GES 2007 German e-Science
Available online at http://www.ges2007.de

Providing Remote Access to Robotic Telescopes by Adopting Grid Technology

T. Granzer[1], F. Breitling[1], M. Braun[1], H. Enke[1] and T. Röblitz[2]

[1] Astrophysikalisches Institut Potsdam, An der Sternwarte 16, D-14482, Potsdam, Germany
[2] Zuse Institute Berlin, Takustrasse 7, D-14195, Berlin-Dahlem, Germany
email: `{tgranzer, fbreitling, mbraun, henke}@aip.de, roeblitz@zib.de`
phone: (+49 331) 7499 350, *fax:* (+49 331) 7499 309

Abstract

We present an architecture for enabling remote access to robotic telescopes through the adoption of Grid technology. With this architecture, Internet connected robotic telescopes form a global network and are controlled by a global resource management system (scheduler), similar to individual compute resources in a Grid. By virtualizing the access to these telescope resources and by describing them and observation requests in a generic language (RTML). Astronomers are provided with an interface to a telescope network, from which they can get the appropriate resources for their observations. Moreover, new kinds of coordinated observations become feasible, such as multi-wavelength campaigns or immediate and continuous monitoring of transient astronomical events. This paper describes the architecture, the processing of observation requests and new research topics in a global network of robotic telescopes.

1 Introduction

In recent years, more and more robotic telescopes have come into operation and are providing astronomers with valuable results. At the same time Grid technology has been developed and established in science and industry. This technology provides means to connect and share geographically distributed resources which are independently administered by their local owners. Therefore, Grid technology appears to be a promising development for the integration of individual robotic telescopes into a global network. Such a network is being developed by the AstroGrid-D project [2]. It can provide important advantages, e.g. for the coordination of multi-wavelength campaigns or the immediate response to and continuous monitoring of transient astronomical events. For example, while a single telescope has a chance of only 12% to perform a successful observation of a gamma-ray burst, which is reduced further by e.g. daytime, weather, and altitude with respect to the telescope, the probability in a network can reach the theoretical limit of 95%.

Research in Grid technology addresses two main aspects, namely the virtualization of the resources, which aims at a uniform interface for access via the Internet, and the coordination, which aims at an optimized usage and sharing of the available resources. Initially the development of middleware made good progress in virtualization. Popular examples are the Globus Toolkit [3] and UNICORE [13]. Currently this coordination is subject of ongoing research which focuses on components such as resource brokers or data-management systems. For example, in high-energy physics gLite [4] provides basic coordination capabilities.

The research and development in AstroGrid-D aims at:

- providing a uniform interface to telescope resources (virtualization),
- optimizing the distribution of observation requests by using enhanced scheduling (coordination), and
- performing new types of observations e.g. coordinated multi-wavelength campaigns or 24 h monitoring of transient astronomical events (testing).

In this paper we describe the approach to virtualization of robotic telescopes and briefly discuss basic coordination mechanisms. We developed an architecture which inherits it's basic mechanism from the Globus Toolkit e.i. from MDS, the Gatekeeper, the Jobmanager and the Grid-Mapfile. In our network, telescope resources as well as observation requests are specified in RTML (Remote Telescope Markup Language) [5], which is then converted to the RDF format [8] to store its metadata in the information service developed for AstroGrid-D. For the basic coordination mechanisms we will use two parallel approaches: first, the protocol for negotiating observation time for a scientist with a telescope in the network, and second, mechanisms for reserving compute resources.

In Sec. 2 we briefly describe the robotic telescopes STELLA-I and II and in Sec. 3 the architecture of their integration. In Sec. 4 we sketch two approaches for coordinating the use of multiple-robotic telescope resources.

2 The Robotic Telescopes STELLA-I & II

The Astrophysical Institute Potsdam (AIP) has constructed and is operating three robotic telescopes: STELLA-I and STELLA-II in Tenerife (Canary Islands, Spain, cf. Fig. 1) and the smaller RoboTel in Potsdam (Germany) [12]. With a diameter of 1.2 m, STELLA-I and STELLA-II are among the world's largest robotic telescopes. The combination of the high resolution spectrograph of STELLA-I with the wide-field image of STELLA-II is unique. RoboTel is an 80 cm telescope, which will also be accessible for local schools with about half of the observation time dedicated to education.

Robotic operation of these telescopes means that the software of the local scheduler takes local weather information into account for the generation of the observation schedule. The different hardware and operation modes of the STELLA telescopes require a rather general approach to the software development. A generalized description of the observation request and the scheduling of the individual observations do not contain information about a specific resource. A unified description of hardware and observation request has been approached in the last two years within the Heterogeneous Telescope Network (HTN) initia-

Figure 1: The robotic telescopes STELLA-I and STELLA-II as installed in Tenerife (Canary Islands, Spain).

tive [1]. As a first step the Remote Telescope Markup Language (RTML) [5, 6] has been developed for describing telescope resources and observation requests which are the basis for further work.

Currently, STELLA-I & II are operated by the local STELLA control system (SCS) whose layered structure is shown in Fig. 2. Whenever an observation request is made, the scheduler runs through its list of known targets and selects the target of highest priority. This scheduling schema is known as *dispatch scheduling*. The request is then sent to the sequencer which breaks it down into a sequence of individual commands which are designated for the individual hardware devices attached to the main host. The main host includes the telescope itself, the instruments required for the observation and some peripheral devices (Adapter in Fig. 2) which provide guiding and maintain stable observation conditions. In addition to a local database of observable targets the on-site scheduler has a receiving interface which is accessed via Java-RMI when new targets arrive. Targets added in this way are available to the dispatch scheduler as soon as a new target is requested. For security reasons, currently only local access to this receiving interface is possible. It requires an authorized user, i.e. the operator, to copy the target description file, which is an XML-formatted ASCII file, to the telescope server at Tenerife. The operator then invokes a parser at the server which in turn sends the target to the local scheduler. To ease the operator's decision at which priority the new target should enter the local queue, a simulator is in use. It uses the same robotic software and is attached to a simulator of the telescope hardware. In rare cases, when a newly requested target is only visible for a short time, the operator may decide to cancel the ongoing observation in favor of the new target.

Because astronomical observations strongly depend on unpredictable weather conditions, months may pass between submission of a target and its execution.

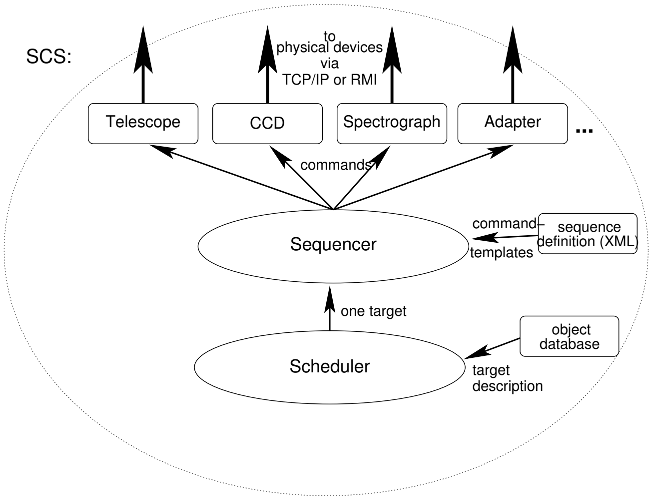

Figure 2: Components of the Stella control system (SCS) for STELLA-I & II.

Thus, the astronomer is notified via e-mail, when the observation of the target has begun and when it has completed. Health-monitoring of the system is done by the telescope operator similar to a system administrator on computer resources.

3 Architecture for the Grid-Integration of STELLA-I

Fig. 3 illustrates the architecture for integrating the robotic telescope STELLA-I into the Grid. The telescope gateway located at the AIP serves three main purposes: the communication with the robotic telescope does not need to be changed, a uniform interface to astronomers is provided through Grid technology and the access to the robotic telescope can easily be secured via a distinguished gateway host. Main components of the telescope gateway are the gatekeeper, the observation managers and the information provider (cf. Fig. 3). In addition, the architecture provides an information service which stores the information of the telescopes such as geographic location, observation capabilities, current weather conditions, availability etc. An astronomer can contact a Grid-enabled telescope from any host which is connected to the Grid.

3.1 The Information Provider

Initially, the information provider registers the static information e.g. geographic location, observation capabilities, access restrictions etc. with an information service (step ①). After a telescope has been registered the information provider periodically determines its current state (step ②) by remotely querying the telescope controller and auxiliary services such as the weather forecast. This

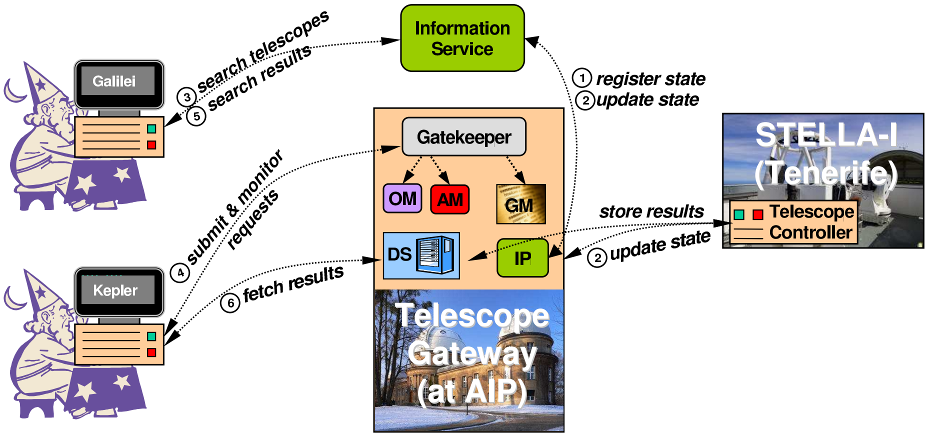

Figure 3: Integration of a telescope resource (STELLA-I) into the Grid. Illustrated is the interaction of the different components: observation manager (OM), administration manager (AM), information provider (IP), Grid-Mapfile (GM) and data storage (DS).

dynamic information complements the static information stored by the information service (step ②).

Since the telescope information is provided via an RTML document but the information service stores metadata in RDF format a conversion is necessary. We developed a generic RTML to RDF converter based on the OwlMap[7] package. For example, Listing 1 shows an extract of the description of the telescope STELLA-I. The corresponding description in RDF is shown in Listing 2.

Listing 1: Outline of the RTML description for the telescope resource STELLA-I.

```
<RTML version="3.1a" mode="resource" uid="rtml://DE.aip.STELLA-I"
     ...>
<Telescope name="STELLA-I">
  <Aperture type="geometric" units="meters">1.2</Aperture>
  <SpectralRegion>optical</SpectralRegion>
  <Location name="Izana Observatory, Tenerife, Spain">
    <Latitude units="degrees">28.300000</Latitude>
    <EastLongitude units="degrees">-16.509722</EastLongitude>
    <Height units="meters">2480</Height>
  </Location>
</Telescope>
</RTML>
```

Listing 2: Outline of the RDF description of the telescope resource STELLA-I.

```
@prefix RTML: <http://www.rtml.org/v3.1a#>.

RTML:RTML [
  RTML:mode "resource";
  RTML:uid "rtml://DE.aip.STELLA-I";
  RTML:version "3.1a";
  RTML:Telescope [
    RTML:name "STELLA-I";
    RTML:Aperture [
      RTML:type "geometric";
      RTML:value "1.2"^^RTML:meters];
    RTML:SpectralRegion "optical";
    RTML:Location [
      RTML:name "Izana Observatory, Teneriffa, Spain";
      RTML:Latitude "28.300000"^^RTML:degrees;
      RTML:EastLongitude "-16.509722"^^RTML:degrees;
      RTML:Height "2480"^^RTML:meters ; ]]]
```

3.2 Querying the Information Service

Before submitting an observation request to a telescope, an astronomer might be interested in the available telescopes and their characteristics, their current weather conditions and the numbers of observation requests that are queued at each telescope (step ③). All this information can be retrieved from the information service by using its SPARQL [11] query interface. For example, the query shown in Listing 3 extracts information on all telescopes on the northern hemisphere with the capability to observe objects in the optical spectral range and ranks them according to their altitude (higher altitudes are preferred). Currently the astronomer has to submit such queries to obtain a list of eligible telescopes from which he can choose a preferred instrument for his observation. In the future, this process can be further automated by a broker.

Listing 3: A SPARQL query for retrieving a list of telescopes on the northern hemisphere with the capabilities to observe objects in the optical spectral range, ranked by their altitude.

```
PREFIX RTML:<http://www.rtml.org/v3.1a#>

SELECT ?name
WHERE
  {
  ?telescope RTML:SpectralRegion "optical" .
  ?telescope RTML:Location ?loc .
  ?loc RTML:Latitude ?lat .
  FILTER (?lat > 0) .
  ?loc RTML:name ?name .
  ?loc RTML:Height ?height
  }
ORDER BY DESC(?height)
```

3.3 The Gatekeeper

The gatekeeper acts as the entry point for observation or administrative requests. After it has received a request (step ④) it authenticates the requestor and verifies if it is authorized to perform the requested action by comparing the distinguished name of the requestor with the entries in the Grid-Mapfile (GM) (cf. Fig. 3). Thereafter, the gatekeeper starts either an observation manager (OM) to handle observation requests or a Globus fork job manager (AM) to handle administrative requests. Through the latter the software environment of both gatekeeper and telescope controller can be remotely administered. In the following we will discuss the handling of observation requests. One difference of this method to compute job requests is that observation requests do not need additional input data. All necessary information is contained in the RTML document.

3.4 The Observation Manager

Each observation request is managed by an observation manager which implements a GRAM-like interface[1]. This way, the observation manager provides methods for initiating observation requests, for querying their status, handling their results and canceling requests. When an observation manager is started by the gatekeeper it receives the observation request as an RTML document and converts it into the format understood by the telescope controller. Finally, the observation manager queues the request with the local submission facility. When an observation has finished, its execution status is recorded with the information service and if requested is sent to the user by email. If the observation was successful, the result is locally stored in files. Since STELLA has currently no guaranteed permanent Internet connection the observation results are cached by the telescope controller and transmitted to a storage server (cf. Fig. 3) at the AIP when a connection is available. In addition to the produced images, the observation manager inserts metadata about generated results into the information service. The flexibility of the information service allows to associate data results with observation requests, which in turn are linked to particular astronomers or scientific projects. The metadata contains the storage facility for the data and the transmission status. Hence, the data of completed observations can conveniently be located and it can be determined if it has been transmitted from the telescope controller to the storage server (step ⑤).

3.5 Comparison with Job Handling in the Globus Toolkit

The architecture presented in the sections above adopts the virtualization of compute resources provided by the Globus Toolkit (GT). As in the Grid environment three main components exist: an information service, a gatekeeper and a

---

[1] The Grid Resource Allocation and Management (GRAM) service is part of the Globus Toolkit.

request manager. The communication among these components and their activities are essentially the same. However, the virtualized resources for computing and observation are different. Another difference lies in the representation of information. On the one hand it is a metacomputing directory service and a RDF-based information service on the other. Thus, the messages exchanged between the components differ, but the core protocol for managing them is the same.

4 Coordinating a Network of Robotic Telescopes

Once multiple robotic telescopes are virtualized as described in Section 3, their use will be coordinated. The first coordination task is to automate step ③ (cf. Fig. 3) the search for telescopes matching an observation request. In Section 4.1 we briefly describe how we adopt resource brokerage to perform this task. Naturally, the successful execution of an observation request does not only depend on the overall workload, but also on external conditions in particular the weather. Thus, the efficient distribution of observation requests has to take these external conditions into account. In Section 4.2 we describe two mechanisms for probing the future status of resources. Note that the mechanisms described here only present the current state of work in progress and no final solution.

4.1 A Broker for Observation Requests

In the architecture proposed in Section 3, an astronomer queries the information service to determine telescopes which match an observation request. In Grid computing, a user does not query the information service directly, but asks a broker to perform that task. The same mechanism can be applied to assign observation requests to telescopes. Additionally, a broker may also enforce community policies to ensure fair sharing of telescopes among astronomers and projects.

If the broker receives an observation request which is described in RTML, it first generates a SPARQL query for the request's properties. Then it sends the query to the information service and receives an ordered list of matching telescopes. The actual conversion has still to be developed.

4.2 Using Probes to Determine the Best Telescope

Since the execution of an observation request may be prevented by various reasons, in particular bad weather conditions the brokerage mechanism may be enhanced by an additional *probing* step, in order to increase the rate of successful requests. After the list of appropriate telescopes has been received, these telescopes are queried when and with which probability they could perform the requested observation.

For computing resources we have developed a similar mechanism to facilitate the efficient processing of co-reservation[2] requests [9, 10]. The Grid reservation services (GRS) receives a multi-part request each being flexible in a given time window and the actual number of processors to be reserved, i.e. the quality of service (QoS). It then sends probe requests to reservable resources, which determine a list of *(time slot, QoS)-tuples* within the user-specified bounds *(time window, QoS)*. Each pair is attributed with a number of properties such as the likelihood that the slot may be reserved or the cost for reserving it. After receiving the responses from the different resources, the GRS evaluates them to determine the best combination for a co-reservation.

Within the HTN initiative a similar protocol was developed [1]. Given an observation request, a telescope controller determines the probability for performing the observation for a number of time slots (cf. Listing 4). With this information a broker can submit the request to the telescope with the highest probability. Because the weather conditions can be rather unstable at some locations [3], the broker will maintain a database which stores information about the accuracy of probes. Thus, the ranking of resources overestimating their capabilities may be adjusted by the broker.

Listing 4: Outline of a reply to a request as anticipated within the HTN.

```
<...>
<Schedule>
  <Exposure count="1">
    <Value units="second">60.0</Value>
  </Exposure>
  <Observation type="offer">
    <Probability units="percent">80.0</Probability>
    <Cost units="Euro">1.5</Cost>
  </Observation>
</Schedule>
<...>
```

5 Conclusion and Future Perspectives

We have presented an architecture and mechanism to integrate robotic telescopes into the Grid. By adopting Grid technology provided by the Globus Toolkit and by using standards such as RTML, RDF and SPARQL, robotic telescopes can become accessible in a similar way as Grid compute resources.

In contrast to compute jobs, observations depend on external conditions such as time of day or the current weather. Thus, the optimal use of the limited observation time requires more advanced management mechanisms. Here we envision combining two developments, one from research in advance reservation of compute resources and another one from the HTN initiative. These mechanisms try to determine at which time the request might be admitted.

---

[2] A co-reservation consists of multiple parts each reserving a resource.

[3] For STELLA-I and II, the current mechanisms to forecast the precise weather conditions are in average only sufficiently precise for about 300 seconds.

The idea of connecting separate robotic telescopes to a worldwide network offers some unique advantages that reach beyond a mere economic use of resources. Multi-wavelength campaigns, immediate response to transient events and continuous monitoring are only some areas, which can profit from the presented architecture. Its deployment is expected to be an important step in the field of robotic astronomy.

Acknowledgment

This work is supported by the German Federal Ministry of Education and Research within the D-Grid initiative under contracts 01AK804A and 01AK804C.

References

[1] A. Allan, F. Hessman, K. Bischoff, M. Burgdorf, B. Cavanagh, D. Christian, N. Clay, R. Dickens, F. Economou, M. Fadavi, S. Fraser, T. Granzer, S. Grosvenor, T. Jenness, A. Koratkar, M. Lehner, C. Mottram, T. Naylor, E. Saunders, N. Solomos, I. Steele, G. Tuparev, T. Vestrand, R. White, and S. Yost. A protocol standard for heterogeneous telescope networks. *Astronomische Nachrichten*, 327:744–+, September 2006.

[2] AstroGrid-D Project homepage, 2006. URL `http://www.gac-grid.org/`.

[3] Ian T. Foster. Globus toolkit version 4: Software for service-oriented systems. In *NPC*, pages 2–13, 2005.

[4] gLite, 2006. URL `http://glite.web.cern.ch/glite/`.

[5] F. V. Hessman. Remote Telescope Markup Language (RTML). *Astronomische Nachrichten*, 327:751–+, September 2006.

[6] Heterogeneous Telescope Network. RTML Schema, December 2006. URL `http://monet.uni-sw.gwdg.de/XMLSchema/RTML/schemas/RTML-nightly.xsd`.

[7] OwlMap, 2006. URL `http://fresco-www.informatik.uni-hamburg.de/technology/`.

[8] Resource Description Framework, 2006. URL `http://www.w3.org/TR/rdf-schema/`.

[9] Thomas Röblitz and Krzysztof Rzadca. On the Placement of Reservations into Job Schedules. In *12th International Euro-Par Conference 2006, Dresden, Germany*, pages 198–210, 2006.

[10] Thomas Röblitz, Florian Schintke, and Alexander Reinefeld. Resource Reservations with Fuzzy Requests. *Concurrency and Computation: Practice and Experience*, 18(13):1681–1703, November 2006.

[11] SPARQL Query Language for RDF, 2006. URL `http://www.w3.org/TR/rdf-sparql-query/`.

[12] K. G. Strassmeier, T. Granzer, M. Weber, M. Woche, M. I. Andersen, J. Bartus, S.-M. Bauer, F. Dionies, E. Popow, T. Fechner, G. Hildebrandt, A. Washuettl, A. Ritter, A. Schwope, A. Staude, J. Paschke, P. A. Stolz,

M. Serre-Ricart, T. de la Rosa, and R. Arnay. The STELLA robotic observatory. *Astronomische Nachrichten*, 325:527–+, October 2004.
[13] UNICORE, 2006. URL `http://www.unicore.eu/`.