# Monitoring the Connectivity of a Grid

Sergio Andreozzi
Istituto Nazionale di Fisica Nucleare-CNAF
Viale Berti Pichat, 6/2
40126 Bologna (Italy)
andreozzi@cnaf.infn.it

Augusto Ciuffoletti
Department of Informatics, University of Pisa
Via F. Buonarroti, 2
56127 Pisa (Italy)
augusto@di.unipi.it

Antonia Ghiselli
Istituto Nazionale di Fisica Nucleare-CNAF
Viale Berti Pichat, 6/2
40126 Bologna (Italy)
ghiselli@cnaf.infn.it

## ABSTRACT

Grid computing is a new paradigm that enables the distributed coordination of resources and services which are geographically dispersed, span multiple trust domains and are heterogeneous. Network infrastructure monitoring, while vital for activities such as service selection, exhibits inherent scalability problems: in principle, in a Grid composed of $n$ resources, we need to keep record of $n^2$ end-to-end paths. We introduce an approach to network monitoring that takes into account scalability: a Grid is partitioned into domains, and network monitoring is limited to the measurement of domain-to-domain connectivity. However, partitions must be consistent with network performance, since we expect that an observed network performance between domains is representative of the performance between the Grid Services included into domains.

## Categories and Subject Descriptors

D.2.8 [**Software Engineering**]: Metrics—*performance measures*; D.2.11 [**Software Engineering**]: Software Architectures—*Domain-specific architectures*

## Keywords

grid monitoring architecture, grid information service, network monitoring, network service, theodolite service

## 1. INTRODUCTION

Modern high-speed networks enabled the definition of new paradigms for the coordination of geographically dispersed resources. Among these, Grid systems offer distributed computing capabilities on resources not subject to centralized control, heterogenous, and provided by different organizations. These resources and services are grouped into virtual pools that are accessible to users with the right credentials [11].

Grid-aware computations need an accurate and up-to-date view of the available resources. One special case of resource is connectivity. In order to inform applications about connectivity, we need to measure its characteristics and make them available to applications: packet loss rate, or round trip time are representative connectivity characteristics.

Scalability problems arise from the fact that connectivity characteristics are defined point-to-point: if a Grid system offers $n$ services, the dimension of the problem is $n^2$. A hierarchical decomposition of the problem following a divide et impera scheme offers an efficient solution, and is often adopted. In our case, this involves partitioning the system into subsystems, assuming that the characteristics between arbitrary services is obtained by composing characteristics within partitions, and between partitions. This scheme can be made recursive, to obtain solutions with complexity better than $n^2$ (like suggested by [10]).

We argue that hierarchical decomposition is inapplicable to our problem, for the reason that the composition of network characteristics is practically not feasible. The following example in a 3-nodes system should clarify our point: assume that the route from node A to node B partially overlaps (i.e., has some intermediate routers in common with) the route from node B to node C. Knowing the packet loss rate from A to B and from B to C does not help in determining the packet loss rate between A and C, since loss events might occur along the path in common between the two routes, which is by-passed by packets from A to C. In order to avoid inconsistent estimates, the monitoring activity should take into account the link level topology of the system, which is an extremely demanding requirement. But even when such requirement were fulfilled, the successive problem would be the rule to apply to compose characteristics: for instance, depending on network load, the composition of two loss rates may be the sum of the twos, when the network is lightly loaded, or the maximum of the two, when a traffic control mechanism starts dropping packets along the pipe.

In this paper, we illustrate an architecture that controls the production of connectivity characteristics, paying special attention to the scalability of the proposed solutions. We accept a $n^2$ solution, since connectivity characteristics composition is not viable. However, we make an effort to reduce $n$, the number of points in our monitoring topology, in order to improve the scalability of our solution, while providing users with useful observations. In Section 2, we present our model for partitioning a Grid system. In Section 3, we depict the design of a prototype implementation. Finally, in Section 4 we summarize the relevant works and in Section 5 we draw up our conclusions.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
*2nd Workshop on Middleware for Grid Computing* Toronto, Canada
Copyright 2004 ACM 1-58113-950-0 ...$5.00.

## 2. MODEL FOR PARTITIONING A GRID

In this section, we introduce a model for monitoring the connectivity of a Grid system avoiding the composition of connectivity characteristics. It is based on the following observation: the periphery of a Grid infrastructure is usually designed in order to optimize the use of the internal infrastructure, therefore it is over-dimensioned with respect to the performance predictably offered by the internal infrastructure. This authorizes to partition Grid Services into domains, corresponding to groups of peripheral services sharing a common interface to the internal infrastructure, and to introduce the requirement 1.

*Requirement 1.* For each pair of domains, the intra-domain communication infrastructure performs better when compared with the inter-domain communication infrastructure.

Based on the above observation, we consider that the infrastructure within partitions has a marginal impact on such characteristics, and we define a Network Service for each ordered pair of domains as a unidirectional communication service between them; the connectivity characteristics between two Grid services included into different partitions will correspond to those associated to the Network Service between the two partitions. If the internal connectivity of a domain does not satisfy the requirement 1, actions should be taken in order to maintain the representation of the connectivity consistent with the real state (see Section 2.1 for more details).

In order to monitor Network Services connectivity, network monitoring tools should run in appropriate locations, from where they can observe the behavior of the intra-domain communication infrastructure, with minimal interference on/ from traffic and workload within the domain. Network monitoring tools offer to Grid applications a specific service, that we call Theodolite Service. Since the location of a Theodolite Service can be dependent on the monitored Network Service, a domain may contain several Theodolite Services, each specialized in the monitoring of an outgoing branch. This specialization can also reduce the interference between internal traffic and traffic generated by monitoring tools. Figure 2 describes the partitioning model formalized as a UML (Unified Modeling Language) class diagram, more details are provided in Appendix A.

Figure 1 describes the partitioning of a simple Grid into domains: each domain (i.e., D1, D2, and D3) offers a number of Edge Services (e.g., C1 and S1), and provides appropriate, dedicated and over-dimensioned high performance connectivity between the hosts where Edge Services are located, thus satisfying the requirement 1. Connectivity with other domains might be based on an infrastructure that is considered appropriate, but is neither dedicated nor over-dimensioned (e.g., leased lines): these are the Network Services (represented by arrows, e.g., N1a) of the Grid. Theodolite Services (e.g., T1) might be located on the gateway itself, or on hosts with a direct and fast connection with the gateway in order to avoid the interference with internal traffic as described above.

In a real scenario, a domain may correspond to the Intranet of a computing center including all its Grid services. The Intranet may also includes geographical links (see Figure 1, domain D1) under the conditions defined in requirement 1. Another possible scenario is an Intranet split in two or more domains due to the different LAN (Local Area Network) technologies that are in use (e.g., FastEthernet and GigaEthernet). The last scenario that we consider is a domain that consists of Grid services provided by two or more Intranets that are geographically interconnected with high-performance dedicated links.

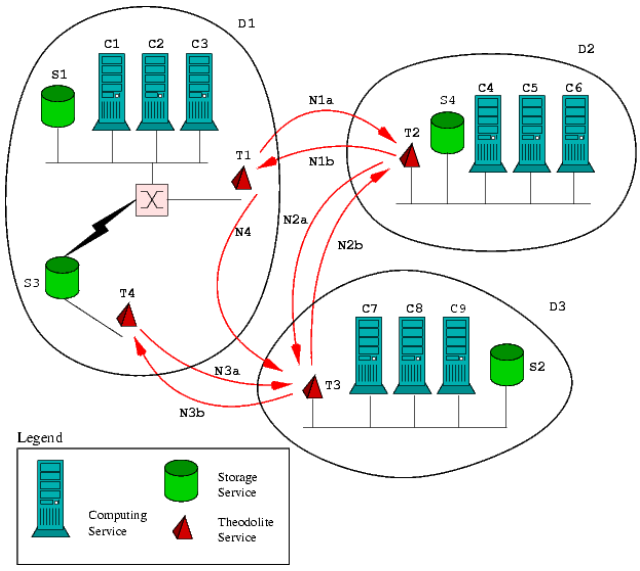


**Figure 1: A partitioning of a Grid into domains**

### 2.1 Fault Tolerant Issues

We must take into account that requirement 1 can be violated due to anticipated or unanticipated events. For instance, (1) a traffic burst may degrade the performance of an internal link, or (2) an hardware failure or routing change may alter the characteristics of an internal link, or (3) an upgrade of an inter-domain link may enable some components of the inter-domains fabric to outperform some domain fabric.

We distinguish two classes of events: internal events (first two cases above), that are due to the degradation of the fabric supporting connectivity inside a domain and external events (latter case), that are due to the upgrade of the intra-domain fabric. We briefly outline detection and recovery strategies for the two classes.

When an internal event occurs, certain Edge Services inside the involved domains may appear closer than they are, since the presence of the internal bottleneck is not revealed by the observations published by the Theodolite Service. Detection of such event is obtained by comparing internal and external observations, and also from those applications that find that an Edge Service does not exhibit the expected performance: they should inform the Theodolite Service which is responsible of the Network Service about the possibility of an internal bottleneck. One way to recover from an internal event is to indicate as unreliable the observations for those Edge Services that exhibit a degraded performance because of the internal bottleneck. This operation is carried out by the Theodolite Service, that indicates the anomaly to the Grid Information Service (GIS): a simple and effective action might be to indicate unreachable such services. An alternative consists in inducing a kind of back-pressure: the degraded performance of an internal link is reflected by the

Theodolite Service of the degraded domain by lowering the published performance of the Network Service edged to this domain. This option might need some sort of cooperation between the Theodolite Services at the edges of the same Network Service.

When an external event occurs, Edge Services in the domain may appear closer than they are, as a consequence of the improved performance of the fabric between the domains: the situation is therefore comparable to the one following an external event. Detection is similar to the case of internal events: either a Theodolite Service detects the inconsistency, or a Grid application signals a failed expectation. It seems inappropriate to tag the Edge Services as unreachable in order to recover from such event: this might produce a sensible degradation as a consequence of an improvement of the fabric. The indication that the observation is unreliable might also induce a similar form of resource misuse. Since external events are, with infrequent exceptions, somewhat planned, the best way to cope with them seems to be to avoid their occurrence: this can be obtained by upgrading the internal fabric before the external one, by trading a lower share of a common resource when it is upgraded, or by reorganizing the domain partitioning around an improved link.

## 3. ARCHITECTURE OF THE PROTOTYPE

In this section, we present the status of a prototype implementation. Its strongly modular design allows partial redesign or replacement of parts that become obsolete, or appear as unfit in a different environment.

As for the monitoring activity, each Theodolite Service is decomposed into several monitoring Sessions subclassed into Periodic and OnDemand. We distinguish two kinds of controls structures for a session: a periodic control, appropriate for the usual ping, and an on demand control, with an external trigger. The characteristics that can be currently measured for each Network Service are: one way jitter, clock skew, round trip delay, and round trip loss. The plan is to add also: one way loss, TCP achievable bandwidth, and UDP achievable bandwidth.

We opted not to implement the production engine, which is an extremely delicate component, but reuse one of the existing ones. We successfully implemented separate adapters for the Globus Monitoring and Discovery Service (MDS) version 2 [8, 12], and for the Relational Grid Monitoring Architecture (R-GMA). The former is based on the Lightweight Directory Access Protocol (LDAP) technology [14]. The latter is an implementation of the Grid Monitoring Architecture (GMA) [13] based on the relational model [3, 4].

The monitoring and topology databases contain the Grid description according to the UML representation in Figure 2. Its replication does not pose serious problems, since it is seldom updated, and partial inconsistencies are tolerated.

The network monitoring host contains a GIS adapter, on the right side of the dotted line in Figure 3. On the left side of the dotted line, a hierarchy of processes implements the monitoring activity: *GlueDomains* (1) is a daemon process that controls the whole monitoring activity of the host. It spawns the processes that implement the theodolite services. The description of the theodolite services is obtained querying the monitoring database hosted by the GlueDomains server, each time a theodolite service is spawned. The query returns the list of all theodolite services that are associated

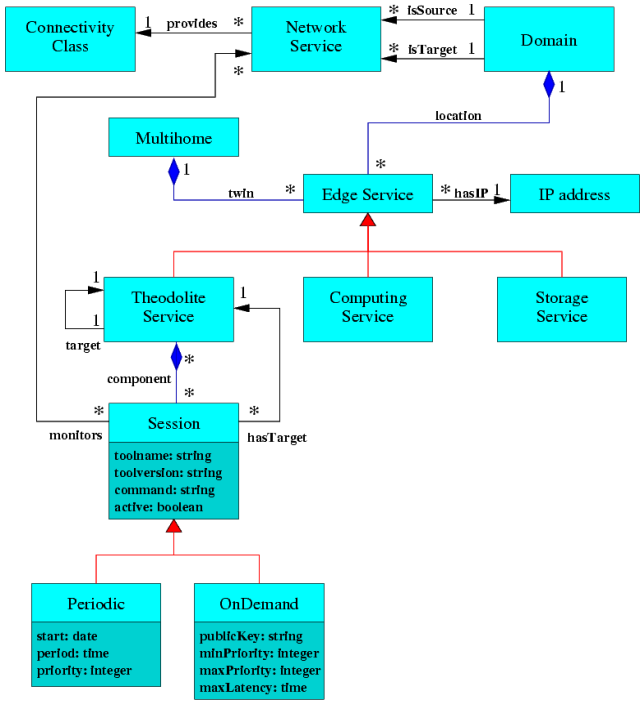


**Figure 2: UML class diagram for modeling the topology**

with any of the IP addresses of the host. *Theodolite* (2) is a process that implements a theodolite service. It spawns - and re-spawns when needed all monitoring sessions associated with a theodolite service. The description of all sessions associated with the theodolite service is retrieved from the monitoring database. *Session* (3) is a process that implements a monitoring session. All parameters that configure a specific session come from the topology database. A session interacts with the GIS adapter to record the observations in the production engine.

A flow of LDIF (LDAP Data Interchange Format) [9] entries is generated by the functions provided by the MDS adapter. Such flow is re-ordered and buffered by a MDS-proxy process, which runs as a daemon. Periodically, the buffer is flushed by the GRIS (Grid Resource Information Service) host using the gdip-client script, which uses a pair of sockets between the two hosts. In this way, the gathered metrics of the connectivity of a Grid are published through the GIS.

The prototype is currently being deployed in a number of sites part of the INFN-Grid (the Grid of the Italian National Institute for Nuclear Physics) [2]. The topology design consists of four domains: Bari, Napoli, Bologna and Padova. The network monitoring data is subsequently collected by the GridICE monitoring service [6]. The latter allows for Web presentation of the current status or historical behavior of the connectivity of the whole domain-based Grid topology. The testing of the prototype and the collection of connectivity characteristics are just started.

## 4. RELATED WORK

The architecture introduced in this paper is in debt with Wolski [15] and the NWS (Network Weather Service). How-

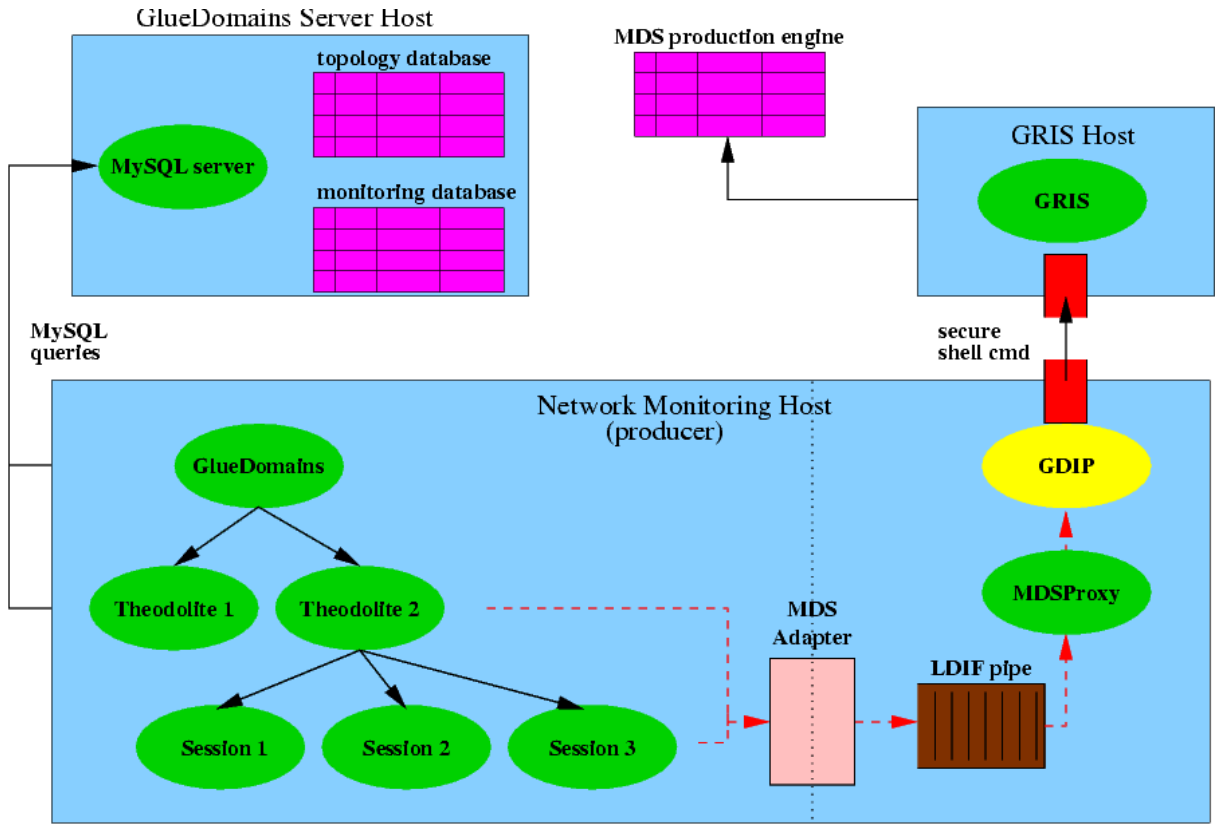


**Figure 3: Architecture of the prototype with the MDS adapter**

ever, although NWS is highly relevant as a proof of concept, it is not useful for a large scale deployment. We tried to put in practice some ideas that characterized such a project, paying special attention to performance: therefore we used existing products to support the configuration database and to store observations, and we introduced the concept of domain to improve scalability.

The concept of domain is not to be confused with a hierarchical decomposition of the Grid, as in [10]. On one side, domains do not have the power to reduce the $n^2$ complexity of the problem; on the other, we do not take as granted the possibility of composing observations in order to obtain significant estimates.

The domain-based architecture should also be distinguished from a site-based architecture, as in [5]: such topology is based on DNS (Domain Name System) naming, and cannot reflect the requirements stated in the introduction, which ensures that observations consistently represent the expected behavior.

## 5. CONCLUSION

Monitoring the communication infrastructure of a Grid is a delicate task that deserves a good deal of attention in order to limit its overhead and produce significant data. In this perspective, we motivated the introduction of a network topology based on the concept of domain, we gave a conceptual framework to the architecture, and we illustrated an implementation that is currently being deployed in a production-quality Grid system, the INFNGRID [2, 1].

Partitioning a Grid into domains is a way to limit network monitoring overhead: this indirectly improves Grid scalability, and usability of network monitoring output. However, a precondition for the application of such option is that intra-domains fabric is characterized by a higher connectivity, with respect to inter domain fabric: this is the basic requirement for domains partitioning.

Future work will focus on the collection of the monitoring data related to the connectivity of the defined Grid topology. Such data will be correlated to the status of Grid services in order to explore how they can be used for service selection. Finally, fault tolerant mechanisms for dealing with the violation of the requirement 1 will be further investigated.

## 6. ADDITIONAL AUTHORS

Additional authors: Cristina Vistoli, INFN-CNAF, email: `vistoli@cnaf.infn.it`

## 7. REFERENCES

[1] GlueDomains Project. http://www.di.unipi.it/~augusto/gluedomains/.

[2] Grid of the Italian National Institute for Nuclear Physics. http://grid.infn.it/.

[3] A. Cooke, A. Gray, L. Ma, et al. R-GMA: an Information Integration System for Grid Monitoring. In *Proceedings of the 11th International Conference on Cooperative Information Systems (CoopIS 2003), Catania, Italy*, Nov 2003.

[4] A. Cooke, N. Nutt, A. Datta, et al. RGMA: First Results After Deployment. In *Proceedings of the Conference on Computing in High Energy and Nuclear Physics (CHEP 2003), La Jolla, CA, USA*, Mar 2003.

[5] A. Alessandrini, H. Blom, F. Bonnassieux, T. Ferrari, R. Hughes-Jones, M. Goutelle, R. Harakaly, Y. Li, C. Pham, P. Primet, and S. Ravot. Network Services: Requirements, Deployement and Use in Testbeds, Technical Report D7.3, DataGrid, 2002.

[6] S. Andreozzi, N. De Bortoli, S. Fantinel, G. Tortone, and C. Vistoli. GridICE: a Monitoring Service for the Grid. In *Proceedings of the 3rd Cracow Grid Workshop (CGW 2003), Cracow, Poland, October 2003.*

[7] M. Campanella. Implementation Architecture Specification for the Premium IP Service, Technical Report D9.1, GN1(GEANT), 2000. http://www.dante.net/geant/publicdeliverables/GEA-01-032av2.pdf.

[8] K. Czajkowski, S. Fitzgerald, I. Foster, and C. Kesselman. Grid Information Services for Distributed Resource Sharing. In *Proceedings of the 10th IEEE International Symposium on High-Performance Distributed Computing (HPDC-10), San Francisco, CA, USA*, Aug 2001.

[9] G. Good. The LDAP Data Interchange Format (LDIF), Technical Specification, RFC 2849, IETF, June 2000.

[10] R. Harakaly, P. Primet, F. Bonnassieux, and B. Gaidioz. Probes Coordination Protocol for Network Performance Measurement in Grid Computing Environment. In *Proceedings of the IEEE International Symposium on Parallel and Distributed Computing (ISPDC 2003), Iaşi, Romania, July 2002.*

[11] Z. Németh and V. Sunderam. Characterizing Grids: Attributes, Definitions, and Formalisms. *Journal of Grid Computing*, 1(1):9–23, 2003.

[12] The Globus Team. Globus Toolkit 2.2 MDS Technology Brief, Draft 4, Jan 2003.

[13] B. Tierney, R. Aydt, D. Gunter, W. Smith, M. Swany, V. Taylor, and R. Wolski. A Grid Monitoring Architecture, GGF Informational Document GFD.7, Jan 2002.

[14] W. Wahl, T. Howes, and S. Kille. Lightweight Directory Access Protocol v.3, RFC 2251, IETF, Dec 1997.

[15] R. Wolski. Dinamically Forecasting Network Performance Using the Network Weather Service, Technical Report TR-CS96-494, University of California at San Diego, January 1998.

# APPENDIX

## A. DESCRIPTION OF THE UML CLASS DIAGRAM

This section provides a description of the classes and attributes part of the UML class diagram presented in Figure 2. Concerning the defined classes:

**Service** superclass that represents any service that a Grid offers to applications

**Edge Service** superclass that represents a service that does not consist of connectivity, but is reached through connectivity

**Multihome** represents an aggregation of Edge Services that share the same hardware support, but are accessible through distinct interfaces

**Computing Service and Storage Service** they are subclasses of Edge Service; their attributes (not shown in figure) include a reference to the production engine record that contains the relevant characteristics of the provisioned service like amount of available storage for Storage Services, or processor share availability for Computing Services

**Network Service** a unidirectional communication service between two domains

**Connectivity Class** different kinds of connectivity offered, like BestEffort or Premium IP [7] in a differentiated services environment

**Theodolite Service** service that monitors a number of Network Services; the service offered by a Theodolite Service is not apparent, since its role is the observation of network characteristics

**Session** a monitoring session, run as part of a theodolite

**Periodic** monitoring session run with a constant timing pattern (possibly randomized)

**OnDemand** ondemand-based monitoring session; such subclass can be used to capture expensive sessions, such as an iperf run

**Domain** partition that composes the Grid

The aggregation associations are defined as follow:

**Location** all Edge Services are included in a domain: this relationship is represented by the location association, that aggregates several Edge Services into a domain

**Twin** Edge Services can also be aggregated according to their hardware support: this is justified when they are accessible through different network interfaces, so they can be differently aggregated to domains; the twin association binds a Service to at most one Multihome that represents its hardware support

**Component** represents a Theodolite Service as an aggregate of sessions

The general associations are defined as follow:

**IsSource and IsTarget** a Network Service represents the fabric between two domains: the source and destination associations reflect this; a domain can be associated as source or destination of many Network Services; instead, each Network Service has unique source and destination domains

**Target** the monitoring activity of a Theodolite Service is performed in cooperation with another Theodolite Service; this fact is represented by the monitors association, that binds two Theodolite Service: one which runs the test, and another that responds

**HasTarget** associates a session to a target Theodolite Service, which should cooperate in the monitoring activity

**Monitors** associates a session to the monitored network service(s); it is not excluded that a monitoring session monitors several Network Service, for instance in case of distinct Communication Classes