VIP ANR-09-COSI-03
Virtual Imaging Platform

M1.2.3: Simulation use-cases joint study with users.

Denis Friboulet	CREATIS	denis.friboulet@creatis.insa-lyon.fr
Alban Gaignard	MODALIS (I3S)	alban.gaignard@i3s.unice.fr
Bernard Gibaud	VISAGES (IRISA)	bernard.gibaud@irisa.fr
Tristan Glatard	CREATIS	tristan.glatard@creatis.insa-lyon.fr
Johan Montagnat	MODALIS (I3S)	johan@i3s.unice.fr

Abstract

The workflow designer used to develop simulation workflows aims at providing an intuitive and accessible graphical interface, by taking into account VIP user needs. This document analyzes the VIP platform simulation use-cases in order to design an adapted workflow editing interface. It covers the needs related to workflow adaptations when including new models and new simulators in the platform, as well as the needs related to simulation execution trace generation and querying.

Contents

1 Introduction

The VIP platform will be used to produce simulation images, from various human body models and using different image modality simulators. A simulation task involves the creation of a simulation workflow and the execution of the resulting workflow to produce a simulated image. Users are interested both in (i) creating simulation use-cases and (ii) analyzing results from former simulations. The creation of simulation use-cases involves the editing of an adapted simulation workflow. Although the generic tooling available for workflow design and enactment makes it possible to create any simulation workflow from scratch, we aim at delivering a simplified interface to compose VIP workflows, well adapted for VIP non-workflow specialist users. Similarly, analyzing results from former simulation experiments is possible through the existing tooling at a low level (analyzing workflow execution logs), but it is cumbersome and not adapted to the expected usage of the platform. The provenance information captured during workflows execution, represented using semantic annotation, can be easily queried and manipulated to deliver to users a higher level view of the platform runs history and access to former results.

This milestone addresses simulation workflows design in section 2 and simulation runs result querying in section 3. It is used as an input to the design of the VIP workflow designer and enactor, in view of fulfilling project milestone M1.2.4 (workflow designer architecture) at PM21 and deliverable D1.2.2 (simulation workflow designer) at PM26.

2 Simulation workflow designer

The VIP project plans the integration of four medical image simulator in the platform:

1. Sorteo (Photon Emission Tomography)
2. Sindbad (Computed Tomography)
3. FIELD-II (Ultrasound)
4. SIMRI (Magnetic Resonance Imaging)

These simulators were developed independently and they consume (respectively produce) different kinds of input (respectively output) data. Prior work has been conducted in the project to align the workflow input data representation as described in section 2.1. Reviewing the expected usage of the VIP platform, we identify current limitations in section 2.2 and we make a proposal to tackle the related requirements in section 2.3.

2.1 Existing material

Each simulator uses a human body model as input in addition to parameters specific to the simulator. These four simulation tools have been decomposed and their simulation processes have been described as scientific workflows in project deliverable D.2.1.1 ("Prototype integration of simulators within VIP"). Each simulator uses specific input formats, including tissue models and simulator-specific parameters. A common IAMF format has been agreed upon by project partners to represent human body models. Preparation workflows converting IAMF models to each simulator input format are described in project deliverable D2.1.2 ("Object preparation workflows for the VIP").

A typical simulation experiments involves chaining a preparation workflow and a simulation workflow. Starting from a IAMF model, such a composed workflow produces a simulated image. All IAMF models cannot be interpreted by all simulators. An IAMF model may be represented by a mesh, an image volume, unstructured data, or any combination of the formers. Although a unique file format will be used for each kind of data (and enforced by automatic conversion if needed when a model is imported to the platform), the representation available may not match the simulator input specified. A given simulator makes assumptions on the data type it receives as input. Similarly, the simulated image produced by simulators may be encoded using different image volume formats, and conversion to a unique format is expected.

2.2 Expected VIP platform usage

Beyond simple simulation experiments involving an already integrated simulator in the VIP platform and a compatible IAMF model as input (as described in section 2.1), there are three main use-cases that are anticipated:

- **UC1**: Using different IAMF formats as input for a simulator.
- **UC2**: Integrating a new simulator in the platform.
- **UC3**: Retrieving different simulated images.

The first use case involves converting IAMF models prior to invoking the simulator preparation workflow. The second use case involves designing a new simulation workflow, possibly inspired by existing ones. The third use-case involves a simple image conversion step.

All three use-cases can be addressed through an abstact simulation workflow depicted in figure 1. In this figure, the two main workflows composing a simulation task are underlined by a dashed box. These workflows are only loosely specified as their concrete implementation really depends on the precise simulator being considered (UC2). However, there is some information on the different functions to be implemented by any simulator that are depicted as yellow boxes. The object preparation workflow deals with models format conversion (UC1), scene definition and physical parameters definition. The simulation workflow deals with data splitting for parallelization, core simulation, simulation results collection, and output format conversion (UC3). Some logical dependencies are defined between these sub-functions.

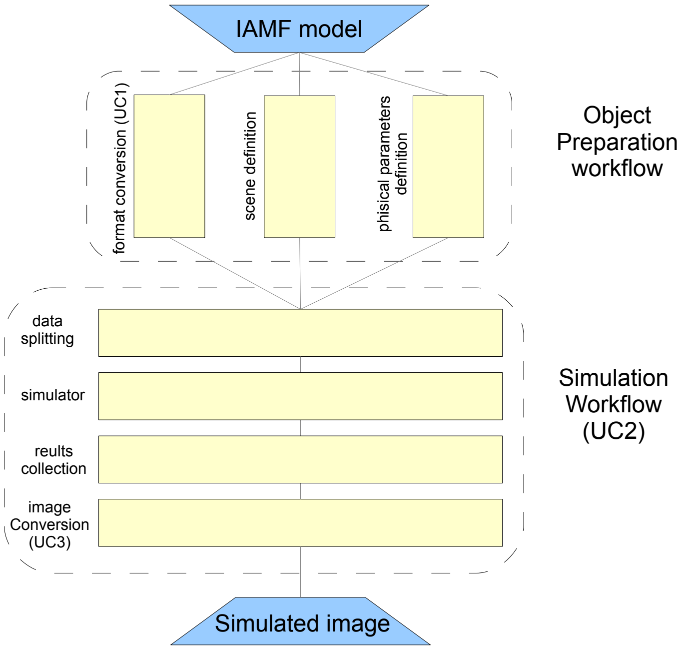

Figure 1: Abstract simulation workflow

2.3 Proposal for addressing VIP workflow design needs

Instantiating a generic workflow such as the one represented in figure 1 requires to specify the concrete services composing each of the functions defined. Each function may be described by a sub-workflow, a single service, or possibly none (if this function is optional). To assist such a generic workflow instantiation task, the workflow designer need to be extended considering the need to:

- Identify the tools that can be used for each function represented (*e.g.* format conversion, data splitting..).
- Transform the logical dependencies into concrete data dependency links.
- Assist the user in selecting the proper tools among the catalogue of tools available.

Annotating data analysis tools with semantic information provide the knowledge necessary to validate the coherency of the workflow composition chain and possibly to assist the user in identifying the conversion tools that can be used to implement each function. Indeed:

- The annotation of each tool function can be used to create a smart tool repository which identifies tools relevant for implementing different simulator functions;
- The annotation of tool input and output formats can be used to validate the coherency of the composed workflows; and
- A semantic reasoner can be used to infer the conversion tool (or the chain of conversion tools) that can be used to link two activities in the workflow editor.

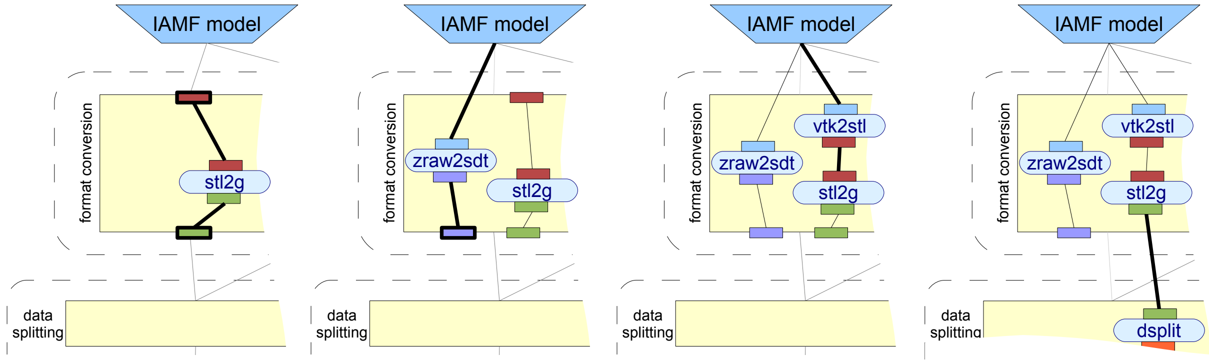

Figure 2: Abstract workflows merging

This approach requires the project ontology to be extended to represent the different functions of tools and the different file formats manipulated. In the case of IAMF models, the ontology of formats should include the representation of different kinds (meshes, volumes, and unstructured data), the file formats used to represent each kind, and possibly a combination of these.

Each function defined in an abstract workflow can be implemented through concrete tools labelled after this function in the ontology. Their input/output ports can be inferred from other functions from the abstract workflows or workflow sources/sinks that are logically connected. Only tools matching the sub-workflow function can be selected and imported in that part of the workflow. Tools with matching output/input ports specification in sub-workflows with logical connections can be automatically linked (unless manually specified otherwise). Figure 2 illustrates a typical workflow editing section, conforming to the abstract workflow represented in Figure 1. In this figure, different workflow tool I/O types are represented with different colors. Bold lines underline the elements that can be automatically inferred at each step exploiting the semantic information attached to workflow tools. First (left), an `stl2g` format converter is inserted. Since the sub-workflows for function "format conversion" has no matching I/O ports, these are inferred and connected to the newly connected activity. Second (center left), a `zraw2sdt` converter is inserted. It has an input port matching the `IAMF model`workflow source that is logically connected to this function. Hence, a link is automatically created. Since there is no matching output port, one is automatically inferred. Third (center right), a `vtk2stl` converter is inserted. This one has an input port matching the workflow source and an output port matching the `stl2g` converter input. Hence, both ports are automatically linked. Finally (right), when a new activity is inserted in the logically-connected data splitting function sub-workflow, links with the subsequent activities from the format conversion function are inferred.

3 Simulation results querying

Provenance information capturing history of simulation experiment runs will be recorded in the VIP platform during execution by producing RDF triplets conforming to the OPM[1] specification. The provenance information will be stored in a repository that can be queried through a semantic query language such as SPARQL. To avoid exposing end-users to the

[1] Open Provenance Model, http://openprovenance.org

complexity of the SPARQL language, a dedicated graphical interface will be implemented to navigate among provenance data. The interface will enable querying execution trace information per (i) data set produced/consumed or (ii) execution period of time. In the first case, it will return the input files and parameters used to produce a given simulated image, or the output files derived from given input files/parameters. In the second case, it will progressively refine the search by identifying workflow runs during a period of time, then activities fired for each workflow and finally input/output parameters for each activity.

4 Conclusions and future work

We will rely on semantic information and semantic data manipulation technologies both to manipulate necessary data format converters and simulation experiments history. In addition, a customized graphical interface will be developed to simplify the simulation workflow adaptation process and the transformation of an abstract workflow into a complete experiment workflow.

The architecture of the extended VIP workflow designer will be proposed as project milestone M1.2.4 (workflow designer architecture) before PM21 and the resulting implementation will be delivered as D1.2.2 (simulation workflow designer) before PM26.