Vijay Dialani, University of Southampton

16th August 2004

**Information Dissemination Service – Abstract Usage Patterns**

Status of This Memo

This memo provides information to the Grid Data Access and Information Dissemination communities. It does not define any standards or technical recommendations. Distribution is unlimited.



**Abstract**

Information Dissemination Service (IDS) uses publish/subscribe paradigm for management of data within grid environments. The IDS is primarily composed of four distinct artifacts – publication, subscription, propagation and consumption, which are manipulated by means of the interfaces provided by the IDS. Interactions between these artifacts determine the semantics of the service.

This paper re-examines the IDS semantics and interfaces to consider: First, possible resource sharing between multiple subscriptions. Second, facilitate maintenance and recovery of service state, and Third, policy based object lifetime management.

The outline of this paper is as follows: First, it describes the standard IDS scenario based on an implicit publication. It then introduces the scenario for creation and management of explicit publications. Second, it suggests modifications to the reflective capability of the service. Third, it describes the policy based object lifetime management of the service entities. Finally, it outlines some of the working assumptions and open issues.

Contents

## 1. Introduction

The Grid Data Distribution (GDD) Model was introduced in [1]. This data dissemination model is based on publish-subscribe paradigm. The GDD model introduces four distinct artifacts publication, subscription, propagation and consumption. Additionally [1] proposes an Information Dissemination Service (IDS) based on the GDD model and describes the IDS service interface.

IDS interface provides a simplified abstraction for manipulating the IDS artifacts. In its present form, the IDS specification provides explicit interfaces to manipulate the publication object, subscription object and relies on use of rules for manipulation of scheduling, event generation, propagation and consumption related artifacts. IDS (like DAIS) promotes use of a rule based descriptive approach to provide an abstraction over the underlying complex data model and scheduling model[1]. For example, it is proposed that the subscription rules will be expressed in SQL syntax or in extended SQL.

The artifacts with declarative interfaces allow type checking and dynamic runtime binding, and are explicitly visible as a part of the service interface. However, the artifacts that use descriptive interfaces are obscure to the scrutiny by the IDS and do not explicitly feature in the service specification. Consequently, we refer to the two types as primary and secondary artifacts to distinguish between the artifacts with either declarative or descriptive interface respectively.

The primary artifacts constitute the visible service state, while the state of the secondary artifacts is maintained within the IDS. The scope of this discussion is restricted to re-examining the semantics of interactions, possible resource sharing between, and lifetime management aspects of these primary artifacts, exposed through the IDS interface. The scope has been restricted due to the following reason:

Only primary artifacts can be directly manipulated through the service interface and necessarily form a part of the service semantics that the client application needs to be aware of. Although, the semantics of the secondary artifacts need to be known to the service users, they remain opaque at the level of service interface.

For example, a publication artifact allows startPublication, stopPublication, createPublication and alterPublication operations, the semantics of which should be explicitly known to the client to effectively utilize the service. Alternately, the publication rule is specified by means of the SQL statement and assumes an underlying relational data model. A client application needs to be aware of the interface language (in this case SQL), but may not necessarily be aware of the exact relational schema being manipulated by the descriptive interface.

The current service interface described in [1], allows the creation of an explicit and an implicit publication. A publication initiated by a request to publish, via the publication interface, results in creation of an explicit publication. However, a subscription request results in the creation of an implicit publication. In the following sections, we examine the creation and maintenance of an implicit and explicit publication, align the DAIS and INFOD terminology, and discuss the lifetime management and reflective aspects of the service.

[1] While the use of partially descriptive interface rather than a completely declarative interface provides the IDS to flexibly contain varying complex resources, it also introduces the need to provide the reflective and verification capability on the underlying systems.

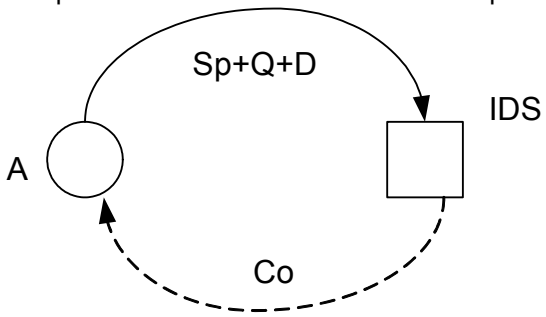
1.1 Notation

| | | | |
|---|---|---|---|
| A | Analyst | Sp | Subscription |
| C | Client | P | Publication |
| S | Subscriber | Q | Subscription rule |
| ——— | Request | ST | Control Signals |
| | | D | Data (parameters) |
| ------ | Response | Co | Consumption of Events |

1.2 Implicit Publication

By default, a subscription request results in the creation of an implicit publication. In such case the lifetime of the subscription determines the lifetime of the publication object. That is to say that the publication is valid for the duration specified by the subscription object.

The following scenario diagram represents the creation and maintenance of an implicit publication. The terms analyst, client, consumer have been used to represent the same interpretations as that used in the DAIS specifications [2].




❖ Figure 1 – An implicit publication
(For notation please refer to section 2)

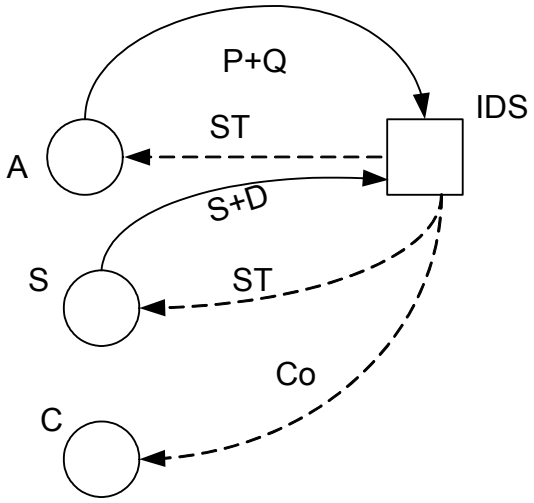
Note: In section 3, we suggest a modification to introduce the reflective capabilities on the implicit publications, please refer to this section for further details.

1.3 Explicit Publication

It is possible to use a combination of subscription and implicit publication to express most of the information dissemination scenarios. However, there exist a number of cases in which it is possible to share a publication source between multiple subscribers. Consider the example of a stockbroker service. The service creates and manages publication of numerous stocks being traded. An analyst (in this case the stock market) creates a publication of specific stocks. The lifetime of such a publication is incumbent to the availability of the stock, the trading hours and the trading on the stock market.

If an explicit publication is created and managed by an analyst through explicit invocations to the publication artifact on the service interface, the source can be managed independently of the subscribers. The following scenario diagram represents an analyst that creates a publication source and specifies the query that can be parameterized. A number of clients can create individually parameterized subscriptions. Termination of the publication invalidates the existence of these subscriptions.

Note: In this case the IDS allows the separation of concerns. The analyst decides the scope and the rate of publication, where as the subscriber can use the query parameters to selectively consume certain events.

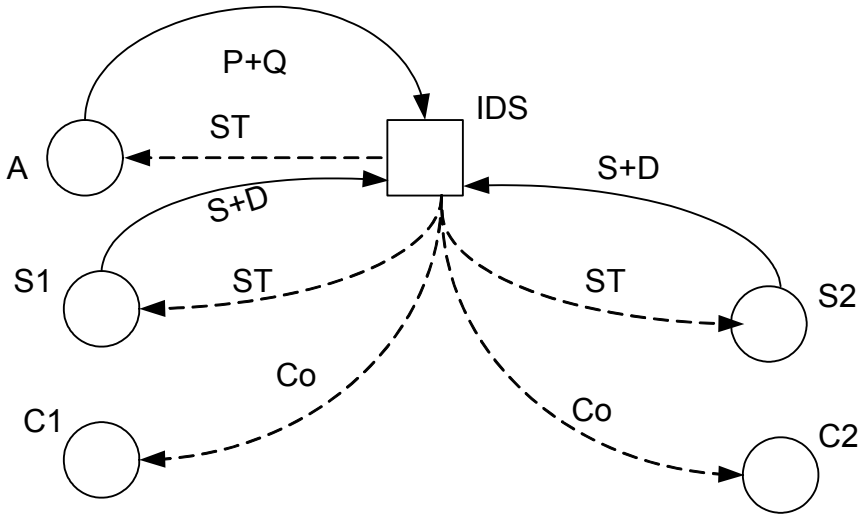



❖ Figure 2 – An explicit publication
(For notation please refer to section 2)

Note: It is assumes that INFOD services are primarily designed for publication of data by persistent data sources. In the case of explicit publication as described above, the subscribers are unable to express the temporal constraints on the data of interest. In such cases, the subscribers could retrieve historical data, by creating alternate publications, provided such operations are supported by the service.

1.4 Explicit Publication with delegation

In the above case, the analyst exclusively managed the publication lifetime. However, in certain cases, the publication lifetime may be determined by the properties of its subscribers. For example, consider the case of sensor networks. In the scenario diagram represented below, an initial subscriber creates a subscription request to obtain the weather data from the sensors employed in the field. As the subscriber allows the sharing of the data with other potential subscribers, a number of subscribers simultaneously subscribe to the data feed. In this case, the termination of the initial subscription does not indicate any cessation of interest in publication. The publication object should be valid till at least one of the subscriber requests is valid.




❖ Figure 3 – An explicit publication with delegation.
(For notation please refer to section 2)

## 2. Differences in Notation

| Role | Description in DAIS | Description in INFOD |
|---|---|---|
| Analyst | 1. Is the initiator of the query and provides the reference to third party delivery.<br>2. Controls the entire lifecycle of the query execution. | 1. By default the Analyst creates and manages the publication.<br>2. It determines the visibility of the Publication object.<br>3. Controls the discovery mechanism for the publication. |
| Client | Recipient the Query Results. | Recipient of the data items |

## 3. Reflective Capability

The current specification of the IDS [1], allows creation and maintenance of the primary artifacts. Creation of the primary artifact results in the creation of the identifier, that is used to uniquely identify the primary artifact that the client has access too and is allowed to manipulate. These identifiers represent the objects that constitute the state of the IDS service. However, the lack of the reflective capability of the service means that disconnection often results in non-recoverable state. It is proposed that the specification be modified to include the reflective capability on the primary artifacts to allow for recovery from failures. It is proposed that most of the system state be maintained by the service and that the recovery from client failures be permitted through reflection on the service state.

### 3.1 Describing Service State

The primary artifacts and their respective properties constitute the visible part service state and the state of the secondary artifacts represents the opaque state of the service. The service interface needs to be modified to provide reflective capability on the visible state, while providing limited reflection on the descriptive specification used to create the opaque state. Additionally, the service state includes the policy definitions and lifetime management of the primary artifacts and may also be considered crucial for recovering the service state.

It should be noted that each subscriber would have its own partial view of the state of the service. Such a partial view also contains the information about the shared/publicized artifacts.

### 3.2 Advertising the Publications

Explicit publication may be shared with a number of potential subscribers. The analyst determines the visibility of the publication object. An analyst needs to disseminate the information about the existence of the publication, for which we propose the use of two techniques.

#### 3.2.1 Brokerage Pattern

In a brokerage pattern, the analyst creates and registers a publication source with a broker service. A client requests a broker to identify an appropriate publication source for its subscription requests. Such requests may be met by either the currently existing publication sources or may be met as and when the publication source becomes available/visible in future. It is assumed that the schemas, data attributes and external events may play a part in defining subscription to future publications.

It is proposed that the broker pattern should extend and not replicate the capabilities that can already be provided by use of filters on the underlying data resource. We explain this with the aide of the two examples:

First, consider a bank broker that provides banking services to customers and allows manipulation on multiple bank accounts held by the customer. The bank associates a number of subscription requests with each customer account. In this example, we assume that the customer is able to associate a funds agent and the mortgage agent. The funds agent associates the rule

like – “Monitor customer’s funds if the balance in less than the customers credit limit”. In this case the scope of the publication is determined by the value of the data attribute, i.e. account balance. While the mortgage agent has the rule “Monitor customers who have defaulted for 3 or more payments”. Second, in the above example substitute the funds agents rule by “Monitor customers identified as potential risks” and mortgage rule as “Identify defaulters for recovery of assets”.

In the first, a data attribute based filter can subsume the publication rule. However, in the second case, the analyst determines how the semantic rule of publication translates into the actual publication artifact. A broker pattern is useful for explicit publications only.

#### 3.2.2 Delegation Pattern

A brokerage pattern requires the creation of the explicit publications and subsequent to their discovery; the analyst manages the publication. However, in certain cases a subscriber may like to share an implicit publication. In this case it may advertise the publication through a discovery service and thereafter delegate the lifetime of the publication source to the service. Unlike the broker pattern, where the discovery information is registered with a third party, the delegation method relies on the services capability to advertise the implicit publications through the reflection mechanisms.

For example, consider a region weather bureau that provides publications on the infrared radar images of this particular region. The weather bureau should only publish the images if it has at least one subscriber. Such a delegation pattern requires that the services provide an explicit interface to control the visibility of the publication source.

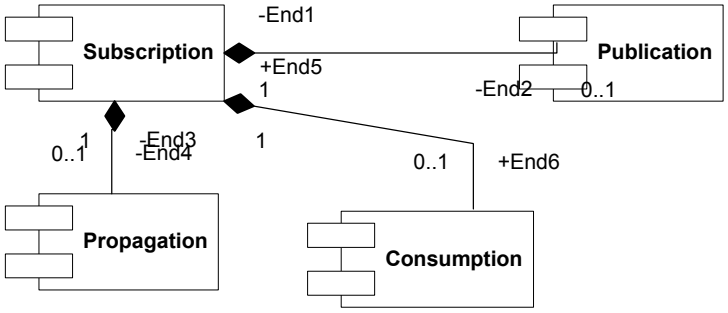
## 4. Policy based Object life-time management

Invocation of the service interface inadvertently leads to the creation of the primary artifacts. The lifetime of these artifacts are interconnected. For example, consider the case of the implicit publication. In this case the subscription object dictates the temporal constraints on the validity of the publication and in effect controls the lifetime of the publication object. The subscription object also controls the propagation and consumption objects created as a result of the subscription. The default policy in this case mandates that the cessation of the subscription object should result in termination of the propagation and consumption activities. However, this type of straightforward lifetime management is not applicable in the cases where multiple subscriptions share a publication.

It is proposed that to enable the use of policies to manage the complex lifetime management issues the service. Candidate rules may assume the following forms:

- Maintain active publication till subscriber > 0
- Maintain active publication for next 3 days.
- Retain consumption for 2 days after subscription.
- Maintain publication for next two years and make visible through reflection.

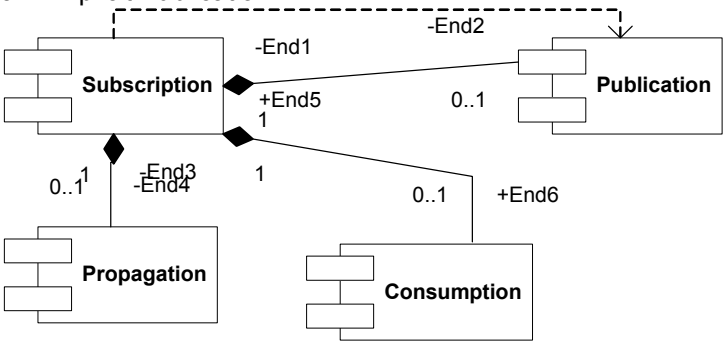
4.1 De-facto Object lifetime management.
The following diagram represents the object lifetime as seen in the case of the implicit subscriptions.




❖ Figure 4 – UML Component Diagram depicting default lifetime management in IDS

4.2 Object lifetime management scenarios.
The following scenarios represent the object lifetime management for two more cases:

a. Implicit Publication




❖ Figure 5 – UML Component Diagram depicting lifetime management for IDS in case of implicit publications

b. Explicit with broker pattern

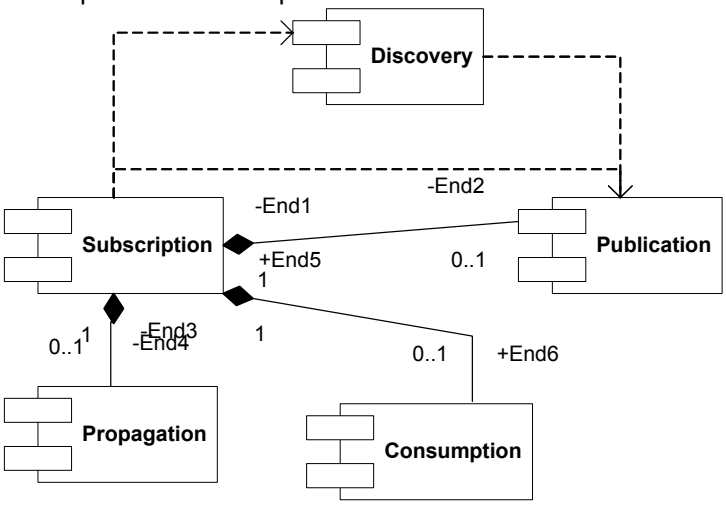



Figure 5 – UML Component Diagram depicting lifetime management for IDS with brokerage pattern.

4.3 Policy based object management and reflective capabilities.

Some of the policy based management of the publication and subscription objects and the various creation and destruction patterns can be transparently managed by the IDS. However, for certain complex patterns IDS should be able to provide the introspection over the underlying data objects. The proposed reflective capability on the IDS should allow introspection on the rules that govern the lifetime of the enclosed objects.

## 5. Discussions

Following is the list of issues that have been highlighted in the INFOD discussions and have been summarized here. Working assumptions have been made for some of these issues, while the others are still being investigated.

Resolved Issues:

- It is assumed that the IDS will provide publications over a persistent data source. A subscriber can retrieve the events in the past by creating the subscription with particular temporal constraints. However, in some cases it may not be possible to have a persistent data source for large amounts of data, for example streaming data. In such cases where the publication source is unable to retrieve the historical information, for example sensor networks, adequate metadata should be provided in the service properties of the IDS.
- Certain systems may require reliable communication and unique consumption of the event. It is proposed that each of the event items generated by the IDS should have a unique identifier. Additionally, the service may support FIFO queues for propagation of the events.

Open Issues:

[This section needs to be completed]

## 6. Acknowledgements

[To be completed].

## Author Information

Vijay Dialani
University of Southampton
Bldg-59, Highfield Campus,
Southampton,
United Kingdom.
vkd00r@ecs.soton.ac.uk

## Glossary

Recommended by not required.

## Intellectual Property Statement

The GGF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this recommendation. Please address the information to the GGF Executive Director.

**Full Copyright Notice**



**References**

[1] D. Gawlick, V. Gogate, C. Kantarjiev, C.Masden, S.Mishra, I.Narang, M.Subramanian, *Information dissemination in the Grid Environment,INFO-D WG Informational Draft, GGF-10, 23 Feburary 2004*.

[2] G.Riccari, M.Subramanian, S.Mishra, S. Laws, DAIS Usage Scenarios, DAIS WG Informational Draft, GGF-10, 24 Feburary 2004.