

IAC-12-D5.2.11

COLLABORATION ON ISS EXPERIMENT DATA AND KNOWLEDGE REPRESENTATION

E.A.Kuijpers
National Aerospace Laboratory NLR, Netherlands, Ed.Kuijpers@nlr.nl

L. Carotenuto
Telespazio s.p.a, Italy, luigi.carotenuto@telespazio.com

J.-C.Malapert
CNES, France, Jean-Christophe.Malapert@cnes.fr

D. Markov-Vetter
MUSC-DLR, Germany daniela.markov-vetter@dlr.de

I. Melatti
Sapienza University Rome, Italy, melatti@di.uniroma1.it

A. Orlandini
Italian National Research Council - CNR,, Italy, andrea.orlandini@istc.cnr.it

R. Pinchuk
Space Applications Services, Belgium, rani.pinchuk@spaceapplications.com

The USOCs (User Support and Operation Centres) are a network of collaborating centres. They have been established in various EU countries with the support of national space agencies and are engaged by the European Space Agency (ESA) to conduct the operations for European scientific experiments on board the International Space Station. The USOCs Knowledge Integration and dissemination for Space Science Experimentation (ULISSE) project developed a platform and tools for improving preservation, valorisation and exploitation of experiment data. The platform integrates tools which are based on a number of formalisms to represent knowledge to be maintained and preserved. Dedicated tools for authoring metadata and knowledge about datasets have been developed to enforce a standard. Knowledge about internal data formats guided a data valorisation framework based on experiment data integration. A timeline based representation to model possible temporal evolutions of scientific payload components as well as to support the automatic definition of on-board activity schedules for performing the associated scientific experiments. Immersive visualization techniques enable intuitive exploration of knowledge about data for better understanding the context. Virtual Reality and Augmented Reality design issues for ordinary end-users were analysed. The project experience in knowledge representation and progress is addressed in support of improving preservation and valorisation of space data, representing intellectual ownership and data dissemination policies, stimulating data reuse and collaboration.

1. INTRODUCTION

In support of improving ISS data experiment utilisation an environment has been created to support distributed access and representation of experiment data sets. This was triggered by the need to improve utilisation of ISS experiment data while addressing the following issues:

- Re-utilisation of data for further deepening of scientific analysis;
- Cross-fertilisation between different scientific areas;
- Generation of spin-off for terrestrial and/or space applications;

- Improving information utilisation

The environment was created in an EU FP7 framework project called ULISSE (USOCs Knowledge Integration and dissemination for Space Science Experimentation).

In this paper, progress and further work is reported. First background of the ULISSE project is described followed up with a discussion of knowledge representation needed for various issues. The integration and experiences conclude the paper.

2. BACKGROUND ULISSE ENVIRONMENT

2.1 Objectives

The following objectives guided the development of the ULISSE environment [1]:

- improving utilisation of space experiment data
- improving preservation of space data
- improving valorisation of research in space
- representing intellectual ownership and data dissemination policies
- stimulating data reuse and collaboration
- developing a demonstrator environment

2.2 ULISSE project overview

The ULISSE consortium is co-ordinated by Telespazio S.p.A. and involves participation of Telespazio SSA (formerly MARS Center S.r.l.) UniRoma and National Research Council of Italy (CNR-ISTC) (Italy), B-USOC, and Space Applications Services(Belgium), MEDES-IMPS and CADMOS-CNES (France), DAMEC (Denmark), the National Aerospace Laboratory NLR (involved in Erasmus USOC, the Netherlands), MUSC-DLR and WERUM(Germany), ETH Zurich(Swiss), N-USOC (Norway), SRC PAS(Poland), UPM(Spain) and ELGRA (European Low Gravity Research Association).

As part of the project a distributed infrastructure has been developed: data repositories and tools for data preservation and exploitation have been developed and integrated through the middleware for the provision of services to the users. The middleware was designed to ensure a continuous enlargement of the ULISSE network by providing the means to integrate any further node that would become available in the future, providing additional data and/or tools.

On this basis, a network architecture for connecting the different ULISSE services has been defined, being compliant with the security requirements of each node and avoiding any overlap with the operative infrastructure of the USOCs. Services were defined towards different typologies of users, mainly in the framework of scientific community, but taking into account also space industries, space agencies, decision-makers, educational bodies and general public.

The project includes specific dissemination activities: scientific as well as general publications for a broader audience, public events and educational activities on space research. In this way the project intended to increase the involvement of specialized communities and the awareness of general public. In this context, a number of dissemination actions, as

congress presentations, publications, presentations at press/special events, have been realized over a three year period. The 3 years of the project resulted in around 40 publications and one book [2].

2.3 Knowledge representation issues

As a first step, an inventory was made of data from previous space experiments on ISS as well as data from other space platforms, like sounding rockets, Foton capsule, Space Shuttle, etc. based on the ULISSE consortium experiment datasets and experiences

The inventory has showed a broad range of formats and distributed datasets in many disciplines requiring a standardized representation. The development of a metadata standard was necessary needed in order to represent the datasets in a uniform way. The datasets coming from space experimentation are typically multi-disciplinary and include both scientific and technical knowledge which need to be represented. The use of Topic Maps (ISO/IEC 13250:2003) as a basic technology to represent this type of knowledge has been elaborated

For the datasets different internal data structure are used. To enable combination of datasets, the data format with corresponding access tools needs to be represented.

The experiments need to be planned and scheduled. This requires knowledge about the facility and the local and remote operations.

The experiment data are generated with various three dimensional and time relationships. Augmented Reality and Virtual Reality techniques for 3-D model visualisation represent knowledge about the relationships.

2.4 Collaboration issues

The datasets are distributed over many locations. The USOCs have a responsibility in ensuring the availability for many years of raw data produced by the respective experiments. Partially intermediate processing results may be stored. In addition science and engineering groups have dedicated archives containing subsets of data relevant for future use. Normally, all documents needed for data processing are collected at USOCs, which have then the expertise to create metadata and semantic information.

Next to the technical access of archives there is a need to have organisational aspects in place related to security and intellectual property rights. The collaboration requires a trade-off between centralized and distributed archiving for data and for metadata.

3. REPRESENTING DISTRIBUTED DATASETS

3.1 Overview

To be able to have access to distributed datasets a dedicated toolset was needed for representation. The data sources can be located at USOCs and other dedicated centres. This resulted in the SITools2 software package [3], an open source framework developed by CNES with AKKA technology as subcontractor. A part of this development has been funded by ULISSE project. The purpose of this framework is dual.

The first aim was to setup easily an archival system based on the Open Archival Information System standard (OAIS). The OAIS model is an international ISO standard (ISO 14721:2003) that has been adopted for guiding the long-term preservation of digital data and documents. The OAIS model was developed by the Consultative Committee for Space Data Systems (CCSDS) in 2002, and was adopted as an ISO standard in 2003. The OAIS model is simply a set of standardized guidelines that breaks down an archive into six functional entities to preserve digital data in a long term:

- Ingest;
- Archival storage;
- Data management;
- Administration;
- Preservation planning;
- Access.

In addition to these six entities, OAIS model defines the concept of information as a combination of data and representation information. The data is the digital data to preserve and the representation information being information that allows for the full interpretation of the data into meaningful information.

The second aim was to federate the development in scientific laboratories. In this perspective, the framework has been designed to be extensible so that developers in scientific laboratories register their own services in the SITools2's API.

3.2 SITOOLS Architecture approach

SITools2 is a client-server architecture based on REST (Representational State Transfer) architecture. RESTlet has been chosen as a REST framework as a result of CNES study in the end of 2009. The main reasons of this choice were the capabilities of this framework in terms of features and extensions on the server side. The server, based on RESTlet, is composed of two APIs: one handling the administration capabilities of the system and another one handling the interaction with the users. At this level, a software client can interact with the system to discover data.

This style of architecture has been chosen for the following features:

- an API independent of specific technologies,
- Uniform Resource Identifier (URI) that identifies a data as well as an unique location over the network. As a consequence, URI can be used for cache and data preservation access mechanism,
- a set of associated data representations (e.g: XML, JSON, HTML) for each URI allowing to adapt the server's response according to the client needs,
- a simple way to secure the resources.

The chosen client to interact with the server is a RIA (Rich Internet Application) client based on AJAX (Asynchronous JavaScript And XML) technology. AJAX offers a better interaction with the user because it is able to run asynchronous tasks. This means that one part of the web page can be used by a user whereas another part is loading some graphical components. EXT-JS is a JavaScript framework which has been selected as the RIA API. The main reasons of this choice were based on the quality and availability of graphical components as well as the inheritance capability of these graphical components. These features allow customization of graphical components by developers. The following picture shows the two web clients (administration web page and user web page) calling the REST APIs. The main business capabilities of SITools2 are presented in the red squares.

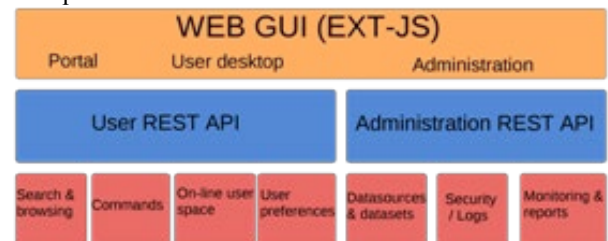


Fig. 3-1. Web-clients calling the REST API

SITools2 can be viewed as a set of layers that is composed of a web client layer, a component server layer, a resource & representation layer and a business & storage layer. Each layer is described as below:

- Web client layer: This layer contains several client applications including the portal, user desktops and the administration panel. These applications are based on both Sencha JavaScript framework – ExtJS and a MVC (Model View Controller) pattern. The “model” role from the MVC pattern is represented by EXTJS Store classes. The “View” role consists in a set of graphical classes (form, panel, grid ...) brought together in the same web page. The “controller”

role is done by both classes and methods in charge of event users and the exchange with the server. The “view” role is considered to be provided by the EXT-JS component.

- Component server layer: In the current version, SITools2 is composed of only one server integrating the administration services and the use of datasets. In a next version, the objective will be to distribute the different data server to ensure a better scalability.
- Server Application layer: This layer includes the high level functional components corresponding to the main functionalities of the system. Other optional applications can be added by a plug-in mechanism. We also consider that Applications are non-volatile objects that own necessary information to process the request by the resources. Therefore, an application must own both a data model in cache and a reference to the objects coming from inferior layers of the persistence.
- Resource & representation layer: This layer includes the responsible classes for the processing of each request. In other words, it consists in the decoding of the provided representation from the body of the request, the invocation of a job and the restitution of the best representation in the response body.
- Business and storage layer: This layer is responsible for either the restitution or the modification of the objects from the application; exposition of an elementary dataset is an application for which a security strategy can be applied.

In this OAIS model, SITools2 implements the following functions: SiTools implements:

- Access, including interfaces with the archival storage and data management entities.
- Administration, including a part of system configuration, active requests functions and customer service.

For functions not available in SITools2, it is possible to add them via application plug-in development. SITools2 has been designed to handle large datasets which includes:

- Streaming and pagination of the response at the server side. The limitation is due to the database Server
- Multi-threading management. On a REST application, an application is instantiated only once when the server starts up. A REST resource is instantiated each time that the resource is called. Next, the resources are attached to the application by a router. RESTlet, the REST framework that SITools is using, allows a certain number of concurrent threads. This number is the limit of simultaneous calls for one application and it can be conFig.d to prevent simultaneous access for large queries
- JSON as Exchange format. JSON has been chosen because this format is a structural format and less verbose than XML format. As consequent, the time to transfer the message from the server to the client is faster in JSON than in XML format.

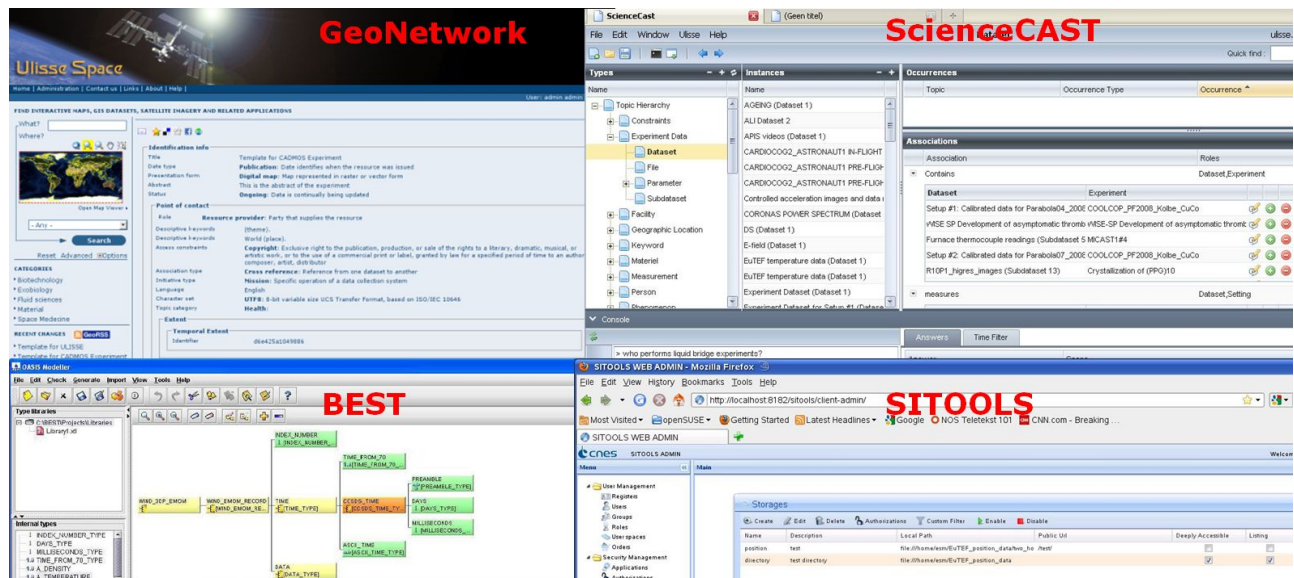


Fig. 3-2. Tool clients for handling metadata, distributed datasets and knowledge representation.

3.3 SITOOLS performance

SITOOLS has been demonstrated for integration of all ULISSE datasets in several demonstrations. Various tests have been performed in the astronomical field on large datasets. The tested dataset was an astronomical catalogue of celestial objects that contained about one million of rows and sixty parameters for each record. The results show interactive behaviour while the user is searching and visualizing this dataset.

3.4 Integration with metadata about datasets

For the metadata describing the location of files, origin, timing information and points of contact a dedicated meta-language was developed. Using the GeoNetwork framework a template was enforced for each dataset. All metadata information describing both the background of the data and its access URL is stored in a centralized database, while the data access capability is provided by SITools2. All formal properties can be edited and remotely accessed via web clients (Figs 3-2), see also following sections.

4. REPRESENTING CONTENTS

4.1 Analysis

Topic Maps technology was used to develop the knowledge representation. Topic Maps are a standardized technology (ISO/IEC 13250) for representing knowledge. It lets us represent the meaning of the data that is stored by including semantics together with the knowledge itself. Having both the technical and scientific information of the experiments in a semantically rich structure like topic maps allows for the creation of smarter applications. This standard of description will support the description of complementary knowledge associated to the experiment and datasets as to the scientific topic from the scientific point of view. The generation of the concerned information could be more dynamic.

In Topic Maps, every subject of a knowledge domain is represented as a topic. Every topic may have a type (which is a topic itself), any number of names and any number of occurrences. An occurrence can be seen as a property of a topic, for instance, a person topic might have a phone number occurrence. Topics can be linked together with associations and each topic that plays a part in an association will be assigned a role. Furthermore, there is the concept of scope. Scopes can be assigned to names, occurrences and associations, and they indicate when the item should or should not be taken into account when interpreting the topic map. Finally, a topic is identified through a set of identifiers (not through its names!) which allow automatic merging of different topic maps. Workshops dedicated to each scientific

discipline have been held for gathering relevant information from scientists and space engineers. During these workshops, Topic Maps ontologies were created for the scientific disciplines that were analysed and for space science experimentation in general. All topics maps are available on their own and merged together from a centralized knowledge base that is integrated in the platform to support navigation, data browsing and searching in the ULISSE platform.

4.2 Metadata relationships with topic maps

Metadata entered in GeoNetwork is of a more technical nature representing standardized information about datasets (people involved, files produced, etc.) and misses scientific information and explicit relationships between the concepts it describes. Topic Maps are used to bridge this gap. Thus two standards are planned to be used in ULISSE as presented previously and two sets of metadata generated and managed. Both verify the XML syntax. It appears necessary to constraint both definitions to keep consistency and information sustainability of the descriptions for a given experiment.

The implementation in ULISSE is based on the identification of the metadata base as the reference base for description by custodian, only source to update if necessary. This base will then be ingested to the Topic Maps Knowledge base, available through ScienceCast, but these concerned "topics" (or information) will there be locked against modification. Besides this initial set complimentary information will be inserted representing additional knowledge.

4.3 Implementation

ScienceCast is a stand-alone, web-based system developed to facilitate sharing, browsing and editing scientific research data. The system is a data browser and editor integrated with a natural language question answering component developed within the LINDO project, to allow users to query data in a natural way. The system front-end is implemented using SmartGWT^{*}, a Google Web Toolkit[†] (GWT) framework with an extensive widget library that offers data-bound components for easier data access, available under the LGPL licence[‡]. The client communicates with a java back-end through HTTP-requests using JSON[§].

^{*} [HTTP://code.google.com/p/smartGWT/](http://code.google.com/p/smartGWT/)

[†] [HTTP://code.google.com/webtoolkit/](http://code.google.com/webtoolkit/)

[‡] [HTTP://www.gnu.org/licenses/LGPL.html](http://www.gnu.org/licenses/LGPL.html)

[§] [HTTP://www.JSON.org](http://www.JSON.org)

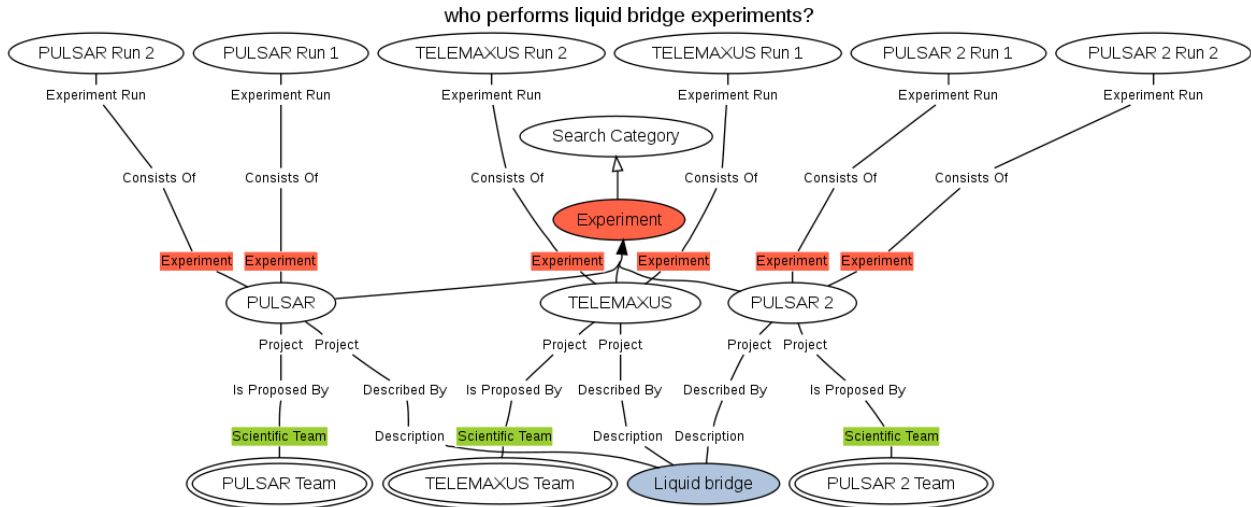


Fig. 4-1. ScienceCast output to an exemplary natural language question.

ScienceCast is a client-server application where the client is built using a modified version of the *Model-View-Presenter* pattern using an event system for inter-component communication and management. The server is built on stateless utility classes. The server functionality is reached through one single service point that then routes the calls to the correct utility. User state is stored in session objects that in turn carry a TopicMapsAccess object through which the topic map data store is accessed. A dedicated REST based interface has been developed for integration[4] similar to other ULISSE tools.

As shown in fig. 4-1, the implementation allows handling natural language queries and generates output in a structured graphical way.

5. REPRESENTING PLANNING FEATURES

5.1 Timeline Representation Framework

Design and implementation of advanced Planning and Scheduling software for space applications is an activity involving a certain amount of developing effort and risk, i.e., the software may fail to meet operational requirements (performance), and/or may fail to capture all the essential aspects of the problem (modelling). A software platform, called Timeline Representation Framework, TRF, for supporting planning and scheduling space applications design has been developed [5, 6, 7] The aim of the framework is to provide help to developers to cope with both software deployment efforts and modelling risks. In fact, meeting operational requirements in challenging space domains often entails coping with conflicting issues. For instance, the need to employ highly efficient software modules may often lead to choose ad-hoc solutions, and this inevitably conflicts with the need of reducing modelling mistakes, which

is best tackled by involving users as much as possible in all the steps of software development. The TRF simplifies the developing effort by providing a library of basic planning and scheduling domain independent solvers, and strengthens the interaction among the specific solvers implemented on top of the framework by providing a uniform representation of the solution database and defining a common inter-module cooperation and coordination interface. Modelling risks are reduced as the framework standardizes and simplifies the process of application deployment fostering a rapid prototyping cycle, which directly helps users to take into account their own feedback during application design.

5.2 Timeline Based Planning

TRF software infrastructure follows an approach to problem solving based on timelines. In this approach the world relevant addressed problem is modelled in terms of a set of temporal functions that describe its evolution over a finite temporal horizon. Posting control decisions can modify these functions.

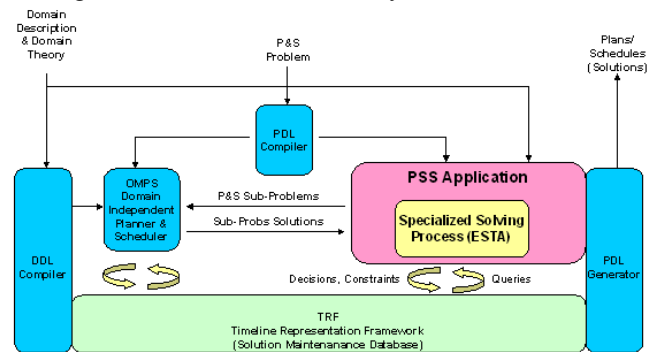


Fig. 5-1. Planning and Scheduling Service (PSS).

Additionally, a domain theory specifies legal patterns of control decisions (i.e., combination of things that necessarily have to be done in a coordinated way to change the evolutions), and the task of the solver is to find a legal sequence of control decisions that brings the entities into a final configuration that verifies both the domain theory and a determined set of desired conditions called goals.

The Timeline-based approach is inspired by classical Control Theory, in which the problem is modelled by identifying a set of relevant features whose temporal evolutions need to be controlled to obtain a desired behaviour. In the TRF such relevant features are described through components, the primitive entities for knowledge modelling. They may represent logical or physical subsystems whose properties may vary in time; therefore, control decisions can be taken on components to define their evolution. The current TRF release provides families of components which enable diversified modelling power. For the current purposes we assume that problems are modelled using components known as multi-valued state variables and renewable resources like those commonly used in constraint-based scheduling.

The TRF provides also a domain definition language (DDL) to specify both the components and the relevant physical constraints that influence their possible temporal evolutions (e.g., possible state transitions over time of a component, synchronization/coordination constraints among different components, maximum capacity of resources, etc.), as well as a Problem Definition Language (PDL) to specify problems as set of temporal landmarks for the temporal functions.

5.3 Solving Fluid Science Laboratory cases

Timeline-based modelling has been exercised for the Fluid Science Laboratory (FSL) in the Columbus laboratory[8]. In Fig. 5-1 different solvers are defined on top of the TRF and shows also how they can exchange information to contribute to a set of external services.

The Planning and Scheduling Service (PSS) application exploits the following solvers: (1) the OMPS (Open Multi-Component Planner and Scheduler Domain Independent Planner to solve specific planning sub problems (endowed with basic scheduling capabilities), and (2) an ESTA domain independent scheduler as a specialized solver for multi-capacity RCPSP-max sub problems. Both solvers, as any solver belonging to a TRF-based application, generally proceeds by interacting with the TRF solution maintenance database through posting queries and adding/removing constraints to/from the database.

The depicted connections among all modules are worth some comments, as they show how the architecture allows using the involved solvers independently from one another or in any combination (for example by means of the exchange of sub problems as underscored in Fig. 5-1). The user has available (a) a domain description language (DDL) for defining and then revising a domain theory, and (b) a problem description language (PDL) for describing the current request to the solver given a domain description. It is also worth saying that a problem solution can be extracted from the temporal database in terms of the same PDL language to enable further uses.

Given the general structure of the PSS, when a FSL timeline-based model (DDL file) is developed and a set of goals described in terms of a set of experiments to be performed on the FSL associated to a description of resources availability during the interesting operative period (PDL file) are given, the PSS is able to produce a temporally flexible sequence of activities whose execution guarantees (i) the correct achievement of all the experiments provided as goals, (ii) the respect of all the given temporal and resource constraints. Moreover, the PSS is also able to promptly produce alternative schedules in the face of a number of modifications to the original resources availability.

Specifically, the PSS solving process proceeds as follows:

- OMPS produces a sub-problem solution as a set of activities that are to be performed in order to guarantee the correct execution of all the requested scientific runs;
- The ESTA scheduler manages the activity plan taking into account all the given facility and operative constraints and, if original resources availability is modified, takes care of rescheduling all the activities;
- When the ESTA scheduler is not able to produce a new feasible schedule, i.e., the new resources availability do not allow to retain the original activity plan, a new P&S sub problems is generated and OMPS is requested to produce a new activity plan. Then, the rescheduling and planning tasks are iteratively repeated until a final activity plan is produced.

5.4 Modelling and validating Operational Procedures

In this section we describe the Procedure Validation and Verification Tool (ProcVT) developed in ULISSE. The aim of ProcVT is to validate and verify on-board operational procedures. A USOC will use ProcVT to: i) model an Operational Procedure (OP), ii) model the Physical Environment (PE) the OP interacts with, iii) model the possible disturbances

for the PE, iv) specify the property P to be verified and v) ask ProcVT whether the uploaded system (OP+PE+disturbances) is correct w.r.t. P or not (see Fig. 5-2).

The verification performed by ProcVT is carried out via Model Checking techniques. Thus ProcVT takes a system description including the OP, the PE and the disturbances models, and verifies if the property P is

satisfied by invoking a Model Checker (see Fig. 5-2). This enables formal verification of OPs, thus improving OPs quality assurance. The FVS service is used for this purpose, hence OP, PE, disturbances and P must be described using the CMurphi input language. This allows to successfully model the closed loop system we consider (i.e. PE + OP), which both possibly involves many complex arithmetical operations and may be nonlinear, thus ruling out symbolic approaches as those used in symbolic model checkers for hybrid systems (e.g. HyTech, UPPAAL and PhaVer).

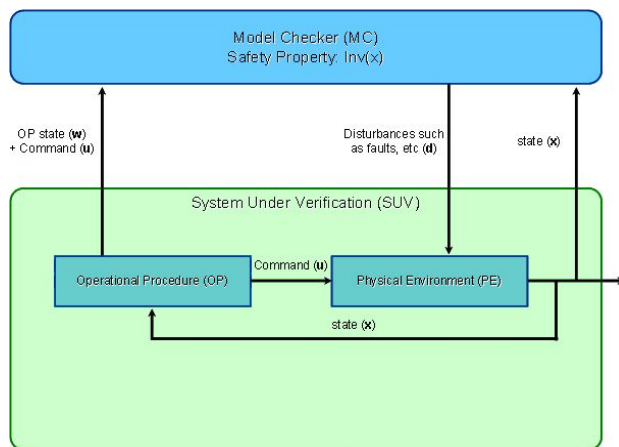


Fig. 5-2. Closed loop Model checking.

The feasibility of our approach has been assessed by verifying the correctness (i.e., termination within a given elapsed time) of a meaningful real-world TCL script as our OP, driving the EMPI experiment inside the FASES set of experiments. The EMPI experiments study the links between the physical chemistry of the droplets interface, by providing the droplet size distribution with the calibration curve and thermograms. Focus was on a meaningful part of the script driving EMPI, namely the one which takes care of experiment samples thermostatisation. By using ProcVT, it was possible show in 6 seconds that if the heating/cooling rate of thermostatisation is disturbed up to 4 times per experiment execution by an offset of 0.5 Celsius/minutes, then the experiment will not terminate within the given elapsed time, whilst it will terminate if the disturbance offset is 0.4 Celsius/minutes.

6. REPRESENTING INTERNAL DATA STRUCTURE

6.1 Approaches

Three approaches are considered at the start of the project. The first approach was to use a generic toolset called BEST [9]. This system is an open source software developed by CNES that allows to describe the internal data structure of binary data files.

The second approach is to reuse existing databases and libraries. In view of the long duration prospect to using ISS data and CCSDS based data formats are of major interest. The YamCS library [10] developed by Space Application Services is being used to have a generic format description compatible with Columbus ISS data available for ULISSE. During the project it has been decided by Space Application Services to make YamCS available as open source which was an additional stimulus for integration as the package also includes interfaces to the standardized experiment data archives used at USOCs.

A third approach was selected during the project for compatibility reasons with other USOCs. The PET (Packet Evaluation Tool) format is a general purpose format based on a header in front of a comma separated value (CSV) format description of the data. This format is considered a suitable common format for exchange of data between USOCs and various partners. The analysis performed by N-USOC in ULISSE showed that TReK (Telescience Resource Kit) used for American ISS payloads could be integrated. The tool is being demonstrated and tested with EDR and EuTEF commissioning data. To integrate with Columbus/ISS system data will also be considered.

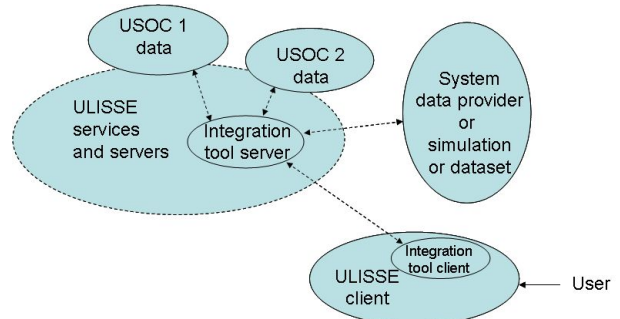


Fig. 6-1. Data integration scenario.

The following summarize the scenario and requirements implemented:

- Selecting facility: The user selects at least two experiment facilities which may be identical.
- Select instrument: The user selects an instrument that belongs to the facility.
- Select time frame: The user selects a time frame for the selected instrument / facility data.

- Select parameters: The user selects parameters that belongs to the selected instrument / facility. Via a drag and drop the users can select parameters and apply basic functions in a spreadsheet.
- Run spreadsheet: The user starts the spreadsheet function for manipulating datasets for data integration.
- Get and store USOC data: The purpose of this use case is to store processed data and data extracted in a local file system available to the user possibly using metadata.
- Installation of integration service: Installation and update of the data experiment integration tool should require a registration activity (war-file).

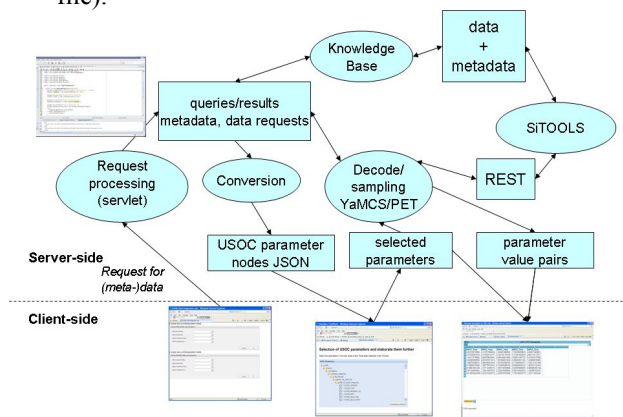


Fig. 6-2. Integration with KB and YaMCS.

The flow of activities in the proposed data integration concept is shown in Fig. 6-2. The initial parameters for the user to start data integration are extracted from meta-data. The parameters are extracted and provide the actual data to the experiment data processing. The actual data are processed within the spreadsheet.

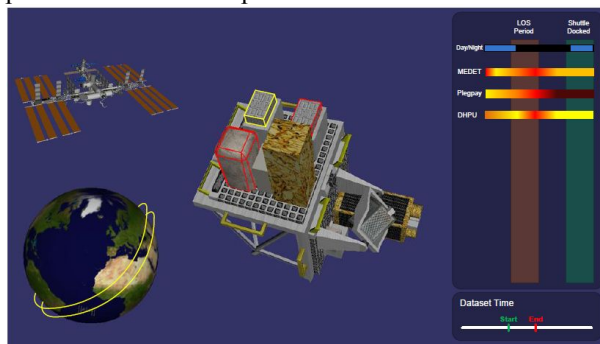


Fig. 6-3. Prototyping 3-D user interfaces.

The case study based on EuTEF data has been extended to include simulation and data visualisation using 3-D models (See Fig. 6-3).

7. REPRESENTING 3D RELATIONSHIPS

For interactive visualization of 3D related ULISSE data, technologies like Virtual Reality (VR) and Augmented Reality (AR) has been explored [11]. The ViRLoc (Virtual Reality Locator) tool was implemented that combines multiple VR and AR features. Because of the complex rendering process of 3D content and the processing and evaluation of real time video data, ViRLoc is provided as standalone client at user's end device. The client tool is available by downloading an installer from the ULISSE platform. ULISSE related data and information are obtained by integrated requests to the ULISSE RESTful webservice.

The ViRLoc tool provides three main features: ISS positioning viewer, 3D model viewer and Augmented Reality viewer. The general interface of ViRLoc is divided in four sections: main menu bar, 3D/AR scene viewer, viewer menu, and the main buttons section.

ISS position viewer

The ISS Positioning Viewer supports the loading, visualization and animation of ISS positions at a certain time range. The viewer provides a full interactive 3D scene. By using a virtual 3D globe, referenced to a geocentric coordinate system, the corresponding ISS orbits are created and displayed. Each read ISS position is marked with a green sphere and the orbits are green paths. At the first path position a 3D model of the ISS is placed. Via controllers the animation can start, stop, pause, forward, and rewind. Stop means the animation will set to reset state (i.e, the ISS model is located at first path position). If the ISS intersects a read position a label is appeared that shows the current latitude and longitude. Additional features, like 'show orbits' or 'show label', allow the display and none-display of the generated green paths and the ISS label.

The viewing perspective of the globe scene can be changed via three different modes: top-down, ISS view and free view. The top-down mode (by default) provides a virtual camera view that is always looking at ISS parallel from orbit. By using the ISS view, the virtual camera is located behind the ISS model and always looking at it. By using free view the virtual camera is not docked at the model and the scene can manipulate freely.

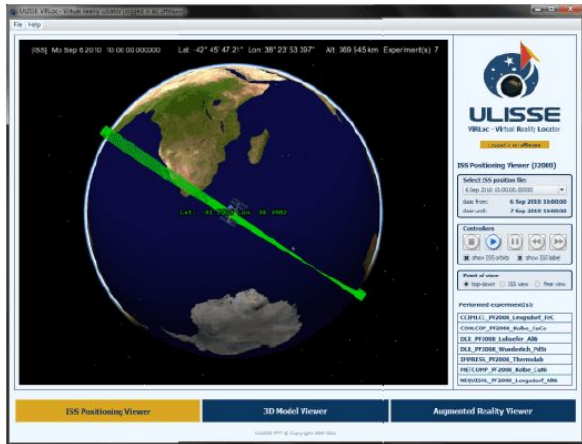


Fig. 7-1. The ISS Positioning viewer of ViRLoc after loading ISS positions with the top-down camera view.

When a position file is loaded, the corresponding performed ISS experiments are loaded and as link in the menu displayed (see Fig. 3). After clicking on the link a separate window appears and shows the detailed information about the experiment selected and linked in the menu to obtain detailed experiment information

The initial state of the ISS Positioning Viewer shows the virtual 3D globe and by moving the mouse over the globe, the corresponding latitude and longitude are displayed at the mouse position. After loading ISS positions, the time, the latitude, the longitude and the number of performed experiment are shown in the scene. While the ISS scene is animated, the information about the time, latitude and longitude are also animated.

Model viewer

The 3D Model Viewer is a generic viewer to load and explore 3D models within a 3D space. The viewer can load different formats, like osg, vrml, wrl, lwo, lws, 3ds, dae, etc. A loaded 3D model can be explored by rotation and zooming in and out via the mouse controller. By using certain shortcuts the model appearance can be changed. There are three different states: the normal state (original texture by default or the transparency state (decreasing and increasing the alpha value) or the wireframe state (the model will be only shown with wire frame). Also the complete scene data are available, such as frame rate and the properties of the loaded model.

The viewer also supports interactive models. That means the models geometry is mapped to its semantic meaning. The ViRLoc Demonstrator enables an interactive 3D model of the Biolab facility. If the Biolab model is loaded and the check box “interactive” is active, by mouse clicking over a Biolab component, a label with its name will appear.

Additional information about the clicked model will appear in the viewer menu.

Augmented Reality viewer

The Augmented Reality Viewer is a generic viewer to load and explore 3D models in physical reality. The viewer provides marker-based tracking. Therefore the end-user device requires a video capture device (e.g., an usual web camera). While a printed pattern is held in the camera’s field of view, synthetic information are put the marker’s position and the combined video streamed is displayed. Marker-based tracking allows the computation of the inverse cameras position and orientation. Resulting from this, it is possible to locate a 3D model or other synthetic information on the computed position to overlay the camera image. Like the 3D Model viewer the AR viewer can load different formats, like osg, vrml, wrl, lwo, lws, 3ds, dae, etc. Also the complete scene data are available, such as frame rate and the properties of the loaded model. The loaded 3D model can be explored by manipulation of the pattern. By using certain shortcuts the model appearance can be changed (model state, wireframe state, and transparency state).

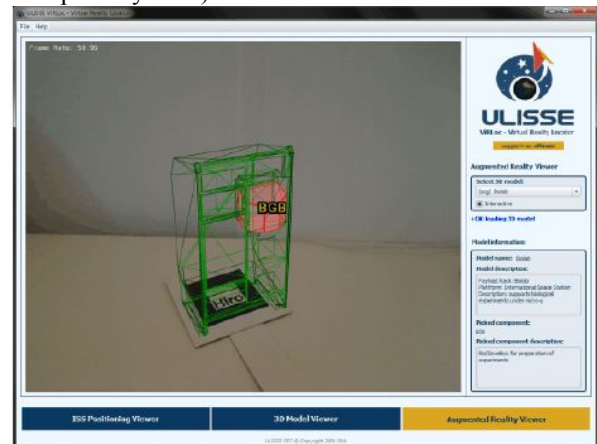


Fig. 7-2. User interface of the Augmented Reality Viewer after loading the inactive Biolab model and selecting a component in wire frame model state.

Like the 3D Model Viewer the AR viewer also supports interactive models by mapping geometry to its semantic meaning. Loading the 3D Biolab model enables a semantic exploration via mouse selection over a Biolab component (see Fig. 7-2). Detailed information about the selected model will appear in the viewer menu.

8. PLATFORM INTEGRATION

8.1 Platform overview

In the following, the architecture and the integration of components shall be illustrated from

various perspectives. The logical view of the components which were used for the ULISSE Demonstrator is shown in Fig. 8-1. The blue cube pictures a schematic view of the ULISSE Knowledge Base, hosting the ULISSE topic maps.

In the Fig., experiments which were described by using the Geonetwork application (and thus the ISO19115 based ULISSE metadata standard) are located on the x-axis (Experiment Descriptions).

They are injected into the Knowledge Base through an automatic conversion mechanism (a proprietary XML to Topic Maps converter, conceived by CNES and Space Application Services). The resulting topic maps can be edited and exploited by the ScienceCast application.

On the left side, access to the distributed data archives is enabled by SITools2, which, in its actual state, map file systems to REST web services. Datasets from five USOC's (i.e. B-USOC, SRC PAS, TPZ Naples, DLR MUSC and E-USOC) were made visible and accessible through the ULISSE portal.

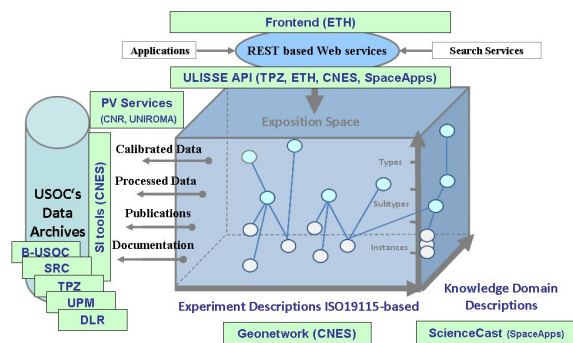


Fig. 8-1. Logical view of integration

On the top of the picture, the Frontend sits on the Platform API. The Frontend consists of the GUI and programmatically addressable web resources.

8.2 Physical view

The physical view of the ULISSE Demonstrator is represented by the ULISSE Network, shown in Fig. 8-2. The ULISSE Network has to provide communications capabilities for the ULISSE services platform.

This network has been released on five European USOCs/data providers and on one gateway centre.

Moreover the ULISSE Demonstrator foresees that some External Users access to data and resources of the ULISSE Demonstrator services platform.

The connected data providers are:

- B-USOC, located at the Royal Observatory of Belgium in Brussels
- SRC PAS, located at Space research Centre in Warsaw
- MUSC USOC, located at DLR premises in Cologne
- MARS USOC, located at Telespazio Naples site
- E-USOC, located at UPM in Madrid
- The Gateway Centre is located at Telespazio premises in Rome.
- The data providers nodes are connected to the “ULISSE Gateway Node” using Internet and VPN. This solution allows secure access for data exchange among nodes.

The B-USOC and SRC Data Sources are already a public ftp server so it has not been necessary to implement the VPN between ULISSE Gateway Node and these sources. The Promotional Web Portal (provided by MEDES) and the Questionnaire server (provided by ETH) were reverse-proxied through the Internet.

Only the “ULISSE Demonstrator Gateway Node” provides Internet access for the External End User through Secure WEB capabilities.

The ULISSE Network has been tested through dedicated tests and demonstration sessions in several European meetings. Management rules of the Network have been defined, also in the perspective of future expansion with the inclusion of additional resources. The Network is monitored with dedicated tools. As example, Fig. 8-2 shows the architecture of the ULISSE Demonstrator.

Front end integration

The installation of the ULISSE prototype, the later migration to the server on Telespazio premises as well as the testing and debugging procedures has been performed by ETH Zurich as planned.

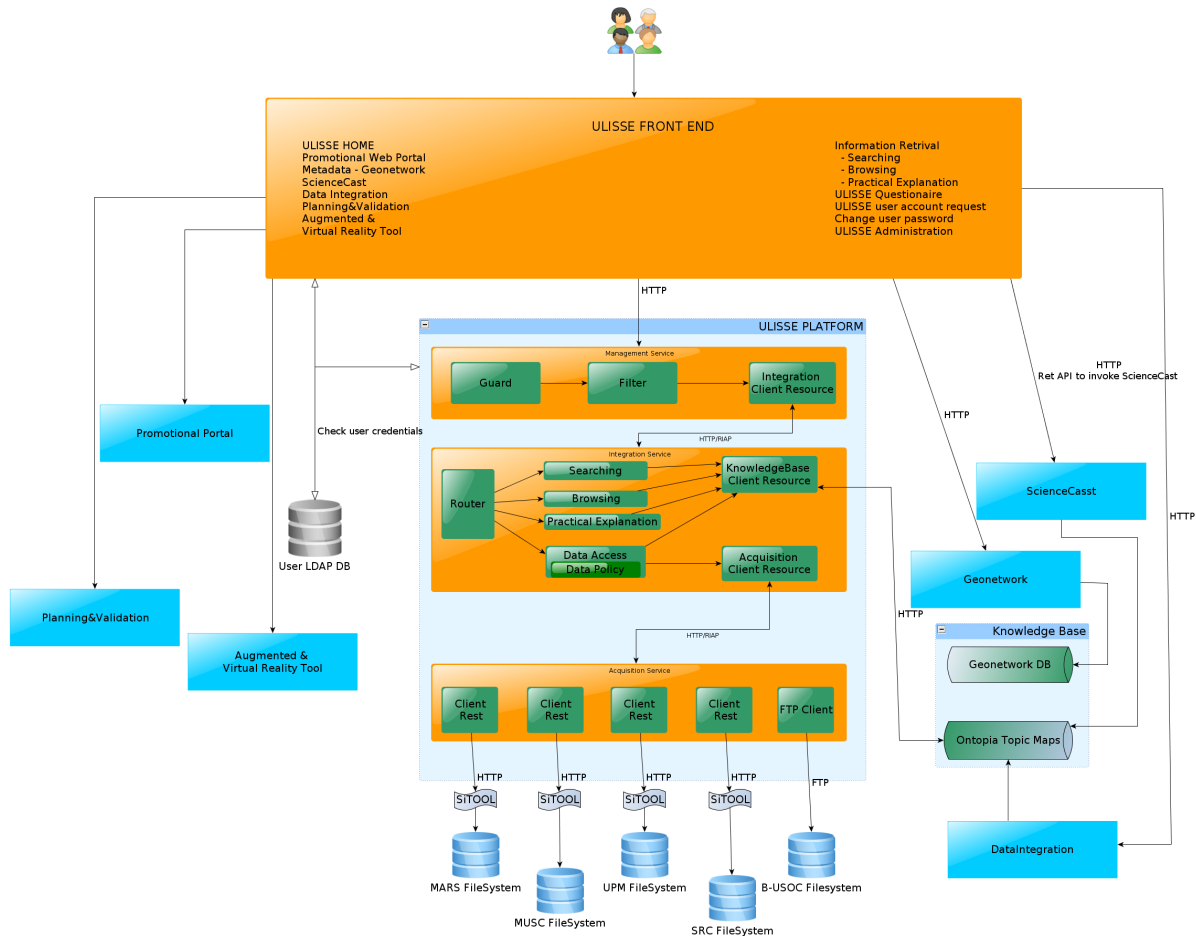


Fig. 8-2. ULISSE Demonstrator architecture

The user questionnaire designed by TPZ Napoli were integrated on the Select Survey Server of ETH and tied into the ULISSE demonstrator. Fig. 8-3 shows the entrance page to the ULISSE portal. An authenticated user can navigate through the different components and is redirected via the reverse Proxy server.

Due to the different technologies used for the different components we have opted for a reverse proxy scenario where multiple ULISSE server applications running on a single server or on multiple servers inside of a LAN can be reverse proxied as to appear as a monolithic system to the WAN (Internet) and hence to the client/user.

The reverse proxy technology was implemented as Virtual Host with Apache2 on a SBG testing server on the premises of the ETH.

Apache2 is configured by placing directives in plain text configuration files. In addition, other configuration files may be added using the “include” directive, and wildcards can be used to include many configuration files. Any directive may be placed in any of these configuration files. Changes to the main

configuration files are only recognized by Apache2 after a restart.



Fig. 8-3. ULISSE Front-end Portal.

Additionally to the first demonstrator implementation the second implementation also allows for instantaneous activation of the entire ULISSE platform on the non secure port 80. Furthermore data sources were registered on a very

granular level that complies with ESA policies. Automatized resource registration was not yet implemented and is considered subject for a follow-up project.

9. EXPERIENCES AND DISCUSSIONS

Many public demonstrations of the ULISSE functionalities have been held on the occasion of scientific congresses and public events. The activities were performed with synergy to various USOC activities, e.g. [12], related to the need to improve knowledge preservation.

The demonstrations offered the opportunity to have fruitful discussions with scientists and to gather feedbacks from the user communities after “hands-on” sessions, also with the support of questionnaires. Almost 100% of replies indicate ULISSE extremely useful; this feedback confirms definitely that a data e-infrastructure is needed for space research.

Scientists recommended that the investigator of the experiment should be involved in the future re-use of the data (pursuing possible cooperation with the new users of the data). This recommendation suggests that cooperative services for the users based on adequate data policies would be advisable for promoting the re-use and exploitation of scientific data.

Presently, data re-use is subjected to the ESA data policy, which gives guidelines and rules for authorising the distribution of space data to third parties (not belonging to ESA or to the investigator team). Accessing data through ULISSE has to be approved by ESA, in compliance with the ESA data policy. Nevertheless, the ULISSE experience has highlighted the opportunity to upgrade the data policy, to pursue data exploitation while protecting intellectual property rights of the investigators.

At the same time, interviewed scientists provided the clear indication that the scientific community cannot sustain the non-negligible task of feeding such an infrastructure with databases, repositories and metadata, because scientists are dedicating already a relevant amount of time to the preparation of research proposals and reports. This outcome confirms the necessity of support entities like the USOCs that, for their role in the execution of space experiments, have the required competences and access to the data and related documentation; therefore the exploitation of the USOC network, as proposed in ULISSE, would allow an efficient implementation of a space data infrastructure.

ULISSE may also stimulate international cooperation. In fact, NASA and JAXA representatives have evaluated ULISSE a very interesting platform and are looking for possible cooperation with Europe to exploit this experience.

International cooperation between space agencies is recommended to develop a standard approach to knowledge representation and to data policy as well as to ensure global coverage. Moreover, UN representatives are interested in exploiting ULISSE for educational programmers to promote space research and activities in developing countries.

The development of the platform and the tools provided many insights in the technology developments needed. They relate to centralized vs. distributed setups, use of open source software, portability, distributed knowledge integration and interfaces to be supported. A representative dataset has been integrated (see Fig. 9-1). However, the population with metadata, knowledge, datasets and related information proved not to be an easy task. New developments for various supporting tools have been identified including version management and automatic extraction of key information useful in browsing.

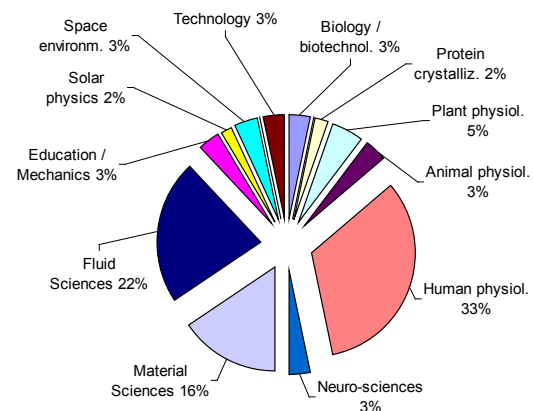


Fig. 9-1: Distribution of the experiment metadata between scientific disciplines.

10. CONCLUSIONS

ULISSE, being the first project addressing the preservation and exploitation of the ISS data, has been a pathfinder for a number of technological, operational and legal issues related to ISS data.

Through its Demonstrator, ULISSE has proven the feasibility and usefulness of a data e-infrastructure for the exploitation of the ISS data.

On this subject, ULISSE has stimulated the interest of the scientific community on one hand, and of space agencies (firstly ESA) on the other hand.

ULISSE contributed in the understanding of interaction with the general public about the benefits produced by the research in space. To this respect, ULISSE has underlined that data re-use would increase the scientific productivity and the return of investment.

Therefore, it is expected that these results will pave the way for the establishment of an e-infrastructure devoted to the preservation and exploitation of the scientific data produced by the experiments on ISS and other space platforms. The experience gained with ULISSE will ease and accelerate the achievement of this objective.

11. ACKNOWLEDGEMENTS

Contributions of the full ULISSE team are gratefully acknowledged. ESA provided support by giving permission of use of various demonstration datasets. The YamCS library was developed by Space Applications Services and kindly made available for the project. The work has received funding from the European Commission through the Seventh Framework Programme (FP7/2007-2013) under the Space Theme, under grant agreement n°218815 for the ULISSE Project.

12. REFERENCES

- [1] ULISSE project page: <http://www.ULISSE-space.eu/>.
- [2] Laboratory Science with Space Data, D. Beysens, L. Carotenuto, J.W.A. van Loon, M. Zell (Eds), Springer 2011.
- [3] SITools 2: <http://sitools2.sourceforge.net/>
- [4] D. Damen, and Patriksson, M. Putting Topic Maps to REST, TMRA 2010 conference, Leipzig.
- [5] A. Cesta and S. Fratini. The Timeline Representation Framework as a Planning and Scheduling Software Development Environment. In PlanSIG-08. Proc. of the 27th Workshop of the UK Planning and Scheduling Special Interest Group, Edinburgh, UK, December 11-12, 2008.
- [6] S. Fratini, F. Pecora, and A. Cesta. Unifying Planning and Scheduling as Timelines in a Component-Based Perspective. Archives of Control Sciences, 18(2):231–271, 2008.
- [7] A. Cesta, A. Finzi, S. Fratini, A. Orlandini, E. Tronci. Flexible Plan Verification: Feasibility Results. Fundamenta Informaticae. 107(2-3): 111-137, 2011.
- [8] A. Cesta, S. Fratini, A. Orlandini, R. Rasconi. A Planning and Scheduling Service for the ULISSE Platform. In proc. of the IEEE 4th International Conference on Space Mission Challenges for Information Technology (SMC-IT 2011). Pages: 35-42. ISBN: 978-1-4577-0712-4. Digital Object Identifier: 10.1109/SMC-IT.2011.26. IEEE Conference Publications. 2011.
- [9] BEST:<http://logiciels.cnes.fr/BEST/EN/best.htm>.
- [10] N. Mihalache and L. Steinicke, YAMCS - A Lightweight Open-Source Mission Control Systems Used In Columbus Payloads Operations, Proceedings DASIA 2011, ESA SP-694.
- [11] D. Markov-Vetter, A. Lehmann, O. Staadt, U. Mittag, Future Interface Technologies For Manned Space Mission, IAC-11.A1.1.11.
- [12] G.I Sören-Schwartz, R. Willnecker, S. Sous - Data and Information Management of ISS Payload And Experiment Data, Proceedings of 62nd International Astronautical Congress, Cape Town, SA, IAC-11-D5.2.8.