



Open System Engineering Environment

The project has been created. [Please visit the project page.](#)

Open System Engineering Environment (OSEE)

Introduction

The Open System Engineering Environment (OSEE) Project is a proposed open source project under the [Eclipse Technology Project](#).

This proposal is in the Project Proposal Phase (as defined in the [Eclipse Development Process document](#)) and is written to declare its intent and scope. This proposal is written to solicit additional participation and input from the Eclipse community. You are invited to comment on and/or join the project. Please send all feedback to the <http://www.eclipse.org/newsportal/thread.php?group=eclipse.technology.osee> newsgroup.

Background

Each large program that develops systems and software is currently required to assemble a wide array of software products to provide piecemeal capabilities necessary for completing different stages of the engineering lifecycle. Once installed, the daunting task of configuring and gluing them together begins. After years of effort, the program may have a toolset that is adequate for its current needs, but is ill-suited for growth and is far from optimized. In addition, the resulting tools are disconnected, sporadically maintained, and use an overwhelming array of disjointed user interfaces to access partially redundant data. Additionally, such a toolset often carries with it multi-million dollar software licensing, customization, configuration, and maintenance costs.

Despite the tremendous effort involved in piecing together an engineering environment, it is not enough. Lean engineering demands a truly integrated engineering environment. There is an increasing acceptance and adoption of lean principles in product development and engineering. Lean principles focus on reducing waste. A major waste when developing complex systems is handing off work between separate functional disciplines and the associated knowledge loss. Someone who knows this first hand is Jonathan Ive, head of design at Apple, who stated that "The historical way of developing products just doesn't work when you're as ambitious as we are. When the challenges are that complex, you have to develop a product in a more collaborative, integrated way." Engineering a product in a more collaborative and integrated way requires an engineering environment that is itself tightly integrated.

OSEE was designed and architected to provide Subject Matter Experts effective access to the data they need across the engineering lifecycle in a single, integrated environment. OSEE is being developed using a spiral methodology and deployed as those capabilities are needed to develop the next generation mission software for an existing Boeing aircraft. Key deployment milestones of OSEE achieved on this program are listed below.

Deployment Milestones

2004 - Nov Test Environment successfully used to verify requirements in simulated environment
2004 - Dec Deployed OSEE Training Tool to organize, schedule, and track program training
2005 - May Action Tracking System (ATS) used for change management of program tools
2005 - Nov System requirements imported using OSEE Define
2006 - Jan Key users began requirements allocation
2006 - Mar Test Environment successfully used to verify requirements on flight hardware
2006 - May Message Tool deployed to high fidelity integration lab
2006 - Aug Action Tracking System (ATS) used as sole tracking mechanism for development of OSEE
2006 - Aug First Software requirements developed and managed in OSEE Define
2006 - Nov Delivered OSEE and partial program database to US Army customer
2007 - Jan Draft of Prime Item Development Specification (PIDS) published using OSEE for the US Army customer
2007 - Feb Draft of Software Requirements Specification (SRS) published using OSEE for the US Army customer
2007 - Feb Extended Action Tracking System to include build/release planning for OSEE

Scope and Objectives

- Provide an extensible application framework and core services which enables a rich and diverse set of applications to share a common persistence layer providing access control, advanced version control, and dynamic object models through standard Eclipse mechanisms while reducing the heavy burden on applications that such capabilities normally entail.
- Provide exemplary applications which demonstrate how the OSEE application framework and services can be used to create tightly integrated applications.
- Provide an integrated environment for Subject Matter Experts to create and utilize data throughout the engineering lifecycle.
- Allow the engineering lifecycle data to be delivered in an environment that supports sophisticated navigation, visualization, and reporting.
- Expand the Eclipse user community to include the broad spectrum of engineering disciplines.
- Foster an ecosystem involving universities and organizations engaged in engineering of products and the customers of these resulting products.

Description

OSEE provides a tightly integrated environment that supports lean engineering across the full product life-cycle. Since OSEE integrates all engineering areas, the full lifecycle data for a product is managed by a common platform allowing this data to be seamlessly combined to form a coherent, accurate view of a project in real-time. At the heart of the OSEE architecture are the OSEE core services and application framework which are available to all the OSEE applications and enable their tight integration.

The vision for OSEE finds concrete grounding in daily use by Subject Matter Experts developing next generation mission software for an existing Boeing aircraft. This direct involvement with a full-scale deployment of OSEE allows important user feedback to be rapidly incorporated back into both OSEE itself and as lessons learned for the development team.

Since OSEE has been developed for use by separate groups of end users, careful attention has been given to providing a proper separation of core OSEE capabilities and specific extensions made for various groups. The requirement to support separate groups has helped ensure that OSEE is extensible through standard Eclipse mechanisms. The OSEE architecture is designed to provide an integrated, extensible engineering environment that can be adopted by any group engaged in engineering.

The functional description of OSEE is provided in the following listings of its proposed components along with their key capabilities.

Proposed Components

OSEE Application Framework

Key Capabilities

- Dynamic and strongly typed artifact model (persistence layer)
- Bidirectional traceability through typed relations model
- Advanced version control including multi-level branching
- Subject-based and object-based Access Control
- User management and authentication
- Dynamic searching API
- Indexing and tagging services
- Views and editors for the creation, manipulation, and navigation of artifacts, attributes, and relations
- Remote event service for communication and synchronization of OSEE instances
- Rules framework for text processing
- Utilities for plug-in developers
- Scheduling framework
- API for the extension and use of this framework to build tightly integrated applications
- Encryption Utility
- Database Utilities
- Jini Utilities
- Scheduling Service
- Datastore Adaptor

OSEE Test Environment (OTE)

OTE is a powerful test solution within OSEE that integrates with existing Java, C, and C++ development environments to provide a seamless flow between developing, debugging, executing, and dispositioning of tests for complex hardware and software. The user is provided a common interface to the simulated and real-time environments for both functional and unit testing.

Key Capabilities

- Supports the execution of multiple simultaneous batches within a single workspace

- Built-in help system extended with test manager user guide
- Message system supporting MIL-STD-1553 MUX, serial, wire, Ethernet, and Data Distribution Service (DDS)
- Message GUIs provide monitoring, manipulation, and recording of messaging data
- Utilizes OSEE application framework to provide traceability to software requirements
- Automatic generation of tests and testing support classes, directly from requirements.
- Leverages off Java Development Toolkit (JDT) and C/C++ Development Toolkit (CDT)
- Provides remote execution of scripts against target hardware and operating system
- Test results are logged in XML
- Transforms test results via built-in or user supplied XSL Transformations
- Built-in XSL Transformations produce interactive HTML result reports
- A test environment service that provides both soft real-time and simulated capabilities, schedules the periodic execution of simulation components, and manages the I/O and testing resources.

Action Tracking System (ATS)

ATS is a tightly integrated change tracking system that manages changes throughout a product's lifecycle. ATS provides integrated change management to all OSEE applications through user customizable workflows.

Key Capabilities

- Built on same OSEE application framework as requirements, code, test development
- Common Workflow Framework that provides for the creation of any number of simple to complex workflow state machine configurations that can work together during the engineering lifecycle
- Workflows are configured through graphical diagrams that ATS uses at runtime
- Configuration of ATS performed through OSEE's common application framework enabling workflows to be created and modified without separate OSEE releases
- Advanced project planning capabilities and release management
- Duplication errors are minimized as items are automatically linked and data is shared
- Menus, Views and Editors give access to ATS while working in any other aspect in OSEE
- ATS is used to track changes and support issues for the development of OSEE itself
- Bug Icon allows quick Action creation against any OSEE integrated tool

Define

Define provides requirements and process management with tightly integrated change management using the Action Tracking System (ATS). OSEE provides publishing capabilities that enable the creation of documents from smaller internal documents to the complex documents needed to meet military requirements for contract deliverables such as the Software Requirements Specification (SRS), System Performance Specification (SPS), Prime Item Development Specification (PIDS), and all the required traceability between them.

Key Capabilities

- Enterprise support for concurrent, distributed requirements development
- Integrated process and workflow
- Programmatic, bidirectional traceability
- End user navigation and search capabilities
- Capture accurate, meaningful review metrics

- Tight integration with lifecycle tools
- Automated change detection capabilities

BLAM

BLAM Lightweight Artifact Manipulation (BLAM) allows non-programmers to graphically construct workflows to automate repetitive tasks. A given workflow can be used for variety of similar tasks by using customizable controls to specify workflow parameters.

Program/Project Management

Program and project management tightly integrated with the Action Tracking System and other OSEE components to provide services necessary for estimation, planning, execution, and delivery of products managed within OSEE.

Key Capabilities

- Integrated management of charge/cost accounting
- Build planning and execution
- Reporting services
- Rules framework for requiring/alerting certain conditions
- Scheduling services for automating reoccurring tasks

Discovery and Learning

Services provided for allowing advanced learning and discovery using OSEE's abundant and inherent lifecycle data and metrics.

Key Capabilities

- Discovery of inefficiencies in lifecycle processes
- Advanced data mining and data fusion
- Advanced estimating
- Advanced export/import of product capabilities between programs including applicable design, requirements, code, and test
- Advanced data visualization
- Prediction of future risks
- Simulation of recommended process and lifecycle changes

Application Development

Provide capabilities needed for external software application development plug-ins, like JDT, to utilize the OSEE persistence layer and integrate with other OSEE-based applications.

Design and Modeling

Provide capabilities needed for external design and modeling plug-ins to utilize the OSEE persistence layer and integrate with other OSEE-based applications.

Relationship with other Eclipse Projects

The test environment leverages off JDT and CDT. OSEE utilizes Mylar's Zest viewer for 2D data visualization. Additional opportunities exist for integrations between ATS and Mylar. BIRT is an obvious choice for providing graphical representations of metrics produced by ATS. OSEE also utilizes advanced SWT widgets that could be contributed to the Nebula project.

Organization

Mentors

- Mark VandenBrink ♦ Chief Architect for systems software at Motorola
- Richard Gronback ♦ Chief Scientist at Borland Software Corporation and GMF Project Lead

Initial Committers

<u>Name</u>	<u>Organization</u>
Ryan D. Brooks	Boeing: co-project lead
Donald G. Dunne	Boeing: co-project lead
Andrew M. Finkbeiner	Boeing
Jeff C. Phillips	Boeing
Ken J. Aguilar	Boeing
Michael P. Masterson	Boeing
Robert A. Fisher	Boeing
Roberto E. Escobar	Boeing

Interested Parties

<u>Organization</u>	<u>Name</u>
Arizona State University	Huan Liu, Ph.D.
Arizona State University	James S. Collofello, Ph.D.
Arizona State University	Peter Wonka, Ph.D.
Arizona State University	Susan D. Urban, Ph.D.
Auburn University	David Umphress, Ph.D.
Auburn University	Dean Hendrix, Ph.D.
Auburn University	James H. Cross II, Ph.D.
EADS	Andreas Keis
General Motors	Srinivas Chande
Lockheed Martin, Advanced Technology Laboratories	Alexei Samoylov
Rockwell Collins	Paul Streit

Developer Community

We expect to extend the initial set of committers by actively supporting a developer community involving both industry and academia. We will seek to extend the developer community by building upon previous collaboration discussions with our interested parties listed above. We will make it easy for others to take an active role in the project by making our planning and process transparent and remotely accessible.

User Community

The OSEE user community includes both developers building applications on top of the OSEE application framework and end users of existing OSEE applications. The OSEE application framework is currently being utilized by a separate development group to develop aircraft maintenance software. OSEE is also currently used by Subject Matter Experts developing next generation mission software for an existing Boeing aircraft. A Principal Engineer at Leapfrog Enterprises expressed interest in using the capability to create BLAM workflows for automating tasks. The OSEE team will invest effort in helping other organizations, including the interested parties listed, to adopt both OSEE end user applications and the OSEE application framework.

Code Contributions

The Boeing Company is offering a suite of Eclipse features (and their associated plug-ins) corresponding to 5 of the proposed components of this project as the initial code-base, namely: OSEE Application Framework, OSEE Test Environment, Action Tracking System, Define, and Program/Project Management.

Tentative Plan

<u>Release Date</u>	<u>Version</u>	<u>Description</u>
2007 - Jul	0.79.x	Remove dependencies on less friendly licenses
2007 - Aug	0.80.x	Separate into components; refactor into more focused features; ensure proper plug-in architecture and separation
2007 - Sep	0.81.x	Sanitize source code, general code cleanup
2007 - Oct	0.82.x	Initial public source-code commit
2007 - Nov	0.83.x	Add additional extension points; Document extension points
2008 - Jan	0.84.x	Public release for EclipseCon 2008 presentations

Copyright © 2016 The Eclipse Foundation. All Rights Reserved.