



D-SA2.2.1: VERCE platform integration: Updated release report of integrated services and tools

01/04/2013

Project acronym: VERCE
Project n°: 283543
Funding Scheme: Combination of CP & CSA
Call Identifier: FP7-INFRASTRUCTURES-2011-2
WP: WP6/SA2, Integration and evaluation of the platform services
Filename: D-SA2.2.1.pdf
Author(s): S.H. Leong (BADW-LRZ)
Location: <http://www.verce.eu/Repository/Deliverables/RP2/>
Type of document: Deliverable
Dissemination level: Public
Status: Final
Due date of delivery: 01/04/2013
Reviewer: J.P. Vilotte/A. Busch (IPGP)
Keywords : Evaluation, Integration, Testing, Tools, Services, Application codes, Software components, First release, Report

Version	Author	Date	Comments
0.1	Siew Hoon Leong	28/02/2013- 06/03/2013	Initial draft for comments
0.2	Andre Gemünd Siew Hoon Leong	07/03/2013	First revision
0.3	Mario David	07/03/2013	Second revision
1.0	Siew Hoon Leong	01/04/2013	Final revision: Implementing recommendations from internal reviewers.

Copyright notice

Copyright © VERCE project, 2011-2015. See www.verce.eu for details on VERCE.

VERCE (“Virtual Earthquake and seismology Research Community e-science environment in Europe”) is a project co-funded by the European Commission as an Integrated Infrastructure Initiative within the 7th Framework Programme. VERCE began in October 2011 and will run for 4 years.

This work is licensed under the Creative Commons Attribution-Noncommercial 3.0 License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc/3.0/> or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, and USA.

The work must be attributed by attaching the following reference to the copied elements: “Copyright © VERCE project, 2011-2015. See www.verce.eu for details on VERCE”. Using this document in a way and/or for purposes not foreseen in the license requires the prior written permission of the copyright holders. The information contained in this document represents the views of the copyright holders as of the date such views are published.

Table of Content

1. Second Release Report	5
1.1. Release Management Schedule	5
1.2. Requested components	5
1.3. Assignments of Evaluators/Testers and Resources	6
1.4. Evaluation and Testing	8
1.4.1. Component Specific Tests	8
1.5. Results & Recommendations	9
1.6. Documentations	12
2. Lessons Learned	13
3. Conclusion and Future Plan	14
3.1. Schedule of next cycle	14
Appendix A: Release Management Schedule	16
References	17

List of figures

Figure 1. Release Management Schedule (Oct 2012 – Mar 2013)	5
---	---

List of tables

Table 1. Requested components for evaluation and testing	6
Table 2. Assigned Resources and Testers of each component	7

Executive summary

One of the objectives of the VERCE project is to provide a service-oriented architecture and framework that wraps the data-infrastructure resources and services with a set of distributed data-aware Grid and HPC resources provided by the European e-Infrastructure and community. To this end, the tools, services and application codes, i.e. software components, which are particularly relevant to the seismologists and the Earth Science community, are selected for integration on the VERCE Platform.

The main aim of this report is to give an update on the second release of the integrated tools and services. The Plan-Do-Check-Act (PDCA) cycle is used to manage the release process. This period corresponds to the second completed PDCA cycle and thus the second release.

The second release successfully completed within the schedule timeframe. In the second release, eight new components were evaluated and five of the delayed components from the first release were re-evaluated. Other than requests from the JRAs, infrastructure tool requests were also received from one of the SAs in this release. Eleven components are approved in this release cycle.

Application code

- SPECFEM3D

Library

- JSAGA

Tools

- Python Basemap (Matplotlib basemap)
- Metis
- GSISSH
- GRAM
- gLite Cream
- gLite WMS
- UNICORE6
- OGSA-DAI
- Dispel Gateway

The suggested improvements after the first release were fully implemented in this release, leading to an improvement in the evaluation procedures and communication with SA1. In addition, the Redmine ticketing system was evaluated and used in this release to foster the communication among the evaluators, the component developers and resource providers.

1. Second Release Report

The second PDCA cycle, corresponding to the second release of integrated services and tools, was completed at month 18, March 2013, of the project. As in the first release, a release management schedule was prepared to ensure that the process was clear to each involved work package. The JRAs and SA2 submitted a total of three application codes and ten tools/libraries for evaluation. In total, three application codes and ten tools/libraries were evaluated in this reporting period. Two application codes were delayed on the developers' request during the evaluation process. All ten tools/libraries were approved in this release. The details of this release are described in the following subsections.

1.1. Release Management Schedule

The work performed in this six months period is depicted in Figure 1. Learning from the experience in the first release, the JRAs and SAs were encouraged to submit their request form as early as possible (before the official deadline on 31st October 2012). All evaluation requests were received much earlier and the selection of the evaluation team for each component took place on 9th October 2012.

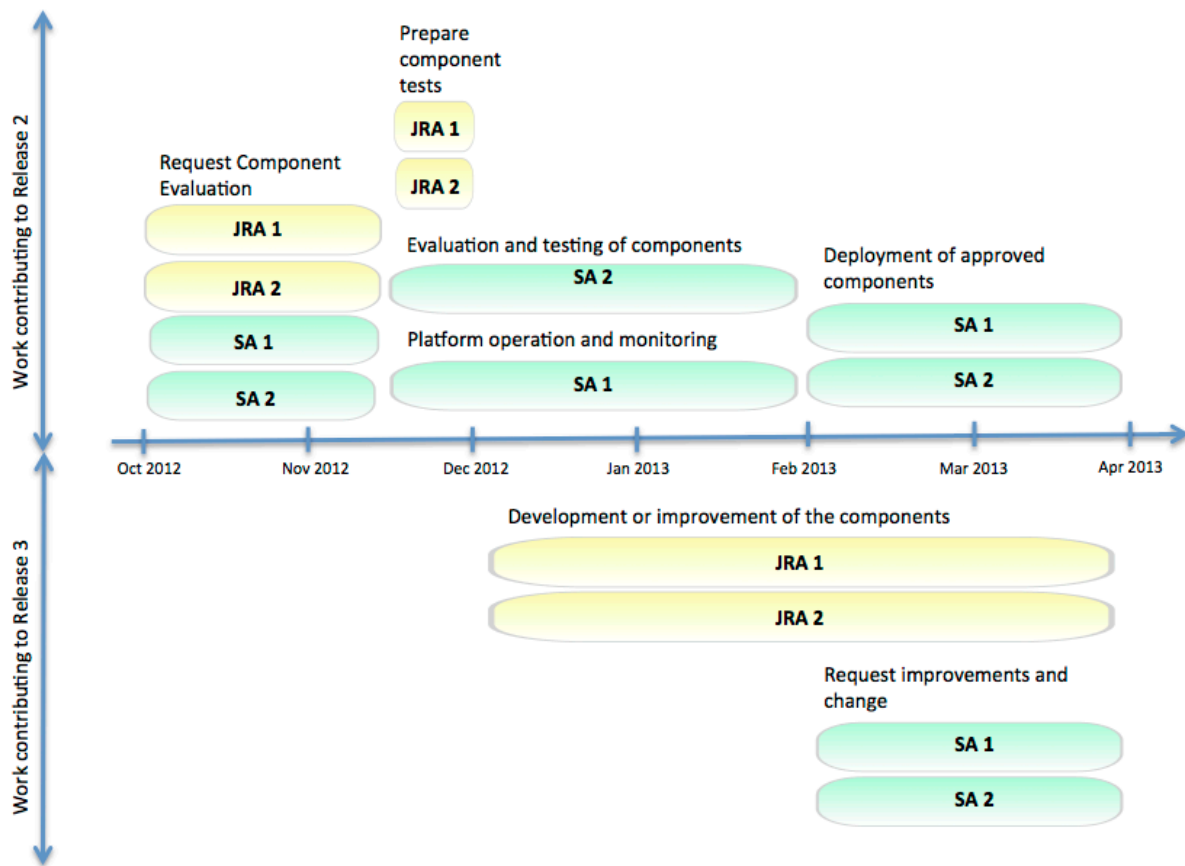


Figure 1. Release Management Schedule (Oct 2012 – Mar 2013)

1.2. Requested components

Three application codes, one library and nine tools evaluation requests were received from JRA1, JRA2 and SA2. A summary of each requested component is shown in Table 4. Detailed information about each component is available at http://www.verce-project.eu/projects/verce1/wiki/RP2b_Evaluation_and_Tests.

Component	Version	Type	Purpose	Submitted by
SPECFEM3D [11]	2.0.1 trunk revision 21447	Application code	Simulate forward and adjoint coupled acoustic-(an)elastic seismic wave propagation on arbitrary unstructured hexahedral meshes	JRA1
AxiSEM [1]	-	Application code	Calculation of 3D wave propagation in spherically symmetric 2D lateral heterogeneities	JRA1
COMCOT [10]	1.7	Application code	Tsunami modelling	JRA1
Python Basemap [13]	1.0.5	Library	Making wave projections of surface waveforms	JRA1
Metis [8]	4.0.3 or 5.0.2	Library	Mesh Partitioning (to be used with SeisSol)	JRA1
JSAGA [7]	0.9.15	Library	A library to interface with middleware Globus, Unicore and gLite	JRA2
GSISSH [6]	SSH 2.0 OpenSSH 6.0p	Tool	Interactive access to resources by using certificates for authentication	SA2
GRAM5 [5]	From GT5.2.1	Tool	Job submission middleware tool for EGI and PRACE resources	SA2
gLite CREAM [3]	1.14.0	Tool	Computing resource execution and management tool used in EGI	SA2
gLite WMS [4]	3.4	Tool	Workload management service tool used in EGI	SA2
UNICORE6 [12]	6	Tool	Grid middleware used in EGI and PRACE. A ready-to-run Grid technology	SA2
Dispel Gateway [2]	1.0	Tool	Delegation and implementation of logical workflows onto distributed compute resources	JRA2
OGSA-DAI [9]	4.2	Tool	Execution of workflows accessing distributed data resources	JRA2

Table 1. Requested components for evaluation and testing

1.3. Assignments of Evaluators/Testers and Resources

Due to the big number of components in this cycle, the partners from other work packages, in particular JRA1, were requested to assist in the evaluation and testing.

Component	Assigned Resource (SA1 Definition)	Assigned Tester
SPECFEM3D	SuperMUC (HPC-LRZ-01)	LRZ: Siew Hoon Leong INGV: Federica Magnoni
	PLX (HPC-CINECA-02)	CINECA: Graziella Ferini INGV: Federica Magnoni
	FERMI (HPC-CINECA-03)	CINECA: Graziella Ferini
AxiSEM	EGI Resources	Unassigned
COMCOT	SuperMUC (HPC-LRZ-01)	LRZ: Gilbert Brietzke
	PLX (HPC-CINECA-02)	SCAI: Andre Gemünd
Python Basemap	SuperMUC (HPC-LRZ-01)	LRZ: Siew Hoon Leong
	EGI Cluster (GRI-LRZ-02)	LRZ: Siew Hoon Leong
	EGI Cluster (GRI-SCAI-01)	SCAI: Andre Gemünd
Metis	EGI Cluster (GRI-LRZ-02)	LRZ: Gilbert Brietzke
	EDIN1 (DEP-UEDIN-01)	KNMI: Luca Trani
JSAGA	Virtual Machine at LRZ (-)	LRZ: Siew Hoon Leong LMU: Marek Simon
	Personal laptop (-)	LRZ: Siew Hoon Leong
	Personal workstation (-)	UEDIN: Iraklis Klampanos
GSISSH	SuperMUC (HPC-LRZ-01)	LRZ: Siew Hoon Leong
	EGI Cluster (GRI-LRZ-02)	LRZ: Siew Hoon Leong
	PLX (HPC-CINECA-02)	LRZ: Siew Hoon Leong
GRAM	SuperMUC (HPC-LRZ-01)	LRZ: Siew Hoon Leong
	EGI Cluster (GRI-LRZ-02)	LRZ: Siew Hoon Leong
gLite Cream	EGI Cluster (GRI-SCAI-01)	SCAI: Andre Gemünd
gLite WMS	EGI Cluster (GRI-SCAI-01)	SCAI: Andre Gemünd
UNICORE6	SuperMUC (HPC-LRZ-01)	LRZ: Siew Hoon Leong
	PLX (HPC-CINECA-02)	UEDIN: Amy Krause
Dispel Gateway	EDIM1 (EDIN)	KNMI: Luca Trani and Alessandro Spinuso
	Institutional Resource (ULIV)	KNMI: Alessandro Spinuso ULIV: Xiao Wang (supporting)
OGSA-DAI	EDIM1 (EDIN)	UEDIN: Iraklis Klampanos
	Institutional Resource (ULIV)	KNMI: Alessandro Spinuso ULIV: Xiao Wang (supporting)

Table 2. Assigned Resources and Testers of each component

The assignment of evaluators to AxiSEM did not take place. The developers were contacted before the assignment to gather more information about the new changes. During this process, we were informed that there were a few administrative issues that had to be resolved before AxiSEM could integrate on the VERCE platform. In addition, the developers would like to stabilise their release by testing it on production

resources. As such, test accounts were set up for the developers on VERCE's EGI resources. The evaluation of COMCOT was also put on hold after the assignment due to the introduction of a new tsunami code in the community that could potentially replace COMCOT. JRA1 and NA2 decided to spend more time in the evaluation and comparison of this new code with COMCOT before making a decision.

1.4. Evaluation and Testing

The evaluation and testing phase commenced in early October 2012 and was completed by the end of February 2013. In anticipation of the Christmas vacation, the team was encouraged to begin the evaluation process as early as possible. As such, several evaluations were completed before end of December 2012.

Similar to the previous release, generic and specific tests were carried out for each component. In this release, all components fulfilled the requirements in the generic tests. The detail of each specific component test is described in the following sub-section.

1.4.1. Component Specific Tests

SPECFEM3D

Since SPECFEM3D is an important code for the HPC use case, priority was given to it. The latest revisions were continuously tested at CINECA, especially on PLX, during this cycle and numerous issues were found. These issues were fed back to the developers for correction and inclusion into newer revisions. The current revision, 21447, is a stable release candidate and is recommended by JRA1 for deployment. Smaller test runs and bigger runs were prepared by the scientists and were used in the verification of this revision.

Python Basemap

To verify the correctness of basemap, the test suites provided by PIL, a python library that basemap requires, and basemap were used. On all test installations, the test suites must complete without any unexpected issues. Additionally, one python script that used the basemap module was run to verify the installation.

Metis

The API and command-line tools provided by Metis were used to verify the installation. Additionally, JRA1 provided a test, emulating the real use case, for the evaluation purpose. Both Metis version 4.0.3 and 5.0.2 were evaluated.

JSAGA

To verify the JSAGA library, the embedded command-line client interface was used. Different certificate formats, VOMS Proxy, JKS context, Globus proxy, were evaluated. Simple jobs were submitted to several job management resources in EGI and PRACE via gLite Cream, gLite WMS, GRAM5 and UNICORE6 interfaces. File transfers using this library via GridFTP (Globus and gLite) on EGI and PRACE resources were also carried out. The current release of JSAGA does not support OpenSSL 1.* PEM formatted keys.

GSISSH

GSISSH was verified by using the Robot Framework test suite that was provided by IGE. The tests in the test suite are based on the tests defined by SA2. The test suite is integrated into our Jenkins Continuous Integration (CI) application. More information about our Jenkins integration is described in Section 1.4.3 of D-SA2.2.

GRAM

GRAM was also verified by using the Robot Framework test suite that was provided by IGE. The tests in the test suite are based on the tests defined by SA2. The test suite is integrated into our Jenkins Continuous Integration (CI) application.

gLite CREAM

gLite CREAM was verified by using the EGI SAM tests via NAGIOS at IPGP and SCAI. More information about the tests can be found at <https://tomtools.cern.ch/confluence/display/SAMDOC/Probes>. More information about the test infrastructure is available at https://wiki.egi.eu/wiki/SAM_Instances.

gLite WMS

gLite WMS was also verified using EGI SAM tests/NAGIOS framework at IPGP and SCAI. More information about the tests and the test infrastructure is available at <https://tomtools.cern.ch/confluence/display/SAMDOC/Probes> and https://wiki.egi.eu/wiki/SAM_Instances respectively.

UNICORE6

To verify UNICORE6, the UNICORE Rich Client was used. A simple MPI job was created and submitted to both SuperMUC and PLX. File creation and file transfer were tested on various file systems on both resources.

OGSA-DAI

To verify OGSA-DAI, the developers created a set of functional tests for SA2. The tests include the basic service availability, simple request submissions and client toolkit verification. File staging is yet to be supported in this release but is expected in future releases.

Dispel Gateway

To verify the Dispel Gateway, the developers created a set of functional tests for SA2. The tests include checking the UI availability, gateway reliability (uptime of at least one week) and request verification. The current release does not support file staging due to its dependency on OGSA-DAI.

1.5. Results & Recommendations

Approved components

Eleven out of the thirteen evaluated components were approved for release after the evaluation and testing phase. In order to aid SA1 in selecting the resources for deployment, the recommendations are categorised as follows.

Resource Type:

- All (Including HPC, Grid and Departmental resources)
- HPC
- Grid
- Departmental
- Embedded – An embedded library or tool and thus explicit deployment is unnecessary.

Recommendation:

- Recommended

- Optional

Each approved component will get one or multiple combinations of “resource type (recommendation)” tags. For example, if an approved component A has a tag of “All (Recommended)”, it implies that SA2 strongly encourage all resource providers (HPC, Grid and Departmental) to deploy A. If component B has two tags of “HPC (Recommended), Grid (Optional)”, it implies that HPC resource providers are strongly encouraged to deploy B while Grid resource providers can choose if they would like to deploy B. Departmental resource providers are thus implicitly not recommended or required to install component B. If component C has a tag of “Embedded”, it implies that C will be embedded within another component and thus no resource providers have to explicitly deploy it. It also implies that the JRAs can safely exploit this component within other tools or services.

The eleven components that were approved in the second release are:

1) SPECFEM3D

The latest revision, 21447, with GPU support, was used for evaluation on three HPC resources, PLX, FERMI and SuperMUC. There are still a few memory problems where arrays are not properly deallocated in this revision. These remaining issues are only in subroutines and modules that support a functionality that will not be used by the VERCE HPC use case. As such, this revision is currently sufficient for our purpose. SA2 thus strongly recommends this component to be deployed on all HPC resources. Grid and departmental resource providers can freely decide if they would like to support this component.

2) Python Basemap

Python Basemap, also known as Matplotlib basemap, was evaluated on two Grid and one HPC resources. Basemap is dependent on python and a number of python libraries, e.g. PIL and Matplotlib. On all three resources, basemap was installed without any major issues. As such, SA2 recommends it to be installed on all resources, including departmental, which have deployed Python and Obspy in the previous release.

3) Metis

Metis was evaluated on an HPC resource and a Grid resource. Both installations and evaluations completed without any issues. One point to note is that the requested version 4.0.3 is not the most up-to-date released version. There is a new version, 5.0.2, which is nearly a complete re-write of the code-base. Version 5 is not totally backward compatible with version 4. Upon discussion with JRA1, both version 4 and 5 comply with our current requirements. The recommendation from SA2 is thus as such. All resources are recommended to install this component. Version 5 is preferable if a resource provider has the option to choose which version to install. Resource providers that have to support users/communities that require version 4 can choose to deploy version 4 or both versions. In principle, the resource providers can freely decide on which version to deploy.

4) JSAGA

JSAGA was evaluated on two local resources (a laptop and a workstation) and a virtual machine. The original jar release had an installation issue, which was later fixed by the developers upon feedback. In general, the installation via the jar release is easy. The required gLite, Globus and UNICORE6 configurations are rather complex due to a lack of comprehensive documentation. All experiences gained were documented in the SA2 wiki. SA2 thus recommends this component to the JRAs as an interface library for their tools and services to interact with the three main middleware tools. Since this component is simply a library, it is not necessary for SA1 to explicitly deploy it.

5) GSISSH

GSISSH was evaluated on one Grid and two HPC resources. GSISSH is chosen to provide a certificate based interactive access mechanism for VERCE developers and members to access the testbed. This service is especially necessary on resources where the JRAs developers and SA2 evaluators have to access to perform installation and test runs. Currently, this service is already available on four resources, SuperMUC, PLX, SCAI EGI cluster and LRZ EGI cluster. IGE offers a pre-compiled version that can be easily installed on numerous operating systems. Alternatively, a source version is also available. SA2 recommends this service to be installed on all resources to allow a unified certificate-based access method on the VERCE testbed.

6) GRAM

GRAM was evaluated on a Grid and an HPC resources. GRAM is used as a job manager on both EGI and PRACE resources. IGE offers a pre-compiled version that can be easily installed on numerous operating systems. Alternatively, a source version is also available. Since there are other job management tools available, this component is optional for all resources. However, each resource provider should choose to deploy at least one of these job management tools, gLite CREAM or UNICORE6 if GRAM is not deployed.

7) gLite CREAM

gLite CREAM was evaluated on two Grid resources. gLite CREAM offers a job manager service and provides other features and functionalities. Its purpose in VERCE is similar to GRAM, to offer an interface for job management. EMI offers releases that are well tested for SL and its variants. Similar to GRAM, this component is optional for all resources. However, each resource provider should choose to deploy at least one of these job management tools, GRAM or UNICORE6 if gLite CREAM is not deployed.

8) gLite WMS

gLite WMS was evaluated on two Grid resources. gLite WMS offers submission and management of jobs on distributed instances of gLite CREAM, including scheduling, data staging and resubmission. As one instance of WMS can service an arbitrary number of CREAM instances (scalability issues aside), it is sufficient to have only one or two instances of this component in the VERCE testbed. Currently, both SCAI and IPGP have a production instance of WMS. As such, no additional gLite WMS needs to be deployed.

9) UNICORE6

UNICORE6 was evaluated on two HPC resources. In addition to a job manager service, UNICORE6 provides other features and functionalities. Its purpose in VERCE is similar to GRAM and gLite CREAM, to offer an interface for job management. EMI offers releases that are well tested. Since UNICORE6 is JAVA based, it can be installed on most operating systems. Similar to the other job manager services, this component is optional for all resources. However, each resource provider should choose to deploy at least one of these job management tools, GRAM or gLite CREAM if UNICORE6 is not deployed.

10) OGSA-DAI

OGSA-DAI was evaluated on two departmental resources. OGSA-DAI is a component that will be used by the Dispel Gateway to support numerous functionalities, e.g. access to data sources, management of data transfers, job submission, etc. As such, it should be deployed together with the Dispel Gateway. This component is only recommended on private or non-production or standalone resources at this moment. For more information, please refer to the Dispel Gateway.

11) Dispel Gateway

Dispel Gateway was evaluated on two departmental resources. Dispel Gateway is highly dependent on OGSA-DAI and thus both should be deployed together. Dispel Gateway does not have its own certificate-based authentication service and thus depends on the web server, e.g. Tomcat or Apache HTTP server, for authentication. Currently, there is no public documentation or recommendation on how best to set up the Dispel Gateway and OGSA-DAI on such a web server. As such, it is only recommended on private or non-production or standalone resources. Once the documentation or recommendation is available, an evaluation will be proposed to check if it satisfies the security requirements before SA2 recommends it on production or shared resources.

Delayed components

1) COMCOT

COMCOT is a tsunami modelling code that was proposed in the first release. However, its evaluation was delayed to the second release. During this time, the research community proposed a new tsunami code that appeared to be very promising. During the evaluation process, JRA1/NA2 contacted SA2 to delay the evaluation until they could internally compare this new code with COMCOT.

2) AxiSEM

The AxiSEM developers decided to delay the evaluation of this code when contacted. Two reasons were raised to justify this delay. The developers would like to stabilise the code and the first results would be internally disseminated before sharing with 3rd party projects, e.g. VERCE. In addition, AxiSEM is currently not open-source as required by VERCE. In order to continue to work together until these issues are resolved, the developers were invited to join the “verce.eu” VO and to test and stabilise their code on VERCE’s EGI resources at LRZ and SCAI.

1.6. Documentations

All issues faced and solutions that the SA2 team encountered during this release cycle were documented in the SA2 wiki and in the request forms for each component. These documents contribute to the technical documentation that SA1 will use to coordinate the deployment of the approved components on the VERCE testbed. The documentation can be found at http://www.verce-project.eu/projects/verce1/wiki/RP2b_Results_Issues_and_Documentations.

2. Lessons Learned

The lessons learned in the previous release cycle were compiled and proposed as improvements to be carried out in this release. Common meetings between SA1 and SA2 were scheduled as needed in order to improve communication between the activities. A meeting was scheduled by SA1 at the end of the evaluation period of the second release to discuss the deployment recommendations. Clearer recommendations were requested by SA1. SA2 promptly added the definition of resource type and recommendation as described in Section 1.5. This helped SA1 to have a better understanding as to where they should deploy each component. The Redmine ticketing system was also being evaluated and used in this release to help improve and foster communications among developers, evaluators and resource providers. As such, relevant discussions and/or decisions are centrally recorded and are traceable by other members and work packages. It also helps to ensure that open issues are not forgotten.

An earlier start or preparation was also recommended to relax the time constraints for everyone involved. SA2 thus prepared the release schedule one month in advance for this release so that some members can have an earlier start. Vacation and holiday season can also delay the evaluation process as we observed in the last release. As such, all members were requested to share their vacation plans as early as possible so that they could be respected in the work plan. This preparation improved the evaluation procedures and allowed many evaluators to complete their tasks before the winter vacation period in December.

The flexibility issue was also addressed in this cycle with regards to SPEC3D. The importance of this code required priority in the evaluation procedures. Since the latest version does not have an official release and the issues found in the previous evaluation cycle had to be addressed by the developers, JRA1's involvement was required. Thus JRA1 actively cooperated with the developers to resolve the issues found and new revisions of the software were continuously evaluated during this time. SA2 contributed by installing and evaluating some revisions before approving the most stable one that satisfied the requirements.

3. Conclusion and Future Plan

The second release of the VERCE software components successfully completed within the scheduled timeframe. Eleven out of thirteen components were approved in the second release as a part of the VERCE platform. Currently, there are a total of fourteen approved components, including the first release, on the platform.

In general, the procedures are effective for managing the release. Improvements have been made in this release as planned and recommended in the previous release. SA2 will continue to carry out these recommendations in future release cycles. If required, a questionnaire will be prepared to gather additional feedback.

The release management schedule for the third release, refer to Section 3.1, is prepared and shared with involved work packages.

3.1. Schedule of next cycle

The schedule for the next cycle, 1st April 2013 to 30th September 2013, for the respective work packages is summarised below.

JRA2

- [1st April – 30th April 2013]: To provide requirements (tools and services to evaluate) via the [Request Form](#). Simultaneously, JRA2 should be receiving new requirements from NA2 and JRA1 (and SAs) to plan for the work in this reporting period.
- [Until 31st May]: To define tests to perform (in particular, functionalities that will be used) for external tools and in-house development components
- [1st June – 30th September 2013] To work on developing the required workflow related tools

JRA1

- [1st April – 30th April 2013]: To provide requirements (application codes to evaluate) via the [Request Form](#). Simultaneously, JRA1 is receiving new requirements from NA2 (and SAs) to plan for their development work in this reporting period.
- [Until 31st May]: To provide the tests (including input files)
- [1st June – 30th September 2013]: To work on developing the application codes

SA2

- [1st April – 30th April 2013]: To identify missing features and functionalities and thus suggest additional tools to be evaluated. Administrative work to prepare for the next release.
- [1st May – 7th May 2013]: To select the team for each component and begin installation on the selected resources
- [8th May – 31st July 2013]: To work on evaluating and integrating the selected components.
- [1st August – 31st September 2013]: To check the service information collected by SA1 and provide feedback/new requirements to the JRAs. Simultaneously, SA2 will provide the accepted list of components (components that have passed the evaluation and tests) to SA1 and assist SA1 in deploying the approved components.

SA1

- [1st April – 30th April 2013]: To identify missing features and functionalities and thus suggest additional tools to be evaluated. Administrative work to prepare for the next release.
- [1st May – 31st July 2013]: Operate and monitor the scientific platform and the tools and services running on it. To collect service information

- [1st August – 31st September 2013]: To work on deploying the released components provided by SA2. To check the service information collected and provide feedback/new requirements to the JRAs. Continue to operate the scientific platform.

Appendix A: Release Management Schedule

Reporting Period: 1st November 2012 – 31st March 2013

JRA2

- [1st October – 31st October 2012]: To provide requirements (tools and services to evaluate) via the [Request Form](#). Simultaneously, JRA2 should be receiving new requirements from NA2 and JRA1 (and SAs) to plan for the work in this reporting period.
- Until 31st October to define tests to perform (in particular, functionalities that will be used) for external tools and in-house development components
- [1st November 2012 – 31st March 2013] To work on developing the required workflow related tools

JRA1

- [1st October – 31st October 2012]: To provide requirements (application codes to evaluate) via the [Request Form](#). Simultaneously, JRA1 is receiving new requirements from NA2 (and SAs) to plan for their development work in this reporting period.
- Until 31st October to provide the tests (including input files)
- [1st November 2012 – 31st March 2013] To work on developing the application codes

SA2

- [1st October – 31st October 2012]: To identify missing features and functionalities and thus suggest additional tools to be evaluated. Administrative work to prepare for the next release.
- [1st November – 6th November 2012] To select the team for each component and begin installation on the selected resources
- [7th November 2012 – 31st January 2013] To work on evaluating and integrating the selected components.
- [1st February – 31st March 2013] To check the service information collected by SA1 and provide feedback/new requirements to the JRAs. Simultaneously, SA2 will provide the accepted list of components (components that have passed the evaluation and tests) to SA1 and assist SA1 in deploying the approved components.

SA1

- [1st October – 31st October 2012]: To identify missing features and functionalities and thus suggest additional tools to be evaluated. Administrative work to prepare for the next release.
- [1st November – 31st January 2013] Operate and monitor the scientific platform and the tools and services running on it. To collect service information
- [1st February – 31st March 2013] To work on deploying the released components provided by SA2. To check the service information collected and provide feedback/new requirements to the JRAs. Continue to operate scientific platform.

[Source from : http://www.verce-project.eu/projects/verce1/wiki/Release_Management_Schedule#Reporting-Period-2b-1st-October-2012-31st-March-2012]

References

- [1] *axisem - Seismology and Geodynamics*. (n.d.). Retrieved March 06, 2013 from <http://www.seg.ethz.ch/software/axisem>
- [2] *Data-Intensive Distributed Computing - ADMIRE*. (n.d.). Retrieved March 06, 2013 from <http://www.admire-project.eu/data-computing/index.html>
- [3] *glite-CREAM*. (n.d.). Retrieved March 06, 2013 from http://glite.cern.ch/R3.2/sl5_x86_64/glite-CREAM/
- [4] *gLite-WMS*. (n.d.). Retrieved March 06, 2013 from <http://glite.cern.ch/glite-WMS/>
- [5] *GT 5.2.4 GRAM5*. (n.d.). Retrieved March 06, 2013 from <http://www.globus.org/toolkit/docs/latest-stable/gram5/#gram5>.
- [6] *GT 5.2.4: GSI-OpenSSH*. (n.d.). Retrieved March 06, 2013 from <http://www.globus.org/toolkit/docs/latest-stable/gsiopenssh/#gsiopenssh>
- [7] *JSAGA*. (n.d.). Retrieved March 06, 2013 from <http://grid.in2p3.fr/jsaga/>
- [8] *METIS - Serial Graph Partitioning and Fill-reducing Matrix Ordering*. (n.d.). Retrieved March 06, 2013 from <http://glaros.dtc.umn.edu/gkhome/metis/metis/overview>
- [9] *OGSA-DAI: Welcome to OGSA-DAI*. (n.d.). Retrieved March 06, 2013 from <http://www.ogsadai.org.uk/>
- [10] *Phil's Group: Research - COMCOT*. (n.d.). Retrieved March 06, 2013 from <http://ceeserver.cee.cornell.edu/pll-group/comcot.htm>
- [11] *SPECFEM3D - Geodynamics.org*. (n.d.). Retrieved March 06, 2013 from <http://www.geodynamics.org/cig/software/specfem3d>
- [12] *UNICORE*. (n.d.). Retrieved March 06, 2013 from <http://www.unicore.eu/>
- [13] *Welcome to the Matplotlib Basemap Toolkit documentation!* (n.d.). Retrieved March 06, 2013 from <http://matplotlib.org/basemap/>