Defines what actions a subject must do
- Subject based ➔ subject interprets policy and performs actions on targets
- Event triggered condition-action rule
- Actions can be remote invocations or local scripts

Obligation Examples

- On a TX circuit failure, replace the circuit with a backup, reconfigure the transceiver and log the failure

```
on    failure (cir, trans, switch) do
      trans.disable(cir);
      trans.enable( “backup”);
      log (cir, trans, switch)
```

- On receiving a temperature event, switch off unit if the temperature is > 50

```
on    tempReading (unit, temp) do
      if      temp > 50 then
              unit.off
```

Roles

- Role groups the rights and duties related to a **position** in an organisation
- E.g., network operator, network manager, finance director, ward-nurse
- Specify policy in terms of **roles** rather than **persons**

➔Do not have to re-specify policies when person assigned to new role

- Use domain to represent role & specify policies in terms of role-domain. Policies can be used to control assignment to role (see ubiquitous systems).

Summary

- **Managed Objects**
  - Object with management interface
  - May be composite component & could be self managed
- **Managers**
  - Interpret & enforce obligation policy (not authorisation policy)
  - May themselves be distributed components
- **Domains**
  - Grouping of objects c.f. directories
  - Partitioning responsibility & structuring of management
  - Scope for application of policy
- **Policy**
  - Independent obligation and authorisation policies
  - Policy objects define subjects, targets, activities & constraints
  - Specify policies in terms of roles rather than people.