



On Policy-based Extensible Management Approaches for Providing QoS

George Pavlou

Centre for Communications Systems Research

University of Surrey, UK

IST TEQUILA Project



- ◆ **Management in the context of enterprise and telecom networks**
- ◆ **An introduction to policies**
 - **Work in the Research Community and IETF**
- ◆ **Policy as a means for programmable, extensible management systems**
- ◆ **Hierarchical Policies**
- ◆ **Policies in the context of the TEQUILA system for QoS management in the context of IP Diff-Serv**
- ◆ **Current state and future work**



Enterprise Networks

- ◆ Typically single centralised “Network Management Centre”(NMC)
- ◆ NMC monitors the managed devices, only rudimentary management intelligence is typically supported
- ◆ Re-configuration is infrequent and driven by the human manager according to predefined “policy”
 - Central human-controlled configuration ensures no conflicts
- ◆ Ok for best-effort IP enterprise networks but not suitable for multiservice networks with increased management needs
 - The latter required frequent re-configuration for traffic engineering, driven from both network state and customer demands (SLSs etc.)



Telecommunication Networks

- ◆ Managed according to the hierarchical TMN model
- ◆ (Re-)Configuration of NEs takes place through Element Managers (EMs) driven by an overall configuration Network Manager (NM)
- ◆ NM applications implement network-wide policy through automated logic
- ◆ Configuration Manager holds the physical and logical network topology
- ◆ Requests from other applications are validated, conflicting configuration requirements are possible but should be avoided through rigorous testing
- ◆ Management intelligence is static, “hard-wired”: system is engineered according to requirements at time T_1 and cannot be easily modified / extended after that



Policy-Based Management

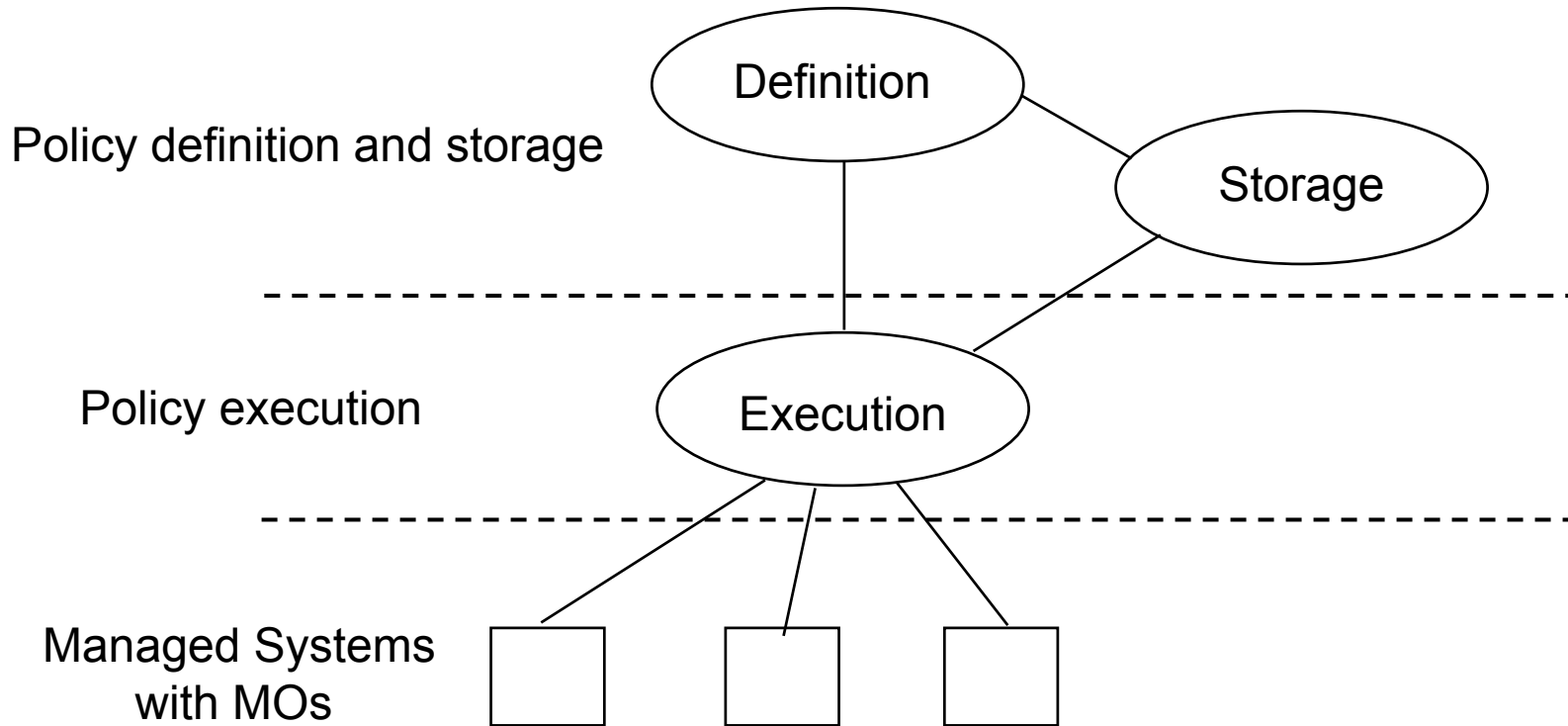
- ◆ **Management logic in Policy-based systems is expressed through declarative policies**
- ◆ **These are transformed in an interpreted fashion to low level actions to managed objects**
- ◆ **Key aspect: management intelligence can be modified, added and removed by manipulating policies**
- ◆ **Results in a highly dynamic system so conflicts are the norm rather than the exception**
- ◆ **Policies can be seen as a means to “late bind” functionality to an existing system**
- ◆ **Key aspect: where to draw the line between static, “hard-wired” management logic and flexible logic realised through policies**



A Policy Conceptual Model

- ◆ **A three-tier architecture**
 - **A policy definition and storage layer**
 - Policies are first defined, checked for consistency, conflicts etc. and then stored
 - **A policy execution layer where decisions are made and configuration actions are taken**
 - Policies are the input from above and network information (events) the input from below
 - **Policies manipulate managed objects in managed elements or in management applications**
- ◆ **A dynamic, fluid model compared to the conventional fixed, static models**

A Policy Conceptual Model (cont'd)





Policies in the Research Community

- ◆ **Distributed models**
- ◆ **Hybrid agent-manager applications with interpreted policy logic realising their management “intelligence”**
- ◆ **Policies: objects defining the relationships between subjects (managers) and targets (managed objects)**
- ◆ **Generic Classification of Policies**
- ◆ **Specification of Policy Notations and Languages**
- ◆ **Issues of policy refinement, conflict detection and resolution**
- ◆ **No convincing real-world applicability yet**



Policies in IETF

- ◆ **IETF sees policies as a means for service management in emerging QoS IP networks**
 - “Intelligent” IP network, not just bit-pushing...
- ◆ **Centralised Model**
- ◆ **Policy Consumers or Policy Decision Points (PDPs) enforce policies by manipulating MOs within Network Elements (NEs) or Policy Enforcement Points (PEPs)**
- ◆ **Policies are seen as the means for flexible service-driven network management**
- ◆ **Policy is defined as an aggregation of Policy Rules**
 - Policy Rule: if <condition> then <action>
- ◆ **Development of O-O information model for representing policies**



Problems with IETF Policies

- ◆ Relatively low-level policies targeted mostly to consistent configuration across a number of devices
- ◆ No policy language yet
- ◆ Policies cannot be triggered from network stimuli
 - Cannot support reactive fault and performance management functionality at least
- ◆ Centralised policy decision point for every different type of policy in a network e.g. QoS, security, etc.
 - Scalability?
 - Conflict detection and resolution? Possibly definition time but what about run-time conflicts
- ◆ Inter-domain policy?
- ◆ In summary, work to date has concentrated on protocols and interoperability given IETF's preoccupation
- ◆ Still a long way to go

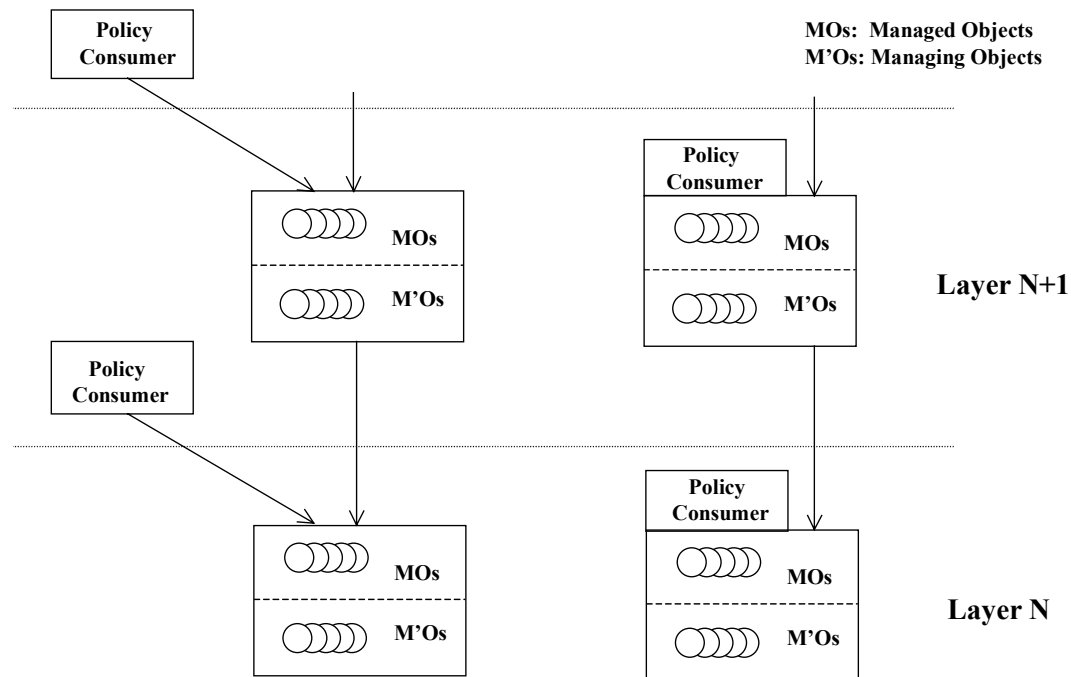


Our View of Policies

- ◆ Try to bring together the IETF model, previous work in the research community + our ideas
- ◆ We see policies as a means for extensible systems that will have to co-exist with static, predefined management logic
 - Impossible to do *everything* with policies
- ◆ We feel the key advantage is the declarative high-level nature and potentially automated transformation
 - IETF focuses on relatively low-level procedural policies
- ◆ Since management systems are typically hierarchical, policies could mirror this hierarchy resulting in hierarchical policies when and where appropriate

Hierarchical Policies

- ◆ Policies may be considered as part of the managing intelligence of layer N+1 (left)
- ◆ Given their interpreted nature, they could execute at the agent-manager having local access to MOs (right)



- ◆ **Policy transformation and refinement**
 - High-level Policies may result in the introduction of related policies at lower layers
 - Guidelines may be devised and followed in the context of a particular type hierarchical system that will assist and possibly automate the process of refinement
- ◆ **We view policy refinement as a key issue for policy-based management**
 - Otherwise, procedural low-level policies are no more than scripts that are checked for potential conflicts
- ◆ **Very difficult problem in general but can be potentially solved in the context of a particular problem area**
 - We hope to do something for dimensioning and resource management of Diff-Serv 😊

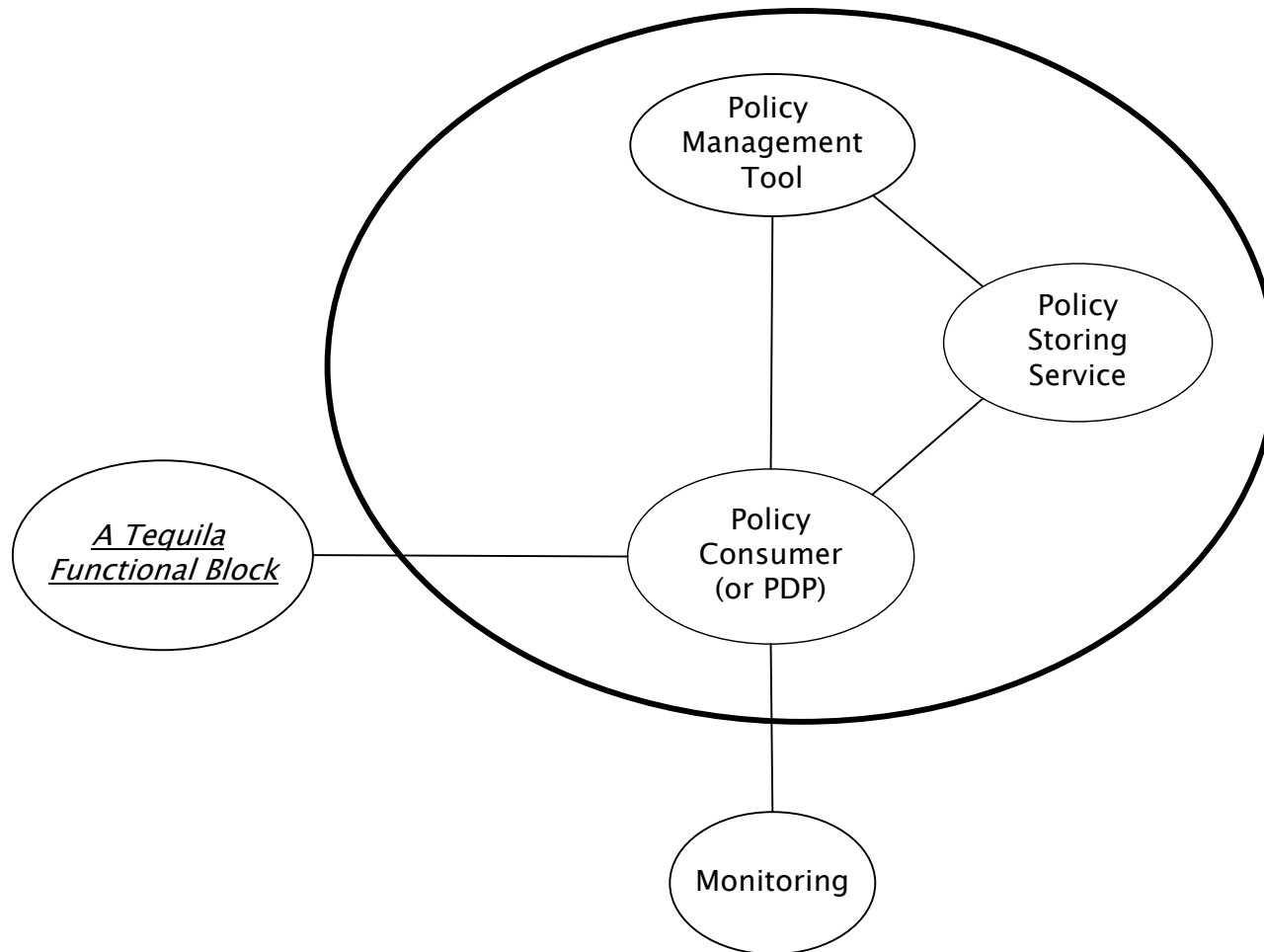


The TEQUILA Project

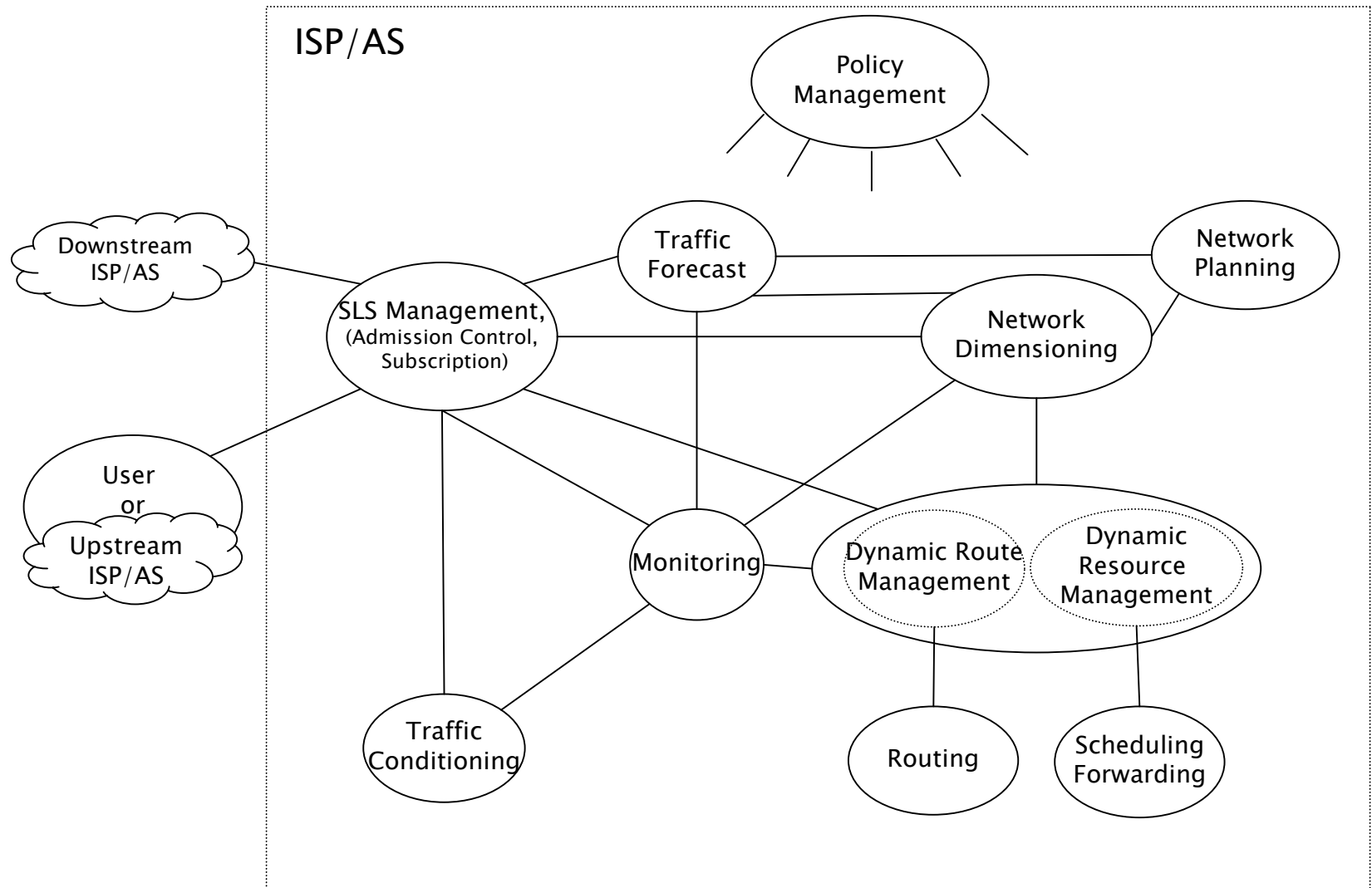
- ◆ **The TEQUILA project (Traffic Engineering for QUality of service in the Internet at LArge scale) is used as the vehicle to explore policy aspects**
- ◆ **TEQUILA has designed an integrated management and control architecture to support end-to-end QoS in IP Diff-Serv networks**
 - **Planning & Dimensioning, SLS Management, Traffic Forecasting, Dynamic Route and Resource Management, Monitoring**
- ◆ **Policy management applies most components of the TEQUILA architecture**
 - **SLS Management policies, Dimensioning policies, Dynamic Route and Resource Management Policies**
 - **Dimensioning and Resource Management policies may have a hierarchical relationship**
- ◆ **Many policy consumers or PDPs with a single policy storing service (for static conflict detection) and a policy management tool for the network operator**

Policy Management in TEQUILA

Policy Management

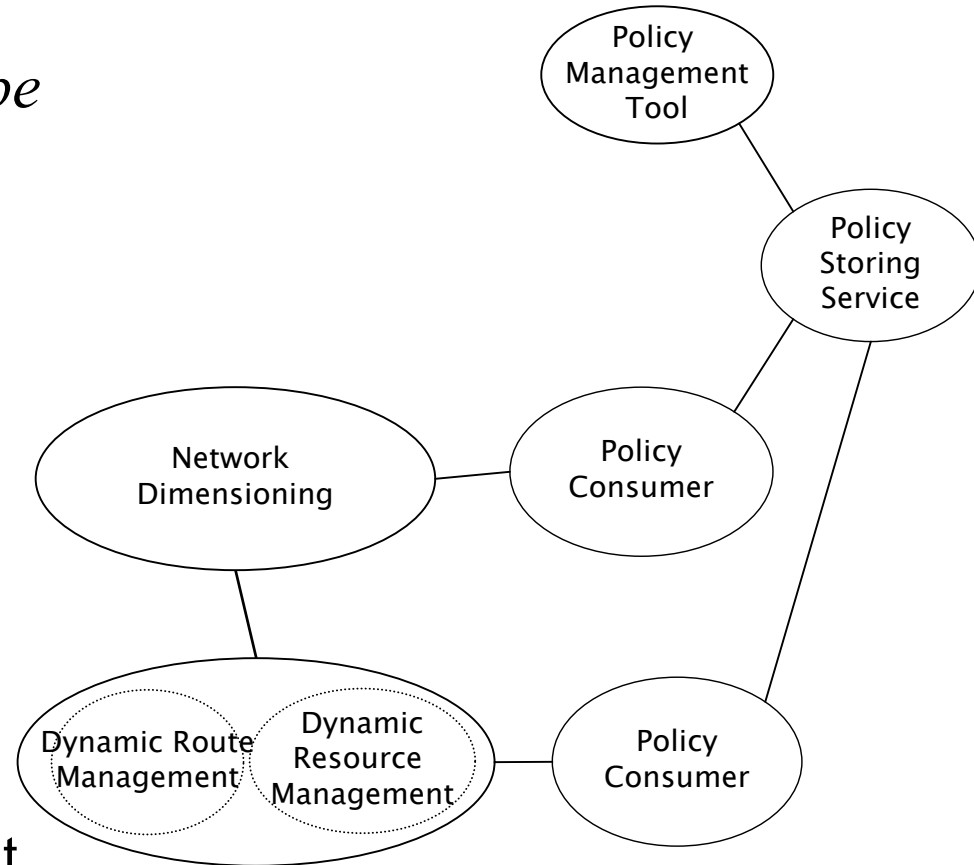


The TEQUILA functional architecture



An Example of Hierarchical Policy

- ◆ *“At least 10% of Network Resources should always be available for EF traffic”*
- ◆ Template of the generic policy class:
<bound><percentage> of Network Resources <period> available for <traffic type>
- ◆ Decomposed into:
 - a dimensioning policy
 - a dynamic resource mgmt policy





Summary and Future Work

- ◆ **Description of the characteristics of policy-based management and its coexistence with static, hierarchical management systems**
- ◆ **Fundamental target is a management system able to sustain requirement changes and evolve gracefully through policies without changing its initial “hard-wired” logic**
- ◆ **Current and future work:**
 - **Definition of O-O Information model representing the capabilities of each layer of the TEQUILA architecture**
 - **Specification of dimensioning and dynamic resource & route management policy classes**
 - **Explore the concept of decomposition and refinement**