Improving the usability of Kenzo, a Common Lisp system for Algebraic Topology

Jónathan Heras†, Vico Pascual†, Julio Rubio† and Francis Sergeraert‡

{jonathan.heras,vico.pascual,julio.rubio}@unirioja.es, francis.sergeraert@ujf-grenoble.fr

†:*Departamento de Matemáticas y Computación, Universidad de La Rioja*
‡: *Institut Fourier, Université Grenoble*

Table of Contents

Symbolic Computation Systems

Traditionally, symbolic computation systems have been oriented to research.

In the area of Computer Algebra Systems.

- Efficiency.
- Extension of the scope of the applications.

In the area of interoperability among symbolic computation systems.

- Universality of the Middleware.

In summary, we are always looking for more powerful systems. However, our systems became so powerful, that we can lose users or interactions.

Symbolic Computation Systems

Traditionally, symbolic computation systems have been oriented to research.

In the area of Computer Algebra Systems.

- Efficiency.
- Extension of the scope of the applications.

In the area of interoperability among symbolic computation systems.

- Universality of the Middleware.

In summary, we are always looking for more powerful systems. However, our systems became so powerful, that we can lose users or interactions.

Symbolic Computation Systems

Traditionally, symbolic computation systems have been oriented to research.

In the area of Computer Algebra Systems.

- Efficiency.
- Extension of the scope of the applications.

In the area of interoperability among symbolic computation systems.

- Universality of the Middleware.

In summary, we are always looking for more powerful systems. However, our systems became so powerful, that we can lose users or interactions.

Symbolic Computation Systems

Traditionally, symbolic computation systems have been oriented to research.

In the area of Computer Algebra Systems.

- Efficiency.
- Extension of the scope of the applications.

In the area of interoperability among symbolic computation systems.

- Universality of the Middleware.

In summary, we are always looking for more powerful systems. However, our systems became so powerful, that we can lose users or interactions.

Symbolic Computation Systems

Traditionally, symbolic computation systems have been oriented to research.

In the area of Computer Algebra Systems.

- Efficiency.
- Extension of the scope of the applications.

In the area of interoperability among symbolic computation systems.

- Universality of the Middleware.

In summary, we are always looking for more powerful systems. However, our systems became so powerful, that we can lose users or interactions.

Symbolic Computation Systems

Traditionally, symbolic computation systems have been oriented to research.

In the area of Computer Algebra Systems.

- Efficiency.
- Extension of the scope of the applications.

In the area of interoperability among symbolic computation systems.

- Universality of the Middleware.

In summary, we are always looking for more powerful systems. However, our systems became so powerful, that we can lose users or interactions.

Symbolic Computation Systems

Traditionally, symbolic computation systems have been oriented to research.

In the area of Computer Algebra Systems.

- Efficiency.
- Extension of the scope of the applications.

In the area of interoperability among symbolic computation systems.

- Universality of the Middleware.

In summary, we are always looking for more powerful systems.

However, our systems became so powerful, that we can lose users or interactions.

Symbolic Computation Systems

Traditionally, symbolic computation systems have been oriented to research.
In the area of Computer Algebra Systems.

- Efficiency.
- Extension of the scope of the applications.

In the area of interoperability among symbolic computation systems.

- Universality of the Middleware.

In summary, we are always looking for more powerful systems.
However, our systems became so powerful, that we can lose users or interactions.

Kenzo

Kenzo is a symbolic computation system devoted to Algebraic Topology.

Weak Points in Kenzo:

- Usability
- Accessibility

Solution:

Construct an *Intermediary Layer*, allowing us an *intelligent access* to some features of the system.

Kenzo

Kenzo is a symbolic computation system devoted to Algebraic Topology.

Weak Points in Kenzo:

- Usability
- Accessibility

Solution:

Construct an *Intermediary Layer*, allowing us an *intelligent access* to some features of the system.

Kenzo

Kenzo is a symbolic computation system devoted to Algebraic Topology.

Weak Points in Kenzo:

- Usability
- Accessibility

Solution:

Construct an *Intermediary Layer*, allowing us an *intelligent access* to some features of the system.

Kenzo

Kenzo is a symbolic computation system devoted to Algebraic Topology.

Weak Points in Kenzo:

- Usability
- Accessibility

Solution:

Construct an *Intermediary Layer*, allowing us an *intelligent access* to some features of the system.

Kenzo

Kenzo is a symbolic computation system devoted to Algebraic Topology.

Weak Points in Kenzo:

- Usability
- Accessibility

Solution:

> Construct an *Intermediary Layer*, allowing us an *intelligent access* to some features of the system.

Kenzo

Kenzo is a symbolic computation system devoted to Algebraic Topology.

Weak Points in Kenzo:

- Usability
- Accessibility

Solution:

Construct an *Intermediary Layer*, allowing us an *intelligent access* to some features of the system.

Noesis method

We have tried to guide our development with already proven methodologies and patterns.

We have followed the guidelines of the Noesis method in the design of the interaction with the user in our GUI front-end.

Noesis models provide modular tools

Noesis method

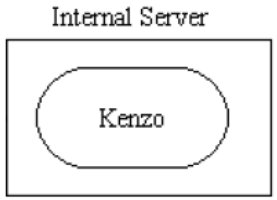
We have tried to guide our development with already proven methodologies and patterns.
We have followed the guidelines of the Noesis method in the design of the interaction with the user in our GUI front-end.
Noesis models provide modular tools

Noesis method

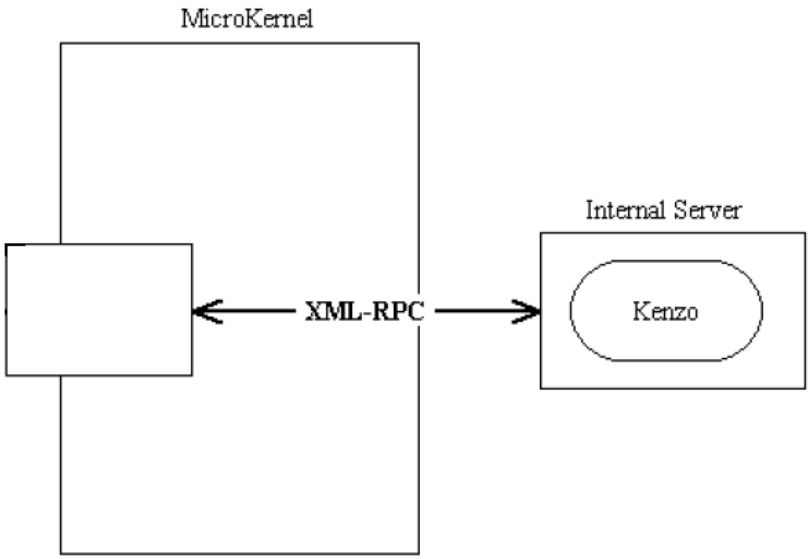
We have tried to guide our development with already proven methodologies and patterns.
We have followed the guidelines of the Noesis method in the design of the interaction with the user in our GUI front-end.
Noesis models provide modular tools

Microkernel architecture of the system

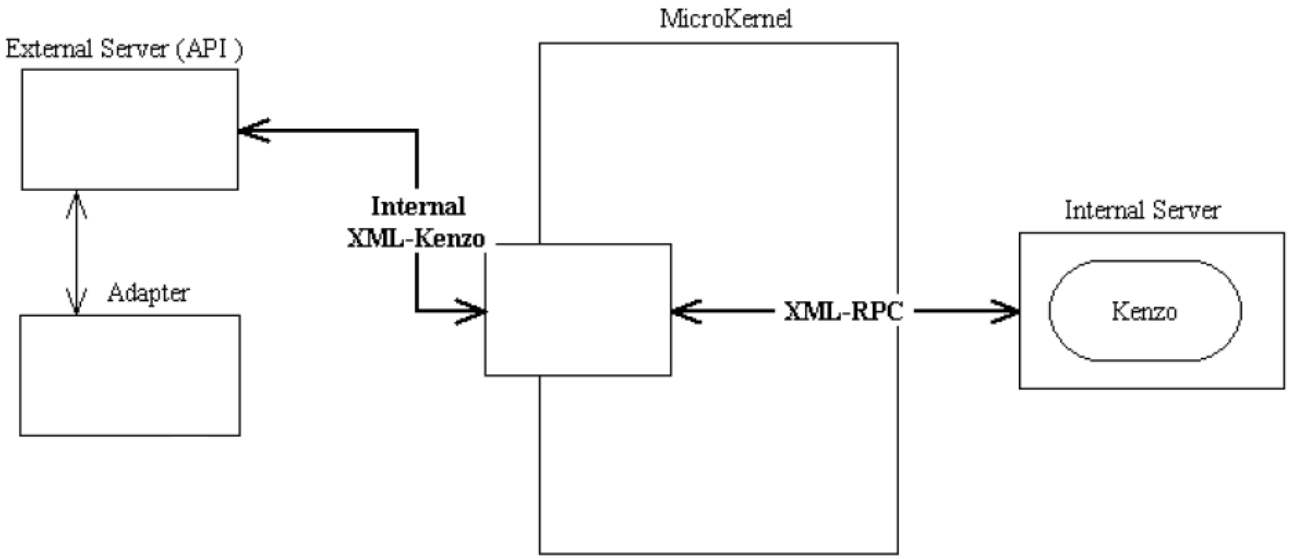
Microkernel architecture of the system



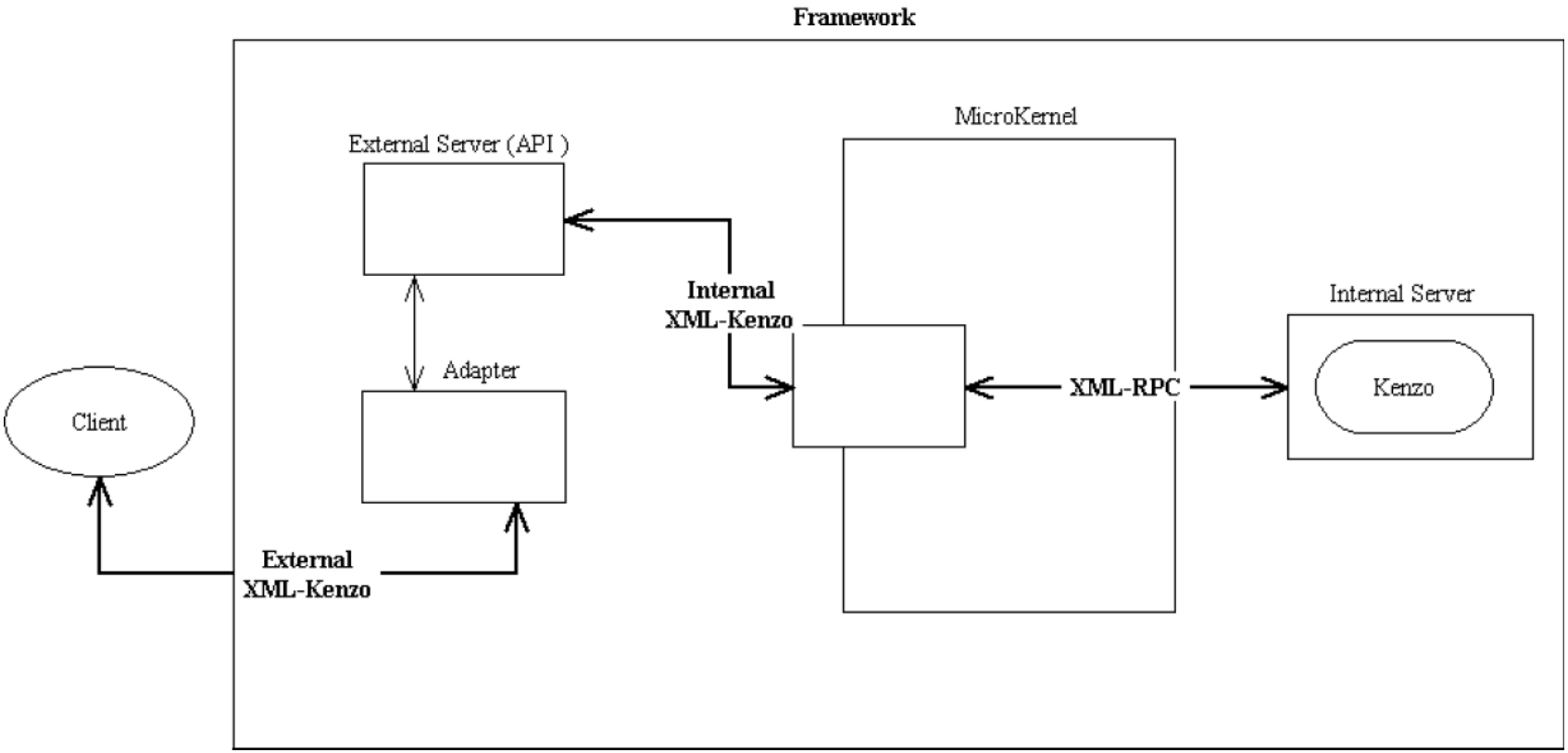
Microkernel architecture of the system



Microkernel architecture of the system



Microkernel architecture of the system



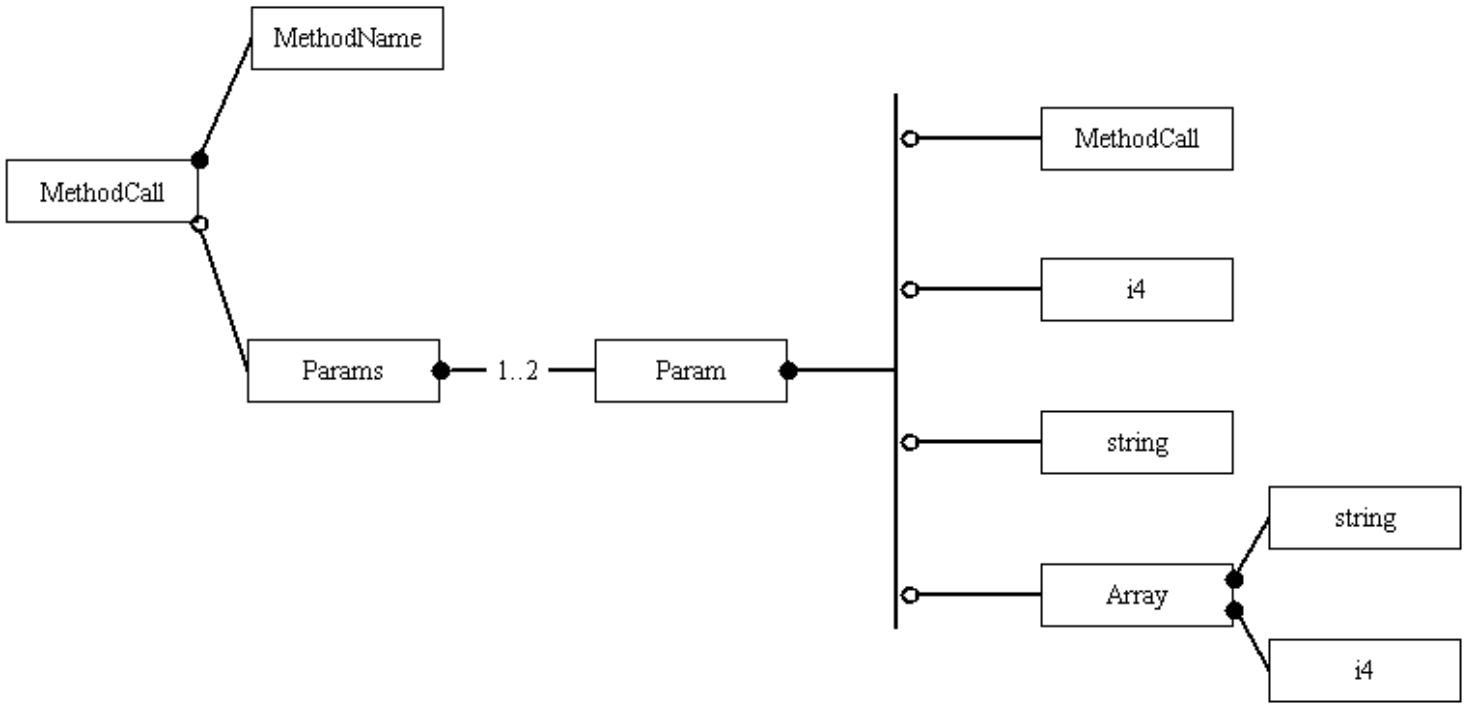
Inspired by the *MicroKernel* architectural pattern

XML

- XML-RPC: is a simple remote procedure call which uses XML to encode its calls.
- Internal XML-Kenzo
- External XML-Kenzo

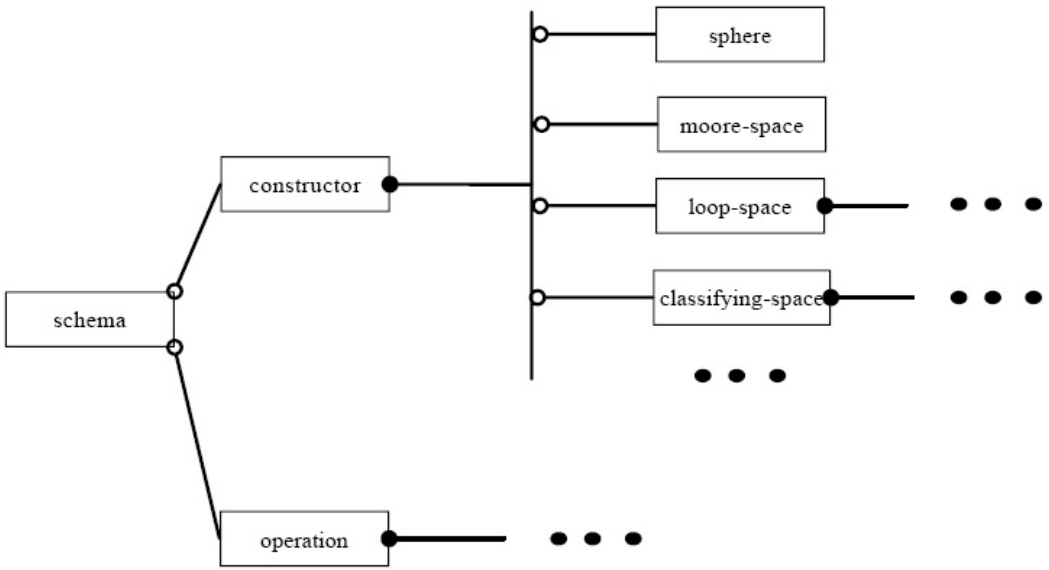
XML

- XML-RPC: is a simple remote procedure call which uses XML to encode its calls.
- Internal XML-Kenzo
- External XML-Kenzo



XML

- XML-RPC: is a simple remote procedure call which uses XML to encode its calls.
- Internal XML-Kenzo
- External XML-Kenzo



Intelligent Enhancements

The system includes the following "intelligent" enhancements:

1. Controlling the input specifications on constructors.
2. Avoiding some operations on objects which will raise errors.
3. Chaining methods in order to provide the user with new tools.
4. Determining if a calculation can be done in a local computer or should be derived to a remote server.

Intelligent Enhancements

The system includes the following "intelligent" enhancements:

1. Controlling the input specifications on constructors.
2. Avoiding some operations on objects which will raise errors.
3. Chaining methods in order to provide the user with new tools.
4. Determining if a calculation can be done in a local computer or should be derived to a remote server.

Intelligent Enhancements

The system includes the following "intelligent" enhancements:

1. Controlling the input specifications on constructors.
2. Avoiding some operations on objects which will raise errors.
3. Chaining methods in order to provide the user with new tools.
4. Determining if a calculation can be done in a local computer or should be derived to a remote server.

Intelligent Enhancements

The system includes the following "intelligent" enhancements:

1. Controlling the input specifications on constructors.
2. Avoiding some operations on objects which will raise errors.
3. Chaining methods in order to provide the user with new tools.
4. Determining if a calculation can be done in a local computer or should be derived to a remote server.

Intelligent Enhancements

The system includes the following "intelligent" enhancements:

1. Controlling the input specifications on constructors.
2. Avoiding some operations on objects which will raise errors.
3. Chaining methods in order to provide the user with new tools.
4. Determining if a calculation can be done in a local computer or should be derived to a remote server.

Objectives

We reach one of the objectives: code reuse.

This reusing has two aspects:

- We have left the Kenzo kernel untouched.
- The intermediary level has been used, without changes, both in the standalone local version and in the light client.

Objectives

We reach one of the objectives: code reuse.
This reusing has two aspects:

- We have left the Kenzo kernel untouched.
- The intermediary level has been used, without changes, both in the standalone local version and in the light client.

Objectives

We reach one of the objectives: code reuse.
This reusing has two aspects:

- We have left the Kenzo kernel untouched.
- The intermediary level has been used, without changes, both in the standalone local version and in the light client.

Objectives

We reach one of the objectives: code reuse.
This reusing has two aspects:

- We have left the Kenzo kernel untouched.
- The intermediary level has been used, without changes, both in the standalone local version and in the light client.

Nowadays

In this moment we have:

- A local version of our interface.
- A remote version of our interface.
- A prototype web application client.

Nowadays

In this moment we have:

- A local version of our interface.
- A remote version of our interface.
- A prototype web application client.

Nowadays

In this moment we have:

- A local version of our interface.
- A remote version of our interface.
- A prototype web application client.

Nowadays

In this moment we have:

- A local version of our interface.
- A remote version of our interface.
- A prototype web application client.

Kenzo Interface

Local or Remote?

When a calculation should be derived to a remote server?

Two kinds of state:

- The state of session.
- The state of space.

We should solve problems of distributed system in order to enable our intermediary layer as an intelligent assistant with respect to the classification of calculations.

Local or Remote?

When a calculation should be derived to a remote server?
Two kinds of state:

- The state of session.
- The state of space.

We should solve problems of distributed system in order to enable our intermediary layer as an intelligent assistant with respect to the classification of calculations.

Local or Remote?

When a calculation should be derived to a remote server?
Two kinds of state:

- The state of session.
- The state of space.

We should solve problems of distributed system in order to enable our intermediary layer as an intelligent assistant with respect to the classification of calculations.

Local or Remote?

When a calculation should be derived to a remote server?
Two kinds of state:

- The state of session.
- The state of space.

We should solve problems of distributed system in order to enable our intermediary layer as an intelligent assistant with respect to the classification of calculations.

Local or Remote?

When a calculation should be derived to a remote server?
Two kinds of state:

- The state of session.
- The state of space.

We should solve problems of distributed system in order to enable our intermediary layer as an intelligent assistant with respect to the classification of calculations.

Conclusions

We have showed how some intelligent guidance can be achieved in the field of CAT, without using standard Artificial Intelligence techniques.

$$\left.\begin{array}{l}\text{Kenzo}\\ \text{Intermediary Layer}\end{array}\right\} \Rightarrow \text{A Framework}$$

With this framework the interaction with Kenzo is easier and enriched.

Conclusions

We have showed how some intelligent guidance can be achieved in the field of CAT, without using standard Artificial Intelligence techniques.

$$\left.\begin{array}{l}\text{Kenzo}\\ \text{Intermediary Layer}\end{array}\right\} \Rightarrow \text{A Framework}$$

With this framework the interaction with Kenzo is easier and enriched.

Conclusions

We have showed how some intelligent guidance can be achieved in the field of CAT, without using standard Artificial Intelligence techniques.

$$\left.\begin{array}{l}\text{Kenzo}\\ \text{Intermediary Layer}\end{array}\right\} \Rightarrow \text{A Framework}$$

With this framework the interaction with Kenzo is easier and enriched.

Conclusions

We have showed how some intelligent guidance can be achieved in the field of CAT, without using standard Artificial Intelligence techniques.

$$\left.\begin{array}{l}\text{Kenzo}\\ \text{Intermediary Layer}\end{array}\right\} \Rightarrow \text{A Framework}$$

With this framework the interaction with Kenzo is easier and enriched.

Conclusions

We have showed how some intelligent guidance can be achieved in the field of CAT, without using standard Artificial Intelligence techniques.

$$\left.\begin{array}{l}\text{Kenzo}\\ \text{Intermediary Layer}\end{array}\right\} \Rightarrow \text{A Framework}$$

With this framework the interaction with Kenzo is easier and enriched.

Improving the usability of Kenzo, a Common Lisp system for Algebraic Topology

Jónathan Heras†, Vico Pascual†, Julio Rubio† and Francis Sergeraert‡

{jonathan.heras,vico.pascual,julio.rubio}@unirioja.es, francis.sergeraert@ujf-grenoble.fr

†:*Departamento de Matemáticas y Computación, Universidad de La Rioja*
‡: *Institut Fourier, Université Grenoble*