TMT Optical Design Key Facts

- Ritchey-Chrétien optical design
- 3.1m convex hyperboloidal secondary mirror
- 30m hyperboloidal f/1 primary mirror w/ 492 segments
- Flat 2.5m x 3.5m tertiary mirror
- f/15 final focal ratio
- Focal surface 20m from tertiary mirror
- 20 arcmin field of view 2.62 m diameter
- Field curvature radius 3.01 m
- 2 Nasmyth platforms hold instruments
TMT Optical Design Key Facts

- Ritchey-Chrétien optical design
- f/15 final focal ratio
- Focal surface 20m from tertiary mirror
- 20 arcmin field of view 2.62 m diameter
- 2 Nasmyth platforms hold instruments

- 3.1m convex hyperboloidal secondary mirror
- 30m hyperboloidal f/1 primary mirror w/ 492 segments
- Flat 2.5m x 3.5m tertiary mirror
Where is TMT?
Operations Plan
End-to-End Science Data Flow

The Observatory Requirements Document (ORD) contains the highest level system requirements for the observatory. Top level requirements are defined for the telescope and instrumentation, summit and support facilities, environmental health and safety, and high level software. The ORD also defines site specific environmental parameters and constraints.

Editors: G. Angel, S. Roberts
Last update: May, 2011

The Observatory Architecture Document (OAD) defines the high-level design for the observatory, including system wide implementation details, and the subsystem decomposition. It partitions function and performance requirements among the subsystems, as necessary, to ensure the integrated systems level performance of the observatory.

Editors: G. Angel, S. Roberts
Last update: December, 2010

The Operations Concept Document (OCD) contains high-level site-independent operations concepts and requirements science and technical operations. Science operations concepts are grouped into a minimum cost baseline service model and a more expensive enhanced service model. It is expected that each major TMT sub-system will have a separate operation concept document that describes operations, calibration, and maintenance processes. The Operations Concept Document (OCD) is currently under significant update to incorporate new developments and refinement of the TMT Operations plans. A new version will be available shortly.

Editors: P. Gray
Last update: April, 2010

Documents available at http://www.tmt.org
Operations Plan
End-to-End Science Data Flow

The Observatory Requirements Document (ORD)
The Observatory Requirements Document (ORD) contains the highest level system requirements for the observatory. Top level requirements are defined for the telescope and instrumentation, summit and support facilities, environmental health and safety, and high level software. The ORD also defines site specific environmental parameters and constraints.

Editors: G. Angel, S. Roberts
Last update: May, 2011

The Observatory Architecture Document (OAD)
The Observatory Architecture Document (OAD) defines the high-level design for the observatory, including system wide implementation details, and the subsystem decomposition. It partitions function and performance requirements among the subsystems, as necessary, to ensure the integrated systems level performance of the observatory.

Editors: G. Angel, S. Roberts
Last update: December, 2010

The Operations Concept Document (OCD)
The Operations Concept Document (OCD) contains high-level site-independent operations concepts and requirements science and technical operations. Science operations concepts are grouped into a minimum cost baseline service model and a more expensive enhanced service model. It is expected that each major TMT sub-system will have a separate operation concept document that describes operations, calibration, and maintenance processes. The Operations Concept Document (OCD) is currently under significant update to incorporate new developments and refinement of the TMT Operations plans. A new version will be available shortly.

Editors: P. Gray
Last update: April, 2010

Documents available at http://www.tmt.org
Operations Plan
End-to-End Science Data Flow

Documents available at http://www.tmt.org
What is TMT Observatory Software?
1. Software architecture for large observatories has converged around a few common ideas and solutions.

2. Reuse known solutions for the low risk areas and focus on aspects unique to TMT.

3. Improve where operations experience with solutions has shown problems or areas for improvement.

4. Some aspects of TMT software complexity scale with aperture size, but a lot of 8m software works fine.
M1 Control System Complexity Scales with Aperture Size

Courtesy: M. Sirota - See M0MK5023
Principal Software Systems

- **Observatory Controls**: Coordinates and parallelizes the activities of the telescope systems, active optics, and instruments during observing using lower tier sequencers.

- **Inst**: Sequences instrument subsystems (one/instrument).

- **Instrument Controls**: Provides all tools needed to capture, store, retrieve and visualize science and engineering data.

- **Telescope Controls**: Provides pointing/tracking kernel for telescope and subsystems. Sequences all telescope subsystems.

- **TCS**: Provides pointing/tracking kernel for telescope and subsystems. Sequences all telescope subsystems.

- **Data Management System**: Sequences AO subsystems.

- **AO Controls**: Sequences AO subsystems.

- **AO**: Sequences AO subsystems.

- Events and Commands
Decomposing into TMT Subsystems
### Areas for Improvement

<table>
<thead>
<tr>
<th>Challenge</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acquisition</td>
<td>There are demanding requirements on the system performance and coordination for target acquisition and observation setup.</td>
</tr>
<tr>
<td>Composition challenges</td>
<td>Some components must be used in a variety of situations or composed in different ways to support different applications. For instance, the MOBIE wavefront sensors must act as part of MOBIE but also work as part of the phasing system.</td>
</tr>
<tr>
<td>Wavefront measurement</td>
<td>Wavefront measurement functionality and hardware is potentially spread throughout telescope systems and instruments making it a challenge to cleanly decompose the system and software. This was a problem with 8m class telescopes as well.</td>
</tr>
<tr>
<td>Distributed development</td>
<td>Multiple, distributed, international partners provides a new level of software construction and management complexity.</td>
</tr>
<tr>
<td>Operations maintenance</td>
<td>The long lifetime of TMT requires an architecture that can be enhanced and modified without impacting the operations system.</td>
</tr>
</tbody>
</table>
## Areas for Improvement

### Table 1: Architecture Challenges

<table>
<thead>
<tr>
<th>Challenge</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acquisition</td>
<td>There are demanding requirements on the system performance and coordination for target acquisition and observation setup.</td>
</tr>
<tr>
<td>Composition challenges</td>
<td>Some components must be used in a variety of situations or composed in different ways to support different applications. For instance, the MOBIE wavefront sensors must act as part of MOBIE but also work as part of the phasing system.</td>
</tr>
<tr>
<td>Wavefront measurement</td>
<td>Wavefront measurement functionality and hardware is potentially spread throughout telescope systems and instruments making it a challenge to cleanly decompose the system and software. This was a problem with 8m class telescopes as well.</td>
</tr>
<tr>
<td>Distributed development</td>
<td>Multiple, distributed, international partners provides a new level of software construction and management complexity.</td>
</tr>
<tr>
<td>Operations maintenance</td>
<td>The long lifetime of TMT requires an architecture that can be enhanced and modified without impacting the operations system.</td>
</tr>
</tbody>
</table>
### Areas for Improvement

<table>
<thead>
<tr>
<th>Challenge</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acquisition</td>
<td>There are demanding requirements on the system performance and coordination for target acquisition and observation setup.</td>
</tr>
<tr>
<td>Composition challenges</td>
<td>Some components must be used in a variety of situations or composed in different ways to support different applications. For instance, the MOBIE wavefront sensors must act as part of MOBIE but also work as part of the phasing system.</td>
</tr>
<tr>
<td>Wavefront measurement</td>
<td>Wavefront measurement functionality and hardware is potentially spread throughout telescope systems and instruments making it a challenge to cleanly decompose the system and software. This was a problem with 8m class telescopes as well.</td>
</tr>
<tr>
<td>Distributed development</td>
<td>Multiple, distributed, international partners provides a new level of software construction and management complexity.</td>
</tr>
<tr>
<td>Operations maintenance</td>
<td>The long lifetime of TMT requires an architecture that can be enhanced and modified without impacting the operations system.</td>
</tr>
</tbody>
</table>
### Areas for Improvement

**Table 1: Architecture Challenges**

<table>
<thead>
<tr>
<th>Challenge</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acquisition</td>
<td>There are demanding requirements on the system performance and coordination for target acquisition and observation setup.</td>
</tr>
<tr>
<td>Composition challenges</td>
<td>Some components must be used in a variety of situations or composed in different ways to support different applications. For instance, the MOBIE wavefront sensors must act as part of MOBIE but also work as part of the phasing system.</td>
</tr>
<tr>
<td>Wavefront measurement</td>
<td>Wavefront measurement functionality and hardware is potentially spread throughout telescope systems and instruments making it a challenge to cleanly decompose the system and software. This was a problem with 8m class telescopes as well.</td>
</tr>
<tr>
<td>Distributed development</td>
<td>Multiple, distributed, international partners provides a new level of software construction and management complexity.</td>
</tr>
<tr>
<td>Operations maintenance</td>
<td>The long lifetime of TMT requires an architecture that can be enhanced and modified without impacting the operations system.</td>
</tr>
</tbody>
</table>
Areas for Improvement

Architecture must allow **flexible integration and construction of components**

software qualities
An observing mode is a well-defined observing task and an associated set of owned resources, procedures, and capabilities.

Observing Mode Example: infra-red diffraction-limited multi-filter integral field spectroscopy

Run as little software as is necessary to execute an observation using a single observing mode.

Eliminate waste code associated with subsystem implementation integration.
The last decade has seen more and more hardware device controllers migrate to network.

Industrial Ethernet-based fieldbus has become common and well-supported.

More and more power and capabilities exist in PLC and PAC systems and their programming.
Hardware Trends

- The last decade has seen more and more hardware device controllers migrate to network.
- Industrial Ethernet-based fieldbus has become common and well-supported.
- More and more power and capabilities exist in PLC and PAC systems and their programming.

TMT will support networked device control via dedicated controllers or PAC/PLC.

Plan is to set standards.
OMOA Structure View
Assembly Example: Device Controller Role
Assembly Example: Composition

Sequencing

Instrument Assembly <<Composition Role>>

Inst System HCD 1
EtherCat Master

Programmable Application Controller (PAC) 1
Hardware

Programmable Application Controller (PAC) 2
Hardware

Enclosure Temperature <<Device Controller Role>>

Temperature I/O HCD 2
I/O HDW Controller Protocol + Transport

I/O-Channel Controller 2
Instrument Input

Enclosure Input
Sequence Layer

- Sequencer is dynamically created for each observing mode

- Software and hardware can be optimally grouped and controlled for each observing mode

- Overall sequencer can consist of multiple focused component sequencers for reuse
Technical and Functional Architecture*
Technical and Functional Architecture*

**Technical Architecture and Design**

Software concepts and infrastructure that provide the foundation for the software system and enables the functional architecture.

**Functional Architecture and Design**

Software concepts, components, and applications that enable the activities of the Observatory from the point of view of the users.

*Adopted from ALMA/ACS and ATST, others*
The integration of these software components requires software infrastructure that is outside the scope of the individual components themselves.

TMT Common Software is the implementation of the technical architecture.

Common Software is a collection of software and services.

A service is a set of related software functionality together with behavior and the policies that control its usage.

The idea of shared software infrastructure based on a set of services and associated software focused on integration is a successful strategy in large observatory software architecture.
## Common Software Integration Services

<table>
<thead>
<tr>
<th>Service</th>
<th>Task</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>User single sign on</strong></td>
<td>Centrally manage user authentication/access control</td>
</tr>
<tr>
<td><strong>Commands</strong></td>
<td>Support for subscribing to, receiving, sending, and completing commands in the form of configurations</td>
</tr>
<tr>
<td><strong>Location/connection</strong></td>
<td>Locate and connect to components within the distributed system</td>
</tr>
<tr>
<td><strong>Events/Telemetry</strong></td>
<td>Enable telemetry/event publishing/subscription</td>
</tr>
<tr>
<td><strong>Alarm/Health</strong></td>
<td>Support monitoring and publishing component alarm and health signals</td>
</tr>
<tr>
<td><strong>Configuration</strong></td>
<td>Manage system and component configuration changes</td>
</tr>
<tr>
<td><strong>Logging</strong></td>
<td>Capture/store logging information</td>
</tr>
<tr>
<td><strong>ODB Access</strong></td>
<td>Common access to Observing Database(s)</td>
</tr>
<tr>
<td><strong>Time</strong></td>
<td>Standards-based, precision time access</td>
</tr>
</tbody>
</table>

A shared software common infrastructure can potentially reduce cost of development, integration, and long-term maintenance.
TMT Software Stack

- Components
  - Component Support
  - User Interface Support
  - CSW Services
  - Package Abstractions and Wrappers
  - Communications Packages
  - Database Packages
  - Other Packages
  - Operating System and Base Hardware
  - Development Tools
  - Supported Hardware
# Event Service Options

- **Base on COTS or Open Source Products and Standards.**
- **100K or more events/second.**

<table>
<thead>
<tr>
<th>Product</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AKKA</td>
<td>Scala-based platform for next generation event-driven, scalable, fault-tolerant architectures on the JVM. <a href="http://akka.io">http://akka.io</a></td>
</tr>
<tr>
<td>DDS</td>
<td>Open Management Group DDS standard (<a href="http://www.omg.org/spec/DDS/1.2">http://www.omg.org/spec/DDS/1.2</a>). RTI company sells this product. In use or proposed for similar role in VLT, EELT, ATST, and LSST. <a href="http://rti.com">http://rti.com</a></td>
</tr>
<tr>
<td>0mq</td>
<td>High-performance, low-level messaging library supporting a variety of message patterns (<a href="http://www.zeromq.org">http://www.zeromq.org</a>). Multi-language support.</td>
</tr>
</tbody>
</table>

- **Starting partner project to evaluate event service options.**
More Choices and Options

<table>
<thead>
<tr>
<th>Product</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time Service</td>
<td>Selected IEEE-1588-2008 - Precision Time Protocol - many vendors</td>
</tr>
<tr>
<td>Configuration Service</td>
<td>Front-end to git or subversion or other version control system for configuration file history and initial values.</td>
</tr>
<tr>
<td>Logging Service</td>
<td>Numerous logging systems are available</td>
</tr>
<tr>
<td>Single sign-on Service</td>
<td>Atlassian Crowd or plain ol’ LDAP.</td>
</tr>
<tr>
<td>Commands</td>
<td>Configuration-based, command-action, HTTP, akka, others.</td>
</tr>
<tr>
<td>Container</td>
<td>OSGi</td>
</tr>
</tbody>
</table>

- Plan is to use Scala and JVM-based languages.
- Reviewing ATST CS and EPICS for use in TMT.
- Linux is inevitable, version is TBD.
Conclusion

- TMT construction is scheduled to start in 2014.
- First light with 492 segments is scheduled for 2021.
- TMT technical architecture is to be set by next year.

Thank-you for listening!