Interactive Visual Analytics of Molecular Data in Immersive Environments via a Semantic Definition of the Content and the Context

Mikael Trellet*
Venise group
LIMSI-CNRS

Nicolas Férey†
Venise group
LIMSI-CNRS

Marc Baaden‡
LBT
IBPC-CNRS

Patrick Bourdot§
Venise group
LIMSI-CNRS

Abstract

Bringing together, in a unique immersive environment, visualization and analysis of scientific and complex data requires a thorough approach in order to fulfill scientists' specific expectations. Such an approach needs to consider the highly heterogeneous nature of data, the dynamic interactions between experts and data, and the large amount of data involved in scientific studies. Whereas small and static scientific datasets can quickly be deciphered thanks to standard immersive tools such as 3D visualization software packages, bigger and dynamic datasets exceed the analytical capacity of these tools, requiring an efficient platform for their manipulation. Through the example of the structural biology field we discuss the need for an approach based on a high-level definition of the content (scientific data) and the context (immersive environments and interfaces). Our design is illustrated by a platform for dynamic and intelligent representation of data to the user. The data hierarchical classification will provide new ways to interact with the data via intelligent and direct relationships between them. This approach is based on the semantic definition of all the concepts manipulated in the virtual environment, either abstract or concrete, which allows for an adaptive and interactive experience of both visualization and analysis.

Index Terms: Computer Graphics [I.3.7]: Three-Dimensional Graphics and Realism—Virtual Reality

1 Introduction

With the increasing complexity of scientific data that experts have to manipulate, the need for platforms capable of handling the intricate data flow is strong. The issue is particularly pressing in the field of structural biology. One central building block in this field, the numerical simulation process, is now able to deal with very large and heterogeneous molecular structures. These molecular assemblies may be composed of several million particles and consist of many different types of molecules, including a biologically realistic environment. This overall complexity raises the need to go beyond common visualization solutions and move towards integrated exploration systems where visualization and analysis can be merged.

These integrated systems must be able to deal with heterogeneous data, rendering them in a condensed working environment for the expert's needs. Immersive environments play an important role in this context, providing both a better comprehension of the 3-dimensional structure of molecules, and offering new interaction techniques to reduce the number of data manipulations executed by the experts.

A few studies took advantage of recent developments in Virtual Reality to enhance some structural biology tasks. Visualization is the first and most obvious task that was improved through new adaptive stereoscopic screens and immersive environments plunging experts into the very center of their molecules [20, 19, 15, 10]. Structure manipulations during specific docking experiments have been improved thanks to the use of haptic devices and audio feedbacks to drive a simulation [6]. Despite these studies, no specific development has been made to setup an immersive platform where the expert could manipulate data coming from different sources to accelerate and improve the development of new hypotheses.

Today, the main workflow that experts follow in structural biology studies compartmentalizes the visualization of data and their analysis into two different and independent sequential tasks based on separate tools. This separation can be partly explained by the significant differences between the data handled by the 3D visualization software packages and the analytical tools.

On one side, 3D visualization solutions such as PyMol [5], VMD [11] or UnityMol [14] explore and manipulate 3D structure coordinates composing the molecular complex that will be displayed. The scene seen by the user is composed of 3D objects reporting the overall shape of a particular molecule and its environment at a particular state. This scene is static if we are interested in only one state of a given molecule, but is often dynamic when a whole simulated trajectory of conformational changes over time is considered.

Analysis tools, on the other side, handle raw numbers, vectors and matrices in various formats and dimensions, from various input sources depending on the analysis pipeline used to generate them. Their outputs are graphical representations of trends or comparisons between parameters or properties in 1 to N dimensions formatted in a way that experts can quickly understand and use such information to guide their hypotheses. During the process that tries to decipher a scientific mechanism, and as we highlighted previously, experts constantly move from visualization tools to analysis tools and vice versa, wasting a precious amount of time along the way.

A major improvement of tools available today would bring into play a scenario where the 3D visualization of a molecular event is coupled to monitoring the evolution of analytical properties, e.g. sub-elements such as distance variations and progression of simulation parameters, into a single working environment. The expert would be able to see any action performed in one space (either 3D visualization or analysis) with a coherent graphical impact on the second space to filter or highlight the parameter or sub-ensemble of objects targeted by the expert.

Immersive environments favor the combination of heterogeneous information sources and provide several ways to merge data and processes for a particular task. The combination of heterogeneous information brings a complexity not at the context level anymore but rather at the content level. This complexity needs to be addressed in order to propose a fully-functional immersive environment where scientists will have the possibility to visualize, analyze and interact with their data.

There is a need to link heterogeneous data around an homogeneous framework where each object or group of objects can be re-

*e-mail: trellet@limsi.fr
†e-mail: ferey@limsi.fr
‡e-mail: baaden@smplinux.de
§e-mail: bourdot@limsi.fr

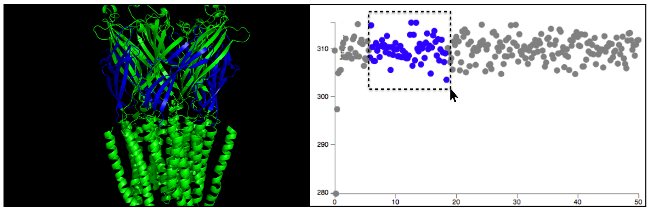


Figure 1: 3D visualization and analytical plot of a protein into two different spaces in the same environment. Any subset of points selected in the analytical space is instantaneously highlighted in the visualization space.

lated to the parameters or properties that define them. These links will first enable to access an object via its parameters, then, conversely, it will enable access to parameters by the object itself. Beyond information access, an interaction of the user with the graphical representation of either the object in the 3D visualization space, or the parameters in the analytical space, will have an impact in the reciprocal space as illustrated in the figure 1.

A database defined by a high-level representation of all the data represents an efficient way to mix visualization and analysis. By classifying knowledge, it is possible to create direct links between individuals and properties, then setup automatic retrievals of data each time an expert would need it.

The semantic definition of the application field is sufficient to implement dynamic links between visual and analytical properties but does not allow for a specific way to interact with the data. As a consequence, we extended the definition of the structural biology content to the immersive context in which the platform will be used. By doing so, we were able to combine information from both the expert knowledge and the user context in order to define intelligent interactions during an immersive working session.

2 Semantic definition of the platform

In Computer Science, knowledge representation is often associated to the notion of ontology. An ontology is defined as a structured and hierarchical ensemble of concepts and relationships allowing to define a partial or a whole field. An ontology must be comprehensible by both computer systems and humans. Indeed, an ontology often needs to be integrated into automatic processes setup by experts of a specific field, not always familiar with the computing side of the ontology.

Ontologies must allow the classification of any specific value or information into a pre-defined scheme as instances of elements of this scheme. It plays the role of a set of grammatical rules for a language made of data.

2.1 State of the Art

The semantic description of content through ontologies in immersive environments has been reported as a very efficient way to setup artificial intelligent systems [21], to design new VR applications [13] or to implement multimodal interactions [12]. Although these developments allowed for a very interesting usage of semantic descriptions in immersive environments, they aim for a generic definition of either the content or the context. Therefore such approaches lack either the expert field knowledge or the interactive possibilities, both required to setup intelligent interactions.

On the application side, the setup of ontologies in order to standardize knowledge in scientific fields has undergone an important and spontaneous growth at the end of the 90s [17].

Bioinformatics, tightly anchored with structural biology, uses ontologies for a long time. The most significant example is the fast-growing genomic field where it became quickly impossible to handle data flow without a proper and standardized organization of the data [18]. *Gene Ontology* [2] was created to regroup genomic data into a uniform format and database. Today it is one of the most cited ontologies.

Only very few expert software packages in structural biology have based their development on the usage of ontologies. DIVE [16] and Avogadro [9] appear both as exceptions, implementing, in different ways, a semantic description of the data they are manipulating. Avogadro is using the Chemical Markup Language (CML) for its semantic capability and adds a description layer on top of the data the software is dealing with. However, no ontologies are set up and there no use is made of the reasoning capabilities of this semantic description.

DIVE offers the possibility to create on-the-fly ontologies and datasets based on the input data. This data representation allows a common data model that the software libraries will use. Then, creation of links between data values and concepts are possible and the different DIVE components (analyses, 3D visualization, etc.) are able to query them at any moment. Beyond value, links and relationships between the dataset elements can be queried. DIVE further implements a powerful and generic ontology creator directly depending on the type of the input data. However, DIVE is limited to quite a simple reasoning on the ontology, based on the multiple inheritance notion. It extends the basic oriented object philosophy found in JAVA, C# or VB.NET, for instance. Consequently, only some ontological relationships are available: *is-a*, *contains*, *is-part-of* and *bound-by*. There is no notion of cardinality or logical operators to define the concept classes. Our needs go slightly beyond and we identified several requirements that our approach should satisfy.

2.2 Formalism choice

The formalism of knowledge representation used in our approach must address the following three rules to properly fit our platform needs:

1. Hierarchical data representation via concepts and properties
2. Reasoning possibility in order to extend the ontology or the dataset ruled by the ontology
3. Efficient query time on the data to stay within interaction time

Several formalisms exist to create ontologies and use them to define databases. A comparison of these formalisms can be found in the table 1.

We chose to use the Semantic Web approach in order to setup our own ontology. The Semantic Web has been created by the *World Wide Web Consortium* under the lead of Tim Berners-Lee, with the aim to share semantic data on the web [3]. It is broadly used by the biggest web companies to uniformly store and share data. It belongs to the family of the Description logics that use the notions of *concepts, roles* and *individuals*. The *concepts* are represented by the sub-ensemble of elements in a specific universe, the *roles* are the links between the elements and the *individuals* are the elements of the universe. In our previous example, ”Atom” and ”Charge” would be *concepts*, ”has-a” a *role* and the oxygen an *individual*, as an instance of the *concept* ”Atom”.

2.3 Semantic Web Layers

The Resource Description Framework (RDF) is the basic model of Semantic Webs. It is a graph-based model aiming to describe in a formal way resources and their associated metadata. It is based on a knowledge representation from triples, a triple being the smallest knowledge division in RDF. Every data description is then an ensemble of triples including (*subject*, *predicate*, *object*) [7].

The *subject* is the resource described, the *predicate* represents a property attached to the resource and the *object* may represent a resource or a data (being a value of the attached property). Any resource is identified by an URI (Uniform Resource Identifier) whereas a data is anonymous since it can be duplicated

Formalism	Domain description	Reasoning on knowledge	Big data management	Efficient	Implementation flexibility
Conceptual Graphs	X	X	-	X	-
Semantic networks	X	-	X	X	-
Classical logics	X	X	X	X	-
Description logics	X	X	X	X	-

Table 1: Comparison of different knowledge representation formalisms with respect to key criteria.

(numerical value, string, etc.). An example of a triple could be: (*#Pierre*, *#age*, *26*)

RDF has been quickly extended by a semantic layer in a model called RDF Schema (RDFS) [4]. It defines classes, sub-classes, properties and sub-properties from which RDF resources will inherit.

The semantic layer provided by RDFS can be complemented by another model named OWL. OWL is a computer standard, supports Semantic Web ontologies, grammatically defines RDF data to ensure their coherency, and sets up a semantic framework. It is complementary to the RDFS even if it goes further in terms of reasoning possibilities and resource description. More expressive than RDFS, OWL adds specific relationships based on logical links between properties or classes. It is then possible to add symmetry, transitivity, similarity or cardinality information on top of concept relationships. As a result, these triples:

```
<my:is-composed-of> <my:is-a>              <owl:TransitiveProperty>
<my:Protein>        <my:is-composed-of>  <my:Amino-acid>
<my:Amino-acid>     <my:is-composed-of>  <my:Atom>
```

will indirectly imply the following statement:

```
<my:Protein>      <my:is-composed-of> <my:Atom>
```

A crucial part of our immersive analytics platorm requirements was to be able to access the data defined by the ontology and stored as RDF triples. Efficient access can be achieved via SPARQL, a query language and protocol used to access databases based on RDF triples. SPARQL not only allows to access data but also to edit, add or remove them. SPARQL is closely related to SQL in its functioning, implementing the SELECT, FROM and WHERE keywords to build a query. Several operations are possible on the final results including SORT, JOIN, DISTINCT in order to filter or sort the results.

3 Architecture

The different layers introduced in the previous section need to be set up in order to create a fully integrated semantic layer around the immersive environment. Once this semantic definition is achieved, a core module will interface user actions and data manipulations within the immersive environment.

3.1 Ontology for the modeling of structural biology concepts

An OWL-based ontology was implemented as core of the platform, thereby creating a broad description of concepts an expert would have to manipulate during visualization and analysis activities. We previously mentioned that several bio-ontologies already exist. We extended one of them, a bio-ontology describing amino-acids and their biophysical and geometrical properties[1]. Each component structuring molecular complexes and each associated property coming from various common bio-informatics tools have been systematically defined in our ontology. However, since needs may vary, we designed this ontology such that it could easily be updated and enriched with new concepts. Our ontology has been designed around five categories, addressing five different parts of our platform:

- **Biomolecular knowledge** - Gather field-related concepts and objects in structural biology
- **3D structure representation** - Gather concepts related to the representation and visualization of 3D molecular complexes
- **2D data representation** - Gather concepts related to the representation of numerical analyses and their results
- **3D interactions** - Gather concepts related to the interactions in 3D environnements
- **2D interactions** - Gather concepts related to the interactions in 2D environnements

The separation of the categories does not induce the absence of relationships between any two categories. For instance, ”Atom” belongs to the **Biomolecular knowledge** category but is directly linked to the ”Sphere” concept from **3D structure representation**. The whole network of connections will then permit to reason on the ontology in order to support the advanced interactivity level required in our platform.

Concepts and properties among the **3D structure representation** and **2D data representation** categories gather the graphical elements that allow for the representation of the **Biomolecular knowledge** category. Shape, colors but also graph types are notions defined in these two categories.

In addition to the biomolecular concepts and representations previously cited, we also defined every concept around the interaction between the user and the data he will directly or indirectly manipulate. These interactions include commands proposed by most of the visualization software packages and analysis tools.

3.2 Molecular database

Once we set up our ontology, it was possible to feed the database by adding biological information gathered by the expert. The new information has to fit the vocabulary and classification defined by the rules present in the ontology in order to be adequately stored in the database.

The description of a molecular system is constructed from the analysis of any biological information that can be described by a character chain or a value and that corresponds to a concept or property identified in the ontology. Each information will be exhaustively gathered in the RDF database as triples. Within the scope of our study, we focused on numerical molecular simulations. These simulations output time series of static snapshots of the molecular system at a regular time step. The Hamiltonian of the simulated model will drive the system towards specific states that experts try to decipher in order to understand underlying molecular mechanisms. The whole simulation creates a trajectory where each state, at a precise time, is associated to a snapshot. Our ontology defines a snapshot by the *Model* concept. A *Model* gathers all the atom coordinates of the molecular system at a defined time step. In order to distinguish the different components of a system, these components are identified by *Chain*, another concept of our ontology. Each *Chain* in the system is composed of a sequence of *Residues* (also known as *Amino-acids* in proteins). The different inference rules present in the ontology save us to specify all the links between the different hierarchical components of a specific *Model* explicitly. As a result, a *Residue* that belongs to a specific *Chain* will be automatically associated to the corresponding *Model* where the *Chain* appears.

[1] http://bioportal.bioontology.org/ontologies/AMINO-ACID

Every geometrical property (position, angle, distance, etc.), physicochemical property (solvent accessibility, partial charge, bond, etc.) or analytical property (interaction energy, RMSD, temperature, etc.) is then integrated in the database and associated to individuals created from 3D structures (Model/Chain/Residue/Atom) for each step of the simulation. As a reminder, any individual is an instance of concepts defined in the ontology. Individuals and their properties form the population of the molecular database.

3.3 Interrogation and queries for the direct interaction

Once all data has been integrated in the RDF database, it is necessary to setup an interrogation system able to retrieve the data for visualization and processing following interaction events in the working space. Our implementation of the query system mainly relies on the usage of SPARQL, as introduced before, and provides several ways to address the different needs of our platform. The richness and flexibility of SPARQL queries allowed us to design a keyword to command interpretation engine that aims to transform a list of keywords into a comprehensive application command triggering an action in the working space.

One of the most widely used interactive techniques in immersive environments is the vocal command. Based on a vocal recognition process, it consists in translating a sentence or a group of words said by the user into an application command. Vocal commands have the strong advantage that they can be associated with gestures to express complex multimodal commands.

Most of the actions identified in our platform involve a structural group designated by the expert. These structural groups can be characterized by: identifiers having a biological meaning (residue ids are, by convention, numbered from one extremity of the chain to the other), unique identifiers in the RDF database, or via their properties. The interpretation of commands vocalized by the expert with natural language using a specific field-related vocabulary requires a representation carrying the complexity of the knowledge and linking the objects targeted by the user to the virtual objects involved in the interaction.

For this purpose, we set up a process that takes as input a vocal command of the user and translates it into an application command for the operating system. This procedure can be divided in three main parts:

1. Recognition of keywords from a vocal command
2. Keyword classification in a decomposed command structure
3. Creation of the final and operational command

Our conceptualization effort and the use of the ontology mainly focused on the second part. Parts one and three are more implementation oriented and will not be deeply described.

3.3.1 Keyword recognition

We are using the keyword spotting capability of Sphinx [2], a vocal recognition toolkit, to recognize keywords. Based on a dictionary created from the ontology list of concepts, it aims to detect any word said by the user that would match a word present in the dictionary.

3.3.2 Keyword classification

Each keyword recognized in the previous step is assigned to a category. This classification is based on our ontology splitting which identifies five categories of words that can be found in a vocal command, semantically modeled as:

- **Action**
- **Component**
- **Identifier**
- **Property**
- **Representation**

This classification is achieved through successive SPARQL queries to the ontology. *Action*, *Component*, *Property* and *Representation* categories have their own concepts and can be identified by a unique word (”Hide”, ”Chain”, ”Charged”, ”Sphere”, etc.). At the opposite, the *Identifier* category is linked to a concept instance from the *Component* category. A biological identifier is very likely to be redundant because of the repetition of the molecular system at each time step. Therefore it is mandatory to pair an identifier with a component in the keywords in order to validate its presence. Without component, any identifier is withdrwan from the list. If the identifier and the associated component exist in the database, the couple is validated.

SPARQL commands use the *ASK* operator to define whether a keyword belongs to a category or not. This operator takes one or several triples and returns a boolean that reflects whether the ensemble of triples is true or not with respect to the database. Some examples of queries can be found below:

```
ASK {my:Hide rdfs:subClassOf my:Action}
ASK {my:Alanine rdfs:subClassOf my:Biological_component}
ASK {my:Cartoon rdfs:subClassOf my:Representation}
ASK {my:Aliphatic rdfs:subClassOf my:Property}
```

Reasoning and inference rules are automatically used in SPARQL queries. For instance, the following query:

```
ASK {my:Alanine rdfs:subClassOf my:Biological_component}
```

will output *true* despite the absence of an explicit direct link between the two concepts (*Alanine* and *Biological_component*) since *AminoAcid*, *Residue* and *Molecule* are located between the two concepts (see Figure 2).

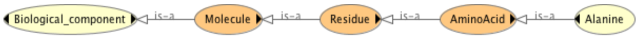


Figure 2: *Extract from our OWL ontology for the Alanine concept.*

Once each keyword is validated and associated to a category, e.g. identified as a concept of the database (or as an individual for identifiers) and eventually grouped with another keyword, it forms a syntactic group. Each syntactic group carries an information corresponding to a specific part of the application command.

3.3.3 From vocal command to application command

In our platform, a vocal command is composed by a succession of syntactic groups linked between them to create an action query to the immersive platform. It is possible to describe the type of command that was defined in the following manner:

$$action\ [parameter]^{+},\ (\ structural\_group\ [identifier]^{+}\ )^{+}$$

Syntactic groups between [] are optional, whereas others are mandatory. The $+$ indicates the possibility to have 0, 1 or several occurrences of the syntactic group. Finally, () indicates a bloc of syntactic groups. This command architecture is present in our ontology under the form of pre-required concepts associated to the *Action* concepts. For instance, the action concept *Color* requires a property of *Colors* type and a structural component to work with. These elements of information are then stored in the ontology, rendering them automatically checkable by the engine to detect whether all requirements are fulfilled for a specific action. This feature simpifies the definition of other actions in the ontology as

[2] http://sourceforge.net/projects/cmusphinx/

the changes that have to be applied to the engine are minimal, typically either no or minor changes. The checking process will stay the same as long as the action is well-defined within the ontology.

At the same level as for an action, a structural group is always mandatory to trigger a command. The different ways to obtain a structural sub-ensemble are:

1. Component only: every individual that belongs to the concept will be taken into account
2. Combination of a component and an ensemble of identifiers: coherency checking between component and identifiers
3. Property only: every individual that possesses the property will be taken into account
4. Combination of a component and a property: coherency checking between component and property

The structural group always refers to a group of individuals in order to disambiguate the results between the commands. This disambiguation implies that final commands are more complex. The hierarchical classification between structural components (Model/Chain/Residue/Atom) has a significant impact on the results of a given command. Indeed, the nature of structural components targeted by an action will be compared to the nature of the structural components currently studied. Depending on whether the command individual will be of a higher or lower hierarchical order, the command might trigger an action either on a subpart of the displayed scene (for lower classified individuals) or as a scene composition changer (for higher or equal classified individuals). For instance, if only two models are studied when a vocal command is transmitted, putative amino-acids individually targeted by an action will be the ones that belong to the two displayed models. If the individuals targeted by the command action would have been models, different from the displayed ones, an update of the displayed molecular complexes would have occurred first.

Once the different checks for the command coherency and validity have been carried out, the command is sent to both spaces (visualization and analysis) in order to synchronize the visual results.

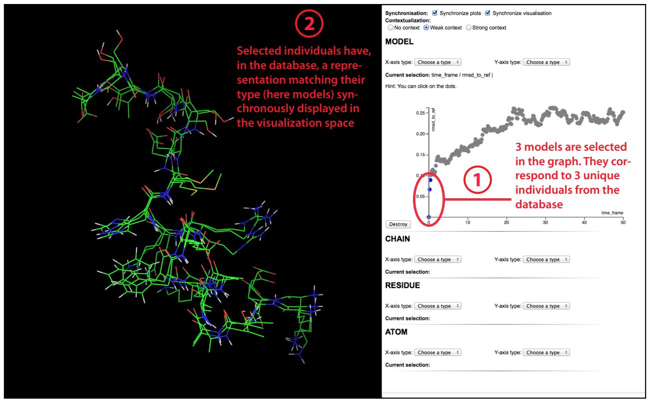
4 Reasoning over semantic data - Scenario

Our interpretation engine, able to translate a list of vocalized keywords into an application command, provides further possibilities through its semantic-based architecture. The visualization (or analytical) filter directly links an individual and its representations to any user interaction with a structural group, a property or an analytic value in either the visualization or analysis space. It is therefore possible to set up intelligent interactions in the immersive environment and synchronize any action of the expert between his two working spaces.

4.1 Test case

To illustrate the full-capacity of our platform, we chose a typical example of a molecular system study. This example sets up a local visualization solution coupled to a distant web server where interactive graphs can be created. Both spaces can be rendered in an immersive environment, either in the same screen space or split on one 3D screen for the visualization and a tablet providing analysis results through a web server. We assume, as it is the case in real studies, that the expert knows the molecular system well and can therefore interact vocally or by selecting elements in one of the spaces.

The first step of our scenario is a query triggered by the analytical space (web server) to retrieve every numerical value that can be represented in a scatter plot. This query will start the connection between the visualization space and the analytical space for the whole session. Once the values are gathered, the expert will choose which structural component hierarchy he is interested in and which combination of properties he wants to plot. Several queries will retrieve the property values and the web server graphs library, D3.js[3], will plot these values. We chose here to focus on the model level and display a simple distance criterion between a reference model and all the models of the studied trajectory.



2
Selected individuals have, in the database, a representation matching their type (here models) synchronously displayed in the visualization space

Synchronisation: ☑ Synchronize plots ☑ Synchronize visualisation
Contextualization:
○ No context ● Weak context ○ Strong context

MODEL

X-axis type: Choose a type   Y-axis type: Choose a type
Current selection: time_frame / rmsd_to_ref |
Hint: You can click on the dots.

rmsd_to_ref
0.25
0.20
0.15
0.1
0.05
0
10
20
30
40
50
time_frame
1
3 models are selected in the graph. They correspond to 3 unique individuals from the database
Destroy

CHAIN

X-axis type: Choose a type   Y-axis type: Choose a type
Current selection:

RESIDUE

X-axis type: Choose a type   Y-axis type: Choose a type
Current selection:

ATOM

X-axis type: Choose a type   Y-axis type: Choose a type
Current selection:

Figure 3: *On the right, selection of interesting models in the analytical space. The visualization state on the left is synchronized thanks to a SPARQL query in order to display the selected models.*

The expert can select several models of interest, as shown in the first step of figure 3. The selection is synchronized over all previously created scatter plots and will trigger a synchronous visualization of the individuals in the visual space (see second step of figure 3).

The expert may then switch to the visualization space and select some elements of the displayed structures he would like to focus on. These sub-elements of the current models will be sent to the analytical space that will ask the expert for the properties to be plotted. Once the choice is made, the selection will be highlighted in the analytical space as shown in the figure 4 for a selection of 3 residues.

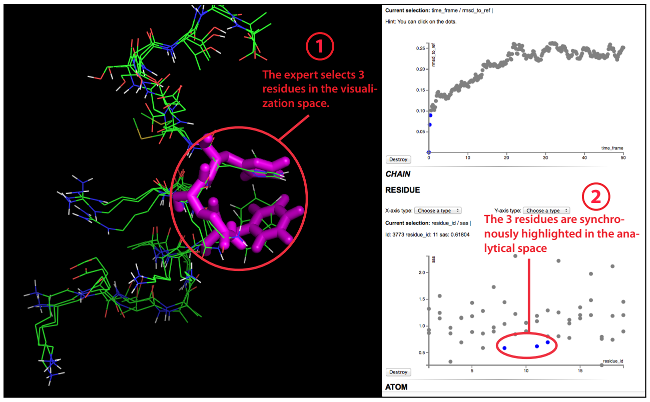
The selection process can be triggered by a vocal command through our interpretation engine or by a manual selection. New graphs can be added at runtime at any moment and are synchronized with the current ones. However, it is important to note that a full synchronization between the visualization and analytical spaces requires the same hierarchy of structural elements to be selected in both spaces. If a new selection is made at a *Model* level, any graphs of lower hierarchy will be reset with the new selected models and the visualization will be reset with the new models at the same time.

4.2 Evaluation

The evaluation process started from the observation that the systematic evaluation of field-related tasks is rather complicated to set up for four reasons. (1) Usage and nature of the evaluated tools, in particular in molecular visualization, differ between experts. (2) Implementation and adaptation of our developments over a representative sample of the tools is complex and very time-consuming. (3) Our approach is biased since it is based on the execution of expert tasks. (4) In order to apply standard statistic methods for evaluation, it is necessary to gather enough participants, yet the number of experts in our application field is rather limited.

We therefore propose an evaluation method that is more theoretically oriented than empirical: the HTA method (for ”Hierarchical Task Analysis”) [1]. The HTA method consists in a division of a primary task into several sub-tasks. Each sub-task can be subdivided again until the sub-tasks reach a degree of precision sufficient to have their execution time evaluated accurately.

[3] `https://d3js.org/`



1 The expert selects 3 residues in the visualization space.

2 The 3 residues are synchronously highlighted in the analytical space

Current selection: time_frame / rmsd_to_ref |

Hint: You can click on the dots.

Current selection: residue_id / sas |

Id: 3773 residue_id: 11 sas: 0.61804

Figure 4: *On the left, selection of amino-acids of interest. The analytical space on the right is synchronized thanks to a SPARQL query and proposes analyses adapted to the visualization context to the expert.*

This method is particularly useful to compare similar tasks performed under different conditions. It allows to evaluate both the task methodology with respect to specific conditions and the performance of the conditions for a specific task. HTA requires only one expert to evaluate the different sub-task execution times.

5 Conclusion

We have set up a semantic layer over an immersive environment dedicated to the interactive visualization and analysis of molecular simulation data. This setup was achieved through the implementation of an ontology describing both structural biology and interaction concepts manipulated by the experts during a study process.

Our architecture, built around heterogeneous components, achieves to bring together visualization and analytical spaces thanks to a common ontology-driven module that maintains a perfect synchronization between the different representations of the same elements in the two spaces.

The knowledge provided by the ontology can significantly improve the interactive capability of the platform by proposing contextualized analysis choices to the user, adapted to the types of elements in his current focus. All along the study process, a set of specific analyses, non redundant with the ones already performed, can be interactively chosen to populate the database. A simple definition of analyses in the ontology, adding input and output types, is sufficient to decide whether an analysis is pertinent or not for a precise selection, and whether the resulting values are already present in the database or not.

The reasoning capability of the ontology allowed us to develop an efficient interpretation engine that can transform a vocal command composed of keywords into an application command. This framework paves the way for a multimodal supervision tool that would use the high-level description of the manipulated elements, as well as the heterogeneous interaction natures, to merge inputs and create intelligent and complex commands in line with the work of M.E. Latoschik [21] or [8]. The RDF/RDFS/OWL model coupled to the SPARQL language allows to enunciate rules of inference, which is particularly important for the decision taking process in collaborative contexts. In these contexts, two users may trigger a multimodal command, in a conjoint way, that can be difficult to interpret without proper rules. An effort would then have to be made to integrate these rules in a future supervisor of the input modality, based on the semantic model, taken users as elements of modality in a multimodal interaction.

Our approach opens the way to a new generation of scientific applications. We illustrated our developments through the field of structural biology but it is worth to note that the generic nature of the Semantic Web allows to extend our developments to most scientific fields where a tight coupling between visualization and analyses is important.

References

[1] J. Annett. Hierarchical task analysis. *Handbook of cognitive task design*, pages 17–35, 2003.

[2] M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, J. T. Eppig, M. A. Harris, D. P. Hill, L. Issel-Tarver, A. Kasarskis, J. E. Richardson, M. Ringwald, G. M. Rubin, and G. Sherlock. Gene Ontology: tool for the unification of biology. *Nature Genetics*, 25(1):25–29, May 2000.

[3] T. Berners-Lee, J. Hendler, O. Lassila, et al. The semantic web. *Scientific american*, 284(5):28–37, 2001.

[4] D. Brickley and R. V. Guha. {RDF vocabulary description language 1.0: RDF schema}. 2004.

[5] W. L. DeLano. The PyMOL molecular graphics system. 2002.

[6] N. Frey, J. Nelson, C. Martin, L. Picinali, G. Bouyer, A. Tek, P. Bourdot, J. M. Burkhardt, B. F. G. Katz, M. Ammi, C. Etchebest, and L. Autin. Multisensory VR interaction for protein-docking in the CoRSAIRe project. *Virtual Reality*, 13(4):273–293, Dec. 2009.

[7] E. GODEL. Bach: An eternal golden braid. *Douglas R. Hofstadter.(Originally published by Basic Books, 1979.) Basic Books*, 1999.

[8] M. Gutierrez, F. Vexo, and D. Thalmann. Semantics-based representation of virtual environments. *International journal of computer applications in technology*, 23(2-4):229–238, 2005.

[9] M. D. Hanwell, D. E. Curtis, D. C. Lonie, T. Vandermeersch, E. Zurek, and G. R. Hutchison. Avogadro: An advanced semantic chemical editor, visualization, and analysis platform. *J. Cheminformatics*, 4(1):17, 2012.

[10] J. D. Hirst, D. R. Glowacki, and M. Baaden. Molecular simulations and visualization: introduction and overview. *Faraday discussions*, 169:9–22, 2014.

[11] W. Humphrey, A. Dalke, and K. Schulten. VMD: visual molecular dynamics. *Journal of molecular graphics*, 14(1):33–38, 1996.

[12] S. Irawati, D. Calderón, and H. Ko. Semantic 3d object manipulation using object ontology in multimodal interaction framework. In *Proceedings of the 2005 international conference on Augmented tele-existence*, pages 35–39. ACM, 2005.

[13] F. Kleinermann, O. De Troyer, H. Mansouri, R. Romero, B. Pellens, and W. Bille. Designing semantic virtual reality applications. In *Proceedings of the 2nd INTUITION International Workshop, Senlis, France*, volume 61. Citeseer, 2005.

[14] Z. Lv, A. Tek, F. Da Silva, C. Empereur-Mot, M. Chavent, and M. Baaden. Game on, science-how video game technology may help biologists tackle visualization challenges. *PloS one*, 8(3):57990, 2013.

[15] S. I. O'Donoghue, D. S. Goodsell, A. S. Frangakis, F. Jossinet, R. A. Laskowski, M. Nilges, H. R. Saibil, A. Schafferhans, R. C. Wade, E. Westhof, and A. J. Olson. Visualization of macromolecular structures. *Nature Methods*, 7:S42–S55, Mar. 2010.

[16] S. Rysavy, D. Bromley, and V. Daggett. DIVE: A Graph-Based Visual-Analytics Framework for Big Data. *IEEE Computer Graphics and Applications*, 34:26–37, Mar. 2014.

[17] S. Schulze-Kremer. Ontologies for molecular biology and bioinformatics. *In silico biology*, 2(3):179–193, 2002.

[18] N. Schuurman and A. Leszczynski. Ontologies for Bioinformatics. *Bioinformatics and Biology Insights*, 2:187–200, Mar. 2008.

[19] J. Stone, A. Kohlmeyer, K. Vandivort, and K. Schulten. Immersive molecular visualization and interactive modeling with commodity hardware. *Advances in Visual Computing*, pages 382–393, 2010.

[20] A. Van Dam, A. S. Forsberg, D. H. Laidlaw, J. J. LaViola Jr, and R. M. Simpson. Immersive vr for scientific visualization: A progress report. *Computer Graphics and Applications, IEEE*, 20(6):26–52, 2000.

[21] D. Wiebusch and M. E. Latoschik. Decoupling the entity-component-system pattern using semantic traits for reusable realtime interactive systems. In *IEEE VR Workshop on Software Engineering and Architectures for Realtime Interactive Systems*, IEEE VR, 2015.