PERIODIC TECHNICAL REPORT I (18 MONTHS), PART B

LUCA DE FEO, MICHAEL KOHLHASE, MIN RAGAN-KELLEY, DMITRII PASECHNIK, CLÉMENT PERNET, BENOÎT PILORGET, VIVIANE PONS, AND NICOLAS M. THIÉRY

CONTENTS

1. EXPLANATION OF THE WORK CARRIED OUT BY THE BENEFICIARIES AND OVERVIEW OF THE PROGRESS

1.1. Objectives

For reference, let us recall the aims of OpenDreamKit.

Aim 1: Improve the productivity of researchers in pure mathematics and applications by promoting collaborations based on mathematical **software**, **data**, and **knowledge**.

Aim 2: Make it easy for teams of researchers of any size to set up custom, collaborative *Virtual Research Environments* tailored to their specific needs, resources and workflows. The VRE should support the entire life-cycle of computational work in mathematical research, from initial exploration to publication, teaching and outreach.

Aim 3: Identify and promote best practices in computational mathematical research including: making results easily reproducible; producing reusable and easily accessible software; sharing data in a semantically sound way; exploiting and supporting the growing ecosystem of computational tools.

Aim 4: Maximise sustainability and impact in mathematics, neighbouring fields, and scientific computing.

Those aims are backed up in our proposal by nine objectives; we now highlight our main contributions during this reporting period toward achieving each of them.

Objective 1: "To develop and standardise an architecture allowing combination of mathematical, data and software components with off-the-shelf computing infrastructure to produce specialised VRE for different communities."

This objective is by nature multilevel; achievements include:

- Collaborative workspaces: major JUPYTERHUB developments, see **T4.2**: "Notebook improvements for collaboration"; study and documentation of the SAGEMATHCLOUD architecture, see **T3.6**: "Document and modularise SAGEMATHCLOUD's codebase";
- User interface level: enabling JUPYTER as uniform interface for all computational components; see **T4.1**.
- Interfaces between computational or database components: short term: refactoring of existing ad-hoc interfaces, see **T4.12**; long term: investigation of patterns to share data, ontologies, and semantics uniformly across components, see **T3.2**, and Section 1.2.6 about WP6.

Objective 2: "To develop open source core components for VRE where existing software is not suitable. These components will support a variety of platforms, including standard cloud computing and clusters. This primarily addresses Aim 2, thereby contributing to Aim 1 and 3."

At this stage, it has been possible to implement most of the required developments within existing components or extensions thereof. New software components includes the tools nbmerge, nbdiff and nbval (see D4.8 and D4.6).

Objective 3: "To bring together research communities (e.g. users of JUPYTER, SAGE, SINGULAR, and GAP) to symbiotically exploit overlaps in tool creation building efforts, avoid duplication of effort in different disciplines, and share best practice. This supports Aims 1, 3 and 4."

We have organized or coorganized a dozen users or developers workshops (see **T2.3**) which brought together several communities. Some key outcomes include:

- Enabling JUPYTER as uniform interface for all computational components; see **T4.1**.
- Sharing best practices for packaging and building containers; see **T3.3**;

- A smooth collaboration between JUPYTERHUB and SAGEMATHCLOUD; see **T3.6** and Section 1.4;
- Work on interfaces between systems; see **T3.2** and **T4.12**;
- Sharing best practices when using VRE's like SAGEMATHCLOUD or JUPYTER for research and education.

Objective 4: "Update a range of existing open source mathematical software systems for seamless deployment and efficient execution within the VRE architecture of objective 1. This fulfills part of Aim 2."

Achievements include:

- Packaging efforts: Docker containers (delivered and regularly updated), Debian and Conda packages (beta); see **T3.3**.
- Portability of SAGE and its dependencies on Windows (beta); see **T3.1**.
- Integration of all the relevant mathematical software in the uniform JUPYTER user interface, in particular for integration in the VRE framework (delivered, ongoing); see **T4.1**.
- Ongoing work in WP5 to better support HPC in the individual mathematical software system and combinations thereof; see Section 1.2.5.

Objective 5: "Ensure that our ecosystem of interoperable open source components is *sustainable* by promoting collaborative software development and outsourcing development to larger communities whenever suitable. This fulfills part of Aims 3 and 4."

Achievements include:

- Outsourcing of the SAGE user interface by migrating to JUPYTER (delivered); see **T4.1**;
- Refactoring SAGE's documentation build system to contribute many local developments upstream (SPHINX) **T4.4** (ongoing);
- Outsourcing and contributing upstream as PYTHON bindings the existing SAGE bindings for many computational systems (delivered, ongoing); see **T4.12**.

Objective 6: "Promote collaborative mathematics and science by exploring the social phenomena that underpin these endeavours: how do researchers collaborate in Mathematics and Computational Sciences? What can be the role of VRE? How can collaborators within a VRE be credited and incentivised? This addresses parts of Aims 3, 1, and 2."

This objective is at the core of WP7; see Section 1.2.7 for details. Achievements include:

- Methodology, data, and tools needed to assess development models of academic open-source systems have been surveyed in D7.1.
- Game-theoretic aspects of ways to incentivise collaboration are studied—ongoing, also see [**Pavlou:2016:MCI:2936924.2936934**].
- Implementing in SAGE constructions from the database of strongly regular graphs by Andries Brouwer, see paper [**2016arXiv160100181C**].

Objective 7: "Identify and extend ontologies and standards to facilitate safe and efficient storage, reuse, interoperation and sharing of rich mathematical data whilst taking account of provenance and citability. This fulfills parts of Aims 2 and 3."

This objective is at the core of WP6; see Section 1.2.6 for details.

Objective 8: "Demonstrate the effectiveness of Virtual Research Environments built on top of OpenDreamKit components for a number of real-world use cases that traverse domains. This addresses part of Aim 2 and through documenting best practices in reproducible demonstrator documents Aim 3."

Most of the work toward this objective is by nature planned for later in the project execution. Nevertheless, work has started e.g. toward the OOMMF demonstrator; see **T2.7 T2.8**, **T3.8**.

Objective 9: "Promote and disseminate OpenDreamKit to the scientific community by active communication, workshop organisation, and training in the spirit of open-source software. This addresses Aim 4."

This objective is at the core of WP2, with in particular 22 meetings, developer and training workshops organized during the first reporting period. See Section 1.2.2 for details.

1.2. Explanation of the work carried per Work Package

1.2.1. Work Package 1: Project Management

The general objectives of Work Package 1 are:

- Meeting the objectives of the project within the agreed budget and timeframe and carrying out control of the milestones and deliverables
- Ensure all the risks jeopardising the success of the projects are managed and that the final results are of good quality
- Ensuring the innovation process within the project is fully aligned with the objectives set up in the Grant agreement

WP1 has been divided into three tasks. In the following, progress is reported with respect to these individual tasks. Key results of WP1 are the following:

- A Consortium Agreement signed by all partners
- A kickoff meeting and three progress meetings organised
- A successful interim review at month 9 with the grade 3/4
- All milestones have been reached and deliverables achieved within the 1st Reporting Period timeframe
- The setting up of a new version of the OpenDreamKit website, with a more end-user friendly interface
- Success in the recruitment of highly qualified staff
- Many successful workshops open to different communities organisedl
- An Advisory Board and Quality Review Board set up to control the quality and the relevance of the software development relative to the end-user needs.

Concerning the recruitment: the strategies we used (tayloring of the positions according to the known pool of potential candidates, in particular among previous related projects, strong advertisement, ...) seem to have paid off, and we are really happy with the top notch quality of our recruits. However, despite many steps to foster women applications to apply (e.g. through reaching personally toward potential candidates or including women in the committees), we had almost no female candidate, and none made it to the short list. This is alas unsurprising in the very tight segment of experienced research software engineers for mathematics on temporary positions which is highly gender imbalanced; this is nevertheless a failure.

T1.1: "Project and financial management". A consortium agreement was signed between partners, stating precise rules about topics such as: responsibilities, governance, access to results and the background included. This consortium agreement respects the state of mind of the opensource software communities and does not plan to commercially exploit the Intellectual Property produced in the frame of OpenDreamKit.

During the 1st Reporting Period, a kick-off meeting was organised in Orsay, followed by 3 progress meetings at which partners presented status reports, and the steering committee got together. The first progress meeting was organised in St Andrews (January 2016), the second one was located in Bremen (June 2016), and the last one in Edinburgh (January 2017). The second meeting coincided with the interim project review, planned at month 9, where deliverables due by then were presented to the Project Officer and Reviewers. The OpenDreamKit project was granted the grade 3 out of 4 for this interim review: “Good progress (the project has achieved most of its objectives and technical goals for the period with relatively minor deviations)”.

As planned in WP1, UPSud has been coordinating OpenDreamKit. The UPSud relevant administration body, the D.A.R.I. (Direction des Activités de Recherche et de l’Innovation) and its finance service took care of the budget repartition in November-December 2015. The D.A.R.I., with Florence Bougeret, is also leading the Financial Statement of this reporting period. Due to the length of the first reporting period (18 months), the UPSud administration had decided to organise an internal and interim breakdown of costs at the middle of the Reporting Period. This exercise aimed at raising potential questions from partners early on and to make sure partners do follow the EC rules for the eligibility of costs. The coordinator is therefore confident that all partners will be able to declare their costs for this Reporting Period.

UPSud also lead the amendment number AMD-676541-5 which added to the consortium UGent, and is currently leading the amendment number AMD-676541-13 for the addition of two new sites: Friedrich-Alexander Universität Erlangen-Nürnberg and the new laboratory European XFEL. The addition of these three new partners is due to the moving of key permanent and/or non-permanent researchers who are key personnel for the success of OpenDreamKit. A collateral effect is the termination of SOUTHAMPTON’s participation to the consortium since no relevant staff for OpenDreamKit remained in this institution.

Concerning the communication, intern communication tools are described in D1.1: “Basic project infrastructure (websites, wikis, issue trackers, mailing lists, repositories)”. As for external communication the website for the project has been continuously updated with new content, and virtually all work in progress is openly accessible on the Internet to external experts and contributors (for example through open source software on Github). A new version of the website was released on the 15/03/2017. Its end-user friendly interface and content makes it a tool not only for internal communication but very much for dissemination and progress tracking by the reviewers and the community.

Furthermore D1.2: “Data Management Plan V1” gave a first version of the management of data produced OpenDreamKit.

T1.2: “Quality assurance and risk management”. The Quality Assurance Plan is described in detail in D1.3: “Internal Progress Reports year 1, including risk management and quality assurance plan”. We will describe the main points below. UPSud launched a Quality Review Board which is chaired by Hans Fangohr. The four members of the board have a track record of caring about the quality of software in computational science. This board is responsible for ensuring key deliverables do reach their original goal and that best practice is followed in the writing process as well as in the innovation production process. The board will meet after the end of each Reporting Period (RP), and before the Review following that RP.

The other structure supporting OpenDreamKit to ensure the quality of the infrastructure is the End-user group that is composed of some members of the Advisory Board. It is composed of seven members:

- Lorena Barba from the George Washington University
- Jacques Carette from the McMaster University
- Istvan Csabai from the Eötvös University Budapest
- Françoise Genova from the Observatoire de Strasbourg
- Konrad Hinsen from the Centre de Biophysique Moléculaire
- William Stein, CEO of SageMath Inc.
- Paul Zimmermann from the INRIA

This Advisory Board being composed of Academics and/or software developers from different backgrounds, countries and communities, it will be a strong asset to understand the needs of a variety of end-user profiles. This Technical Report for the first Reporting Period will be the first occasion to ask for their feedback on the potential of the VRE and our strategy to promote its use around the world. According to the Consortium Agreement, all Advisory Board members have signed a lightweight Non-Disclosure Agreement with the consortium.

UPSud has also been managing risks. In D1.3 all potential risks were assessed by the Coordinator at Month 12. Here is a brief update on Risk 1 concerning the recruitment of highly qualified staff. This risk has been globally well managed thanks to a flexible workplan enabling adjustments in the timing of some tasks or deliverables, and thanks to legal actions taken by the Coordinator to allow key personnel, permanent or not, to remain in the Consortium even though their positions changed. The addition of the three partners is representative of these actions. The assessment for the other risks remain valid at Month 18, and we refer to D1.3 for details.

T1.3: "Innovation management". D1.4: "Innovation Management Plan v1" was produced at month 18 and is mainly focused on:

- The open source aspect of the innovation produced within OpenDreamKit
- The various implementation processes the project is dealing with
- The strategy to match end-users needs with the promoted VRE.

The second version of the Innovation Management Plan will add content to explain all the innovations that the VRE is bringing to end-users. However the open source approach and the "by users for users" development process will not change. One of the assessed risks for OpenDreamKit is to have different groups not forming effective teams. Put in other words, having developers of the different pieces of software working solely for the benefit of the programme they were initially working on and for. This risk is tackled by the Coordinator in order to reach the final goals of the VRE which are the unification of open source tools with overlapping functionality, the simplification of the tools for end-users without coding expertise, and the development of user-friendly interfaces. For this, the Scientific Coordinator is for example willfully pushing for joint actions and workshops. Even if it takes time to bend some of the old implementation processes and coding habits, more and actions are taken by OpenDreamKit participants from different communities to work together. More information on joint workshops can be found in the section below.

1.2.2. WorkPackage 2: Community Building, Training, Dissemination, Exploitation, and Outreach

As planned in **T2.1**: "Dissemination and Communication activities" and **T2.5**: "Dissemination: reaching towards users and fostering diversity", 22 meetings, developer and training workshops have already been organized and co-organized by OpenDreamKit (including 14 have for year 1 described in D2.2: "Community building: Impact of development workshops, dissemination and training activities, year 1") , and complemented by many presentations and activities in external events. This includes the first Women in Sage workshop in Europe and the first major dissemination event of the project (Computational Mathematics with Jupyter in ICMS). This testifies of the vibrant activity and energy of the OpenDreamKit participants.

More specifically, we have targeted two specific communities:

- The Mathematics community through specific Sage Days.
- The micro-magnetic community through specific workshops presenting the JUPYTER and Python interface to the widely used OOMMF **T2.7**: "Open source dissemination of micromagnetic VRE" (5 workshops so far).

OpenDreamKit is also working on its visibility and communication strategy with an active website keeping track of all project activities including conferences, workshops, talks, blogposts, press releases, reports, etc.

T2.1: "Dissemination and Communication activities". The D2.1: "Starting press release" was delivered and a page for the E-infrastructure booklet was written jointly by OpenDreamKit members. The website has been entirely rebuilt to better promote and centralize project activities. At this date, we have 58 posts on our website which correspond to blogposts (14), conferences and workshops (11), talks (14), and other project communication. The website also keeps track of press releases and deliverable reports. We have recently added an open-source web analytics system (Piwik) to track some basic informations about our visitors.

T2.2: "Training and training portal". Training is a core and transversal aspect of our project. It is carried out through interventions and events as we discuss in **T2.5**: "Dissemination: reaching towards users and fostering diversity" but also by writing documentation, tutorials (D2.4), blogposts, etc. All this communication is centralized on our website. We have created a specific page listing the different software programmes which could eventually host training material. Furthermore, each programme has its own tag which links to all related project activities.

T2.3: "Community Building: Development Workshops". Development workshops are a key aspect of OpenDreamKit development model. The aim of these workshops is to bring together developers from the different communities to design and implement some of the wanted features. As reported in D2.2: "Community building: Impact of development workshops, dissemination and training activities, year 1", we have organized or co-organized 5 of these workshops during year 1 of the project. Since then, 2 more have happened. The thematics varies for each event: packaging and portability, Sage and Jupyter, Sage-GAP days, PARI/GP, knowledge representation, etc. To this, we can add 4 project meetings which we always turn into an occasion for more coding sprints and development discussions.

T2.4: "Reviewing emerging technologies". This task has been started during our initial Kick-Off meeting where we organized a session of short talk presenting many different technologies (Jupyter, SageMathCloud, Docker, Sphynx, Cython, Pythran, and many more). Some of them were further developed into D2.3. The goal is to keep up to date with most recent breakthroughs related to our work in OpenDreamKit. When we felt it was relevant, we turned part of the document into blogposts on our website gathered under a specific tag. At this date, we have published 4 articles and plan on publishing more throughout the project.

T2.5: "Dissemination: reaching towards users and fostering diversity". Dissemination is a key aspect of the success of OpenDreamKit. Indeed, our development is carried out to help and support mathematical communities. One of the goals is to bring more users and more developers to the different projects we are involved in. The events that took place during Year 1 have been reported in D2.2: "Community building: Impact of development workshops, dissemination and training activities, year 1", since then, still more happened.

- **Organization of Sage Days in established mathematical communities.** Sage Days have long been part of the SageMath tradition. By organizing and supporting Sage Days, OpenDreamKit can stay close the mathematical community, understand its needs, gather more users and developers, and improve the overall quality of the software. We have been involved in 5 different such events since the beginning of the project.
- **Training activities in developing countries.** OpenDreamKit has a long term plan of fostering a SageMath community in the Mediterranean area where 3 different events where organized (Algeria, Lebanon, and Tunisia) and some more are planned in Morocco. We were also present at ECCO 2016 in Columbia and have been invited to come back for the next conference in 2018.
- **Women in Sage.** OpenDreamKit is concerned with the gender gap in mathematic software development. We have organized the first Women in Sage conference in Europe, inviting 20 women to participate to a week of Sage development. We plan at least one other such event during the project. You can read our report on our website.
- **Computational Mathematics with Jupyter.** This was the first event of the 3 main dissemination conferences planned throughout the project. It gathered a large mathematics community in ICMS, Edinburgh, in collaboration with the CoDiMa project.
- **Jupyter Day in Orsay.** This one day training event in Orsay was fully booked shortly after it was announced, testifying of the growing interest of the community for the Jupyter project. It featured demos and talks from experts of the field as well as an afternoon of tutorials.
- **Other training and communication activities.** The participants of OpenDreamKit are very active in spreading their knowledge and the project's news to the different open source and mathematical communities they belong to. We count 11 talks on the website. We have been present to many major events: PyCon, EuroScyPy, CICM, ISSAC, CoDiMa school and more.

T2.6: "Introduce OpenDreamKit to Researchers and Teachers". At the occasion of the KickOff Meeting, Viviane Pons presented SageMathCloud: an online solution for collaborative work on OpenDreamKit software such as SAGE and JUPYTER. In particular, it offers a basic course management system that appeared to be a very good solution for some of the challenges described in this task. Following this talk, Michael Croucher implemented this solution for many courses at Sheffield university. This system is also used at Paris-Sud. As a result, the first goal of the task was achieved when we delivered D2.4 on how to use SageMathCloud for teaching with OpenDreamKit technologies.

We are also looking at nbgrader: at Southampton, the Jupyter Notebook and nbgrader were used to support teaching of large (ca. 500 students) and more specialized engineering design teaching modules for students in the first year of their engineering degree programs.

T2.7: "Open source dissemination of micromagnetic VRE". We created a GitHub organisation named JOOMMF, where the micro-magnetic VRE code is publicly hosted (JOOMMF repo). For each JOOMMF package we use continuous integration on Travis CI where we perform tests and monitor the test coverage, which we then make available on Codecov. Documentation for each package consists of APIs (automatically generated from the code) and different tutorials created in Jupyter notebooks. Both of them are tested on Travis CI. Documentation is built and made publicly available on Read the Docs. After every major milestone, we upload each package to the Python Package Index repository. We encourage the early use of our software and invite for feedback for which we provide several different communication channels: Google group (joommf-news), Gitter channel, GitHub help repository, Twitter account, and a website.

T2.8: "Micromagnetic VRE dissemination workshops". We had several workshops and tutorials so far where we demonstrated the use of our Micromagnetic VRE, received feedback and feature requests from the community:

- Two workshops at the 61st Annual Conference on Magnetism and Magnetic Materials in New Orleans, LA, USA (2nd and 3rd November 2016).
- Tutorial at the Deutsche Physikalische Gesellschaft Fruehjahrstagung (Spring Meeting) of the Condensed Matter Section in Dresden, Germany on 19th March 2017.
- Workshop at the Institute of Physics Magnetism 2017 conference in York, UK on 5th April 2017.
- Workshop at the IEEE International Magnetics Conference - Intermag 2017, Dublin, Ireland. 24th April 2017.

T2.9: "Demonstrator: Interactive books". The first book we committed to deliver is *Linear Algebra (lectures for physicists)* D2.8. It is under active development as you can see on its git repo. Our main issue is now to complete the English version (it already exists in polish). We chose to use Sphinx and the Sagecell plug-in. The remaining books are at earlier stage. They will include some ideas and teaching materials which are developed at the moment.

Moreover, at Southampton, we have started to create interactive notebooks that support teaching of mathematics and computing to engineers. In particular, we have studied the feasibility of converting existing teaching materials (in LaTeX) containing code snippets into interactive notebooks. This is work in progress. We worked together with the Simula team in directing the design of the (NBVAL) software developed in **T4.3** of the OpenDreamKit project (D4.9), which addresses one of the questions outlined in this task: "How can we facilitate automatic testing of all code examples, plots, etc ?".

T2.10: "Demonstrator: Computational mathematics resources indexing service". We have started to list relevant resources related to the OpenDreamKit project and started discussions on the best course of action to follow. We are looking closely at the GEANT project to compare it with the SageIndex and understand their common goals and main differences.

1.2.3. WorkPackage 3: Component Architecture

T3.1: "Portability". The first task of this workpackage is to improve the portability of computational components. A particular challenge is the portability of SAGE (and therefore all its dependencies) on Windows, which has remained elusive for a decade, despite many efforts of the community.

No deliverable is due for the evaluation period, but we are happy to report that Erik Bray of UPSud has made considerable progress on D3.7: "One-click install SAGE distribution for Windows with Cygwin 32bits and 64bits" by producing a one-click Windows installer based on Docker. Although the Docker based installer has proven itself a viable solution, our plan is still to deliver a Cygwin-based installer on month 24. This project is at a very advanced stage: we have a patched version of SAGE compiling and runnning in Cygwin, and a beta version of the installer was made available in December. We expect to deliver D3.7 on time.

T3.2: "Interfaces between systems". In this task we investigate patterns to share data, ontologies, and semantics across computational systems, possibly connected remotely.

The Symbolic Computation Software Composability Protocol (SCSCP) is a remote procedure call protocol by which a computer algebra system (CAS) may offer services to a variety of possible clients, including e.g. another CAS running on the same computer system or remotely. The goal of D3.3 was to bring SCSCP support to all relevant components of OpenDreamKit. Thanks to the joint efforts of USTAN, UVSQ, JacobsUni, UGent and CNRS, SCSCP is now supported in GAP, SAGE and MATHHUB, and plans have been made to extend support to SINGULAR and PARI/GP at an appropriate time.

This task benefits from the work done in Task **T4.12** on low level interfaces to software components. In particular, the results of deliverables D4.1: "Python/Cython bindings for PARI and its integration in Sage" and D4.10: "Second version of the PARI Python/Cython bindings" will be crucial to supporting SCSCP in PARI/GP.

Furthermore, experimental work on a semantic interface between GAP and SAGE (D3.9, due on month 36) has started during the joint GAP-Sage days, and a working prototype is already available. The current prototype uses *ad hoc* language mechanisms to transfer the semantics from one system to the other; these mechanisms will be replaced with a generic API (Application programming Interface) once the MitM (Math-in-the-Middle) approach developed in WP6 will be mature enough.

T3.3: "Modularisation and packaging". In this task we investigate best practices for composing, sharing and interfacing computational components and data for connected mathematical systems.

The first deliverable, D3.1: "Virtual images and containers" was delivered on time early in the project, producing the expected results.

A focused workshop in March 2016 (Sage Days 77, followed by Sage Days 85 in March 2017) also triggered much work and progress on the packaging side, both by OpenDreamKit participants and the community. Partly thanks to this, SAGE is now on its way to be available as a standard package in major distributions such as Debian and Anaconda. We stress the fact that this task is very much ahead of schedule: having SAGE in Debian was the goal set for D3.10, which is only due in month 48.

The workshops were also the occasion to clarify the modularization, packaging, and distribution needs and challenges. This has paved the way for D3.6: "Open package repository for SAGE", due in month 24.

T3.4: "Simulagora integration". The goal of this task is to deliver every six months a new Simulagora VM image containing all the software components released over the period.

To this date, three OpenDreamKit VMs have been released in Simulagora.

T3.5: "Component architecture for High Performance Computing and Parallelism". Not applicable for this period.

T3.6: "Document and modularise SAGEMATHCLOUD's codebase". From its inception in 2013, SAGEMATHCLOUD has quickly developed into a full featured VRE. Because of the tight development cycles, it is quite difficult to keep track of SAGEMATHCLOUD's structure and goals. The goal of this task is to participate in the evolution of SAGEMATHCLOUD, by helping with documentation and interoperability.

The first deliverable of the task, D3.2: "Understand and document SAGEMATHCLOUD backend code." was delayed because of the slow recruitment process in UPSud. We were nevertheless able to catch up on time for the first reporting period, the result being a remarkable contribution[1] by Erik Bray to the internal documentation of SAGEMATHCLOUD.

On the other hand, D3.4: "*Personal* SAGEMATHCLOUD: single user version of SAGEMATHCLOUD distributed with SAGE.", due in month 24, was achieved by the SAGEMATHCLOUD developers *before the start of OpenDreamKit*. We are currently evaluating the most useful way to re-allocate the planned effort.

T3.7: "Improving the development workflow in mathematical software". This task seeks new ways of accepting contributions to mathematical software in a scalable way. No deliverable is due for the evaluation period.

T3.8: "Python interface for OOMMF micromagnetic simulation library". This task provides a Python interface to the Object Oriented Micromagnetic Framework (OOMMF). This allows to access the capabilities of this package as one component of a virtual research environment, together with the existing ecosystem of scientific python libraries and tools. The task has been completed [**Beg2017a**], and the resulting software is available online on GitHub[2] and through the Python packaging index under the name `oommfc`. The interface has been presented to users in the micromagnetic community through our dissemination workshops (**T2.7**). This task is part of a number of steps towards a virtual research environment for micromagnetic simulations (D2.13).

[1] https://github.com/sagemathinc/smc/blob/master/src/doc/design_overview/overview.rst

[2] https://github.com/joommf/oommfc

1.2.4. WorkPackage 4: User Interfaces

The objective of WorkPackage 4 is to provide modern, robust, and flexible user interfaces for computation, supporting real-time sharing, integration with collaborative problem-solving, multilingual documents, paper writing and publication, links to databases, etc. This work is focused primarily around the JUPYTER project, in the form of:

- Enhancing existing JUPYTER tools (**T4.2**)
- Building new tools in the JUPYTER ecosystem (**T4.3**, **T4.2**, **T4.8**)
- Improving the use of OpenDreamKit components in JUPYTER and SAGE environments (**T4.1**, **T4.4**, **T4.5**, **T4.12**)
- Demonstrating effectiveness of WorkPackage 4 results in specific scientific applications (**T4.9**, **T4.11**, **T4.14**, **T4.13**)
- Work on Active Documents, which have some goals in common with JUPYTER notebooks (**T4.6**, **T4.7**)

Progress across WorkPackage 4 has been highly successful thus far. Several new software packages have been created, and existing projects in the SAGE and JUPYTER communities have been improved toward sustainability to serve OpenDreamKit objectives.

T4.1: "Uniform notebook interface for all interactive components". The first task for this workpackage is to enable the use of JUPYTER as uniform notebook interface for the relevant computational components. D4.4 has been delivered, providing basic JUPYTER kernels for GAP, PARI, SAGE, and SINGULAR. They are being further improved in preparation for D4.7: "Full featured JUPYTER interface for GAP, PARI/GP, Singular".

D4.5: "SAGE notebook / JUPYTER notebook convergence" has been delivered, further enhancing SAGE's JUPYTER integration and preparing for the systematic transition from the legacy custom-built SAGE notebook application to JUPYTER in the coming months. Beside all the benefits of a uniform and actively developed interface for the user, outsourcing the maintenance of this key but non disciplinary component will save the SAGE community much needed resources and is an important step toward the sustainability of the OpenDreamKit ecosystem (Objective 5).

Progress was particularly fast thanks to a very active involvement of the SAGE community.

T4.2: "Notebook improvements for collaboration". D4.6: "Tools for collaborating on notebooks via version-control" has been delivered in the form of a new JUPYTER package, nbdime, enabling easier collaboration on notebooks via version control systems such as Git. This project was presented at the major Scientific Python conferences SciPy US in July 2016 and EuroSciPy in August 2016, and has been met with enthusiasm from the scientific Python community for its prospect of solving a longstanding difficulty in working with notebooks.

The JUPYTERHUB package has received significant updates and further development, specifically a *Services extension point*, which enables shared workspaces for collaboration, a step on the path toward real-time collaboration for D4.15.

T4.3: "Reproducible Notebooks". D4.8: "Facilities for running notebooks as verification tests" has been delivered in the form of a new Python package, nbval, which enables testing and verification of existing notebooks via a plugin to the Python testing framework pytest. nbval integrates with nbdime from D4.6 to deliver testable, reproducible notebooks via traditional software development testing practices. This work furthers OpenDreamKit objective 5 of promoting sustainable software in math and science.

T4.4: "Refactor SAGE's SPHINX documentation system". As part of the Sage Days 77 workshop, we began the long-term work of refactoring the Sage documentation build system for D4.13: “Refactorisation of SAGE’s SPHINX documentation system” in collaboration with a Sphinx developer. A great deal of work has been done toward the goal of sustainability for SAGE’s vast documentation, including improving reproducibility of builds, updating contents, and increasing reliance on community-standard tools instead of less maintainable bespoke implementations. Improving the maintainability of the SAGE documentation build system is an important part of ensuring high quality documentation going forward.

T4.5: “Dynamic documentation and exploration system”. Due M36 (D4.16) Not applicable for this period.

T4.6: “Structured documents”. Active structured documents are a common need with many use cases, and has many potential solutions. Requirements and venues for collaborations were explored through discussions between participants, in particular at the occasion of Sage Days 77 workshop (see the notes), and June’s ODK meeting in Bremen. The findings were reported in D4.2: “Active/Structured Documents Requirements and existing Solutions”.

In D4.9: “In-place computation in active documents (context/computation)”, We have presented a general framework for in-situ computation in active documents. This is a contribution towards using mathematical documents – the traditional form mathematicians interact with mathematical knowledge and computations – as a user interface for a mathematical virtual research environments. This is also a step towards integrating the two main UI frameworks under investigation in the OpenDreamKit project: JUPYTER notebooks and active documents – see D4.2 – at a conceptual level. The system is prototypical at the moment, but can already be embedded into active documents via a Javascript framework and is ready for use in the OpenDreamKit project. The user interface and SCSCP connections are quite fresh and need substantial testing and optimizations.

T4.7: “Active Documents Portal”. One of the most prominent features of a virtual research environment (VRE) is a unified user interface. The OpenDreamKit approach is to create a mathematical VRE by integrating various pre-existing mathematical software systems. There are two approaches that can serve as a basis for the OpenDreamKit UI: computational notebooks and active documents. The former allows for mathematical text around the computation cells of a real-eval-print loop of a mathematical software system and the latter makes semantically annotated documents active.

MATHHUB is a portal for active mathematical documents ranging from formal libraries of theorem provers to informal – but rigorous – mathematical documents lightly marked up by preserving LaTeX markup.

As the authoring, maintenance, and curation of theory-structured mathematical ontologies and the transfer of mathematical knowledge via active documents are an important part of the OpenDreamKit VRE toolkit, the editing facilities in MATHHUB play a great role for the project, as delivered in D4.3: “Distributed, Collaborative, Versioned Editing of Active Documents in MathHub.info”.

T4.8: “Visualisation system for 3D data in web-notebook”. The current landscape for 3D visualization in Jupyter has been explored and reported on, in order to identify where OpenDreamKit can best contribute to 3D visualization in the notebook, towards D4.12: “JUPYTER extension for 3D visualisation, demonstrated with computational fluid dynamics”. There have been many community-led developments in this area, including the ipyvolume and K3D packages, and OpenDreamKit will further enhance existing community tools to best serve the needs of the community. In particular, developing tools for unstructured mesh visualization and CFD simulations. The SciViJS tool for interactive visualization in a webbrowser has been developed, and will gain better support for use in JUPYTER, also as part of D4.12.

T4.9: "Visualisation of 3D fluid dynamics data in web-notebook". Not applicable for this period.

T4.10: "Common option system for various displays in Sage". Not applicable for this period.

T4.11: "Case study: micromagnetic VRE built from OpenDreamKit". The micromagnetic virtual research environment is hosted in the JUPYTER Notebook. The computational engine is the (existing) OOMMF (Object Oriented MicroMagnetic Framework) which is accessible through the new Python interface that has been created as part of OpenDreamKit (**T3.8**). The JUPYTER Notebook allows us to integrate the problem specification, the execution of the calculation, and the postprocessing and data representation within a single executable document; providing a new computational research environment for micromagnetic simulation that uses the most widely used simulation code. We have enhanced this environment further by exploiting that the notebook allows objects to represent themselves in different ways within the notebook. For example, Python objects that represent mathematical equations in the micromagnetic VRE appear rendered as LaTeX in the notebook. It allows users to interactively compose and explore computational models, and to be able to inspect what they have put together in the language of the scientist (i.e. through equations) rather than through the language of the computer (i.e. code). The addition of this representation options does not stop the code from being valid PYTHON that can be run outside the notebook. We have also provided a graphical representation of the mesh and discretisation cell as the appropriate representation of a finite difference mesh to further assist the effective communication between code and science user, graphical representation of vector field objects, and GUI elements for data exploration. We have used dissemination workshops to seek feedback from users and to refine interface.

T4.12: "Python/Cython bindings for PARI". Completing D4.1: "Python/Cython bindings for PARI and its integration in Sage" happened to be more difficult than originally planned. The high level of coupling between SAGE internals and the PARI interface made it very delicate to pull the latter out of the SAGE codebase. Initially planned to be delivered in month 6, it was only completed in month 16.

The process of making this deliverable possible led to a great amount of refactoring inside the SAGE project. The benefits go beyond this deliverable: implementations of PPL and fpLLL Python interfaces indirectly relied on this work.

As summarized in Trac ticket 20238, it required close to 50 *tickets*, which fell in the following categories:

- Moving SAGE's C interrupts and signals API to a separate PYTHON/CYTHON package called cysignals.
- Decoupling SAGE's PARI interface from the *coercion model*.
- Upgrading the PARI interface to the latest upstream version (2.9.2).
- Cleaning up the PARI interface API, by removing unneeded object orientation and external dependencies.
- Moving the PARI interface to a separate PYTHON/CYTHON package CyPari2, depending on cysignals.

The end results of this work are the packages cysignals and CyPari2, both installable in a pure PYTHON environment via the standard tool `pip`. Starting from version 8.0, installation via `pip` is SAGE's default way of providing the PARI interface.

T4.13: "Demonstrator: micromagnetic VRE notebooks". Not applicable for this period.

T4.14: "Online portal for micromagnetic VRE demonstrator". Not applicable for this period.

1.2.5. WorkPackage 5: High Performance Mathematical Computing

Workpackage 5 is about the development of high performance computing tools in mathematical virtual research environments. It is addressed at the level of each kernel library composing the computational tools of the project (PARI, GAP, LINBOX, MPIR, SAGE, SINGULAR, ...), and also at the level of interfacing and exposing core parallel features to higher level programming interfaces.

Key results obtained over the period for WorkPackage 5 are the following:

- A fine grain parallelisation of matrix Fast Fourier Transform code in FLINT, delivering high and scalable performances.
- Parallelization of relation Sieving code in FLINT.
- A new super-optimizer for vectorized assembly code and its exploitation to improve the performances of the MPIR code.
- A MapReduce framework implementation to parallelize huge (out of core) datasets from combinatorics presented as recursion trees.
- CYTHON can now use PYTHRAN to compile and vectorize NUMPY code.
- A new Sun Grid Engine notebook spawner for JUPYTER to drive interractive computations on HPC clusters from a Jupyter notebook.

T5.1: "PARI". No deliverable is due for the reporting period. The generic paralellization engine in the PARI C-library has reached its maturity. It is now part of the standard releases of PARI and D5.10: "Devise a generic parallelisation engine for PARI and use it to prototype selected functions (integer factorisation, discrete logarithm, modular polynomials)" will be delivered on time on month 24.

T5.2: "GAP". No deliverable is due for the evaluation period but steady progress was made on Deliverable D5.15: "Final report and evaluation of all the GAP developments.". Over this period, six releases were cut incorporating contributions to Deliverable D3.11: "HPC enabled SAGE distribution".

Another major direction of efforts is the HPC-GAP integration: HPC-GAP is a fork of GAP initiated during the SCIEnce project, which enables multithreaded calculations. Now that HPC-GAP has reached maturity, it's critical for its widespread use and long term maintainability to merge it back into GAP's master branch. The first step towards this long-standing goal, which is at the core of Task **T5.2**, is the next major release of GAP 4.9, planned in 2017 and allowing compilation in HPC-GAP compatibility mode. The progress towards this release can be tracked on the GAP 4.9.0 milestone on GitHub: `https://github.com/gap-system/gap/milestone/4`. A culmination of the work in this direction was the merge of a major refactorisation of GAP's build system mainly developed by our external collaborator Max Horn (Giessen) into GAP's master branch during the Sage-GAP-Days-85 we organized in March 2017. Some GAP packages also required updates to work in GAP 4.9.

T5.3: "Linbox". No deliverable is due for this evaluation period. The design of a domain specific language (DSL) exposing parallel features through the library stack up to the SAGE interface is making good progress (deliverable D5.9: "Library design and domain specific language exposing LINBOX parallel features to SAGE"); the demonstrator code for matrix inverse over a finite field is under review. Progress on deliverable D5.12: "Exact linear algebra algorithms and implementations. Library maintenance and close integration in mathematical software for LINBOX library" is twofold:

- Maintenance, and close integration of the Library with SAGE has been active (Ticket #17635 finally updates SAGE to LinBox-1.4.2 after a long and technical review process).

- Several algorithmic innovations have been produced: a new matrix invariant for rank profiles [**DumPerSul:fcrpmgbd16**], more efficient representation for quasiseparable matrices [**Pernet:cqm16**; **PerSto:tsegqm17**], and new developments on interactive certificates for the security of large scale distributed computations on unsafe resources [**DumKalTho:lticmpds DumLucPer:cftearp17**]. These results have or will be presented in the main venues in the domain: the international conference ISSAC'16-17 and the Journal of Symbolic computation.

T5.4: "Singular". The SINGULAR library, for commutative algebra, already has a framework to design high level parallel algorithms and handle shared memory safely. Yet, a few kernel routines, such as matrix Fast Fourier transform, relation sieving algorithms need a special attention as they have to rely on a highly optimized fine grained parallelization. Deliverables D5.7: "Take advantage of multiple cores in the matrix Fourier Algorithm component of the FFT for integer and polynomial arithmetic,and include assembly primitives for SIMD processor instructions (e.g. AVX, etc.), especially in the FFT butterflies." and D5.6: "Parallelise the relation sieving component of the Quadratic Sieve and implement a parallel version of Block-Wiederman linear algebra over GF2 and implement large prime variants" for this reporting period only deal with these kernel routines, in the FLINT library, a dependency of SINGULAR.

Deliverable D5.7 is a parallelization of the matrix fast Fourier transform in two ways: using Open-MP standard for shared memory multi-core parallelization on one hand and the SIMD vectorized instructions of each core on the other hand. Performance experiments demonstrate the effectiveness of the design of thread parallelization, reaching speed-up factors up to 3 out of 4 cores, and up to 5 out of 8 cores, which is remarkable given the high memory bandwidth of the kind task considered here. The SIMD optimization of the FFT butterfly partly relies on the super-optimizer results of Deliverable D5.5: "Write an assembly superoptimiser supporting AVX and upcoming Intel processor extensions for the MPIR library and optimise MPIR for modern processors.". Here again, experiments demonstrate a strong improvement in practice for the new code produced.

Deliverable D5.6 now focuses on one major type of application of the library: relation sieving for factoring medium sized integers. There, use of SIMD optimizations did not provide strong enough improvements and the work mostly focused on the thread paralellization of the reliation sieving. The resulting code in FLINT improves over PARI's implementation for large enough integers (above 170 bits) in sequential already, and benefits from a speed-up factor of about 3 out of 4 cores. On the contrary, the parallelization of the block Wiedemann algorithm mostly benefitted from SIMD optimizations and much less from multithreading.

T5.5: "MPIR". The MPIR library is a core component of many computational mathematics software providing highly optimized basic arithmetic over multiprecision integers.

Deliverable D5.5 is a super-optimizer, performing exhaustive search of permutations in a sequence of assembly instructions, combined with a fine clock cycle count to elect the most efficient variant.

The main unexpected obstacle has been the production of a precise clock cycle counter, on which to base the benchmarks: recent. Frequency scaling strategies of the CPU led to noise in the measurements, while the real CPU cycle counter would require patching a kernel module to allow access in user mode. This problem has been solved thanks to the pmu-tools API to the performance system of the Linux kernel.

Then super-optimizer could reliably produce optimized code for the main functions of the MPIR library (multiplication, gcd, and extended gcd) on the most common recent vectorized architectures: AMD Bulldozer, Intel Skylake and Haswell. Experiments comparing the resulting new routines with the current ones, showed a significant improvement in performances in almost all cases, thus validating this approach and motivating its application to a broader set of routines.

T5.6: "HPC infrastructure for combinatorics". The goal of this task is to use combinatorics as a source of challenges to experiment on various HPC techniques. In the first deliverable D5.1: "Turn the Python prototypes for tree exploration into production code, integrate to SAGE.", we successfully implement a MapReduce programming model on large datasets described by a recursion tree, which are too big to fit in memory. After chasing around some bug on MacOS posix support, the code was integrated in Sage (Trac Ticket 13580) and presented at the "journée du groupe de travail LaMHA" at the Université Pierre et Marie Curie on November the 26th of 2016.

Since it is written purely in Python, the code doesn't perform well when the computation in each node is short. A good technology for handling such situations with fine grain parallelism, seems to be the CILK++ runtime, and we are currently doing experiment which goal is to have a two stage parallel computation, where CILK++ is doing the load balancing on a machine (shared memory) and Python the load balancing among machines. As a base for our experiment, we are working on the enumeration of numerical monoids; indeed, it is a very challenging problem as the explored recursion tree is extremely unbalanced. We are currently able to have a code generating and process 50Gio/s on a single 8 core i7 machine. When the problem of re-balancing the work among several machines will be solved we hope to be able to have throughput higher than 1To/s on a network of machines. To this end, we are also experimenting with the `Spark` technology for distributed computations.

T5.7: "Pythran". Mathematical software, such as SAGE, intensively rely on the Python language for its expressivity. In order to harness most of a CPU computing efficiency, critical code in such interpreted languages need to be compiled into C code. This is precisely what CYTHON and PYTHRAN are offering. The former supports a broad range that PYTHON constructs, while the latter focuses on optimizing NUMPY constructs for linear algebra. The purpose of this task is to:

(1) Implement a convergence between these two compilers
(2) Expose the new capacity to optimize NUMPY code to the developers and users of SAGE.

Concerning target 1, the delivery of D5.2: "Facility to compile PYTHRAN compliant user kernels and SAGE code and automatically take advantage of multi-cores and SIMD instruction units in CYTHON" successfully implements the proposed convergence: CYTHON is now able to delegate compilation tasks to PYTHRAN whenever NUMPY code is detected. In practice, benchmarks show that resulting code executes faster. For instance computing euclidean norm of a large floating point vector is sped up by a factor of 2.5 without vectorization and a factor 3.7 with AVX2 vectorization enabled.

As for target 2, a technical lock is that PYTHRAN's analysis of PYTHON typing used to be too weak to support the breadth of Object Oriented programming style of SAGE. Deliverable D5.4: "Make PYTHRAN typing better to improve error information." addressed this issue by strongly enhancing PYTHRAN's type inference system in two ways: first, the compiler now more accurately tracks the identifier $\leftrightarrow$ value binding, which in turns makes it possible to generate strongly typed code for a wider class of PYTHON kernels. Second, an unsound type checker for PYTHRAN has been developed. It provides human-readable error report when a type error is detected at compile time, when a cryptic internal error was previously reported. Both algorithms have been extensively detailed in separated blog posts and the resulting implementation is part of the official PYTHRAN 0.8.0 release.

Further work on PYTHRAN's integration to SAGE and the exposition of parallelism via CILK++ is still in progress and should be delivered in D5.8: "Explore the possibility to interface smoothly PYTHRAN, CYTHON and `Cilk++`." at M24.

T5.8: "Sun Grid Engine Integration in Project JUPYTER Hub". It is common for academic High Performance Computing (HPC) clusters to make use of schedulers based on Sun Grid Engine with Son of Grid Engine as one of the most popular. It is used, for example, on the institutional HPC systems in the Universities of Sheffield and Manchester in the United Kingdom. It is also used on the regional N8 HPC facility, a system shared by the eight most research intensive universities in the North of England.

In deliverable D5.3: "Sun Grid Engine support for Project JUPYTER Hub", we have developed and demonstrated a Sun Grid Engine notebook spawner for Project Jupyter, allowing users to easily access Jupyter notebooks on HPC clusters directly from the web-browser. This development allows users with no background in High Performance Computing to easily migrate workflows from laptop to HPC cluster, allowing them to access greater resources with no additional training required.

1.2.6. WorkPackage 6: Data/Knowledge/Software-Bases

In a series of workshops (September 2015 in Paris, January 2016 in St. Andrews, June 2016 in Bremen, and July 2016 in Białystok) the participants working on WP6 met and discussed the topic of integrating the OpenDreamKit systems into a mathematical VRE toolkit. Key results were

R1. the observation that *knowledge-aware interoperability of software and database-systems is the most critical objective* for WP6 in the OpenDreamKit project.

R1. the consensus that this can be achieved by *aligning the mathematical knowledge underlying the various systems*.

This requires explicitly representing the three aspects of math VREs – Data (D), Knowledge (K), and Software (S) – and basing computational services and inter-system communication on a joint $\mathcal{DKS}$-base. These results are engrained in the "Math-in-the-Middle" (MitM) paradigm [**DehKohKon:iop16**], which gives a representational basis for specification-based interoperability of mathematical software systems – so that they can be integrated in a VRE toolkit. In the MitM paradigm, the mathematical knowledge underlying the VREs (K) and the interface for each system (S) are represented as modular theory graphs in the OMDoc/MMT format. For the data aspect (D) we have extended the concept of OMDoc/MMT theories to "virtual theories" that allow the practical management of possibly infinite theories, see [**ODK-D6.2**] for details.

A side effect of **R1.** is that the verification aspects anticipated in the proposal are non-critical to the OpenDreamKit project. In particular the value of the exemplary verification of an LMFDB algorithm in **T6.8** and deliverable D6.8: "Curated Math-in-the-Middle Ontology and Alignments for GAP/ SAGE/ LMFDB" seems highly questionable.

Correspondingly we have refined the notion of "triformal theories" coined in the proposal into the concept of "$\mathcal{DKS}$ theory graphs", which can be formalized and implemented without the extension of OMDoc/MMT for "biformal theories" anticipated in the proposal.

Through the concerted effort of the WP6 participants, we have been able to implement this design into prototypical $\mathcal{DKS}$ base patterned after the MitM paradigm with virtual theories, generating interface theory graphs for the GAP and SAGE systems and integrating the LMFDB system via the MitM codec architecture described in [**ODK-D6.2**]. Based on this, we were able to generically integrate GAP, SAGE, and LMFDB via the standardised SCSCP protocol [**HorRoz:ossp09**] – essentially remote procedure calls with OpenMath Objects. This case study shows the feasibility of the initial design of $\mathcal{DKS}$-bases; further investigations and the integration of additional systems will determine the practicability.

T6.1: "Survey of existing $\mathcal{DKS}$ bases, Formulation of requirements". This task was directly addressed in the WP6 workshops in the first year.

T6.2: "Triform Theories in OMDoc/MMT". For this task we have specified and implemented the concept of virtual theories that can contain large – theoretically even infinite – numbers of declarations and objects (e.g. 3.5M declarations in the LMFDB data base for elliptic functions) in OMDoc/MMT. Virtual theories are characterized by the fact that they are too large to keep in main memory of the MMT System and have to be partially and lazily imported from an external data store. We have reported on the results in D6.2: "Initial $\mathcal{DKS}$ base Design (including base survey and Requirements Workshop Report)".

T6.3: "$\mathcal{DKS}$ Base Design". This task was directly addressed in the WP6 workshops in the first year and has led to the design and implementation in D6.2. The design has been published at an international conference (CICM 2016) and is now being implemented.

T6.4: "Computational Foundation for Python/Sage". In the course of the deliberations in the WP6 workshops we saw a shift from the development of computational foundations and verification towards API/Interface function specifications to enable semantic system interoperability via the Math-in-the-Middle Ontology. Consequently, emphasis has changed to the generation of API Content Dictionaries (API CDs) for GAP, LMFDB and SAGE. We have a prototypical set of GAP and SAGE Content Dictionaries in OMDoc/MMT form (GAP: 218 CDs, 2996 entries; SAGE: 512 CDs, 2800 entries overall). The computational foundations exist but are rather more simple than originally anticipated. Much of the functionality has been offloaded to the SCSCP standard – remote procedure call with OpenMath representations of the mathematical objects – developed in the SCIENCE Project. As a direct consequence of the work in OpenDreamKit the OpenMath Society has promoted the SCSCP protocol into as an OpenMath Standard.

Conversely, the GAP and SAGE CDs are rather more elaborated than anticipated in the proposal, and thus form a viable basis for alignment with the MitM Ontology.

T6.5: "Knowledge-based code infrastructure". This task has just started, and the MitM architecture developed in WP6 has given important impulses to make the code infrastructure of SAGE and GAP more declarative (knowledge-based). In SAGE, the category infrastructure was validated (it seems to be the right level of abstraction to generate API CDs) and extended; further annotations were added into the SAGE code base and accepted by the community. In GAP, the facilities for "constructors" was reformed, extended by an infrastructure for documentation and static typing/type analysis, and the code base refactored for 6000 constructors. Similarly, the online documentation subsystem for GAP has been regularized and synchronized with the constructor level. Already at this early stage of the task the new "knowledge-based perspective" has revealed a plethora of errors and inefficiencies and has contributed to the code quality in both systems.

T6.6: "OEIS Case Study (Coverage and automated Import)". For the OEIS case study we have parsed the OEIS data and converted it into OMDoc/MMT theories (ca. 260,000). The main problem solved here was to parse the formula section (generating functions, relations between sequences, ...): they are represented in a human-oriented ASCII syntax, which is highly irregular, ill-separated from surrounding text, and interpunctuation. Nonetheless we managed to recover ca. 90% of the formulae and

i) generate ca. 100,000 new relations between sequences and

ii) provide a package of ca. 50,000 generating functions to Sage (which can be used e.g. in the FindStat database).

We use this theory set to test the functionalities of "virtual theory graphs" (one step up from the "virtual theories" developed in **T6.3**).

T6.7: "FindStat Case Study (Triformal Theories)". Not applicable for this period

T6.8: "LMFDB Case Study (Triformal Theories)". Work on this task has started. Given the concept of virtual theories developed in **T6.2** the task is to build a database connector that converts the MongoDB tables in LMFDB into "mathematical objects". We have identified the problems – e.g. that objects are reduced to ad-hoc database records: for instance elliptic curves are represented as a quadruple of integers, where the last is represented as a string of digits as the range of MongoDB integers is too small. We have developed an architecture of language-specific Codecs which mitigate these problems in a knowledge-centered way (Codecs are OMDoc/MMT objects) that interpret database records as OMDoc/MMT objects and can thus be used populate virtual theories. The next step is to extend the existing MMT query language by a query compiler into the underlying data store system; concretely to MongoDB underlying LMFDB for **T6.8**.

T6.9: "Memoisation and production of new data". Not applicable for this period

T6.10: "Math Search Engine". Work on the first work phase has proceeded as planned and has culminated in D6.1: "Full-text Search (Formulae + Keywords) over LaTeX-based Documents (e.g. the arXiv subset)". The second work phase on this task presupposes the Math-in-the-Middle ontology (as we call it now.) Where we already have that, e.g. for the OEIS (see **T6.6**) we already have a running search engine. The main problem here is to devise intuitive query interfaces and integrate them into the OpenDreamKit VRE framework.

1.2.7. WorkPackage 7: Social Aspects

Ursula Martin has stepped down in anticipation of her upcoming retirement, and Dmitrii Pasechnik has become the lead PI for this work package. This somewhat slowed down the takeoff of this work package, but the important deliverables are well on track, if not ahead of schedule. Nevertheless, the consortium may need to request for a new amendment after the Review of the 1st Reporting Period to clarify the situation.

As planned, the work was mainly focused on bootstrapping **T7.1** and **T7.3** whose early outcomes will nurture the design of OpenDreamKit's VREs in other work packages. As well, good progress was booked on **T7.2**.

T7.1: "Social Science Input to Design". For deliverable D7.1: "The flow of code and patches in open source projects", we surveyed the methodology, data, and tools needed to assess development models of large-scale academic open-source systems like model system SAGE. Part I of D7.5: "Report on relevant research in sociology of mathematics and lessons for design of OpenDreamKit VRE, parts I-III, with part I (resp. II) due at month 3 (resp. 24)" is also ready and available as an internal publication.

T7.2: "Implications of VREs for Publication". Early work was done on connections of reproducibility, crowdsourcing, and a VRE as a mean to test and control the former: a paper [**2016arXiv160100181C**] analysing a concrete well-established area of combinatorics in this respect, and describing the implementation of the corresponding meta-database in SAGE system, has been published. Another paper [**2016arXiv160301710P**], again touching upon reproducibility and correctness of mathematical results, and utilizing extensive computations in GAP, has been published.

T7.3: "Mechanism Design for Free Software Development". Deliverable D7.2: "TRAC add-on to manage ticket prioritisation" is largely ready and is to be tested on the system SAGE; a paper [**Pavlou:2016:MCI:2936924.2936934**], forming a part of D7.7: "Game-theoretic analysis of development practices in open-source VREs" has been published.

T7.4: "Evaluation of Micromagnetic VRE". Not applicable for this period.

1.3. Impact

All the information of section 2.1 of the DoA is still relevant. There is for now no change to bring to Key Performance Indicators. The evolution of the measures between Month 18 and Month 36 will allow the Coordinator to evaluate if the selected KPI are appropriate.

1.4. Infrastructures

Per design, OpenDreamKit focuses on delivering "a flexible toolkit enabling research groups to set up Virtual Research Environments". As such, there is no e-infrastructure deployed and managed by OpenDreamKit. Instead, there are many e-infrastructures that use the software developed or contributed to by OpenDreamKit, and we regularly help with new or updated deployments.

Some of the typical content of this section (e.g. Selection Panel, ...) is therefore irrelevant for OpenDreamKit, and we simply provide some informal information and figures on the main existing deployments and their typical public, together with some assessment of the impact we had on them.

- cloud.sagemath.org With 500k accounts worldwide and 30k active projects both for research and education, SAGEMATHCLOUD is the largest Virtual Research Environment based on the ecosystem OpenDreamKit contributes to. Predating OpenDreamKit, it benefits back from most of our actions. OpenDreamKit has been contributing to a healthy collaboration/competition relation between JUPYTERHUB and SAGEMATHCLOUD, with the competition occurring only at the level of specific individual components and both teams learning from each other.

- jupyter.math.cnrs.fr We have helped setup this JUPYTERHUB service, deployed by the French CNRS for the benefit of the personnel of all math labs in France. This service includes all the OpenDreamKit computational components.
- mybinder.org Binder is a web service that makes it easy for any user to publish live notebooks based on an arbitrary reproducible executable environments. It thus fosters dissemination and reproducible research. The current main instance (`http://mybinder.org/`) is overloaded by the demand, proving that it has identified just the right service for a critical need.

  Our work on packaging **T3.3** and JUPYTER integration **T4.1** is about to enable the easy definition of executable environments including OpenDreamKit's computational math software.

  We are further reaching toward EGI/EUDAT to use their e-infrastructure to contribute additional computing resources to the main mybinder instance or setup a new one for the EC community. `https://github.com/OpenDreamKit/OpenDreamKit/issues/205`
- JupyterHub at USFD JUPYTERHUB instance deployed on USheffield's HPC system.

Many other instances are being deployed by universities (e.g. university Paris Sud) for their personnel. We are keeping track of those we are aware of at `https://github.com/OpenDreamKit/OpenDreamKit/issues/174`.

2. UPDATE OF THE PLAN FOR EXPLOITATION AND DISSEMINATION OF RESULT (IF APPLICABLE)

Not applicable

3. 4. FOLLOW-UP OF RECOMMENDATIONS AND COMMENTS FROM PREVIOUS REVIEW(S) (IF APPLICABLE)

Recommendation 1. (i) "An advisory statement should be present at the beginning of each document stating that any links given were in place and reference-able as at the date of submission of the deliverable."

Each deliverable ends with a disclaimer saying "Disclaimer: this report, together with its annexes and the reports for the earlier deliverables, is self contained for auditing and reviewing purposes. Hyperlinks to external resources are meant as a convenience for casual readers wishing to follow our progress; such links have been checked for correctness at the time of submission of the deliverable, but there is no guarantee implied that they will remain valid."

Recommendations 2 and 3. (ii) "All links should be fully expanded and given in the text, together with an outline of their significance and relevance/importance to the project. If a document uses links as references, this should be treated as if they were normal academic references and should be set out in a separate section as if they were academic references in conformance with current best academic practice for such links."

(iii) "The same should also apply to change requests/records (eg Trac Records) with their effect being set out in detail. This is particularly important for links to external databases of changes."

Reports are now fully consistent for the understanding for the carried out work. Hyperlinks are now meant merely as convenience for the reader if they wish to go into further details. Furthermore, all referred work is now presented according to academic norms of bibliography.

Recommendation 4. (iv) ”Efforts should be made by the project and the Commission to find a mechanism to ensure the long-term sustainability of the outputs and results beyond the lifetime of the project. This may necessitate maintaining duplicates of these databases by the Commission.”

Not in the hands of the consortium.

4. Deviations from Annex 1 (if applicable)

There was no major deviation from Annex 1. All deliverables due for M18 were delivered within the timeframe of the 1st Reporting Period, and all milestones in this period were reached. Slight modifications were brought to WP5 and WP6 and were included in the AMD-676541-13.

Some deliverables names in WP5 were modified

(1) D5.5 from ”Extend the existing assembly superoptimiser for AVX and upcoming Intel processor extensions for the MPIR library” to ”Write an assembly superoptimiser supporting AVX and upcoming Intel processor extensions for the MPIR library and optimise MPIR for modern processors”
(2) D5.6 from”Parallelise the relation sieving component of the Quadratic Sieve and implement a parallel version of Block-Wiederman linear algebra over GF2 and the triple large prime variant” to ”Parallelise the relation sieving component of the Quadratic Sieve and implement a parallel version of Block-Wiederman linear algebra over GF2 and implement large prime variants”
(3) D5.7 from ”Take advantage of multiple cores in the matrix Fourier Algorithm component of the FFT for integer and polynomial arithmetic, and include assembly primitives for SIMD processor instructions (AVX, Knight’s Bridge, etc.), especially in the FFT butterflies” to ”Take advantage of multiple cores in the matrix Fourier Algorithm component of the FFT for integer and polynomial arithmetic, and include assembly primitives for SIMD processor instructions (e.g. AVX, etc.), especially in the FFT butterflies”

Some titles of the WP6 deliverables were modified to respect the change in priority on system interoperability and distributed computing in the “Math-in-the-Middle” Paradigm over algorithm verification: D6.5: “GAP/SAGE/LMFDB Interface Theories and Alignment in OMDoc/MMT for System Interoperability”, D6.7: “GAP/SAGE Interface Views OMDoc/MMT”, and D6.8: “Curated Math-in-the-Middle Ontology and Alignments for GAP/ SAGE/ LMFDB”

4.1. Tasks

No deviation from the tasks. All workplan is on time at the end of the Reporting Period.

4.2. Use of resources

All changes of use of resources were included in the two amendments previously cited and were due to modifications in the personnel. Those adjustments were due to the change of positions of some key OpenDreamKit participants and expected difficulties in hiring planned staff. The work plan has been updated accordingly, with no foreseeable impact on the achievement of tasks, deliverables, and milestones.

4.3. Unforeseen subcontracting (if applicable)

Not applicable.

4.4. Unforeseen use of in kind contribution from third party against payment or free of charges (if applicable)

Not applicable.

Disclaimer: this report, together with its annexes and the reports for the earlier deliverables, is self contained for auditing and reviewing purposes. Hyperlinks to external resources are meant as a convenience for casual readers wishing to follow our progress; such links have been checked for correctness at the time of submission of the deliverable, but there is no guarantee implied that they will remain valid.