



Configuration Description, Deployment and Lifecycle Management Working Group (CDDLIM-WG)

Final Report

Status of This Document

This document provides information to the Grid community. It does not define any standards or technical recommendations. Distribution is unlimited.

Copyright Notice

Copyright © Open Grid Forum 2007. All Rights Reserved.

Trademarks

Open Grid Services Architecture and OGSA are trademarks of the Open Grid Forum.

Abstract

This document is the final report from the CDDLIM working group. It summarizes the purpose, progress and deliverables of the group.

Contents

1. Introduction to the CDDLIM-WG	3
2. History of the CDDLIM-WG	3
2.1 Origins and initial target deliverables	3
2.2 Summary of progress	4
2.3 Next steps	4
3. Inventory of CDDLIM-WG Deliverables	5
3.1 Documents in the formal OGF process	5
3.2 Other documents	5
3.3 Implementations	6
4. Contributors	6
5. Intellectual Property Statement	7
6. Disclaimer	7
7. Full Copyright Notice	7
8. References	8

1. Introduction to the CDDLW-G

The CDDLW Working Group was initiated to develop standards and reference implementations to meet the requirement of deploying potentially complex, distributed software systems on Grid resources. This section provides a brief overview of the CDDLW technology.

As an example, imagine that you have acquired the use of several Grid servers through some Grid resource allocation mechanism and you now wish to deploy a multi-tier web application on these machines. Your complete application could consist of a database tier running on one server, an application server tier running on two servers configured for failover, and a web server tier running on a variable number of servers depending on workload. Assume also that since you will need to relinquish the Grid resources periodically, you want to automate the process of deploying the complete application, and removing it cleanly when done.

CDDLW-G defined a **Configuration Description Language** that is used to describe the desired application. It allows you to specify which software application components (e.g., web servers) are to be deployed, onto which resources and in which order. It allows you to specify the individual configuration parameters for each software component and to link configuration parameters across the set of components that comprise the application. Description files can be reused and employed as templates for other deployments. There are two description language variants, one based on the language from the SmartFrog system [SMARTFROG], the other, called CDL, developed specifically for CDDLW and based on XML.

Application descriptions are passed to the Grid resources via a **Deployment API**. The Deployment API takes in descriptions and realizes the required systems by interacting with the Grid resources to install, configure, start and manage the required software components. It also allows management of deployed systems, including termination.

Finally, the **Component Model** specifies the requirements on CDDLW-deployable software components or *deployment objects*. It defines the interfaces the objects must provide in order to be managed by the CDDLW deployment system, and it defines the capabilities offered to running objects by the deployment runtime. It should be noted that CDDLW deployment objects are typically management wrappers for the actual functional components – for example, a management wrapper for a web server would know how to configure, start and stop the web server.

Taken together, the description language, deployment API and the component model provide a complete framework for automatically deploying complex, distributed applications on Grid resources.

2. History of the CDDLW-G

2.1 Origins and initial target deliverables

The CDDLW-G has its origins in a BoF session held at GGF9 in Chicago in October 2003. The motivation for the BoF was to explore whether ideas being developed in HP Labs around data center automation were of interest to the Grid community. Specifically, the ideas in the SmartFrog (Smart Framework for Object Groups [SMARTFROG]) technology for describing, deploying and managing software systems looked like they could have applicability to Grid standards.

The BoF generated good levels of interest, and a formal working group was approved in November 2003, with Dejan Milojicic of HP Labs and Takashi Kojo of NEC as the co-chairs of

the group. An aggressive time schedule was proposed, with a focus on creating the following deliverables:

1. Configuration Language Specification based on the SmartFrog language
2. Configuration Language Specification using XML syntax (became known as CDL)
3. Basic Services Specification (which became known as the Deployment API Specification)
4. Object Model Specification (which became known as the Component Model Specification)
5. Two, independent interoperable reference implementations of each of the specifications above

Deliverables 1-4 were intended for the OGF standards track, with the reference implementations acting as the required evidence that the standards could be translated into separate operational systems that exhibited compliance with the CDDL standards.

2.2 Summary of progress

Between November 2003 and the end of 2006, the working group made good progress. The group generated and submitted four documents comprising a complete proposed standard, and revised them based on comments.

The group also set out to create a total of four reference implementations, developed by various group participants. The reference implementations were developed to varying degrees of completeness, and three of them were subjected to a range of interoperability tests, the results of which are documented. The scope and complexity of the CDDL standards mean that neither the implementations nor the test coverage can be considered complete, but substantial progress was made.

However, overall, the working group was unable to meet its schedule aspirations. There was more generative work (i.e., new thinking) required than had been expected in taking the SmartFrog concepts and mapping them into the Grid space. Another factor that delayed progress was the shift from OGS to WSRF as the web service standard for the GGF/OGF. This required significant reimplementations. An additional, and perhaps more important factor, was the sheer immaturity and instability of the set of WS standards with which we needed to work, with follow-on consequences for the libraries on which the implementations were expected to depend.

The consequence of things taking longer than expected was that the group lost critical mass as people were assigned to other activities by their sponsoring organizations.

2.3 Next steps

It is our view that the CDDL-WG made interesting contributions and generated significant artifacts that may be of continuing interest to OGF and the Grid community. However, the group is no longer proposing that CDDL become an OGF standard, and our view is that the group should close.

The artifacts generated by the working group will continue to be available via the normal OGF mechanisms, and the CDDL authors will be available to address any questions that arise.

3. Inventory of CDDL M-WG Deliverables

3.1 Documents in the formal OGF process

The following documents are (at the time of writing) part of the formal OGF document process.

Document Title	Description	Status
CDDL M Foundation Document (GFD.50 : INFO)	Introduces the goals, use cases and proposed architecture for CDDL M; sets out the deliverables.	Complete.
CDDL M SmartFrog-Based Language Specification (GFD.51 : P-REC)	Specification of the SmartFrog variant of the configuration language	Complete. Note, however, there is only one implementation of the language processor (from the SmartFrog system itself).
CDDL M Component Model (GFD.65 : P-REC)	Outlines the requirements for creating a deployment object responsible for the lifecycle of a deployed resource	Complete. There are multiple implementations
CDDL M Deployment API (GFD.69 : P-REC)	Specification of the API used to deploy and manage configured applications	Complete. There are multiple implementations and interoperability tests.
CDDL M CDL Specification (GFD.85 : P-REC)	Specification of the XML-based variant of the description language	Complete. There are multiple implementations and interoperability tests.

3.2 Other documents

The following documents have not been injected into the formal OGF document process, but they are available on the CDDL M-WG GridForge site.

Document Title	Description	Status
CDDL M Deployment Test Plan	Test plan for interoperability testing of the Deployment API	Complete
URL	https://forge.gridforum.org/sf/docman/do/downloadDocument/projects.cddl-m-wg/docman.root/doc14946	
CDDL M Deployment API Interoperability Test Results	Results of interoperability testing for the Deployment API implementations from HP, UFCG and NEC.	Complete
URL	https://forge.gridforum.org/sf/docman/do/downloadDocument/projects.cddl-m-wg/docman.root/doc14947	
Testing a Specification: The CDDL M Experience	Covers the different ways that the CDDL M specification was tested during its transformation from a specification to a working implementation.	Complete

URL	https://forge.gridforum.org/sf/docman/do/downloadDocument/projects.cddlmgw/docman.root/doc14949
-----	---

3.3 Implementations

Four implementations were created for the CDDLML-WG specifications.

3.3.1 The HP Labs Implementation

The HP Labs implementation is a complete implementation of the CDL language and deployment API, based on the Java platform and developed under the LGPL open source license. The implementation is available from the SourceForge development site at the following URL:

<http://sourceforge.net/projects/smartfrog>

The HP Labs implementation is capable of accepting deployment descriptors in both CDDLML-CDL and CDDLML-SmartFrog. The deployment components are independent of the syntax used to describe the system configuration, and so can interoperate.

3.3.2 The UFCG Implementation

The computer science department at UFCG (Universidade Federal de Campina Grande, Brazil) developed an implementation, also based on the Java platform. This was done as a collaborative project with HP Brazil R&D. The UFCG implementation is not publicly available at the time of writing.

3.3.3 The NEC Implementation

The NEC implementation is also Java based. It implements the CDL language, the deployment API and the component model. The front-end GUI is all web based; this client has successfully deployed applications to other implementations.

3.3.4 The Softricity Implementation

The Softricity implementation was led by Stuart Schaeffer of Softricity, now part of Microsoft. The implementation was not made public, and did not participate in interoperability testing. The implementation was based on the Microsoft .NET platform. It demonstrated that the language, component model and deployment API was not restricted to the Java platform.

3.3.5 Supporting Test data and XML documents

The teams found their work was aided by collaborating on a set of XML test documents, as well as the XSD and WSDL files of the specifications themselves. All of these files were developed under a BSD-style open source license, and hosted at the SourceForge development site at the following URL:

<http://sourceforge.net/projects/deployment/>

This repository includes an implementation-neutral JUnit test runners for verifying language compliance across all the Java implementations, and a set of valid and invalid SOAP messages to test some aspects of the deployment API.

4. Contributors

The working group would like to recognize and thank the following people for their direct contributions to CDDLML-WG:

Ayla Debora Dantas de Souza
Dejan Milojicic
Edmund Smith
Flavio Roberto Santos
Guilherme Germoglio
Hiro Kishimoto
Jun Tatemura
Keisuke Fukui
Patrick Goldsack
Paul Anderson
Peter Toft
Satish Bhat
Steve Loughran
Stuart Schaeffer
Takashi Kojo

5. Intellectual Property Statement

The OGF takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the OGF Secretariat.

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

6. Disclaimer

This document and the information contained herein is provided on an “As Is” basis and the OGF disclaims all warranties, express or implied, including but not limited to any warranty that the use of the information herein will not infringe any rights or any implied warranties of merchantability or fitness for a particular purpose.

7. Full Copyright Notice

Copyright © Open Grid Forum 2007. All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the OGF or other organizations, except as needed for the purpose of developing Grid Recommendations in which case the procedures for copyrights defined in the OGF Document process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the OGF or its successors or assignees.

8. References

[SMARTFROG] See <http://www.smartfrog.org>