Created:September 24, 1998
Modified:September 16, 1999

Gemini High-Level Software Group Report

Description of an XML-based Phase 1 Document

Kim Gillies, Shane Walker, Darrell Denlinger, Dayle Kotturi

ocs.kkg.065-Phase1XMLDTD/05

This report describes an XML based file format for sharing and processing Phase 1 telescope proposal information. The format was created for use with Gemini telescope proposals, but the authors have tried to make document format flexible and general, making it a good candidate for use at other observatories as well.

Table of Contents

1.0 Introduction 2
- 1.1 Project Goals 2

2.0 Technical Overview 3
- 2.1 Document Features and Definitions 3
- 2.2 The Common Element 5
- 2.3 The Observatory Element 6
- 2.4 Validating a Proposal 6
- 2.5 XML References 7

3.0 DTD Description 9
- 3.1 The Phase1Document Element 9
- 3.2 The Common Element 10
- 3.3 Extension and Constraint Elements 11
- 3.4 The ResourceList Element 12
- 3.5 ObsList and Observatory Elements 13

4.0 Authoring Tools 16
- 4.1 Tool Support is Required for Proper Documents 16
- 4.2 Associated XML Data Files 17
- 4.3 Authoring Tool Support 18
- 4.4 Gemini Phase 1 Tool 18

5.0 Conclusions 18

6.0 References 19

7.0 The DTDs and Examples 19
- 7.1 Phase 1 DTD - AstronomyPhase1.dtd 19
- 7.2 The Gemini DTD 29

7.3 An Example Phase 1 Proposal Using AstroPhase1.dtd and Gemini.dtd. 32

1.0 Introduction

Most observatories follow a similar process for allocating telescope time that consists of a series of steps or phases. Astronomers describe a scientific problem and request facility access during Phase 1 of the process. A *Telescope Proposal Process Workshop*, hosted by National Optical Astronomy Observatories, was held in Tucson on 10 and 11 August, 1998. At the meeting several observatories compared and contrasted their Phase 1 process. It turns out that much of the information requested at different observatories during Phase 1 is similar, but the representation of that information varies among the different observatories. Many use a submission system with documents based on TeX, but the keywords used within the documents vary from observatory to observatory. At this time, there is no standard keyword set. One of the conclusions of the meeting was that a common Phase 1 keyword set was desirable and that an eventual move to Extensible Markup Language, or XML, was a reasonable goal.

Gemini needed a document format that would allow the partner countries to submit their Phase 1 proposals to Gemini. In current Phase 1 systems, proposals are represented by a list of attributes and values. The attribute is often called a keyword. However, Gemini Phase 1 needs more information than many observatories, and this requires a flexible, structured document. The keyword list approach does not easily support the structure and associations that Gemini Phase 1 requires.

XML [1] was designed to provide precisely the capabilities that are needed by the Gemini Phase 1 document; therefore, a document file format has been defined using XML, with the goal of satisfying the need determined at the Tucson meeting. The document format is defined by publishing a Document Type Definition (DTD). This report defines the Phase 1 DTD and supporting XML files. Gemini plans on using a similar XML document for its Phase II science program definition.

The authors offer the DTD described here as the basis for a common Phase 1 document that can be used by other observatories. If that is to happen, it is understood that aspects of this design may require changes. The details of Gemini's use of this DTD are described in another document.

In this report, an overview of the document format will be presented, and this will be followed by a detailed look at the DTD and an example document. The DTD and data files discussed in this document are available from the Gemini FTP site at the following URL: http://ftp.gemini.edu/pub/gemini_Central/OCS/phase1DTD.tar.Z. This document covers version 1.1X of the AstronomyPhase1.dtd where X is greater than 1 (i.e. 1.11 or above).

1.1 Project Goals

The most important goal for the development of the Phase 1 XML document was to produce a document that can describe the information needed for Gemini Phase 1. We examined proposals from several observatories making sure that the DTD supported the content that the various forms had in common, but content clearly specific to any one observatory (including Gemini) was not included. Many document elements are optional allowing observatories to pick the features they support or need. Consideration was given to the process of conversion to a new XML-based system by making the document as simple as possible and by using as few XML features and technologies as possible.

A secondary, important goal was that the document format support multiple observatory proposals within the same document. For example, a Phase 1 document can be created by one observatory's authoring tool. It can then be augmented to add proposal information for another observatory while continuing to be readable by the first observatory's authoring tool. A proposal based on the same targets can be submitted to multiple observatories, each including some observatory-specific information. To accomplish this does require some work on the part of the authoring tool, but not too much.

The multi-observatory capability does add some complexity to the document, but this complexity is needed to allow for a document that is general and not tied too closely to Gemini.

2.0 Technical Overview

An XML document is not merely a set of attributes and values; it is a structured document with content that describes what is present in the document as well as the order and hierarchy of the items. This is what sets XML apart from methods currently in use. The XML items, called *elements*, use start and end tags as markup to structure the document. An XML document looks like an HTML document, but the tags are not familiar HTML tags. Instead, the tags are application specific (the X in XML is for extensible after all). XML refers to the tags as elements. What is between the tags is referred to as content. Element and content are used in the remainder of this document to distinguish between a kind of information and its values.

XML is not too concerned with what appears between the tags; ensuring that the content makes sense is the job of the application processing the document. For instance, the XML document type definition doesn't say what appears in the phone number of an address; however, it can ensure that an address includes a phone number, and the address isn't valid unless it includes a phone number. XML is not too complex, but it is too much to cover here. Therefore, this document assumes the reader has some knowledge of XML. There are a number of books and on-line sources available to learn more about XML [1], [6], [7], [8].

The content and structure of the Phase 1 document can be described using a DTD or some other way of representing a schema for a class of documents. A DTD uses familiar regular expression/BNF notation to define the document format. An application uses an XML parser and the DTD (optionally) to read and validate that a document is correctly formed. The application can also use the DTD to write a valid XML document. The DTD is the agreement on what the document can contain. More sophisticated schema definitions methods for XML are on the horizon but are not yet defined well enough for general use [2],[3].

Many publicly available parsers and tools exist to read and write XML files. Java and C language parsers exist as well as extensive tools support in script languages like Perl and Python (see [7]). One of the advantages of using XML is that an existing parser may be used to process XML documents. This makes the software used to read and write the documents more robust and the document easier to modify. Formatting and display of the XML documents is also possible using style sheets written in CSS or XSL [4], [5].

In this report, DTD element and attribute names appear with emphasis as in `entity`. This should make it easier to detect when the text is discussing XML syntax.

2.1 Document Features and Definitions

The advantage of supporting more than one proposal within one document is that some content can be shared among all the proposals. For example, the principle and co-investigator content is the same for all the proposals in the document. In the Phase 1 document, the shared elements and content are contained in an element called `common`.

Many of the elements that are needed by each proposal are the same for each observatory, it's just that the content supplied for the elements differs. For instance, since each observatory has different instrument capabilities, each observatory's proposal needs its own technical justification element and instrument list. The content for the technical justification element of one observatory will describe how the science can be accomplished with the instruments of that observatory.

Some of the elements that are important to one observatory may not be needed by another. Some of the elements in the common part are optional, and most of the elements in the observatory sections are optional. This means that the document can remain valid and still be flexible. The side effect of making elements

optional is that higher level, observatory-specific software must be used to do complete semantic and content checking of a document.

Figure 1 shows how the Phase 1 document is structured at the highest level. The `phase1Document` element consists of a `common` element and zero or more `observatory` elements. The `common` element and one `observatory` element make a single *proposal*. The number of proposals contained in a document is the number of included `observatory` elements.

One important `common` element is called `targetCatalog`. This element contains a list of all of the targets that are to be observed in any of the proposals. This list includes the science objects as well as any support objects such as wavefront sensor objects. Entering target information is time consuming so it is important to make targets common.

The Phase 1 DTD uses the concept of a resource to describe something that an observatory provides that must be scheduled with an observation or program. A set of resources is represented by a `resourceList` element.

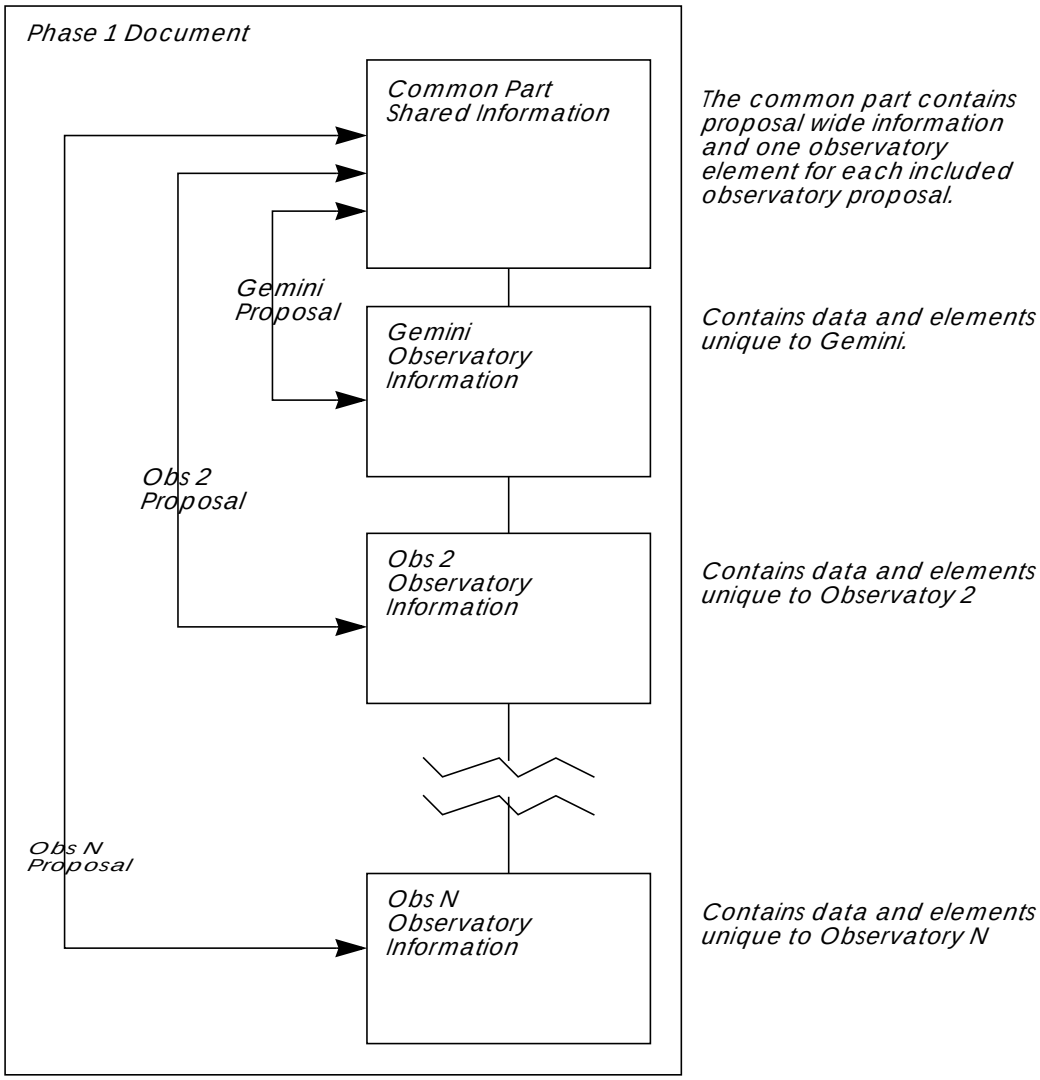
It is impossible to create a single document that has a set of elements that are broad enough to satisfy every observatory. Therefore, the Phase 1 document must be extensible to allow observatories to add private information as needed. This capability is provided through two optional types of elements: constraints and extensions. Each `observatory` can add any number of `extension` and `constraint` elements.

Elements that add observatory specific content are called extensions in the Phase 1 document. As an example, Gemini defines an element called `geminiSubDetailsExtension` that indicates the partner country that is receiving the proposal and whether or not the proposal is being submitted to multiple partners.

Constraints can also add observatory specific content, but are used differently. A constraint is an element that is tied to a single observation or to the entire observation list—its function is to constrain how the observation is executed. Gemini defines a constraint called `geminiObsConditionsConstraint` that describes the observing conditions the proposer wishes for the observations. The constraint can apply to the entire observation list or a single observation.

Generally, an observatory's constraints and extensions are of no use or interest to other observatories, and should not depend on or use any elements that are not also observatory specific.

FIGURE 1. The document is a common element and one element for each included observatory proposal.



2.2 The Common Element

The `common` element includes elements that either are or can optionally be shared by all the proposals using the document. These elements include:

- `title`, `abstract`, and `scienceJustification` with attachments. These elements are required, but attachments are optional.
- `keywords`, which help the proposal processing and review process. The keywords content includes a category with one or more keywords. The shared set of categories and keywords are listed in an associated XML-based data file that is used by authoring tools. Keywords is an optional item.
- `investigators` includes one required principal investigator with complete information and zero or more co-investigators and their contact information.
- the `targetCatalog`, which contains the positions of all the objects that can be used by the proposals in the document. The coordinate information for each target can be specified in a number of ways with varying levels of detail.

2.3 The Observatory Element

As shown in Figure 1, there is one `observatory` element for each proposal in the document. The observatory element is designed as a group of optional features that an observatory can use if appropriate. The required observatory elements are:

- a `site` element specifying the address of the observatory and observatory contact information.
- an `obsList` element that includes zero or more observations along with the constraints and resources used by the observations.

The following elements can be optionally included in the observatory element.

- The observingMode element states whether the proposal is queue or classical.
- The requirements element describes things the proposer needs from the observatory such as support from the observatory staff.
- The `proposalSupport` element contains items relating to publications and previous allocations that support the current proposal request.
- The `scheduling` element contains a number of optional items related to preferences or requirements for scheduling the proposal.
- `techicalJustification` describes how the proposal science can be accomplished with the resources of the observatory.
- The `resourceList` is a complete list of the resources (instruments, etc.) that are needed to complete the proposal at the observatory.
- The `extension` and `constraint` elements describe any other observatory information. For instance, an observatory might allow the proposer to request a particular room and board plan or specify the observing conditions for the observation list.

2.4 Validating a Proposal

In XML, a distinction is drawn between the notions of *well-formed* and *valid*. To be well-formed, a document must only obey a few rules about document structure such as always providing a closing tag:

```
<item>Item content here</item>
```

A well-formed document is required to nest elements properly and have a single root element. To determine if a document is well-formed requires only that it obey the well-formedness rules.

If a document is well-formed *and* valid, it is valid with respect to a schema, which is defined in terms of a Document Type Definition or DTD. This means that the structure (and content to a degree) are valid with respect to a specific DTD. A document can easily be well-formed but not valid according to a specific DTD.

The following example shows the beginning, or prolog, of a Phase 1 XML document that is written to be validated with respect to the AstronomyPhase1 DTD.

DTD Fragment 1: A prolog with two included observatory DTDs.

```
1 <?xml version="1.0" ?>
2 <!DOCTYPE phase1Document PUBLIC
3    "-//Gemini 8-m Telescopes Project//DTD for astronomy phase 1//EN"
4     "AstronomyPhase1.dtd" [
5 <!ENTITY % gemini PUBLIC
6    "-//Gemini 8-m Telescopes Project//DTD for Gemini phase 1//EN"
7    "Gemini.dtd">
8 %gemini;
```

```
 9 <!ENTITY % obs2 PUBLIC
10    "-//Observatory 2//DTD for Observatory 2 phase 1//EN"
11    "Observatory2.dtd">
12 %obs2;
13 ]>
```

The document is associated with the AstronomyPhase1 DTD in lines 2-4. The root element is always called `phase1Document`. The ENTITY in lines 5-9 is an external parameter reference that points to the DTD that describes the Gemini additions to the proposal. The DTD locations can be local files as shown or URLs, allowing the XML processing program to fetch the DTD from a web server. The Gemini definitions are included in line 8. A second observatory called observatory2 is defined and included in lines 9-12. The process is repeated whenever a new observatory is added to the proposal.

Checking for validity of the document is important in some circumstances, but using a DTD also has its drawbacks. Phase 1 Proposals should be stand-alone documents. Bundling each document with a set of DTDs is not practical. Therefore, the prolog must be written such that the external parameter references use URLs. Then the problem is that whenever a document is to be opened, the DTD must be available requiring a network connection. This is also impractical.

The suggested use for the DTD is that authoring tools write well-formed documents with no DTD references. The documents can be validated against a DTD when submitted to an observatory if necessary, or a service could be provided to check for validity. This is not a loss. Authoring tools used to create and edit Phase 1 documents must examine a document more carefully than an XML parser can in order to do semantic checking of the content. In the case of no DTD references, the prolog is simple.

```
<?xml version="1.0" standalone="yes"?>
```

The `standalone="yes"` attribute indicates to the XML processor that no DTDs are needed to process the following document.

2.5 XML References

Each observatory defines its proposal in an `observatory` element. This element contains XML references to targets and resources defined in both the `common` element of the document and in observatory-specific elements. The connections to elements such as `targets` are made through XML ID references. References allow one element to refer to another element in the document. The concept of references is illustrated in the following proposal excerpt, Example 1.

Example 1. The use of references in an XML document.

```
<phase1Document>
  <common>
    …
    <targetCatalog>
      <target id=”ref-1” type=”science”>
        <targetName>NGC90210</targetName>
        <hmsdegSystem type="J2000">
            <c1>13:37:05.12</c1>
            <c2>-31:23:13.2</c2>
          </hmsdegSystem>
      <target id="ref-2" type="wfs">
        <targetName>1023456</targetName>
        <hmsdegSystem type="J2000">
            <c1>13:37:15.32</c1>
            <c2> 31:24:0.2</c2>
          </hmsdegSystem>
      </target>
    </targetCatalog>
  …
  </common>
  <observatory observatoryId=”gemini”>
    …
    <obsList>
      <constraintRef constraintId=”gemref-4”/>
      <resourceRef resourceId=”geminiNorth”/>
      <observation>
        <constraintRef constraintId="gemref-15"/>
        <targetRef targetId=”ref-1”/>
        <targetRef targetId=”ref-2”/>
        <expTime units=”hours”>1.2</expTime>
        <expTotalTime units=”hours”>1.5</expTotalTime>
      </observation>
    </obsList>
    …
  </observatory>
  …
</phase1Document>
```

This example shows a fragment of the `obsList` that is part of a Gemini `observatory` element. The `obsList` shows an `observation` that references two targets: a science object, “ref-1”; and its guide star, “ref-2”.The circled `targetRef` element references the `target` element with the `id` of “ref-1”. Since the targets are included in the `targetCatalog` of the `common` element, they are available to all observatory proposals in the document.

Authoring tools must guarantee that `ID`s are unique within the document. One approach is that each observatory begin their `ID`s with a shortened version of their observatory name. However, the only requirement is that the `ID`s be unique and the references be correct.

This linkage between `target` and `targetRef` is setup in the DTD as follows in the trimmed down version of element `target`.

```
1 <!ATTLIST target
2   id ID #REQUIRED
3   (parts excluded)
4 >
```

The “ID” type indicates the value of `id` must be a unique reference. Then, in the `targetRef` element used in the `observation`, the `targetId` attribute is defined to be type “IDREF”. The parser guarantees that `targetID` contains the name of a valid, unique ID.

```
1 <!-- A targetRef is used to refer to a target in the targetCatalog -->
2 <!ELEMENT targetRef EMPTY>
3 <!ATTLIST targetRef targetId IDREF #REQUIRED>
```

3.0 DTD Description

The only way to understand the document format is to examine the DTD line by line along with an example document. The full DTD and example document are attached at the end of this paper. The DTD can be found on page 19. The Gemini DTD is on page 29. The example proposal document includes one proposal for Gemini North. page 32.

The DTD should serve as the documentation for the Phase 1 document format. For this reason, it is heavily commented. Creating a detailed discussion of the DTD here is problematic because as soon as the DTD changes the document describing the DTD must be painstakingly examined. Therefore, the approach here is to examine key features and constructs in the DTD by showing sections of an example proposal. The more simple elements are not described here and can be found in the DTD description.

3.1 The Phase1Document Element

Every XML document must have a root element that tells the parser where to start. When using a DTD, the DOCTYPE tag tells the parser the root element. When no DTD is referenced, the parser starts at the first tag. The root element for the Phase XML document is called `phase1Document`.

DTD Fragment 2: The phase1Document root element.

```
1 <!ELEMENT phase1Document (common, observatory*)>
2 <!ATTLIST phase1Document
3           dtdVersion   CDATA #FIXED "&PHASE1_DTD_VERSION;"
4           created      CDATA #IMPLIED
5           lastModified CDATA #IMPLIED
6 >
```

The phase1Document element consists of a `common` element, and zero or more `observatory` elements. Without an `observatory`, there is no complete proposal. A document with no `observatory` could be a useful template.

A namespace attribute will be added to `phase1Document` at a later time. Namespaces are a W3C recommendation [8], but do not work well with the use of DTDs making validation of documents difficult. Future schemas, will allow both validation and the use of namespaces.

An authoring tool is required to write attributes into the element as is shown in lines 3-5 above. The attributes specify the version of the DTD used to create the document, the date on which the document was created, and the date of the last modification. The DTD version is available as an ENTITY in the DTD named PHASE1_DTD_VERSION. Here’s an example.

Example 2. The phase1Document element in an example document.

```
<phase1Document dtdVersion="1.1" created="1999-05-21"
      lastModified="1999-07-16">
```

The dates are strings that should be written as YYYY-MM-DD. The `lastModified` attribute should be updated whenever an authoring tool changes the document.

3.2 The Common Element

The `common` element contains general proposal information that is common to all the proposals in the document. The `title`, `abstract`, and `scienceJustification` elements are PCDATA, which means they are parsed by the XML parser. As in HTML, instances of characters such as <, >, or & must be replaced with their entities: <, >, and &, respectively. Tools reading the document must return the entities back to their actual values. If this isn't done, even a parser that is only checking for well-formedness will object and generate an error.

DTD Fragment 3: The `common` element.

```
<!ELEMENT common
          (title,
           abstract,
           scienceJustification,
           keywords?,
           investigators,
           targetCatalog)>
```

The `targetCatalog` consists of a list of `target` elements. For most Phase 1 applications, only a right ascension and declination are required. The Phase 1 DTD provides more involved coordinates, if needed. The following fragment shows the definition of a `target` element.

DTD Fragment 4:

```
<!ELEMENT target (targetName, (namedSystem |
                               conicSystem |
                               hmsdegSystem |
                               degdegSystem |
                               nonSidSystem ))>
<!ATTLIST target
          id ID #REQUIRED
          type ( science | guide | wfs | oiwfs ) "science"
>
```

A `target` is a `targetName` followed by a coordinate system. A target has two attributes: `id` and `type`. The `id` is used in the `observation` elements to refer to the target. The `type` is a hint to authoring tools indicating how the target is to be used. Some authoring tools may choose to only view targets of a particular `type`. The available coordinate systems are briefly described here.

namedSystem. A target using this coordinate system is identified only by its name, which is well known.

hmsDegSystem. A coordinate system where the first coordinate is given in hours, minutes, and seconds. The second is in degrees, minutes, and seconds.

degdegSystem. A coordinate system where both coordinates are given in degrees, minutes, and seconds.

nonSidSystem. A low accuracy coordinate for indicating the position of a non sidereal target. The position is given for a particular date.

conicSystem. The position of the target is given by its orbital elements.

Most applications will be satisfied with the simple forms of `hmsDegSystem` and `nonSidSystem`. The others are provided for completeness at the request of reviewers.

3.3 Extension and Constraint Elements

Observatory specific information is included in the proposal through the use of `constraint` and `extension` elements. Constraints are referenced through `constraintRef` elements that can appear in the `obsList` and `observation` elements.

- The `extension` elements describe additional observatory information. For instance, an observatory might allow the proposer to request a particular room and board plan. The observatory supports zero or more extensions.
- The `constraint` elements contain observatory specific information that must be tied to the entire list of observations or individual observations. For instance, Gemini has a constraint that allows the observation list or individual observations to be tied to a specific set of observing conditions.

XML provides a way to include unknown elements without restrictions through the ANY keyword. ANY allows a well-formed fragment of elements to be added to a document while retaining the ability to validate the document with the DTD. Only the observatory writing the extension or constraint knows how to use the contained information so the use of ANY is not a detriment. The use of ANY provides a flexible way to allow unknown data to be included in the document.

The following DTD fragments show the `extension` and `constraint` elements.

DTD Fragment 5: The `extension` and `constraint` elements.

```
<!ELEMENT extension ANY>
<!ATTLIST extension
          type  CDATA #REQUIRED
>

<!ELEMENT constraint ANY>
<!ATTLIST constraint
          id    ID    #REQUIRED
          type  CDATA #REQUIRED
          name  CDATA  "no name"
>
```

The attribute `type` provides identification for the extension or constraint. The attribute `id` is used to refer to the constraint in the `constraintRef` element. The `name` attribute provides a way to identify a particular instance of a constraint. Extensions appear only once in a document and don't need a `name` attribute.

Constraints and extensions are constructed in the same way within a document. The following example shows the `geminiObsConditionsConstraint` enclosed by a `constraint` element.

Example 3. An example `constraint` element.

```
<constraint type="observingConditions" name="Global Default"
  id="gem-ref-4">
  <geminiObsConditionsConstraint imageQuality="any"
       skyBackground="any" waterVapor="any" cloudCover="50"/>
```

```
5 </constraint>
```

When this fragment is validated, the parser checks for the required `id` and `type` attributes, but it only checks that the elements within the constraint are well-formed. The content and elements within an `extension` or `constraint` is entirely up to the creators of the `observatory` element. The following example shows the `geminiTACExtension`, used to contain partner TAC information.

Example 4. An example `extension` element.

```
1  <extension type="tac">
2    <geminiTACExtension>
3      <partnerReferenceNumber>100010001A</partnerReferenceNumber>
4      <partnerRanking>22</partnerRanking>
5      <partnerRecommendedTime units="nights">4</partnerRecommendedTime>
6      <partnerReceivedDate>
7        <date><year>2000</year><month>1</month><day>22</day></date>
8      </partnerReceivedDate>
9    </geminiTACExtension>
10 </extension>
```

The `extension` and `constraint` differ only in that `constraint` includes a `name` and `id` attribute. Constraints can be referred to by multiple `observation` elements, which requires they have an `id`.

3.4 The ResourceList Element

Each observatory provides a set of resources for use by proposers. This Phase 1 document considers a resource to be anything that must be scheduled with the proposal or observations. Examples are telescope facilities, instruments, gratings, filters, or even computer time.

Resources must be defined by the observatory so a flexible implementation is required. The Phase 1 document provides elements that allow the observatory to hierarchically organize an arbitrary collection of resources. The `resourceList` element is a list of `resourceCategory` elements. A `resourceCategory` contains a `resouceType` (instrument, for instance) and a list of `resource` elements.

A `resource` is the element that can be tied to the observations through its `id` attribute. Resources can contain any number of components that have types and names through the `resourceCompType` and `resourceCompName` elements. An example follows the DTD fragment.

DTD Fragment 6: The `resourceList` element.

```
1  <!ELEMENT resourceList (resourceCategory)*>
2
3  <!ELEMENT resourceCategory (resourceType, resource*)>
4  <!ELEMENT resourceType (#PCDATA)>
5
6  <!ELEMENT resource (resourceName, resourceComp*)>
7  <!ATTLIST resource
8            id ID #REQUIRED
9  >
10
11 <!ELEMENT resourceName (#PCDATA)>
12 <!ELEMENT resourceComp (resourceCompType,
13           (resourceComp* | resourceCompName*))>
14
15 <!-- The component type is something like "Camera" or "Filter". -->
16 <!ELEMENT resourceCompType (#PCDATA)>
17 <!-- The resourceCompName is something like "blue" or "GG485". -->
```

```
18 <!ELEMENT resourceCompName (#PCDATA)>
```

The assumption is that each observatory provides a data file containing a `resourceList` of its available resources for use during a scheduling period. The proposer selects a subset of the available resources to associate with his observations. The selected `resourceList` is inserted into the document as part of the `observatory` element. The `obsList` or individual `observation` elements are tied to the resources in the selected `resourceList`. The following example shows a selected `resourceList`.

Example 5. An example selected `resourceList` element.

```
 1 <resourceList>
 2   <resourceCategory>
 3     <resourceType>Facility</resourceType>
 4     <resource id="geminiNorth">
 5       <resourceName>Gemini North</resourceName>
 6     </resource>
 7   </resourceCategory>
 8   <resourceCategory>
 9     <resourceType>Instrument</resourceType>
10     <resource id="geminiNIRI">
11       <resourceName>Near Infra-red Imager (NIRI)</resourceName>
12       <resourceComp>
13         <resourceCompType>Camera</resourceCompType>
14         <resourceCompName>f/32 (0.02 arcsec)</resourceCompName>
15       </resourceComp>
16       <resourceComp>
17         <resourceCompType>Disperser</resourceCompType>
18         <resourceCompName>R=600 K-band</resourceCompName>
19       </resourceComp>
20       <resourceComp>
21         <resourceCompType>Filter</resourceCompType>
22         <resourceComp>
23           <resourceCompType>Broad-Band</resourceCompType>
24           <resourceCompName>K'(2.150 um)</resourceCompName>
25           <resourceCompName>K_s(2.200 um)</resourceCompName>
26         </resourceComp>
27       </resourceComp>
28     </resource>
29   </resourceCategory>
30 </resourceList>
```

The `resourceComp` elements can be nested to support ownership of components and organization by category. The example shows that the `resource` geminiNIRI contains Camera, Disperser, and Filter `resourceComp` elements. The Filter `resourceComp` contains a nested component of type Broad-Band. Two filters, K' and K_s, are requested for use with NIRI.

3.5 ObsList and Observatory Elements

The `observatory` element ties together all the information needed for a single, scheduled, time allocation at a particular observatory. This is the definition of an `observatory` element in the DTD.

DTD Fragment 7: The `observatory` element.

```
1 <!ELEMENT observatory (site,
2                        obsList,
3                        observingMode?,
4                        requirements?,
```

```
                        proposalSupport?,
                        scheduling?,
                        technicalJustification?,
                        resourceList?,
                        extension*,
                        constraint*)
>
<!ATTLIST observatory
          observatoryId CDATA #REQUIRED
>
```

There is one `observatory` element for every observatory that will receive a proposal. `Observatory` has one required attribute called `observatoryId`. This string is the agreed upon name for the observatory. It is there to allow authoring tools to quickly locate a specific observatory or test whether the observatory exists in the document. Each observatory should define its `observatoryID` in an entity as part of its observatory-specific DTD. However, if the document is being written without DTD references, the text of the entity should be used. For example, in the Gemini DTD:

```
<!-- This entity is used as the observatoryId for the Gemini
     north and south facilities. -->
<!ENTITY geminiId "gemini" >
```

Each `observatory` has two required elements: the `site` element includes name, address, phone, etc. for the observatory. The `site` element also appears in the DTD as the contact information for the PI. The `obsList` element collects the information for the observatory-specific list of observations. Again, the elements are optional. An empty `obsList` makes sense for an observatory that doesn't need the concept of observations and uses only a list of targets.

DTD Fragment 8: The `obsList` element.

```
<!ELEMENT obsList (totalTime?,
                   constraintRef*,
                   resourceRef*,
                   observation*)>
```

The `totalTime` element is present in the `obsList` element to contain the total amount of time requested in the proposal. It is the sum of all the `expTotalTime` elements in the observations or, for classical proposals, it is the total number of time requested.

The `constraintRef` list and `resourceRef` list elements contain references to any `constraint` elements and resources, respectively, that are tied to the entire list of observations.

The `observation` element is defined in Fragment 9.

DTD Fragment 9: The observation element.

```
<!ELEMENT observation (constraintRef*,
                       resourceRef*,
                       targetRef*,
                       expTime?,
                       expTotalTime?)>
```

An `observation` consists of optional items. Although any number of `targetRef` elements are allowed in the `observation` element, it's hard to imagine an observation with zero targets, but it is allowed. The `observation` is tied to the targets through the ID reference mechanism discussed in a previous section. `Targets` have a `type` attribute that takes on values such as "science" or "wfs" to indicate how the `tar-`

get is to be used. For instance, a Gemini observation may need to be tied to one or more wave front sensor stars.

If needed, an observation can be tied to a set of resource elements in the selected resourceList defined elsewhere in the observatory element. These could be anything an observatory treats as a resource. For instance, a proposal using queue observing could tie each observation to a different instrument configuration. This kind of capability is not yet used in current Phase 1 implementations but could be useful in the future.

The observation can include a shutter open estimate (expTime) as well as the open time plus observatory overhead (expTotalTime), which is calculated by the authoring tool. Both are optional so an observatory that doesn't wish to bother with this information is not required to support it.

An observation can be tied to any number of constraints through a list of constraintRef elements. The constraintRef uses the same ID/IDREF mechanism as the targetRef. However, constraint elements and the selected resourceList are defined within the observatory element.

Example 6 shows a part of a Gemini observatory section. The site content can be added from information in the observatory data file. An observing conditions constraint called "gem-ref-12" applies to the entire obsList. There is a more specific site quality constraint in the first observation in line 18. There are references to three targets in the first observation and one in the second.

Example 6. An observatory showing constraints and extensions.

```
1  <observatory observatoryId="gemini">
2    <site>
3      <institution>Gemini Observatory Northern Operations</institution>
4      …
5    </site>
6    <obsList>
7      <constraintRef constraintId="gem-ref-12"/>
8      <resourceRef resourceId="geminiNorth"/>
9      <resourceRef resourceId="geminiNIRI"/>
10     <observation>
11       <constraintRef constraintId="gem-ref-15"/>
12       <targetRef targetId="ref-0"/>
13       <targetRef targetId="ref-4"/>
14       <targetRef targetId="ref-5"/>
15       <expTime units="hours">3.0</expTime>
16       <expTotalTime units="hours">3.25</expTotalTime>
17     </observation>
18     <observation>
19       <targetRef targetId="ref-2"/>
20       <expTime units="hours">2.0</expTime>
21       <expTotalTime units="hours">2.25</expTotalTime>
22     </observation>
23   </obsList>
24   … resourceList would be here
25   <extension type="tac">
26     <geminiTACExtension>
27       <partnerReferenceNumber>100010001A</partnerReferenceNumber>
28       <partnerRanking>22</partnerRanking>
29       <partnerRecommendedTime units="nights">4</partnerRecommendedTime>
30     </geminiTACExtension>
31   </extension>
32   <constraint type="observingConditions"
```

```
33                name="Global Default" id="gem-ref-12">
34     <geminiObsConditionsConstraint imageQuality="any"
35          skyBackground="any" waterVapor="any" cloudCover="any"/>
36   </constraint>
37   <constraint type="observingConditions"
38                name="Best Cond" id="gem-ref-15">
39     <geminiObsConditionsConstraint imageQuality="20" skyBackground="20"
40          waterVapor="20" cloudCover="20"/>
41     </constraint>
42 </observatory>
```

The `resourceRef` elements in lines 8-9 refer to the `resourceList` example in Example 5. The set of observations are to be done with the NIRI instrument on the Gemini North telescope.

An example extension is shown in lines 25-31. Gemini has added an element to contain TAC information provided by the partner country submitting the proposal.

Following in lines 32-41 are two instances of the observingConditions constraint. The constraint, “Global Default”, applied to the entire obsList (referenced in line 7), is very general indicating that any conditions are acceptable. The constraint called “Best Cond”, requiring the best conditions available, is tied to the first observation in line 11.

4.0 Authoring Tools

The DTDs contained in this document were developed using the Java-based xml4j package available at http://alphaworks.ibm.com. A similar package exists at the Javasoft home page: http://www.javasoft.com. Parsers exist for C and other languages (see references). This document also assumes that the parser supports the Document Object Model (DOM level 1) [1]. This feature provides a common programmer interface in the parser that allows accessing the parsed document tree as well as modifying the document and creating new elements.

Two types of tools are needed to process Phase 1 documents. Authoring tools allow the observer to create new proposals and modify old ones. Back-end tools exist at an observatory site. They are primarily used to examine and extract information from the proposals; generally, for insertion into database applications. Both of these tools use an XML parser to access the document. The state of XML tools is very good and improvements and new options appear at a rapid rate.

4.1 Tool Support is Required for Proper Documents

XML deals with document structure well, but it can not understand the semantics of the content within tags. The authoring tools must ensure that semantic rules of the document are correct. XML calls these rules *validity constraints*. The AstronomyPhase1 DTD is augmented with text comments instructing tool programmers on what rules they must enforce. `Observatory` element creators should also add validity constraints to their DTDs. The following fragment shows a validity constraint placed upon the keywords.

DTD Fragment 10: Keywords must be valid.

```
1 <!ELEMENT keywords (keyword+)>
2 <!ATTLIST keywords
3           category (none|solarSystem|galactic|extraGalactic) "none"
4 >
5
6 <!--VC: keyword value should a keyword for the category. -->
7 <!ELEMENT keyword (#PCDATA)>
```

```
8
```

The `keywords` element consists of a `category` attribute and one or more `keyword` elements. The set of possible keywords comes from the AstronomyPhase1 data file. They are just PCDATA so they can be easily changed.

With the help of the DTD, the parser can detect if at least one `keyword` is present in the `keywords` element, but it can't tell that the `keyword` belongs to the selected `category`. It can only tell that the keyword is valid PCDATA. That restriction is a semantic rule that should be ensured by authoring tools. The validity constraints do not influence back-end tools unless they modify or create the XML document.

4.2 Associated XML Data Files

Authoring tools read, write, or modify the Phase 1 document. Most need to display observatory-specific information as well as shared, database information such as keywords and site address information. Rather than hard-coding this information into the authoring tools, this information should be made available as data files with names and root elements that are predictable allowing sharing and hopefully, standardization.

The rule for locating a data file is to append "Data" to the DTD name. For instance, the Phase 1 document DTD is named AstronomyPhase1.dtd. The associated data file is known as AstronomyPhase1Data.xml. The root element of the data file is known as `astronomyPhase1Data`. It is defined in the AstronomyPhase1 DTD as follows.

```
1 <!ELEMENT astronomyPhase1Data (keywords+, site*)>
2 <!ATTLIST astronomyPhase1Data
3           lastModified CDATA #IMPLIED
4           version CDATA #FIXED "1.1"
5 >
```

Placing information in a data file allows information to be added by non-programmers. The AstronomyPhase1Data.xml file is distributed with AstronomyPhase.dtd.

Most observatories have, at minimum, resources and site information. Each observatory needs to be able to manage their own observatory content. Each observatory DTD can have an associated data file with a name following the same rule as in the previous paragraph. For example, the Gemini DTD is called Gemini.dtd, and the Gemini data file is called GeminiData.xml. The root element in the data file is `geminiData`.

The content of the observatory data file is largely up to the observatory, but it must be defined apart from the common DTD in the observatory DTD. This is the current description of the GeminiData document. It includes a `resourceList` element that holds Gemini's available resources. This file will be updated for each scheduling period.

DTD Fragment 11: The `GeminiData` element definition.

```
1 <!ELEMENT geminiData (astronomyPhase1Options,
2                       submissionDetails,
3                       resourceOverheads,
4                       resourceList)>
5 <!ATTLIST geminiData lastModified CDATA #REQUIRED>
```

Authors must ensure that all element and entity names defined in an observatory DTD are unique (e.g. geminiPartners) until XML namespaces [9] are used some time in the future.

4.3 Authoring Tool Support

DTD Fragment 11 shows the geminiData element includes elements `astronomyPhase1Options`, `submissionDetails`, and `resourceOverheads`. These options are defined in AstronomyPhase1.dtd for inclusion in observatory data files as a suggested way to provide necessary support for authoring tools.

The `submissionDetails` element contains information an observatory uses to characterize a proposal scheduling period. `submissionDetails` includes `semesterTitle`, `semesterLimits`, and `dueDate` elements. The `semesterLimits` is the start and end of the scheduling period. The `dueDate` is the date on which proposals are due. Also included is a list of URLs authoring tools can use to submit or fetch proposals. The content of the URLs is entirely up to each observatory since each observatory's backend software is different.

The `resourceOverheads` element provides information an authoring tool needs to calculate the `expTotalTime` element in the `observation` from the `totalTime`. A list of overheads is provided; each tied to a specific `resource` in the available `resourceList`. Authoring tools simply use the `id` of resources tied to an `observation` to look up the resource's overhead and calculate `expTotalTime`. Placing this information in the data file allows support staff to easily modify the overhead values.

The astronomyPhase1Options is provided to help authoring tools understand what features are needed to view and edit an observatory's proposals. The following fragment shows the options as a list of boolean attributes.

DTD Fragment 12: The astronomyPhase1Options element.

```
<!ELEMENT astronomyPhase1Options EMPTY>
<!ATTLIST astronomyPhase1Options
          usesKeywords (true|false) "true"
          usesObservingMode (true|false) "true"
          usesProposalSupport (true|false) "true"
          usesScheduling (true|false) "true"
          usesTechnicalJustification (true|false) "true"
          usesResourceList (true|false) "true"
          usesExtensions (true|false) "true"
          usesConstraints (true|false) "true"
>
```

An authoring tool can look for the `astronomyPhase1Options` and possibly alter the user interface based upon the required features.

4.4 Gemini Phase 1 Tool

The Gemini 8m Telescopes Project has produced a Java-based application for the creation and modification of Phase 1 proposals that use the AstronomyPhase 1 XML document. We have provided a Gemini plug-in to allow the creation of Gemini proposals. The application itself is extensible allowing other observatories to add on to the tool in order to create their own proposals based upon the XML document format. The tool is available from the Gemini FTP site (ftp.gemini.edu).

5.0 Conclusions

XML is clearly a standard that will be valid for some time. This report defines a general document format for a multi-observatory Phase 1 document based on XML. The format is flexible and extensible. Processing of the documents is made easier through use of tools that are used and supported outside of the astronomy community. Use of this document would provide Gemini and other observatories with capabilities that are currently not possible.

6.0 References

[1] World Wide Web Consortium, http://www.w3c.org. XML specification and all things happening regarding XML.

[2] *XML Schema Part1: Structures*, http://www.w3.org/1999/05/06-xmlschema-1

[3] *XML Schema Part2: Datatypes*, http://www.w3.org/1999/05/06-xmlschema-2

[4] See http://www.w3.org/Style/CSS for information on CSS use with XML.

[5] *Extensible Stylesheet Language (XSL) Specification*, W3C Working Draft 21 April 1999, http://www.w3.org/TR/WD-xsl.

[6] www.xml.com. This site contains the annotated XML specification.

[7] www.oasis-open.org/cover/publicSW.html. This site contains a comprehensive listing of publicly available software for XML/SGML/DSSSL.

[8] Harold, Elliot Rusty, *XML Extensible Markup Language*, IDG Books, 1998.

[9] *Namespaces in XML*, W3C recommendation, http://www.w3.org/TR/1999/REC-xml-names-19990114, 14 January, 1999.

7.0 The DTDs and Examples

This section includes the AstronomyPhase1.dtd, the Gemini.dtd, and an example Gemini proposal based on the AstronomyPhase1 DTD.

7.1 Phase 1 DTD - AstronomyPhase1.dtd

```
<!--  *****     *****     *****     ******     *****     *****     *****

   Document Type Declaration for Phase 1 Proposal Information.
   This DTD describes a general structure for describing information
   required by astronomy observatories during Phase 1.

   An instance of this DTD is required as part of submitting a proposal to
   the Gemini 8-m Telescopes Project.  The Gemini proposal DTD is more
   restrictive than this DTD.  It is described in Gemini.dtd.

    Authors:  Kim Gillies       (Gemini 8-m Telescopes Project)
              Shane Walker      (Gemini 8-m Telescopes Project)
              Darrell Denlinger (Gemini 8-m Telescopes Project)

    $Id: AstronomyPhase1.dtd,v 1.13 1999/07/29 06:56:06 cvs-tuc Exp $

    For a description of changes between versions see the “CHANGES” document.

    The goals of this DTD are the following:
    1.  To provide a common specification of proposals that will satisfy the
        needs of many observatories.
    2.  To provide a description that can specify a multi-observatory
        proposal in a single document.  This should allow a proposal to be
        written once and sent to more than one observatory while still
        allowing observatory-specific differences.

    Rather than specify a single format that observatories must
    adopt, the DTD describes a flexible document with many features.
    Observatories, can use the parts in the common part of the document and
    then add their own extensions.

    VC: The VC comments are validity constraints.  These are things that
        must be true, but that must be constrained by the tool writing the
```

```
34        proposal, not the DTD.  Generally, these are comments about what
35        the content should look like when it isn't reasonable to specify
36        it in the DTD.  Future schemas should allow many of these
37        restrictions to be handled in the parser.
38 -->
39
40 <!ENTITY DTD_LAST_MODIFIED "1999-8-15 KG/SW/DD">
41 <!ENTITY PHASE1_DTD_VERSION "1.12">
42
43 <!-- Modules for the DTD -->
44 <!ENTITY % Phase1.targets SYSTEM "target.mod">
45
46 <!-- Include Mods -->
47 %Phase1.targets;
48
49 <!--============================================================================-->
50 <!-- Some basic elements and entities used in multiple places. -->
51
52 <!--Most time units should be specified as one of the values in
53     entity timeUnits
54 -->
55 <!ENTITY % timeUnits 'units (nights|hours|minutes|seconds) "nights"'>
56 <!ENTITY % xtimeUnits '(nights|hours|minutes|seconds) "nights"'>
57
58 <!--Each date is a year month and day element-->
59 <!--VC: Applications must ensure that the date content is in the
60         correct format
61  -->
62 <!ENTITY % dateFormat "(year, month, day)">
63 <!--VC: All dates should be in the format YYYY-MM-DD. -->
64 <!ELEMENT date %dateFormat;>
65 <!ELEMENT dateRange (startDate, endDate)>
66 <!ELEMENT startDate %dateFormat;>
67 <!ELEMENT endDate %dateFormat;>
68 <!ELEMENT year (#PCDATA)>
69 <!ELEMENT month (#PCDATA)>
70 <!ELEMENT day (#PCDATA)>
71
72 <!--Comments are associated with several other elements. -->
73 <!ELEMENT comment (#PCDATA)>
74
75 <!--Attachments are optional.  An attachment can be the name of a
76     file.  The types of attachments are restricted.
77     Attributes:
78     name: is the name of the attachment in the text (e.g. Figure 1)
79     src:  is the file name or URL of the attachment (e.g. figure.eps)
80     type: is the type of the attachment (e.g. EPS )
81 -->
82 <!ELEMENT attachment EMPTY>
83 <!ATTLIST attachment
84           name CDATA #REQUIRED
85           src  CDATA #REQUIRED
86           type (rtf|eps|ps|gif|txt|jpg|tex|pdf) #REQUIRED
87 >
88
89 <!--Some items can have attachements or text entered in a program -->
90 <!ELEMENT embeddedText (#PCDATA)>
91
92 <!--Elements like science justification are text and/or attachements. -->
93 <!ENTITY % textOrAttachments "(embeddedText?, attachment*)">
94
95 <!--============================================================================-->
96 <!--  *****     *****     *****     ******     *****     *****     *****
97     PHASE1DOCUMENT ELEMENT
```

```
98     A phase1Document element is the root of every phase1 document.  The
99     document consists of a common part, where all the information that is
100     common to all the proposals is kept, and zero or more observatories.
101     Each observatory goes with one destination for the proposal.
102     Attributes:
103     created:      the date when the document was created (YYYY-MM-DD)
104     lastModified: the date when the document was last modified (YYYY-MM-DD)
105     version:      the version of the DTD used to construct this document
106                   The DTD version is not the CVS version.  The number to the
107                   left of the decimal will be changed if the DTD changes
108                   such that older documents are not compatible.
109 -->
110 <!ELEMENT phase1Document (common, observatory*)>
111 <!ATTLIST phase1Document
112           dtdVersion   CDATA #FIXED “&PHASE1_DTD_VERSION;”
113           created      CDATA #IMPLIED
114           lastModified CDATA #IMPLIED
115 >
116
117 <!--=========================================================================-->
118 <!--  *****     *****     *****     ******     *****     *****     *****
119     COMMON ELEMENT The common element consists of the parts of the document
120     that are the same for all the proposals within the document.
121 -->
122 <!ELEMENT common
123           (title,
124            abstract,
125            scienceJustification,
126            keywords?,
127            investigators,
128            targetCatalog)>
129
130 <!--Title and abstract are plain text -->
131 <!ELEMENT title (#PCDATA)>
132 <!ELEMENT abstract (#PCDATA)>
133
134 <!--The scienceJustification is common to all proposals.  It can be
135     embedded text or one or more attachements. The scienceJustification
136     describes the proposal’s scientific case. -->
137 <!ELEMENT scienceJustification %textOrAttachments;>
138
139 <!--The keywords are defined here.  It is assumed that the set of keywords
140     for each category is available to the Phase 1 viewer from the
141     AstronomyPhase1 data file.  One category is required for the keywords.
142     Attributes:
143     category: keyword category for the proposal
144 -->
145 <!ELEMENT keywords (keyword+)>
146 <!ATTLIST keywords
147           category (none|solarSystem|galactic|extraGalactic) “none”
148 >
149
150 <!--VC: keyword value should a keyword for the category. -->
151 <!ELEMENT keyword (#PCDATA)>
152
153 <!--=========================================================================-->
154 <!--  *****     *****     *****     ******     *****     *****     *****
155     INVESTIGATORS ELEMENT
156     The investigators element contains contact information for the PI and CoIs.
157 -->
158 <!ELEMENT investigators (pi, coi*)>
159
160 <!--The pi must have complete contact information including his
161     associated site information.
```

```
    Attributes:
    status  : pi background
    visiting: is this pi going to the site?
-->
<!ELEMENT pi (name, contact, site)>
<!ATTLIST pi
   status (phd | gradThesis | gradNoThesis | other ) “phd”
   visiting ( true | false ) “false”
>

<!--The co-investigator just has contact information.
    Attributes:
    visiting: is this coi going to the site?
-->
<!ELEMENT coi (name, contact, institution)>
<!ATTLIST coi
   visiting ( true | false ) “false”
>

<!--The investigator’s name. -->
<!ELEMENT name (first,last)>
<!ELEMENT first (#PCDATA)>
<!ELEMENT last (#PCDATA)>

<!--Contact is an email, phone, and fax information.  This can
    be any number of phones, emails, or faxes.  -->
<!ELEMENT contact (email? | phone? | fax?)*>

<!--Other investigator contact information.
    The fax number is optional.  The investigator may have one or
    more email addresses. -->
<!ELEMENT email (#PCDATA)>
<!ELEMENT phone (#PCDATA)>
<!ELEMENT fax (#PCDATA)>

<!--A site contains all the address information for an institution.
    A database of site information is available in the common data file.
    The contact information for the site should be someone at the
    institution who fields general requests for information.  -->
<!ELEMENT site (institution, address+, country, contact)>

<!--The investigator’s institution and address. An address can
    span multiple lines. At the moment the address is not broken
    up into more detail than this although it may be needed in the future.
-->
<!ELEMENT institution (#PCDATA)>
<!ELEMENT address (#PCDATA)>
<!ELEMENT country (#PCDATA)>

<!--============================================================================-->
<!--  *****     *****     *****     ******     *****     *****     *****
   TARGETCATALOG ELEMENT
   targetCatalog is a list of targets (or objects).  The various observatory
   sections reference some or all of the targets.  The target element
   description is in the target.mod module.
-->
<!--The targetCatalog consists of 1 or more targets. -->
<!ELEMENT targetCatalog (target*)>

<!--============================================================================-->
<!--  *****     *****     *****     ******     *****     *****     *****
    OBSERVATORY ELEMENT
    An observatory element contains content related to observing
    some of the targets at a specific observatory.  Observatory-specific
```

```
226     information is present only within the observatory element.  There is
227     one observatory element for each observatory receiving the proposal.
228     The common element is followed by one observatory element for each
229     observatory receiving the proposal.
230
231     The observatory element consists of the site contact information
232     and the list of observations.
233     These elements are following by optional observingMode, requirements,
234     proposalSupport, scheduling, technicalJustification, resourceList, and
235     any observatory extensions and constraints. Some observatories may
236     not use these features.
237 -->
238 <!ELEMENT observatory (site,
239                        obsList,
240                        observingMode?,
241                        requirements?,
242                        proposalSupport?,
243                        scheduling?,
244                        technicalJustification?,
245                        resourceList?,
246                        extension*,
247                        constraint*)
248 >
249 <!--The observatoryId is present to allow authoring tools to easily
250     determine which observatories are present in the proposal.
251 -->
252 <!--VC: observatoryId should be an ENTITY defined by an observatory DTD. -->
253 <!ATTLIST observatory
254           observatoryId CDATA #REQUIRED
255 >
256
257 <!--The obsList element is the observatory-specific list of observations.
258     Tied to the observation list are: an optional totalTime, a list of
259     constraints, and a list of resources.  Either the entire
260     obsList can be tied to constraints or resources or they can be tied
261     to individual observations.
262     An example constraint is a site quality requirement.  At the
263     obsList level this constraint applies to all the observations.
264 -->
265 <!ELEMENT obsList (totalTime?,
266                    constraintRef*,
267                    resourceRef*,
268                    observation*)>
269
270 <!--Some observatories may wish to have their proposers specify an
271     estimate for the total time the obsList should require rather than
272     specify times for each observation.  In this case, totalTime is the
273     sum of the expTotalTime for each observation.
274     This could be used in classical proposals for the requested time.
275  -->
276 <!ELEMENT totalTime (#PCDATA)>
277 <!ATTLIST totalTime
278           %timeUnits;
279 >
280
281 <!--A constraintRef is used to refer to a constraint, which is an
282     observatory-specific element that is tied to an observation.
283     Constraints are defined later in the observatory element.
284  -->
285 <!ELEMENT constraintRef EMPTY>
286 <!ATTLIST constraintRef
287           constraintId IDREF #REQUIRED
288 >
289
```

```
<!-- The resourceRef is used in the obsList to tie an observation to one
     or more resources. Resources are defined in the resources.mod
     module.
-->
<!ELEMENT resourceRef EMPTY>
<!ATTLIST resourceRef resourceId IDREF #REQUIRED>

<!--An observation element associates items that are used for one
    proposed observation.  Each observatory section of the proposal has
    one obsList, which contains zero or more observations.
    An observation can be tied to zero or more constraints and targets.
    Optionally, resources may also be tied to specific observation.
    Observatories may wish to specify and exposure time and a total
    time (exposure + overhead) with each observation.
-->
<!ELEMENT observation (constraintRef*,
                       resourceRef*,
                       targetRef*,
                       expTime?,
                       expTotalTime?)>

<!--A targetRef is used to refer to a target in the targetCatalog in common -->
<!ELEMENT targetRef EMPTY>
<!ATTLIST targetRef
          targetId IDREF #REQUIRED
>

<!--expTime is an optional item in the observation.  It is the observer's
    estimate of the time the observation requires. This could come from
    other online tools such as an integration time calculator.
-->
<!ELEMENT expTime (#PCDATA)>
<!ATTLIST expTime
          %timeUnits;
>

<!--expTotalTime is an optional item in the observation.  It is the
    observer's estimate of the time the observation requires plus any
    observatory overheads. This is usually calculated by the authoring tool
    used to generate the proposal.
-->
<!ELEMENT expTotalTime (#PCDATA)>
<!ATTLIST expTotalTime
          %timeUnits;
>

<!--============================================================================-->
<!--  *****     *****     *****     ******     *****     *****     *****
   Start of optional observatory elements.
-->

<!--============================================================================-->
<!--  *****     *****     *****     ******     *****     *****     *****
    OBSERVING MODE ELEMENT
    Observing Mode can be set to queue or classical.
    Attributes:
    mode: is the proposal queue or classical?
-->
<!ELEMENT observingMode EMPTY>
<!ATTLIST observingMode
          mode (queue|classical) "queue"
>

<!--============================================================================-->
```

```
<!--  *****     *****     *****     ******     *****     *****     *****
    REQUIREMENTS ELEMENT
    Requirements contains items the observer needs from the observatory.
-->
<!ELEMENT requirements (staffSupport?, travelNeeds?)>

<!--staffSupport is used to indicate that a classical observer needs
    staff assistance on site.
    Attributes:
    need: level of supported requested.
-->
<!ELEMENT staffSupport (#PCDATA)>
<!ATTLIST staffSupport
          need (none|firstNight|everyNight) “none”
>

<!--travelNeeds is an observer-specific travel needs or information -->
<!ELEMENT travelNeeds (#PCDATA)>

<!--============================================================================-->
<!--  *****     *****     *****     ******     *****     *****     *****
   PROPOSAL SUPPORT ELEMENT
   proposalSupport is items relating to publications and allocations
   that promote or support the current proposal.
   publication - the name of the publication.
   allocation  - a previous facility use.
   awardedTime - what amount of time was previously awarded.
-->
<!ELEMENT proposalSupport (publication*, allocation*)>
<!ELEMENT publication (#PCDATA)>
<!ELEMENT allocation (allocationReference,
                      awardedTime,
                      percentUsefulTime,
                      comment)>
<!ELEMENT allocationReference (#PCDATA)>
<!ELEMENT awardedTime (#PCDATA)>
<!ATTLIST awardedTime
          %timeUnits;
>
<!ELEMENT percentUsefulTime (#PCDATA)>

<!--============================================================================-->
<!--  *****     *****     *****     ******     *****     *****     *****
   SCHEDULING ELEMENT
   Proposal time allocation information occurs in the scheduling element.
   This element contains a number of optional items related to
   scheduling the proposal.
-->
<!ELEMENT scheduling
          (minimumUsefulAllocation?,
           futureTimeRequirements?,
           optimalDates?,
           impossibleDates?,
           syncObservingDates?)
>

<!--minimumUsefulAllocation can be used to specify the minimum amount of
    useful time for the proposal.  This is usually found in classical
    proposals.
-->
<!ELEMENT minimumUsefulAllocation (#PCDATA)>
<!ATTLIST minimumUsefulAllocation
          %timeUnits;
>
```

```
418
419 <!--futureTimeRequirements indicates that the proposal spans multiple
420     scheduling periods.  The amount of time needed is specified along
421     with the value's units.
422 -->
423 <!ELEMENT futureTimeRequirements (#PCDATA)>
424 <!ATTLIST futureTimeRequirements
425           %timeUnits;
426 >
427
428 <!--optimalDates indicate when the observer would prefer the proposal to
429     be scheduled. Everything other than impossibleDates are still okay.
430 -->
431 <!ELEMENT optimalDates
432      (dateRange*)>
433
434 <!--impossibleDates are dates that can not be used for the proposal.
435     A comment can be included to give a reason for the impossible dates.
436  -->
437 <!ELEMENT impossibleDates
438      (dateRange*, comment?)>
439
440 <!--This item indicates dates or date ranges when observations must be
441     synchronous with observations at other observatories.  The comment
442     explains the synchronous needs.
443 -->
444 <!ELEMENT syncObservingDates
445      (dateRange*, comment?)>
446
447 <!--=========================================================================-->
448 <!--Technical justification is an optional entry for an observatory.  The
449     technical justification depends on the equipment at a specific observatory
450     and describes how the observatory equipment will satisfy the science goals
451     in the scienceJustification.
452     In includes embedded text or one or more attachements. -->
453 <!ELEMENT technicalJustification %textOrAttachments;>
454
455 <!--=========================================================================-->
456 <!--  *****     *****     *****     ******     *****     *****     *****
457    RESOURCELIST ELEMENT
458    resourceList contains a a list of observatory resources.
459    The list contains observatory equipment.
460 -->
461 <!ELEMENT resourceList (resourceCategory)*>
462
463 <!ELEMENT resourceCategory (resourceType, resource*)>
464 <!ELEMENT resourceType (#PCDATA)>
465
466 <!--  *****     *****     *****     ******     *****     *****     *****
467    Each observatory provides a definition of their available resources
468    in an observatory data file (defined in this DTD).  An authoring tool
469    selects some of those resources and includes them in the observatory
470    element as a resourceList.  A resource has a category (e.g. instrument
471    or facility).  It can also have components (e.g. camera, filter) and
472    the components have values (e.g. GG485 filter).
473 -->
474 <!ELEMENT resource (resourceName, resourceComp*)>
475 <!-- Resources must have an ID that is used to tie resources to
476      observations. -->
477 <!-- VC: resource ID's must be unique within the program.  Authoring
478          tools must generate unique observatory-specific ID's
479          (e.g. gemResource1). -->
480 <!ATTLIST resource
481           id ID #REQUIRED
```

```
482 >
483
484 <!ELEMENT resourceName (#PCDATA)>
485 <!ELEMENT resourceComp (resourceCompType, (resourceComp* | resourceCompName*))>
486
487 <!-- The component type is something like “Camera” or “Filter”. -->
488 <!ELEMENT resourceCompType (#PCDATA)>
489 <!-- The resourceCompName is something like “blue” or “GG485”. -->
490 <!ELEMENT resourceCompName (#PCDATA)>
491
492 <!--=========================================================================-->
493 <!--  *****     *****     *****     ******     *****     *****     *****
494      The extension element wraps elements that are specific to one
495      observatory.  The AstronomyPhase1 allows anything as long as it’s
496      wrapped in an extension.  Extensions contain new observatory-specific
497      information that does not depend upon and is not tied to information
498      in other elements (i.e. no IDs).
499 -->
500 <!ELEMENT extension ANY>
501 <!ATTLIST extension
502           type  CDATA #REQUIRED
503 >
504
505 <!--=========================================================================-->
506 <!--  *****     *****     *****     ******     *****     *****     *****
507      The constraint element wraps elements that are specific to one
508      observatory.  The AstronomyPhase1 DTD allows anything as long as it’s
509      wrapped in a constraint.  Constraints contain new observatory-specific
510      information that is tied to either observation elements or the obsList
511      through ID references.  This allows, for example, resources to be
512      tied to individual observations if needed. The name is text that gives
513      the user a hint about how the constraint is used.
514 -->
515 <!ELEMENT constraint ANY>
516 <!ATTLIST constraint
517           id    ID    #REQUIRED
518           type  CDATA #REQUIRED
519           name  CDATA  “no name”
520 >
521
522 <!--=========================================================================-->
523 <!--  *****     *****     *****     ******     *****     *****     *****
524     PHASE1 DOCUMENT DATA ELEMENT
525     Document element for creating common data for Phase 1 authoring programs.
526     This element should be the the root element of the data file.  The
527     data file contains the list of keywords and a catalog of site data.
528     The data file included in the distribution is called
529     AstronomyPhase1Data.xml.
530 -->
531 <!ELEMENT astronomyPhase1Data (keywords+, site*)>
532 <!ATTLIST astronomyPhase1Data
533           lastModified CDATA #IMPLIED
534           version      CDATA #FIXED “1.1”
535 >
536
537 <!--=========================================================================-->
538 <!--This element must be included in each observatory’s data file so that
539     a tool can learn about proposal submission details.-->
540 <!ELEMENT submissionDetails
541           (semesterTitle,
542            semesterLimits,
543            dueDate,
544            urlData+)
545 >
```

```
546
547 <!--semesterTitle is a text name for the submission period.  Something like
548     “Fall Observing Period”.-->
549 <!ELEMENT semesterTitle (#PCDATA)>
550 <!--semesterLimits indicates the first and last date of the submission
551     period.-->
552 <!ELEMENT semesterLimits (dateRange)>
553 <!--dueDate is the date when all proposals must be submitted.-->
554 <!ELEMENT dueDate (date)>
555
556 <!--  *****     *****     *****     ******     *****     *****     *****
557     Each observatory must have one or more valid urlData entries
558     for submission and fetch of proposals. -->
559 <!ELEMENT submissionUrls (urlData)*>
560
561 <!--Each urlData should consist of at least one submission URL and
562     one retrieval URL. The hint can be used to place information that can
563     identify a specific urlData element -->
564 <!ELEMENT urlData (submitUrl | fetchUrl)*>
565 <!ATTLIST urlData
566           hint CDATA #IMPLIED>
567
568 <!-- A submitUrl has a type (currently we only support Rfc1867) that is
569      used by the submission process to find the correct code to post.
570 -->
571 <!ELEMENT submitUrl (#PCDATA)>
572 <!ATTLIST submitUrl
573           type CDATA “Rfc1867”>
574
575 <!-- A fetchUrl has a type (we currently only support Jar) that is used
576      by the submission process to find the correct code to post.
577 -->
578 <!ELEMENT fetchUrl (#PCDATA)>
579 <!ATTLIST fetchUrl
580           type CDATA “Jar”>
581
582 <!-- This element can be included in an observatory’s data file if the
583      observatory uses the expTotalTime, totalTime feature. overheads
584      is used by the authoring tools to calculate exposure time overheads
585      dependent upon the use of a resource (usually an instrument).
586      If no instrument is present, the default value is used.
587      The overhead id should match the resource id of the instrument.  One
588      id should be called “default”.  This overhead is used when there is no
589      other match.  The resourceId is not an IDREF so that the two are not
590      too tightly coupled.
591 -->
592 <!ELEMENT resourceOverheads (overhead*)>
593 <!ELEMENT overhead (#PCDATA)>
594 <!ATTLIST overhead
595           resourceId CDATA #REQUIRED
596           %timeUnits;
597 >
598
599 <!--=========================================================================-->
600 <!--This element can be included in an observatory’s data file to enable
601     authoring tools to determine which features of the AstronomyPhase1.dtd
602     the proposal uses.
603 -->
604 <!ELEMENT astronomyPhase1Options EMPTY>
605 <!ATTLIST astronomyPhase1Options
606           usesKeywords (true|false) “true”
607           usesObservingMode (true|false) “true”
608           usesProposalSupport (true|false) “true”
609           usesScheduling (true|false) “true”
```

```
610           usesTechnicalJustification (true|false) “true”
611           usesResourceList (true|false) “true”
612           usesExtensions (true|false) “true”
613           usesConstraints (true|false) “true”
614 >
615
```

7.2 The Gemini DTD

```
 1 <!--  *****     *****     *****     ******     *****     *****     *****
 2
 3    Document Type Declaration for Gemini 8-m Telescopes
 4    Phase 1 Proposal Information.
 5
 6    Authors:  Kim Gillies  (Gemini 8-m Telescopes Project)
 7              Shane Walker (Gemini 8-m Telescopes Project)
 8
 9    $Id: Gemini.dtd,v 1.6 1999/07/22 21:51:52 cvs-tuc Exp $
10
11 -->
12
13 <!--  *****     *****     *****     ******     *****     *****     *****
14    Entities that define the Gemini site information.
15 -->
16
17 <!-- This is the version of the Gemini DTD         -->
18 <!ENTITY GEMINI_DTD_VERSION “1.40”>
19
20 <!-- Gemini Telescopes Project Northern Operations -->
21 <!ENTITY geminiNorthInstitution “<institution>Gemini Observatory Northern
Operations</institution>”>
22 <!ENTITY geminiNorthAddress     “<address>670 A Ohoku Place</
address><address>Hilo, HI 96720</address>”>
23 <!ENTITY geminiNorthCountry     “<country>USA</country>”>
24 <!ENTITY geminiNorthPhone       “<phone>808-935-9235</phone>”>
25 <!ENTITY geminiNorthFax         “<fax>808-974-2500</fax>”>
26 <!ENTITY geminiNorthEmail       “<email>jpurcell@gemini.edu</email>”>
27 <!ENTITY geminiNorthSite “&geminiNorthInstitution;
28                          &geminiNorthAddress;
29                          &geminiNorthCountry;
30                          &geminiNorthPhone;
31                          &geminiNorthFax;
32                          &geminiNorthEmail;”>
33
34 <!-- This entity is used as the observatoryId for the Gemini
35      north and south facilities. -->
36 <!ENTITY geminiId “gemini” >
37
38 <!--  *****     *****     *****     ******     *****     *****     *****
39     geminiObsConditionsConstraint ELEMENT
40     This constraint defines the observing conditions needed for the entire
41     obsList or a single observation within the obsList.  See the Gemini
42     web site for the meaning of the numbers for the observing condition
43     attributes.
44 -->
45 <!--VC: A Gemini proposal must include at least one
46          geminiObsConditionsConstraint with the name=”global” in the parent
47         constraint element.  Other geminiObsConditionsConstraints can be added
48          and referenced as needed.
```

```
-->
<!ELEMENT geminiObsConditionsConstraint EMPTY>
<!ATTLIST geminiObsConditionsConstraint
          imageQuality (any|20|50|80) “any”
          skyBackground (any|20|50|80) “any”
          cloudCover (any|20|50|70|90) “any”
          waterVapor (any|20|50|80) “any”
>

<!--  *****     *****     *****     ******     *****     *****     *****
    geminiSubDetailsExtension ELEMENT
    This extension element adds Gemini-specific proposal information
    indicating the source partner country for the proposal.
    If the proposal going to multiple partner country TACs, multiplePartners
    is set to “true”.
    The geminiReferenceNumber and geminiReceivedDate are added by Gemini
    when the proposal is received.
    The semesterTitle, semesterLimits, and dueDate are retrieved from the
    Gemini data file and should be reproduced in the output file.
-->
<!--VC: A Gemini proposal must include at one and only one
        geminiSubDetailsExtension.
-->
<!ELEMENT geminiSubDetailsExtension (semesterTitle?,
                                     semesterLimits?,
                                     dueDate?,
                                     geminiReferenceNumber?,
                                     geminiReceivedDate?)>
<!ATTLIST geminiSubDetailsExtension
          hostPartner (Argentina|
                       Australia|
                       Brazil|
                       Canada|
                       Chile|
                       UK|
                       USA|
                       UH) #REQUIRED
          multiplePartners (true|false) “false”
>

<!-- Entities defining the Gemini partner country names -->
<!ENTITY geminiAR “Argentina”>
<!ENTITY geminiAU “Australia”>
<!ENTITY geminiBR “Brazil”>
<!ENTITY geminiCA “Canada”>
<!ENTITY geminiCH “Chile”>
<!ENTITY geminiUH “University of Hawaii”>
<!ENTITY geminiUK “United Kingdom”>
<!ENTITY geminiUS “United States”>
<!--  *****     *****     *****     ******     *****     *****     *****
    geminiHostPartner should have the value of one of the host counties.
-->
<!--VC: The name of the country should be the value of one of the entities
       shown above.
-->
<!ELEMENT geminiHostPartner (#PCDATA)>

<!--  *****     *****     *****     ******     *****     *****     *****
    geminiTACExtension ELEMENT
```

```
    This extension element adds information added by partner TAC committees
    rating the proposals.
-->
<!ELEMENT geminiTACExtension (partnerReferenceNumber,
                              partnerRanking,
                              partnerRecommendedTime,
                              partnerReceivedDate,
                              partnerComment?)>

<!-- partnerReferenceNumber is a partner-defined string that is unique
     to this proposal.
-->
<!ELEMENT partnerReferenceNumber (#PCDATA)>

<!-- partnerRanking is an integer ranking the proposal with a range of
     1 to the number of proposals.
-->
<!ELEMENT partnerRanking (#PCDATA)>

<!--partnerRecommendedTime the amount of time the TAC suggests should
    be allocated to the proposal.
-->
<!ELEMENT partnerRecommendedTime (#PCDATA)>
<!ATTLIST partnerRecommendedTime
          units (nights|hours) 'nights'
>

<!--partnerReceivedDate indicated when the proposal was received by the
    partner.
-->
<!ELEMENT partnerReceivedDate (date)>

<!-- partnerComment is additional information that can be added by
     the partner country TAC.  It is not required.
-->
<!ELEMENT partnerComment (embeddedText)>

<!-- geminiReferenceNumber is a Gemini specified reference number for the
     proposal.  It's added by Gemini, not the partner TAC.
-->
<!ELEMENT geminiReferenceNumber (#PCDATA)>

<!--geminiReceivedDate indicated when the proposal was received by Gemini.
    It's added by Gemini, not the partner TAC.
-->
<!ELEMENT geminiReceivedDate (date)>

<!--  *****     *****     *****     ******     *****     *****     *****
   geminiData ELEMENT
   This is the root element for the Gemini data file.  The data file is
   used by authoring tools to provide correct data to users.
-->
<!ELEMENT geminiData (astronomyPhase1Options,
                      submissionDetails,
                      resourceOverheads,
                      resourceList)>
<!ATTLIST geminiData lastModified CDATA #REQUIRED>
```

7.3 An Example Phase 1 Proposal Using AstroPhase1.dtd and Gemini.dtd.

```
<?xml version="1.0" standalone="yes"?>
<!--
   Phase1 Document produced by the Gemini Phase 1 Tool
   This document was created on: 1999-07-16
-->
<!--
   This document contains a Gemini 8-m Telescopes Proposal (gemini)
   Semester Title        : 2000-Spring
   Proposal Due Date     : 2000-03-31
    Principal Investigator: PJ Puxley
    Partner Reference     : 100010001A
    Partner Ranking       : 22
    Gemini Reference      : unassigned

    This comment section will change.
-->
<phase1Document created="1999-05-21" lastModified="1999-07-16">
  <common>
    <title>Molecular Hydrogen Excitation in Star-forming Dwarf Galaxies</title>
    <abstract>We propose to observe a small sample of dwarf galaxies.</abstract>
    <scienceJustification>
      <embeddedText>Background
H2 Observations of Galaxies
  Text Removed for inclusion.
</embeddedText>
    </scienceJustification>
    <keywords category="galactic">
      <keyword>Dark Matter</keyword>
      <keyword>Absorption Lines</keyword>
    </keywords>
    <investigators>
      <pi status="phd" visiting="true">
        <name><first>PJ</first><last>Puxley</last></name>
        <contact>
          <email>ppuxley@gemini.edu</email>
          <phone>808-555-1212</phone>
          <fax>808-555-1213</fax>
        </contact>
        <site>
          <institution>Gemini Observatory Northern Operations</institution>
          <address>670 A Ohoku Place</address>
          <address>Hilo, HI 96720</address>
          <country>USA</country>
          <contact>
            <email>jp@gemini.edu</email>
            <phone>808-555-1212</phone>
            <fax>808-555-1213</fax>
          </contact>
        </site>
      </pi>
      <coi visiting="true">
        <name><first>Matt</first><last>Mountain</last></name>
        <contact>
          <email>mmountain@gemini.edu</email>
          <phone>808-555-1212</phone>
          <fax>808-555-1213</fax>
        </contact>
        <institution>Gemini Observatory Northern Operations</institution>
      </coi>
      <coi visiting="false">
        <name><first>Sam</first><last>IYam</last></name>
        <contact>
          <email>sam@roe.ac.uk</email>
```

```
64            <phone>520-555-1212</phone>
65          </contact>
66          <institution>Royal Observatory, Edinburgh </institution>
67        </coi>
68      </investigators>
69      <targetCatalog>
70        <target id=”ref-0” type=”science”>
71          <targetName>NGC5253</targetName>
72          <hmsdegSystem type=”J2000”>
73            <c1>13:37:05.12</c1>
74            <c2>-31:23:13.2</c2>
75          </hmsdegSystem>
76        </target>
77        <target id=”ref-1” type=”science”>
78          <targetName>Haro2</targetName>
79          <hmsdegSystem type=”J2000”>
80            <c1>10:29:22.67</c1>
81            <c2>54:39:30.8</c2>
82          </hmsdegSystem>
83        </target>
84       <target id=”ref-4” type=”wfs”>
85          <targetName>GSC0726501274</targetName>
86          <hmsdegSystem type=”J2000”>
87            <c1>13:36:58.93</c1>
88            <c2>-31:19:05.48</c2>
89          </hmsdegSystem>
90        </target>
91        <target id=”ref-5” type=”wfs”>
92          <targetName>GSC0726501323</targetName>
93          <hmsdegSystem type=”J2000”>
94            <c1>13:37:07.387</c1>
95            <c2>-31:18:45.0</c2>
96          </hmsdegSystem>
97        </target>
98        <target id=”ref-6” type=”wfs”>
99          <targetName>GSC0381600860</targetName>
100          <hmsdegSystem type=”J2000”>
101            <c1>10:29:02.335</c1>
102            <c2>54:44:17.34</c2>
103          </hmsdegSystem>
104        </target>
105        <target id=”ref-7” type=”wfs”>
106          <targetName>GSC0381601019</targetName>
107          <hmsdegSystem type=”J2000”>
108            <c1>10:28:44.474</c1>
109            <c2>54:40:29.5</c2>
110          </hmsdegSystem>
111        </target>
112     </targetCatalog>
113 </common>
114   <observatory observatoryId=”gemini”>
115     <site>
116       <institution>Gemini Observatory Northern Operations</institution>
117       <address>670 A Ohoku Place</address>
118       <address>Hilo, HI 96720</address>
119       <country>USA</country>
120       <contact>
121         <email>jp@gemini.edu</email>
122         <phone>808-555-1212</phone>
123         <fax>808-555-1213</fax>
124       </contact>
125     </site>
126     <obsList>
127       <constraintRef constraintId=”ref-12”/>
```

```
128       <resourceRef resourceId=”geminiNorth”/>
129       <resourceRef resourceId=”geminiNIRI”/>
130       <observation>
131         <constraintRef constraintId=”ref-15”/>
132         <targetRef targetId=”ref-0”/>
133         <targetRef targetId=”ref-4”/>
134         <targetRef targetId=”ref-5”/>
135         <expTime units=”hours”>3.0</expTime>
136         <expTotalTime units=”hours”>3.5</expTotalTime>
137       </observation>
138       <observation>
139         <targetRef targetId=”ref-1”/>
140         <targetRef targetId=”ref-6”/>
141         <targetRef targetId=”ref-7”/>
142         <expTime units=”hours”>2.0</expTime>
143         <expTotalTime units=”hours”>2.5</expTotalTime>
144       </observation>
145     </obsList>
146     <observingMode mode=”queue”/>
147     <requirements>
148       <staffSupport need =”firstNight”/>
149       <travelNeeds>none</travelNeeds>
150     </requirements>
151     <proposalSupport>
152       <publication>Puxley, Doyon &amp;amp; Ward (1996). The spatial distribution of stel-
lar CO absorption in M83, ApJ, 476, 120.</publication>
153
154       <allocation>
155         <allocationReference>UKIRT 97A</allocationReference>
156         <awardedTime units=”nights”>2.0</awardedTime>
157         <percentUsefulTime>100</percentUsefulTime>
158         <comment>Data shown in text, paper in preparation</comment>
159       </allocation>
160     </proposalSupport>
161     <scheduling>
162       <minimumUsefulAllocation units=”hours”>12.0</minimumUsefulAllocation>
163       <impossibleDates>
164         <dateRange>
165           <startDate><year>2000</year><month>8</month><day>5</day></startDate>
166           <endDate><year>2000</year><month>8</month><day>18</day></endDate>
167         </dateRange>
168         <dateRange>
169           <startDate><year>2001</year><month>2</month><day>14</day></startDate>
170           <endDate><year>2001</year><month>2</month><day>27</day></endDate>
171         </dateRange>
172         <comment>Gemini science retreat; SPIE meeting</comment>
173       </impossibleDates>
174     </scheduling>
175     <technicalJustification>
176       <embeddedText> Text not included...</embeddedText>
177       <attachment name=”Fig. 1” src=”figure1.gif” type=”txt”/>
178     </technicalJustification>
179     <resourceList>
180       <resourceCategory>
181         <resourceType>Facility</resourceType>
182         <resource id=”geminiNorth”>
183           <resourceName>Gemini North</resourceName>
184         </resource>
185       </resourceCategory>
186       <resourceCategory>
187         <resourceType>Instrument</resourceType>
188         <resource id=”geminiNIRI”>
189           <resourceName>Near Infra-red Imager (NIRI)</resourceName>
190           <resourceComp>
```

```
191             <resourceCompType>Camera</resourceCompType>
192             <resourceCompName>f/32 (0.02 arcsec)</resourceCompName>
193           </resourceComp>
194           <resourceComp>
195             <resourceCompType>Disperser</resourceCompType>
196             <resourceCompName>R=600 K-band</resourceCompName>
197             <resourceCompName>R=2000 K-band</resourceCompName>
198           </resourceComp>
199           <resourceComp>
200             <resourceCompType>Filter</resourceCompType>
201             <resourceComp>
202               <resourceCompType>Broad-Band</resourceCompType>
203               <resourceCompName>K’(2.150 um)</resourceCompName>
204               <resourceCompName>K_s(2.200 um)</resourceCompName>
205               <resourceCompName>K_l(2.240 um)</resourceCompName>
206             </resourceComp>
207           </resourceComp>
208         </resource>
209       </resourceCategory>
210     </resourceList>
211     <extension type=”subDetails”>
212       <geminiSubDetailsExtension hostPartner=”USA” multiplePartners=”false”>
213         <semesterTitle>First Semester 2000</semesterTitle>
214         <semesterLimits>
215           <dateRange>
216             <startDate>
217               <year>2000</year><month>6</month><day>1</day>
218             </startDate>
219             <endDate>
220               <year>2000</year><month>12</month><day>31</day>
221             </endDate>
222           </dateRange>
223         </semesterLimits>
224         <dueDate>
225           <date><year>2000</year><month>3</month><day>31</day></date>
226         </dueDate>
227         <geminiReferenceNumber>unassigned</geminiReferenceNumber>
228         <geminiReceivedDate>
229            <date><year>2000</year><month>4</month><day>1</day></date>
230         </geminiReceivedDate>
231       </geminiSubDetailsExtension>
232     </extension>
233     <extension type=”tac”>
234        <geminiTACExtension>
235           <partnerReferenceNumber>100010001A</partnerReferenceNumber>
236           <partnerRanking>22</partnerRanking>
237           <partnerRecommendedTime units=”nights”>4</partnerRecommendedTime>
238           <partnerReceivedDate>
239              <date><year>2000</year><month>1</month><day>22</day></date>
240           </partnerReceivedDate>
241           <partnerComment>
242             <embeddedText>We think this proposal should be scheduled.</embeddedText>
243           </partnerComment>
244        </geminiTACExtension>
245     </extension>
246     <constraint type=”observingConditions” name=”Global Default” id=”ref-12”>
247       <geminiObsConditionsConstraint imageQuality=”any” skyBackground=”any” water-
Vapor=”any” cloudCover=”any”/>
248     </constraint>
249     <constraint type=”observingConditions” name=”Best Conditions” id=”ref-15”>
250       <geminiObsConditionsConstraint imageQuality=”20” skyBackground=”20” water-
Vapor=”20” cloudCover=”20”/>
251     </constraint>
252   </observatory>
```

```
253 </phase1Document>
```