S. Gullapalli,
K. Czajkowski,
C. Kesselman
USC/ISI

Global Grid Forum Draft

S. Fitzgerald
CSUN

Document: draft-grid-notification-01.txt
Category: Informational July 5, 2001
Expires: September 2001

Grid Notification Framework (VB0.1)

Status of this Memo

This document specifies the Grid Notification Framework developed for the Grid community. We welcome a discussion of the issues and suggestions for improvements.

Abstract

This document describes a generic notification framework through which information about the existence of a grid entity, as well as properties about its state can be propagated to other grid entities. Grid entities include---but are not limited to---compute devices, hardware devices, services, storage, instruments, and people such as users or system administrators. The framework is based on periodic messages being dispatched by a declaring entity to one or more receiving entities. Upon receipt of these messages, receiving entities can interpret the contents of the message to generate or update their beliefs about the grid entity being described.

Table of Contents

Status of this Memo:

This document specifies the Grid Notification Framework developed for the Grid community, and requests discussion and suggestions for improvements.

1. Introduction

This document describes a generic notification framework through which information about the existence of a grid entity, as well as properties about its state can be propagated to other grid entities. Grid entities can include---but are not limited to---compute resources, hardware devices, services, storage, instruments, and people such as users or system administrators. Notification is based on periodic messages being delivered from a declaring entity to one or more receiving entities. Upon receipt of these messages, receiving entities can interpret the contents of the messages to generate or update their beliefs about the grid entity being described.

This Document describes:

i) The structure of the message used within the framework.
ii) The naming mechanism used for describing Grid entities (URL and subject names).
iii) The timestamp mechanism that limits the scope of each message within the framework.
iv) A soft-state mechanism developed to interpret information from a stream of timestamped messages along with a state model (Finite State Automata) of the beliefs at a receiving entity.
v) Example uses of this framework to develop services such as registrations and invitations for Virtual Organizations (VOs).

2. Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC-2119 [RFC-2119].

3. Approach taken by the Framework:

Transmittal and receipt of periodic messages is the basis of this framework. Receipt of messages about a particular entity provides credence to the belief that the entity exists (is alive). The payload within each message provides information about the properties (state) of the entity. Two levels of information are conveyed through the receipt of each message 1) the existence and 2) state information of an entity.

Separate intervals of time are associated with the validity of each of the above pieces of information. Typically these time intervals are selected such that if no further messages are received, one’s belief in the existence of an entity stays within the framework longer than the confidence in its state information.

These intervals can be adjusted to balance between the accuracy (validity) of information versus the cost of generating, propagating, and processing each message. Generating and propagating messages with small confidence intervals would result in more up to date information, however it is compute and communication intensive. Increasing the time intervals associated with the confidence of each piece of information results in a possibly less accurate picture, but reduces the computational and communications overhead.

This framework can be used to capture relatively steady state conditions with the Grid, where a majority of messages get delivered in a timely manner.

A soft-state model is constructed and maintained at each receiving entity using the information contained within messages along with the two time intervals mentioned above. This soft-state model provides the basis for aggregating information as well as for making the information available to other applications and services.

Note that this framework does not define how to interpret the state information contained within the message, nor does it specify what processing must be performed by the entity receiving the message. For instance, receivers might simply note the existence of the message, or initiate some additional interactions with the described entity, and/or simply log the information to stable storage, or perform some other unspecified operation.

4. Key Aspects of the Framework Include:

1. Unambiguous identification of Grid entities using their URLs and/or Grid security names. Belief about remote entities is dependent not only on information expressed by remote entities, but also on a receiver’s local trust policy for remote information sources.
2. Knowledge based on a ***soft-state model*** to provide a robust notification mechanism coupled with a graceful degradation of stale information.

3. Standard representation of global time. The framework requires the interpretation of message content over time, but does not rely on any fixed accuracy or clock synchronization. A suitable time representation is adopted that expresses global time value, value precision, and value accuracy to account for locale-specific synchronization to the global time.
4. Application selectable (tunable) time intervals that determine the longevity of each datum within the system.

5. Definitions:

Message (MSG):

A MSG is a unit of data that transmitted between entities within the Grid for the purpose of notification.

Auxiliary Data (AUX):

AUX is a single datum associated with a notification message that provides optional state information about a described entity.

Described Entity (DES):

DES is the entity whose existence is being asserted and whose state is being described by auxiliary data (AUX), included within the framework message (MSG).

Declaring Entity (DCL):

DCL is the entity that asserts the validity of the information about DES contained within the MSG.

Sending Entity (SND):

SND is the entity that initiates the propagation of the MSG. Often, but not always, it is the same entity as DCL.

Receiving Entity (RCV):

RCV is the entity that receives and interprets the MSG. RCV is also the entity that can provide the information within the MSG to other external services (such as search engines).

5.1. Timestamps

Each framework message includes three timestamp values labeled TS1, TS2 and TS3. The section below on “Message Semantics” describes what each timestamp value represents. Timestamps are relative to the global time as perceived by DCL. The values of the three timestamps are related via the constraint:

TS1 <= TS2 <= TS3

Here we define the operator "<=" to specify the relationship between two timestamps. TS1 <= TS2 is defined to be true if and only if the time specified by TS2 is the same as or occurs after the time specified by TS1. We also use the operator "<", to denote that the time specified by TS1 which occurs before the time specified by TS2.

6. The General Notification Exchange:

6.1. Information Flow Diagram:

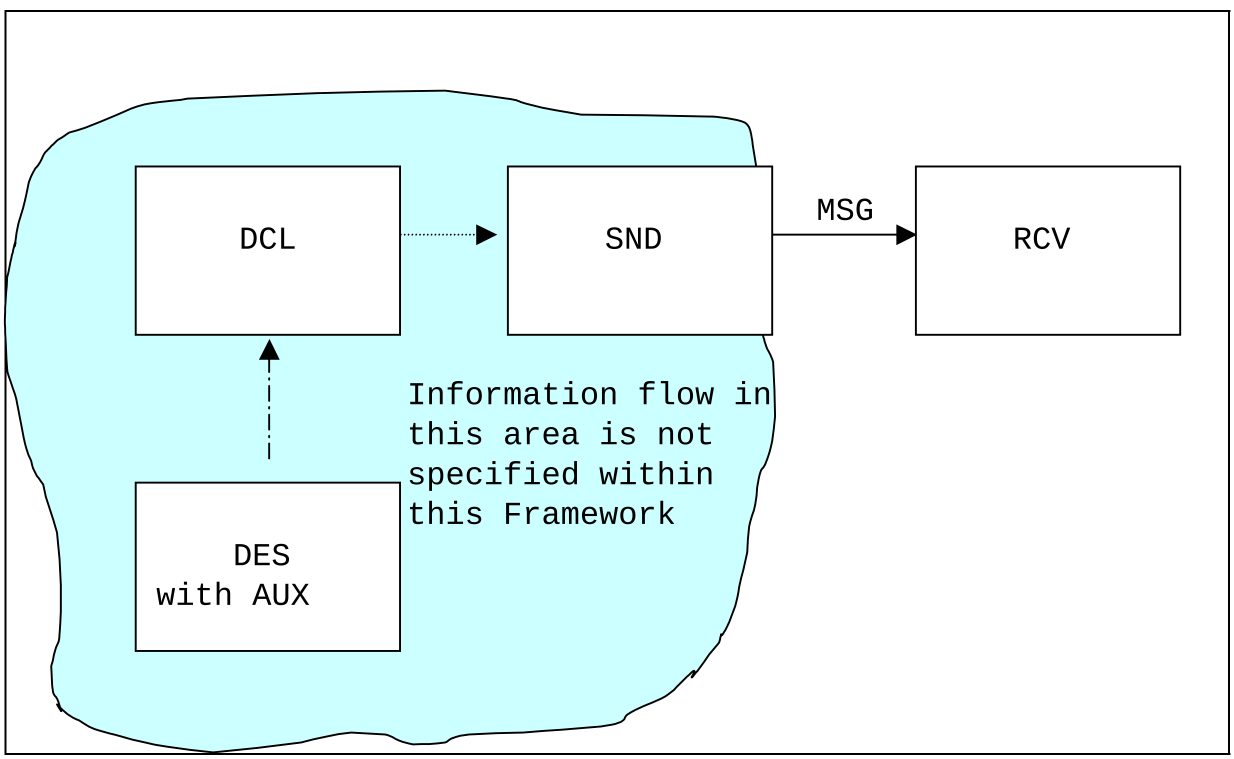

Figure 1: Information flow diagram.

In Figure 1, we show the flow of information about an entity in the schematic below. Please note that the framework does not specify the flow of information between the DES, DCL and SND. The SND (sender) gets the MSG to dispatch to the RCV.

6.2. Description of a Notification Event:

A notification event within this framework is defined as a message (MSG) generated by DCL, based on knowledge about DES, transported by SND and received by RCV. The framework does not specify how DCL obtains information about DES.

The RCV entity accepts the message from SND. The message asserts the existence of DES with the state property AUX at time TS1 (defined below in Message Semantics).

Note that by the time the message reaches RCV, there is no guarantee that DCL's assertion will continue to be valid. Independent verification of the information could be performed using a variety of methods. One such method is to utilize the enquiry operation associated with the Grid Information Service (GIS). The enquiry can be directed towards either DES or DCL.

Each message contains the required information to locate, contact and securely communicate with the DES and/or DCL entities to obtain additional information. Thus each message has to provide a name and/or a communications protocol for DES and DCL, and a security mechanism, either through a trusted channel or via the use of security credentials.

6.3. Message Semantics

Each notification message within this framework is interpreted to represent the following statement:

> *"**DCL** asserts that at time **TS1**, **DES** exists and **AUX** is expected to be valid until at least time **TS2** and **DES** is expected to exist until at least time **TS3**."*

The notion of existence indicates that a valid name is associated with an entity. The entity may or may not be "alive" or functioning, regardless of any assertion contained within AUX.

6.4. A Special Case of the Notification Exchange:

In the general case, DCL, DES and SND may be different entities. Notably, DCL could be a system administrator, SND and DES could be a grid service, and RCV could be an application that processes the message in order to provide information to other applications.

In some implementations, however, three of the above four entities can identify a single entity leading to a special case of the framework, as described below.

SND (sending entity), DCL (declaring entity) and DES (described entity) may identify a single entity. In this case, SND informs RCV that it is alive and has state information (AUX) that should be valid from time TS1 until time TS2. RCV may assume that SND exists between time TS1 and TS3.

7. Finite State Automata (FA) at the Receiving Entity:

RCV's knowledge of a described entity can be modeled via a three-state FA. The meaning associated with each state is described below:

S1: There are no applicable assertions provided by DCL about DES.
S2: DCL asserts that DES exists and AUX is valid.
S3: DCL asserts that DES exists and AUX is uncertain.

A state transition occurs when there is a change in RCV's beliefs about a DES entity. RCV's belief changes either through the expiration of time or by the receipt of a valid message (MSG).

This framework performs two types of time comparisons i) between a timestamp in a message and the current time as known at RCV, and ii) between timestamps in two messages from the same DCL about the same DES. In both cases, this framework leaves it to the application to define the comparison functions to determine which value is more recent.

We define **Current_TS** to be time as perceived by the RCV.

The use of Current_TS is then used to determine whether or not a message is relevant. Messages are considered to be relevant, if:

Current_TS <= TS3

If RCV receives multiple relevant messages about a particular DES from a particular DCL, only the message with most recent TS1 is considered.

The other messages are discarded. Consider the case where two messages, A and B, arrive at RCV with the value of timestamp TS1 set to $TS1_A$ and $TS1_B$, respectively. If $TS1_A$ is more recent than $TS1_B$ (i.e., $TS1_B < TS1_A$), message B is discarded. Notice that the order in which messages are received does not determine which message is discarded.

The situation of multiple, distinct DCLs reporting to the same RCV about the same DES is left to the RCV to handle based on its internal policy.

7.1. Transitions within the State Diagram:

In this section, we describe the transitions that occur within the given FA (Figure 2). Each state transition is labeled as a Roman numeral. Additionally, either the string MSG_N or TS value (TS1, TS2, TS3) is provided within a set of parentheses. These strings indicate that the transition may occur when a relevant message (MSG_N) arrives or when a constraint on timestamps changes, respectively.

All state transitions are at the RCV relative to one particular DES and one particular DCL.

Note that the timestamps from the previous relevant message are retained. We denote these timestamps as $TS1_P$, $TS2_P$, and $TS3_P$. A new message is denoted as MSG_N with associated timestamps $TS1_N$, $TS2_N$, and $TS3_N$.

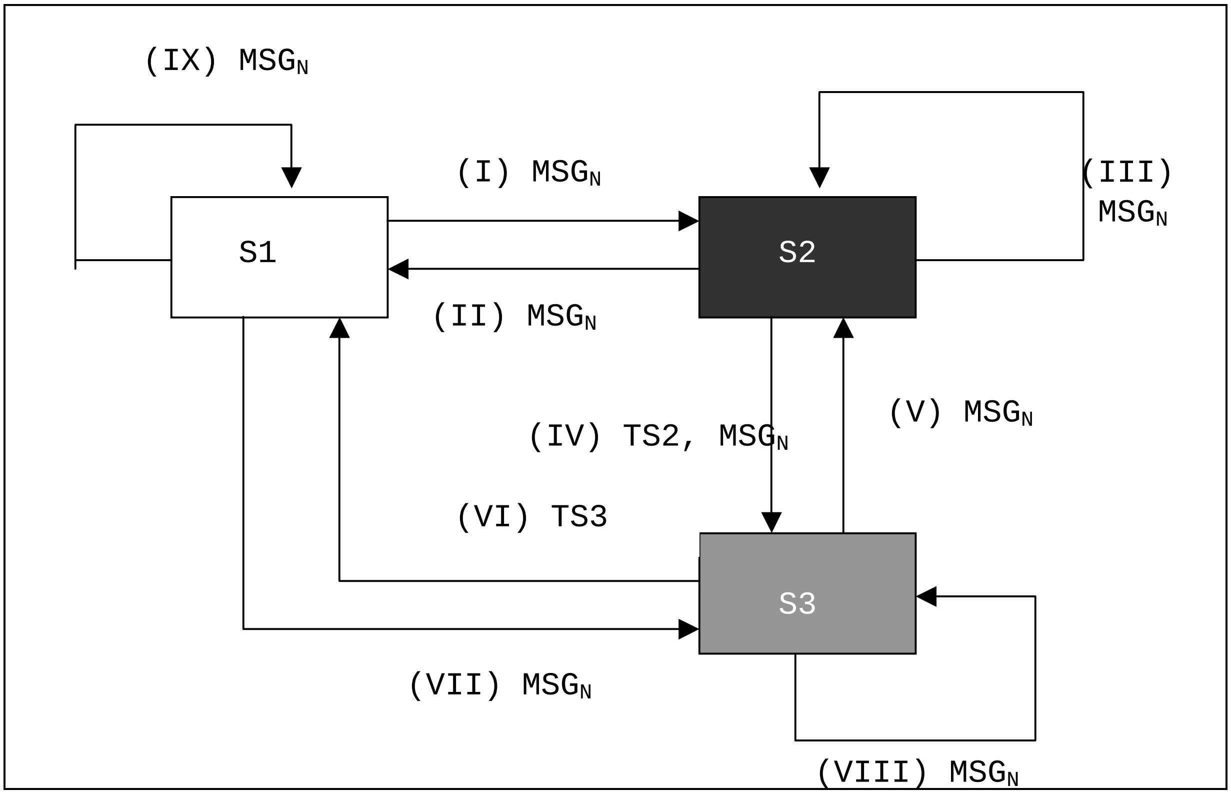

Figure 2: State Transition Diagram of RCV's beliefs

The initial state is S1. The conditions for each transition are as follows:

(I) RCV receives a new message MSG_N that obeys the constraints:

$$TS1_P < TS1_N <= Current\_TS <= TS2_N$$

The new message provides relevant information about DES. Hence, RCV's beliefs about DES moves from state S1 to state S2.

(II) RCV receives a new message MSG_N that obeys the constraints:

$TS1_P < TS1_N$ AND $TS2_N <= TS3_N <= Current\_TS$

The new message invalidates the previous received message. Hence, RCV's belief about DES moves from state S2 to state S1

(III) RCV receives a new message MSG_N that obeys the constraints:

$TS1_P < TS1_N <= Current\_TS <= TS2_N$

The new message provides updated information about DES. RCV's belief of DES remains in state S2, with updated times.

(IV)EITHER:

- RCV receives a new message MSG_N that obeys the constraints:

$TS2_N < Current\_TS < TS2_P <= TS3_P$

OR

- sufficient time passes and no new message is received. The following constraint is obeyed:

$TS2_P < Current\_TS <= TS3_P$

The AUX value is no longer current. Hence, RCV's belief about DES moves to state S3.

(V) RCV receives a new message (MSG_N) that obeys the constraints:

$TS1_P < TS1_N$ AND $Current\_TS <= TS2_N$

The new message is more recent than the previous message. Hence, RCV's belief about DES moves from state S3 to S2.

(VI) No relevant new message is received and sufficient time passes, which validates the constraint:

$TS3_P$ < Current_TS

The stored message is no longer relevant. Hence, RCV's belief about DES is obsolete, and it transitions to S1.

(VII) RCV receives a new message (MSG_N) that obeys the constraints:

$TS1_P$ < $TS1_N$ AND $TS2_N$ < Current_TS < $TS3_N$

The new message indicates that DES exists but the AUX value is not current. Hence, RCV's beliefs about DES moves from state S1 to S3.

(VIII) RCV receives a new message (MSG_N) that obeys the constraints:

$TS2_N$ < Current_TS < $TS3_N$

(IX) RCV receives a new message (MSG_N) that obeys the constraints:

$TS3_N$ < Current_TS

8. Framework Message Format

To develop a protocol based on the above framework all operations are encapsulated in a common envelope, called the MSG.

MSG::= SEQUENCE {	
version	Integer,
dcl_name	SubjectName, (optional)
dcl_location	Location, (optional)
des_name	SubjectName, (optional)
des_location	Location, (optional)
ts1	Timestamp,
ts2	Timestamp,

ts3	Timestamp,
auxiliary_state	OCTET STRING
}	

```
        Location ::= IA5String
            -- a URL as defined in RFC-2396 [RFC-2396]

        SubjectName ::= UTF8String
            -- a representation of a DN as defined in RFC-2253
[RFC-2253]

        Timestamp ::= Grid Forum Representation
```

The MSG is either sent via an authenticated connection using GSI or within messages signed using GSI credentials.

Note: The specification of the above message content (auxiliary state) will be translated into a SOAP(/XML) description of the fixed metadata content and the potentially nested AUX data field. This translation will be appended to this document in the near future. An appropriate standard variation including XML signature syntax will also be specified to support optional DCL-signed messages.

8.1. Description of the elements:

Version: The version number of the framework used by the implementation that created the message. Version numbers are important to maintain interoperability of mixed-version grids. The version number is to be used for verification of compatibility between different implementations.

dcl_name: The name of the declaring entity, in UTF8String format---a representation of a DN as defined in RFC 2253.

dcl_location: The location of declaring entity. It is a URL (DNS Name, protocol and the port number) in IA5String---a URL as defined in RFC2396.

des_name: The name of the described entity, in UTF8String format---a representation of a DN as defined in RFC 2253.

des_location: The location of described entity. It is a URL (DNS Name, protocol and the port number) in IA5String---a URL as defined in RFC2396.

ts1: The time at which the declaring entity asserts information about the described entity. It is in the ASCII Grid Forum representation of time.

ts2: The time past which a declarer no longer has confidence in the information about the described entity. It is in the ASCII Grid Forum representation of time.

ts3: The time past which a declarer suggests that all information about the described entity may be invalid. It is in the ASCII Grid Forum representation of time.

Auxiliary Data: Information about the properties (state) of the described entity. This is suggested to be XML in general implementations.

9. Mapping onto Transport Services

This framework is designed to run at the applications level of the network protocol stack, over a transport layer. For example, SOAP over HTTP/HTTPS can be used as the underlying transport mechanism.

10. Developing An Information System for the Grid

A specific instantiation of the above protocol can be used to develop an information system for the Grid environment.

We introduce two types of notification services:

1. Registrations
2. Invitations

These services facilitate the aggregation of information from grid entities, which can then be queried by external applications and services. In both these services, we describe entities and aggregators of information.

10.1. Registrations:

Registration is the mechanism by which an entity provides information to another entity (aggregator). The information is transported within the auxiliary (AUX) part of the message. The aggregator entity collates information from one or more DEC individual entities, and can be queried by external services and entities for this information.

10.2. Invitations:

Invitation is the mechanism by which an aggregator solicits information from DEC entities. Once invited, a grid entity can register periodically with the newly discovered aggregator. The aggregator can, as described in the above section on registration, then be queried by external services and entities for this collated information.

11. Advice To Implementers

We intend to develop a follow-on protocol specification document based on this framework. That document will specify APIs and implementation details.

12. Acknowledgments

We are grateful to numerous colleagues for discussions on the topics covered in this paper, in particular (in alphabetical order, with apologies to anybody we've missed): Ewa Deelman, Ian Foster, Shelley Henderson, Adriana Imanitchi, Laura Pearlman, Warren Smith and Peter Vanderbilt.

The organization and other basic text of this memo were taken from the IETF PKIX Roadmap Draft [IETF-PKIX].

13. References

[RFC-2119] Key words for use in RFCs to Indicate Requirement Levels. S. Bradner. March 1997.

http://www.ietf.org/iesg/1rfc_index.txt

[RFC-2253] Lightweight Directory Access Protocol (v3): UTF-8 String Representation of Distinguished Names. M. Wahl, S. Kille, T. Howes. December 1997.

http://www.ietf.org/iesg/1rfc_index.txt

[RFC-2396] Uniform Resource Identifiers (URI): Generic Syntax. T.Berners-Lee, R. Fielding,L.Masinter. August 1998.

http://www.ietf.org/iesg/1rfc_index.txt

[IETF-PKIX] draft-ietf-pkix-roadmap-06.txt: Internet X.509 Public Key Infrastructure. A. Aresenault, S. Turner. November 2000.

ftp://ftp.isi.edu/internet-drafts/draft-ietf-pkix-roadmap-06.txt

[GRID-TS] Standard Timestamp for Grid Computing: D. Gunter, B. Tierney. February 2001.

http://www-didc.lbl.gov/GridPerf/papers/GWD-GP-010-1.pdf

14. Contact Information

Sridhar Gullapalli
USC Information Sciences Institute
4676 Admiralty Way, Suite 1001
Marina del Rey, CA 90292-6695
Phone: (310) 448-8790
E-mail: sridhar@isi.edu

Karl Czajkowski
USC Information Sciences Institute
4676 Admiralty Way, Suite 1001
Marina del Rey, CA 90292-6695
Phone: (310) 448-8451
E-mail: karlcz@isi.edu

Carl Kesselman
USC Information Sciences Institute
4676 Admiralty Way, Suite 1001
Marina del Rey, CA 90292-6695
Phone: (310) 448-9338
E-mail: carl@isi.edu

Steven Fitzgerald
Department of Computer Science
California State University Northridge
Northridge, CA 91330
Phone: (818) 677-3314
E-Mail: steve@ecs.csun.edu