

# Policy Definition and Classification: Aspects, Criteria, and Examples

René Wies

Munich Network Management Team

University of Munich, Department of Computer Science

Leopoldstr. 11b, 80802 Munich, Germany

Phone: +49-89-2180-3139

Email: wies@informatik.uni-muenchen.de

## Abstract

To analyse the concept of policies and its use, it is necessary to first clarify the meaning of the notion of policy and derive criteria for policy classification. In this paper we discuss different definitions of policies and show their strengths and weaknesses. We then motivate the classification of policies and provide a list of criteria for a classification. A policy hierarchy is introduced and described. The classification criteria are then used in conjunction with the policy hierarchy to investigate three policies from the area of network and systems management. At the end, conclusions and future work are outlined.

**Keywords:** Policy, Policy Definition, Policy Classification, Policy Hierarchy

## 1 Introduction: Differences in Policy Definitions

As the task of network management started to become more complex and as heterogeneous systems and distributed applications needed management, two notions made their way into the literature — **domain** and **policy**. They were introduced to reduce the complexity of management tasks but their interpretations vary between and within the fields of research, standardization, and industry.

Following the common idea to model network devices, systems, and applications in an object-oriented fashion as Managed Objects (MOs), the ISO in its drafts on standardizing domains and policies ([ISO 10164-19, ISO 10040/2]) places MOs in a logical domain provided they satisfy a certain policy. Thus a policy is merely a number of rules tied in to a domain managed object. A very similar approach can be found in the ODP draft standards ([ISO 10746-1]). Products such as MaestroVision ([MAES 93]) or HP's dolphin ([PGMM 93]) follow the same line of thought where policy "objects" are defined by means of simple rules. Domains are just sets of objects that satisfy a specific policy. However, it is not enough to display a policy object or a domain of objects on a fancy graphical user interface. At present, the management purposes of domains and policies such as "management by delegation" ([YGYE 91]) are almost neglected. There is no (de-facto) standard on how to define, use or manipulate policy and domain objects

which leads to the problem that management applications cannot be written based on these two powerful management concepts.

In the field of research a different approach is taken. The definition of policy is independent from that of a domain and hence a policy may be either applied to or used to define a domain. [SLOM 93] and [MOFF 94] define policy as to *influence* the behavior of a *manager* or *managed objects*. The definition we will use is that policies are derived from management goals and *define* the desired behavior of *distributed heterogeneous systems, applications, and networks* ([WIES 94]). The difference between the above two definitions is simply the level of abstraction. Applying policies to MOs is already at a technical level, whereas specifying the behavior of the overall environment is at a higher level of abstraction and more appropriate for enterprise management tasks. Concerning the definition of policy services, [IDSM 93] specifies some services which could be used in writing management applications using the concept of policies.

To summarize, the advantages of the research-definition are as follows:

- policy and domain are two independent concepts which may but need not be combined;
- policies may be used to define a domain but may also be applied to a domain of objects; and
- policies are an active concept. Policies can initiate or change the characteristics of ongoing management activities.

After this brief introduction to the different definitions of policy we will now motivate the classification of policies in Section 2 and present criteria for their classification and some examples of policies with their respective classifications.

## 2 Aspects of Policy Classification

### 2.1 Motivation

As interest grows and research in the field of integrated network and systems management progresses, it becomes increasingly important to clarify what we mean by management policies. A first step towards this was the definition of the term policy. However, this only allows to distinguish whether a statement is considered to be a policy or not ([MOSL 91]).

The vast number of policies, some examples of which we will present later, calls for a classification i.e. a well-defined set of (orthogonal) grouping criteria. The goals of such a classification of policies are listed below:

1. to find commonalities of and similarities between different classes of policies;
2. to derive and verify the components of a formal definition of policies;
3. to derive a policy hierarchy for the process of policy refinement and transformation;
4. to allow a (semi-) automatic processing (enforcement and monitoring) of policies; and
5. to guarantee an efficient and effective management of policies.

The manner in which a policy is applied can be very different depending on the class of policy, i.e. its characteristics. For example, the classification aspect *life-time of policy* can be used to distinguish between short, medium, and long-term policies for which the realization may range from a polling strategy to check the enforcement of a short-term policy, to a more complex configuration of asynchronous notifications (traps) in network probes and proxy systems for long-term policies. Thus, our goal is to define classification criteria which influence or even determine the way in which certain classes of policies are defined, activated, enforced, monitored, changed, deactivated, and managed or the way in which policy conflicts are resolved. This then leads to the definition of policy services, in other words services used by management applications to employ the concept of policies. Thus, a precise and detailed classification is a prerequisite in the process of deriving parameters for policy templates, policy processing, and their application.

## 2.2 Criteria for the Classification of Policies

Extensive analysis of policy catalogues from numerous network and system providers (such as the FidoNet, VirNet, etc.) and talks with network and system managers, administrators, and operators (LRZ, BMW, debis) have allowed us to collect the following list of criteria for the classification of policies which are illustrated in Figure 1 in form of a multi-dimensional diagram. The precise labels of the axes, i.e. the different categories for each criterion inevitably depend on the type of target objects the policy deals with and the functional area to which the policy can be assigned. Thus, the criteria are not always perfectly orthogonal. This aspect will be discussed further in Section 2.3.

**time:** Time considerations are important because a policy for example may be active throughout the lifecycle of its target objects or may only be activated for a short while, e.g. at the start when a new network device goes into operation (configuration policy). Policy enforcement instructions may need to be dispatched repeatedly, e.g. whenever new devices are added to the configuration. Further refinement of this criterion are

**life time:** The duration of a policy may be characterized by a short, medium, or long-term application (i.e. enforcement and monitoring). A more detailed consideration of this idea can be based on separating policy enforcement and policy monitoring ([WIES 94]), both of which may be short, medium, or long-term activities. Short-term policy application in this context may be a once-only management policy enforcement (e.g. which userinterface to install on a PC), medium-term for example a policy applicable until a new software release is installed, and long-term for example once-and-for-all policies applicable to a compute-server independent from its replacement by a different vendor's system. As stated above, these categories must be put in concrete terms depending on the type of target object(s) effected by the policy.

**trigger mode:** The question here is whether the enforcement and especially the monitoring are constantly active, repeated periodically for a specific time interval, triggered by asynchronous events or a combination of the last two. Another aspect could be a policy's relationship to other policies. Examples of this are: no relationship, sequential execution, simultaneous execution, etc.

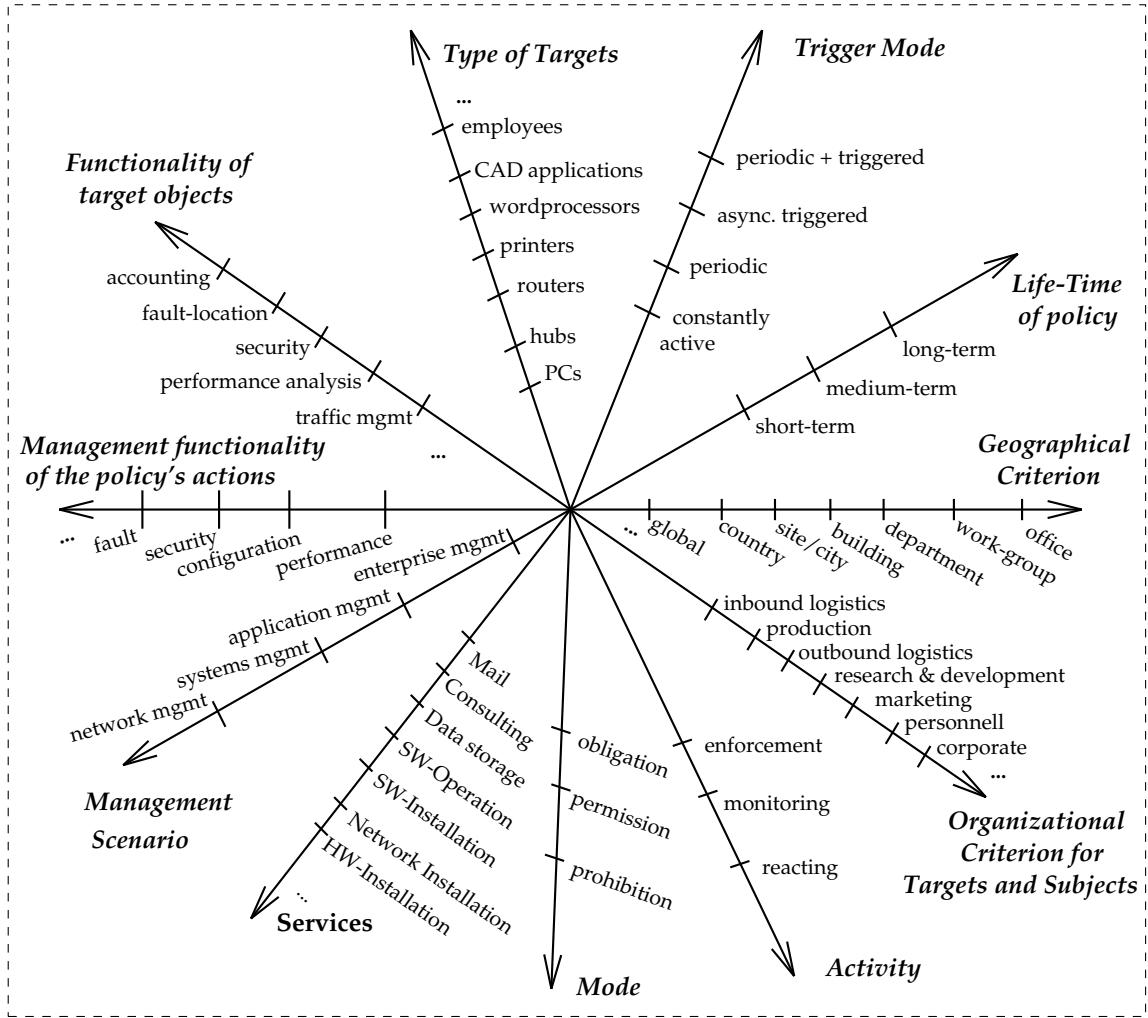


Figure 1: Criteria for Policy Classification

**activity:** A policy may be only monitoring or enforcing actions on its target objects or reacting to an event. The monitoring policy will only report its observations but never take on actions whereas enforcing and reacting policies can initiate management activities (actions and reactions).

**mode:** Policies can be a constraint or an empowerment. We will use the distinction between an obligation, a permission, and a prohibition.

**geographical criterion:** With this criterion, policies are grouped according to their location, i.e. affecting co-located physical and logical resources along geographical boundaries. Examples are policies for systems within a LAN segment, or a policy for all systems in a virtual private network. Typical aspects for a grouping of systems and resources are also: office, work-group, department, building, site/city, country, global.

**organizational criterion:** This grouping of policies reflects the organizational structure of the environment, e.g. policies for specific business units of an enterprise or policies which only

need to be obeyed in high security departments. Other categories such as inbound logistics, operations, outbound logistics, marketing and sales, service, procurement, research and development, or corporate can be derived from the value chain ([POMI 85]) of an organization. The category 'corporate' would qualify a policy for the overall corporation.

**service criterion:** Policies are often specific to certain services an organization offers its customers, buys from service providers, or provides for internal use. Thus, the service for which a policy is specified can help to identify a certain set of management tools to be used or the resources to be taken action on in order to enforce the policy. Services are data storage, email, network information services or software installation to name just a few.

**type of targets:** This criterion could include policies applicable to all endsystems from one vendor, or all PCs in one department, i.e. target objects with common characteristics. Familiar categories here are resources such as workstations, PCs, hubs, router, laserprinters, wordprocessing applications, CAD applications, employees, etc.

**functionality of targets:** This includes all policies which apply to resources with a set of common functionalities although possibly of different characteristics otherwise. Targets with common characteristic functionalities could be all network devices capable of routing, all systems (PCs, printers, hubs, etc.) whose user-interface is protected by a password mechanism. In other words, functional characteristics such as accounting, fault-location, security, traffic management, performance analysis etc.

**management scenario:** Policies may be associated with a particular management scenario such as network management, systems management, or application management. Some policies may overlap and should thus be grouped together as enterprise management policies. For the scenario of network management, a group of policies may be those applicable to one specific layer of the OSI basic reference model ([ISO 7498]) or policies applicable to several adjacent layers. The next criterion is based on the management scenario:

**management functionality within a management scenario:** Within each of the above scenarios, we can distinguish different functional areas, e.g. configuration management within systems management, or configuration management within network management, or security management for enterprise management.

As mentioned earlier, the above criteria can be used to derive components (attributes) of a policy template. However, the values of these attributes depend on the policy's level of abstraction i.e. its position in the policy hierarchy.

## 2.3 Policy Hierarchy and Transformation Process

Before presenting a few examples with their classification, the important issue of a policy hierarchy or better the level of abstraction must be raised. When analyzing policies, we must differentiate between (see also Figure 2 and [MACA 93]):

- Corporate policies or high level policies: These are directly derived from corporate goals and thus embody aspects of strategic business management rather than aspects of technology oriented management. To allow their application within the management environment, they have to be refined to one of the three policy types below.

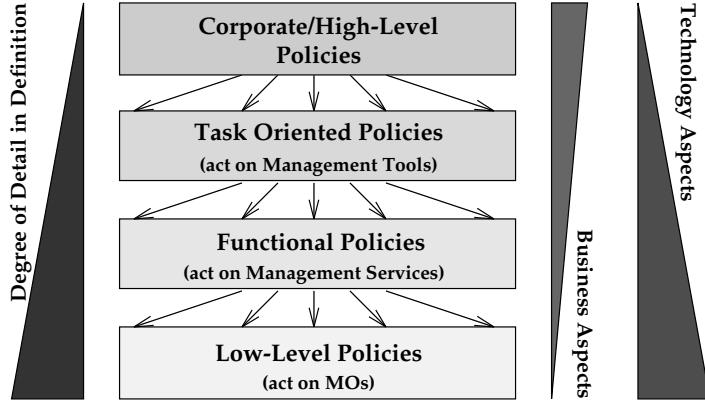


Figure 2: The Policy Hierarchy

- Task oriented policies: Their field of action is sometimes referred to as task or process management, where they define the way how management tools are to be applied and used to achieve the desired behavior of the resources.
- Functional policies: These policies operate at the level of and define the usage of management functions, such as the OSI *systems management functions* ([ISO 10164-X]), the OSF/DME *distributed services* ([DME 92]), or OMG's *object services* ([OMG 92a, OMG 92b]); and
- Low level policies: They operate at the level of managed objects (MOs). MOs in this context refer to simple abstractions of managed network and system resources, and not MOs for e.g. systems management functions.

Thus we find "simple policies" with MOs as their target objects, policies that operate on more complex managed objects such as SMFs, and policies that act on task objects (e.g. abstractions of management tools). We will not consider corporate policies any further in this paper but believe that the other three types of policies can be derived from these.

A policy definition (or a policy object) will have the same components at each level in the hierarchy, but the possible values for each component / attribute will depend on the level of abstraction. In other words, the lower the level of abstraction, the more precise and detailed will the definition become, i.e. the granularity of the criteria increases.

The transformation process of a policy definition is a process of stepwise refinement, moving from high-level policy definitions down to low-level (MO-based) policy definitions which can be more easily automated and applied to the managed environment. The question to whether this transformation process can be automated cannot be answered at this stage. However, for any automation of this process (fully computerized or human guided), management information on the managed environment and the management capabilities of the involved systems is necessary.

Some policies can be assigned to exactly one level of the hierarchy, yet other (less well defined) policies can be assigned to different levels of abstraction and thus need to be split into separate policies before the transformation process can be applied. Examples, including one containing more than one level of abstraction, are presented in the next section.

### 3 Examples

All of the following examples are taken from real life situations from different corporations where the operators applied these policies without a systematic and structured approach. Thus, the values for each classification criterion were derived manually, since none of these policies were systematically refined. For each example, the level of abstraction is given and possible values for each of the above classification criteria are indicated. Examples 1 and 2 are used to show the components of a policy definition, whereas example 3 illustrates the splitting of a “composite” policy into separate policies after which the transformation and refinement process can be applied.

#### Example 1:

“The exchange of data between the company’s headquarters and its remote production sites is to be done between 18:00 and 22:00 hours in encrypted mode.”

The degree of detail in this policy is very limited and thus, we can only record it as a high level policy of the following format with several dimensions to be further specified.:

- **level of abstraction:** high level policy

- **classification criteria:**

*life time*: long-term (no end specified)

*trigger mode*: periodic (daily between 18:00 and 22:00 hours)

*activity*: enforcement (no reaction is specified if the time intervall or the security level are not obeyed – a separate policy for this purpose would be necessary)

*mode*: obligation

*geographical criterion*: corporate headquarters and production sites

*organizational criterion*: unspecified

*service criterion*: unspecified

*type of targets*: unspecified

*functionality of targets*: unspecified

*management scenario*: enterprise management

*management functionality within a management scenario*: security management for enterprise management

Analyzing and refining this policy further leads to a number of low level policies, depending on the way the encryption is achieved. The following two policy descriptions illustrate this, the first enforcing the encryption by activating either encryption modems or scramblers, the second by activating the encryption mode for data transfer in the application software. This also shows, that a policy can be applied in several different ways without changing the management goal.

- **level of abstraction:** low level policy

This is because the policy applies to MOs which, in this case, are abstractions of network devices, i.e. modems or scramblers.

- **classification criteria:**

*life time*: long-term (no end specified)

*trigger mode*: periodic (daily between 18:00 and 22:00 hours)

*activity*: enforcement  
*mode*: obligation  
*geographical criterion*: corporate headquarters and production sites  
*organizational criterion*: networking department  
*service criterion*: data transfer service  
***type of targets*: encrypting modems or scramblers**  
*functionality of targets*: data transfer or encryption  
*management scenario*: network management  
*management functionality within a management scenario*: security management for network management

- **level of abstraction:** low level policy

This is because the policy applies to MOs which, in this case, are abstractions of the application software based on a client-server architecture e.g. distributed CAD or word-processing applications.

- **classification criteria:**

*life time*: long-term (no end specified)  
*trigger mode*: periodic (daily between 18:00 and 22:00 hours)  
*activity*: enforcement  
*mode*: obligation  
*geographical criterion*: corporate headquarters and production sites  
*organizational criterion*: systems department  
*service criterion*: application software installation and software maintenance  
***type of targets*: general distributed applications based on a client-server architecture, which therefor transfer data across the network.**  
*functionality of targets*: **applications with encryption**  
*management scenario*: application management  
*management functionality within a management scenario*: security management for systems and application management

Looking back at the policy hierarchy introduced in Section 2.3, it can be noted that the above policy was refined to different low-level MO-based policies, without specifying task oriented policies nor functional policies. This is because there were no management tools or management functions which could have been used to enforce this policy at a higher level. However, if these had been available, a task oriented policy could have specified the way to use a management tool for the configuration of modems or scramblers, or a functional policy could have defined the manner in which to use a certain encryption management function.

### Example 2:

”If workstation access is protected by a password mechanism, passwords must be at least 6 characters long, if they combine upper-case and lower-case letters, or at least 8 characters long, if in monocase. No other password structure is allowed.”

- **level of abstraction:** managed-object based policy

This is a low-level or managed-object based policy, as it specifies the characteristics of the specific password mechanism, i.e. a specific implementation of e.g. an authentication

management function. Provided a Managed Object for the password mechanism exists, the policy can already be used to set the attributes' values. It is not a functional policy, because the attributes and not the functionality of the password mechanism is effected by the policy.

- **classification criteria:**

*life time*: long-term

*trigger mode*: asynchronously triggered (e.g. by execution of the UNIX-command *passwd*)

*activity*: monitoring, reacting (to a wrong password structure), and enforcing (setting the password mechanism's characteristics)

*mode*: obligation

*geographical criterion*: global

*organizational criterion*: corporate

*service criterion*: data processing (authentication)

*type of targets*: workstations

*functionality of targets*: authentication/password mechanisms

*management scenario*: systems management

*management functionality within a management scenario*: security management within systems management

### **Example 3:**

"Travel agencies are to be connected to the central booking office through leased lines. In case of failure, dial-in lines are to be provided, and the agencies must authenticate themselves with their login-IDs and login-keys."

This policy obviously mixes aspects of several levels of abstraction, the level of corporate policies, the level of functional policies, and the level of MO-based policies. The policy should be split into separate policies of specific levels of abstraction e.g.: (3a, corporate) "the network operations center at the central booking office is to provide and maintain leased lines to the agencies, and modems for dial-in connections", (3b, functional) "in case of failure of a leased line, modems are to be activated for dial-in connections", (3c, functional) "dial-in connections are to be protected by an authentication procedure." and (3d, MO-based) "the authentication mechanism MO must guarantee the use of non-empty login-ids and login-keys".

For the sake of brevity we will not discuss the classification of these policies here further. Yet, these examples clearly show that this classification allows us to find commonalities among policies and that this form of classification is a necessary first step towards finding the components of a formal policy definition. The transformation process will only be able to refine some components/attributes further, depending on the management information available to the process.

## **4 Conclusions and Future Work**

With the above introduced classification, we can now systematically derive a specification of policy templates with all necessary components e.g. attributes, actions, notifications, packages,

etc. which will aid in the generation of a (semi-) automatic policy transformation and application process.

Policies are a powerful concept for the management of distributed heterogeneous systems, networks and applications. In this paper we did not present finished work, but rather ongoing research. While our definition, concepts and classification criteria appear to be powerful as well as natural, in our next steps we will prove that they are also practical.

Our research is in a state where the conceptual issues are almost finished, or have at least reached a state where the engineering alternatives are being evaluated by means of implementation studies and tests. The next steps are the formalization and implementation of some selected policies in the field of systems management. This work will be SNMP-based using a refined version of the host resources MIB [KRUP 93] customized for the Leibniz-Supercomputing-Center.

## Acknowledgements

The author wishes to thank the members of the Munich Network Management Team and Thomas Berthold for fruitful discussions and valuable comments to the preliminary version of this paper. The MNM Team directed by Prof. Dr. Heinz-Gerd Hegering is a group of researchers of the University of Munich, the Technical University of Munich, and the Leibniz Supercomputing Center of the Bavarian Academy of Sciences.

## References

- [DME 92] Open Software Foundation, *OSF Distributed Management Environment (DME) Architecture*, 1992.
- [DSOM 93] IFIP, *Proceedings of the IFIP/IEEE International Workshop on Distributed Systems: Operations & Management*, October 1993.
- [IDSM 93] “Domain and Policy Service Specification”, IDSM Deliverable D6 / SysMan Deliverable MA2V2, IDSM Project (ESPRIT III EP 6311) and SysMan Project (ESPRIT III EP 7026), October 1993.
- [INM-II 91] I. Krishnan and W. Zimmer, editors, *Proceedings of the 2nd International Symposium on Integrated Network Management*, Washington, IFIP, North-Holland, April 1991.
- [INM-III 93] Heinz-Gerd Hegering and Yechiam Yemini, editors, *Proceedings of the 3rd International Symposium on Integrated Network Management*, San Francisco, IFIP, North-Holland, April 1993.
- [ISO 10040/2] “Information Technology – Open Systems Interconnection – Systems Management Overview – Amendment 2: Management Domains Architecture”, PDAM 10040/2, ISO/IEC, November 1993.
- [ISO 10164-19] “Information Technology – Open Systems Interconnection – Systems Management – Part 19: Management Domain and Management Policy Management Function”, CD 10164-19, ISO/IEC, January 1994.

- [ISO 10164-X] “Information Technology – Open Systems Interconnection – Systems Management – Management Functions”, IS 10164-X, ISO/IEC.
- [ISO 10746-1] “Basic Reference Model of Open Distributed Processing – Part 1: Overview and Guide to Use”, WD 10746-1, ISO/IEC, November 1993.
- [ISO 7498] “Information Processing Systems – Open Systems Interconnection – Basic Reference Model”, IS 7498, ISO/IEC, 1984.
- [IWSM-I 93] Wesley W. Chu and Allan Finkel, editors, *Proceedings of the IEEE First International Workshop On Systems Management, Los Angeles*, IEEE, April 1993.
- [KRUP 93] B. Krupczak, “UNIX Systems Management via SNMP”, In [INM-III 93], pages 289–299.
- [MACA 93] M. Masullo and S. Calo, “Policy Management: An Architecture and Approach”, In [IWSM-I 93].
- [MAES 93] Calypso Software Systems, “MaestroVision 2.0 beta 1”, Release Notes, Calypso Software Systems, Inc., 1993.
- [MOFF 94] Jonathan D. Moffett, *Specification of Management Policies and Discretionary Access Control*, chapter 17, pages 455–481, In [SLOM 94], June 1994.
- [MOSL 91] Jonathan D. Moffett and Morris S. Sloman, “The Representation of Policies as System Objects”, In *Conference on Organizational Computing Systems*, volume 12 of *COCS’91, Atlanta, SIGOIS Bulletin*, pages 171–184, November 1991.
- [OMG 92a] “Object Management Architecture Guide”, Document 92-11-1, Object Management Group, September 1992.
- [OMG 92b] “Object Services Architecture”, Document 92-8-4, Object Management Group, August 1992.
- [PGMM 93] Adrian Pell, Chen Goh, Paul Mellor, Jean-Jacques Moreau and Simon Towers, “Data + Understanding = Management”, In [IWSM-I 93].
- [POMI 85] Michael Porter and Victor Millar, “How information gives you competitive advantage”, *Harvard Business Review*, 63(4):149–160, 1985.
- [SLOM 93] Morris Sloman, “Specifying Policy for Management of Distributed Systems”, In [DSOM 93].
- [SLOM 94] Morris Sloman, *Network and Distributed Systems Management*, Addison-Wesley, June 1994.

- [WIES 94] René Wies, “Policies in Network and Systems Management – Formal Definition and Architecture –”, In Manu Malek, editor, *Journal of Network and Systems Management*, volume 2, pages 63 – 83, Plenum Publishing Corporation, March 1994.
- [YGYE 91] Yechiam Yemini, German Goldszmidt and Shaula Yemini, “Network Management by Delegation”, In [INM-II 91], pages 95–107.