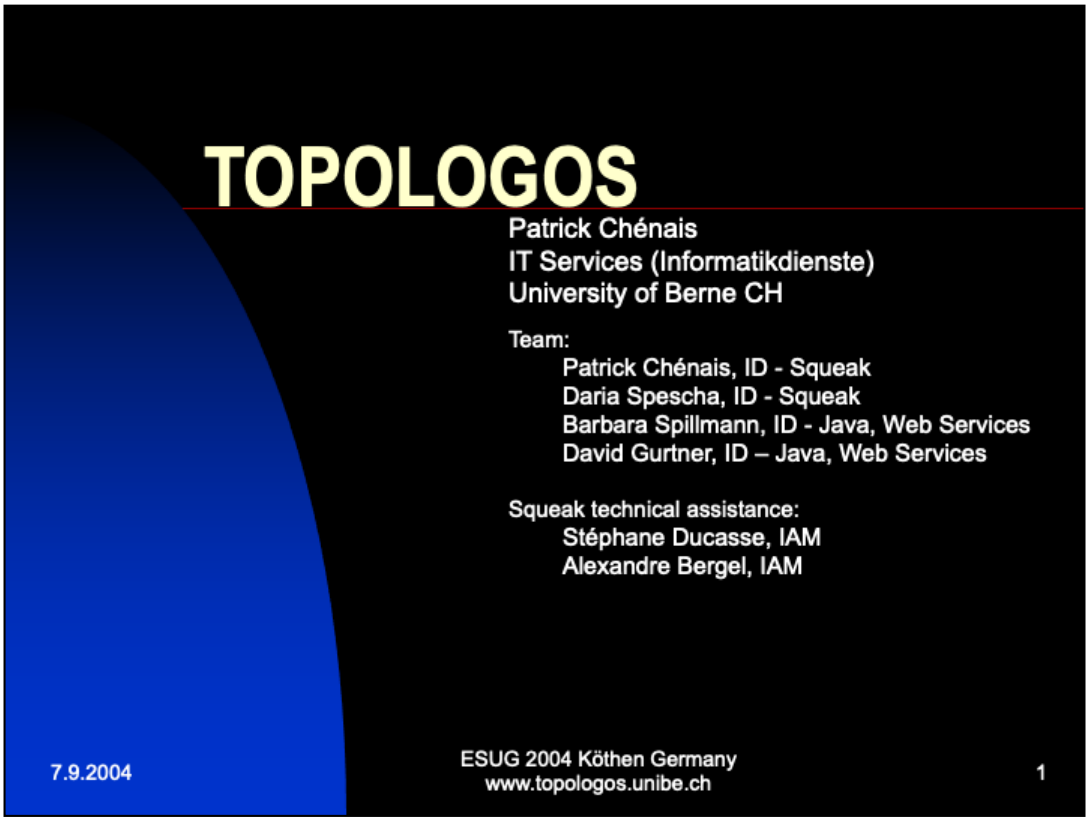


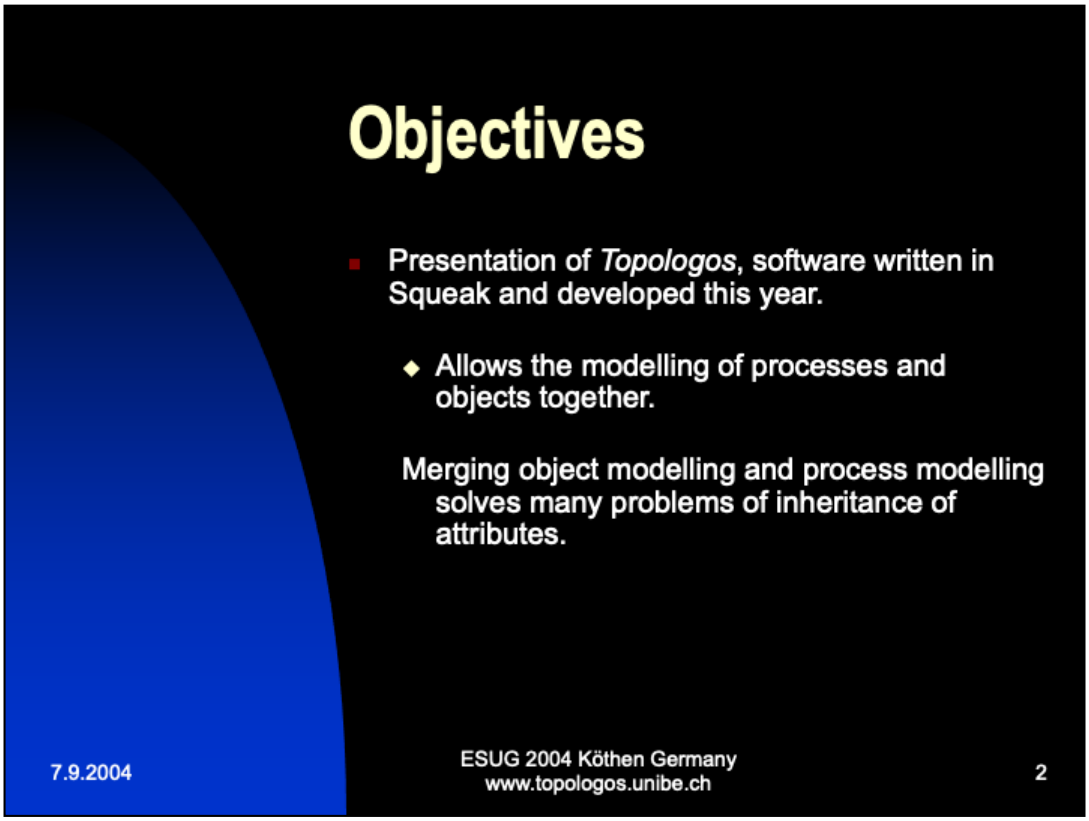
My name is Patrick Chénais. I am working at the university of Berne in the IT Services.

I am responsible for the development of system and applications for the administration of the university.

We are a small team working on that project, but it is not our main activity, we work about one day per week on this project.

I would like to thank Stéphane Ducasse and Alex Bergel from the Institute of Computer Science of the University of Berne for their help.

Without their assistance and motivation this project would not have been possible.



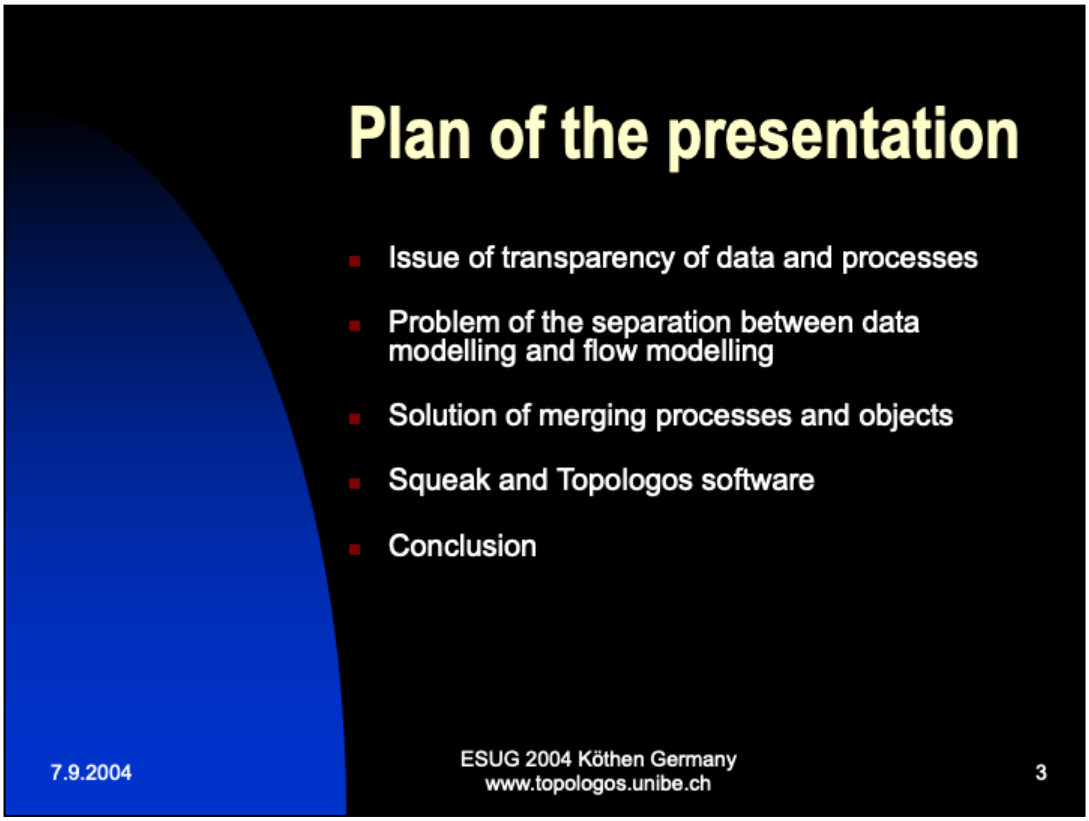
I would like to present you a software we called Topologos (coming from the name Topology) which has been developed this year at the university of Berne in the IT services.

It allows the modelling of processes and objects together.

The main message I would like to give here is following:
Merging object modelling and process modelling solves many problems of inheritance of attributes.

More …

This constitutes, in my opinion, a new approach in the domain of data and process modelling because one model includes process flows, class diagrams and instantiation of objects.



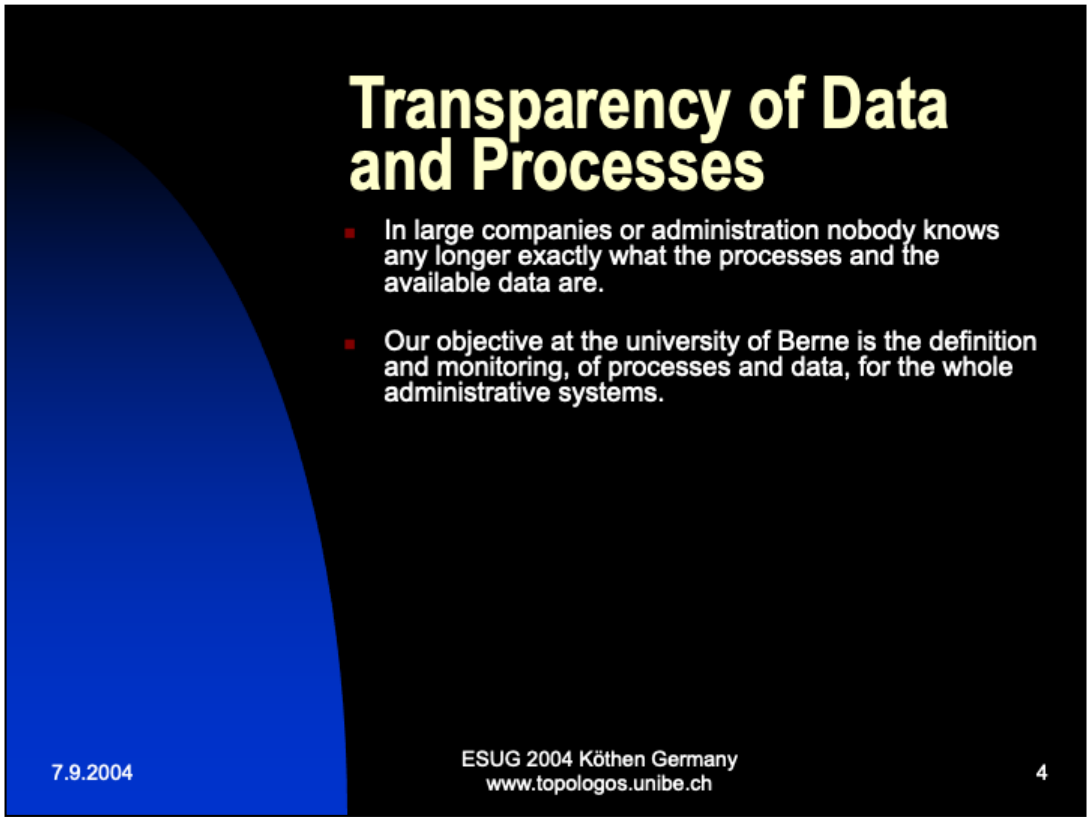
We will first see that the transparency of data and processes is not always guaranteed.

The problem is that object modeling and flow modeling are still two separated activities.

We will see our solution which consists in merging these two activities in one tool.

I will present how Squeak is used to validate the idea.

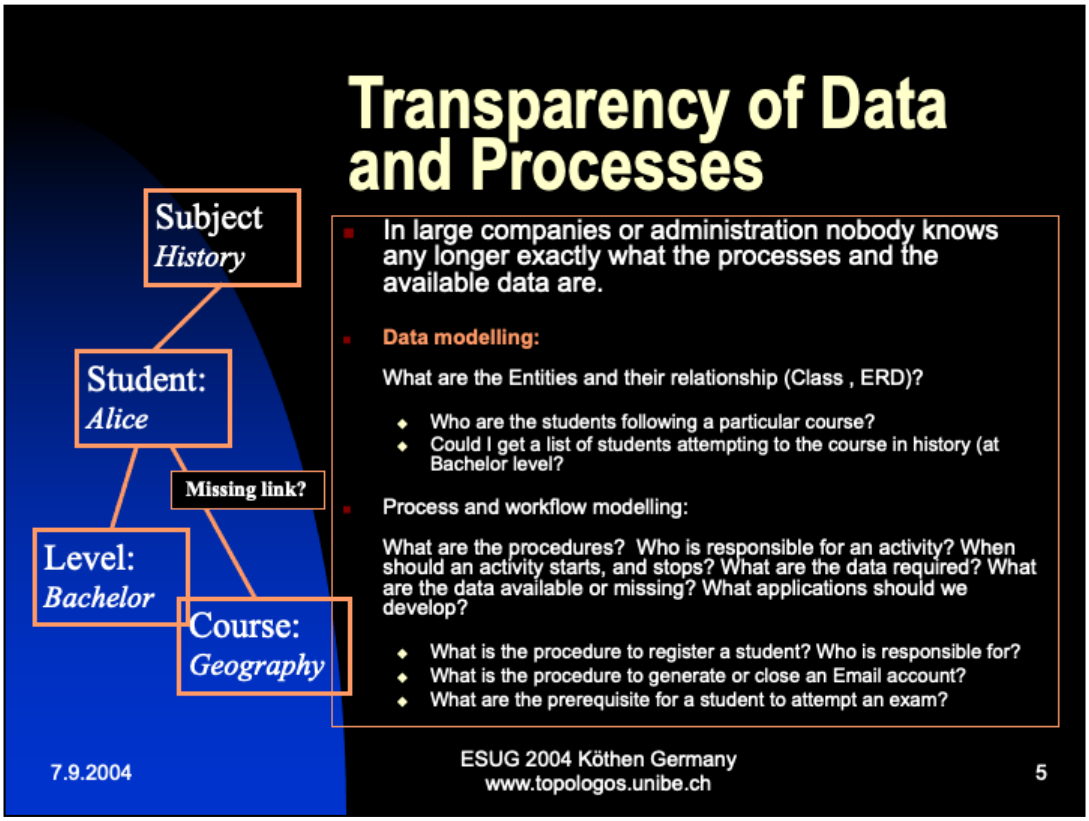
Finally, I will show a short conclusion.



How to ensure the transparency of Data and processes?

Today a big problem in a large company or an administration, such as the university, that nobody knows any longer exactly what the processes and the available data are.

Our objective at the university of Berne is the definition and monitoring, of processes and data, for the whole administrative systems.



What does data transparency mean? (1/2)

Some examples:

Could we know who are the students taking a particular course?

Could we have a list of students attempting to the course in history?

To answer this question we must know what are the data or objects available, in that example the students and the list of subjects, and we have to know which are the available or missing relations between these objects.

As IT engineers we should find a solution to these questions.

We should ensure the transparency of data and processes.

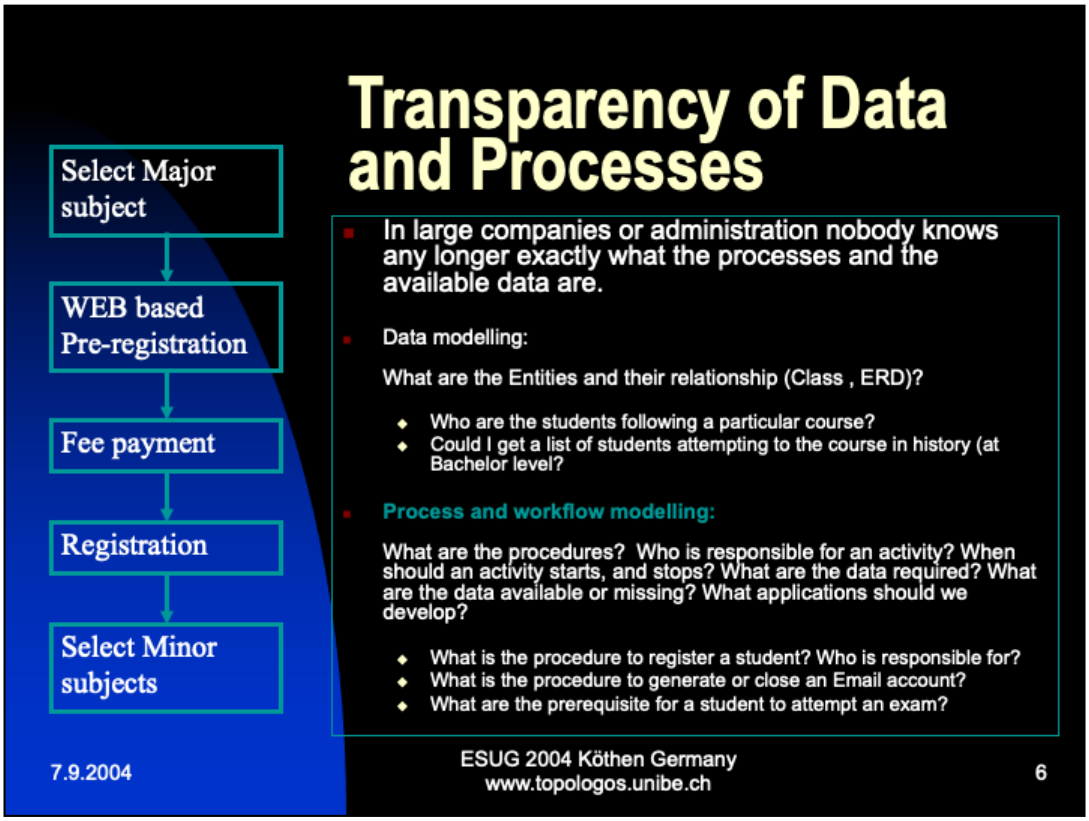
We should be able to answer the simple questions:

-Who is doing what and when?

-What are the required data and applications?

-Where are the available data?

-What are the missing data?



What does data transparency mean? (2/2)

Some examples:

Could we know who are the students taking a particular course?

Could we have a list of students attempting to the course in history?

To answer this question, we must know what are the data or objects available, in that example the students and the list of subjects, and we must know which are the available or missing relations between these objects.

As IT engineers we should find a solution to these questions.

We should ensure the transparency of data and processes.

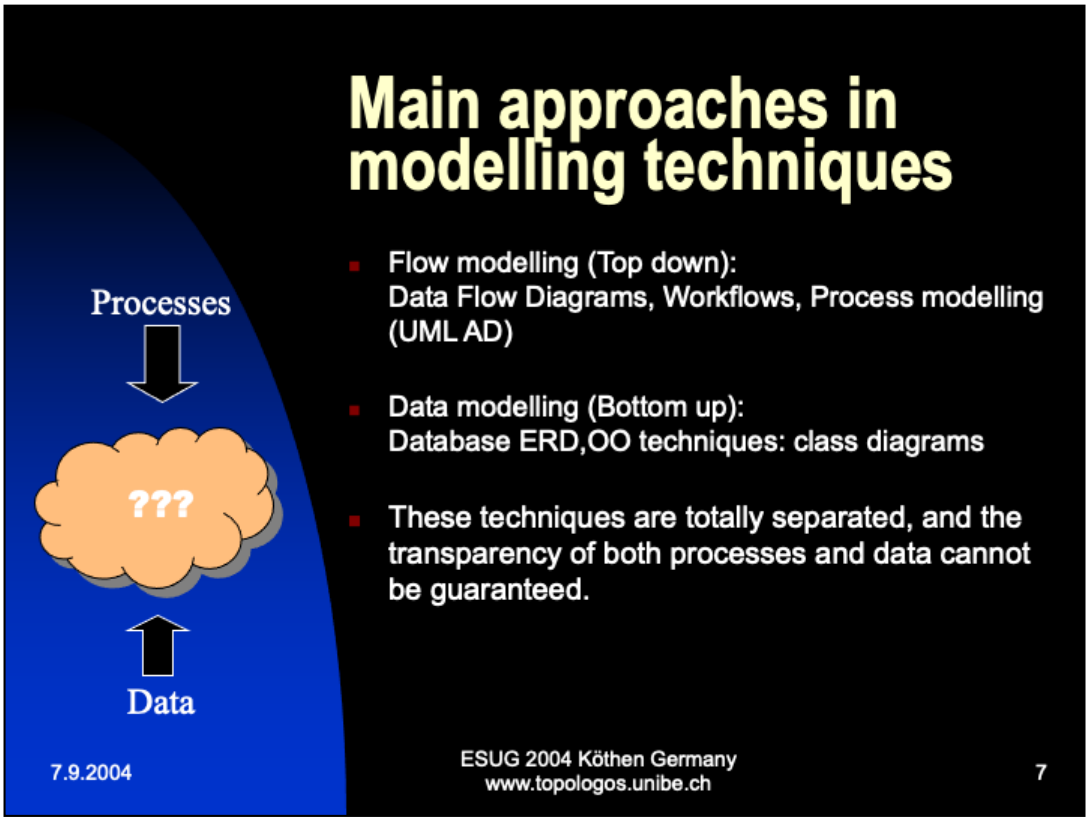
We should be able to answer the simple questions:

-Who is doing what and when?

-What are the required data and applications?

-Where are the available data?

-What are the missing data?



What are the main approaches in modelling techniques?

There are two major approaches in modelling techniques:

-Flow modelling which is top-down,

-and data modelling which is bottom-up.

Up to now these two techniques are totally separated, hence the transparency of both processes and data cannot be guaranteed.

How could we bring these techniques together?

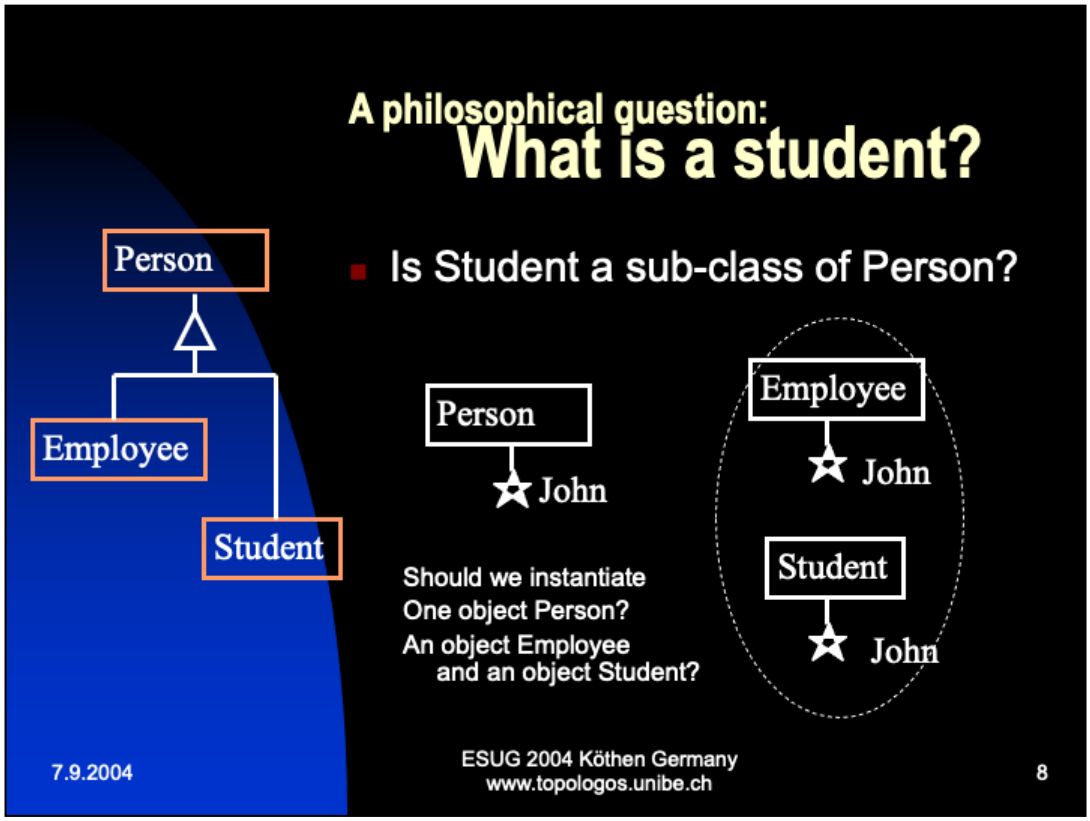
This is not only a technical issue, but also a conceptual.

More…

We will see that our solution at the university of Berne is the combination of process and data modelling by creating what we call OFD Object Flow Diagrams.

Note…

Structured analysis (DFD, Mini specification and Data Dictionary) performed a link between Data Flows and Data through the Data Dictionary.



A philosophical question: What is a student? (1/2)

The conceptual problem can be illustrated with two simple questions:

-What is a student?

-Is a student a sub class of a person?

→ 2

Consider a person who is both employee and student.

Should we instantiate one object person?

Then loose the specific attributes of both employee and student.

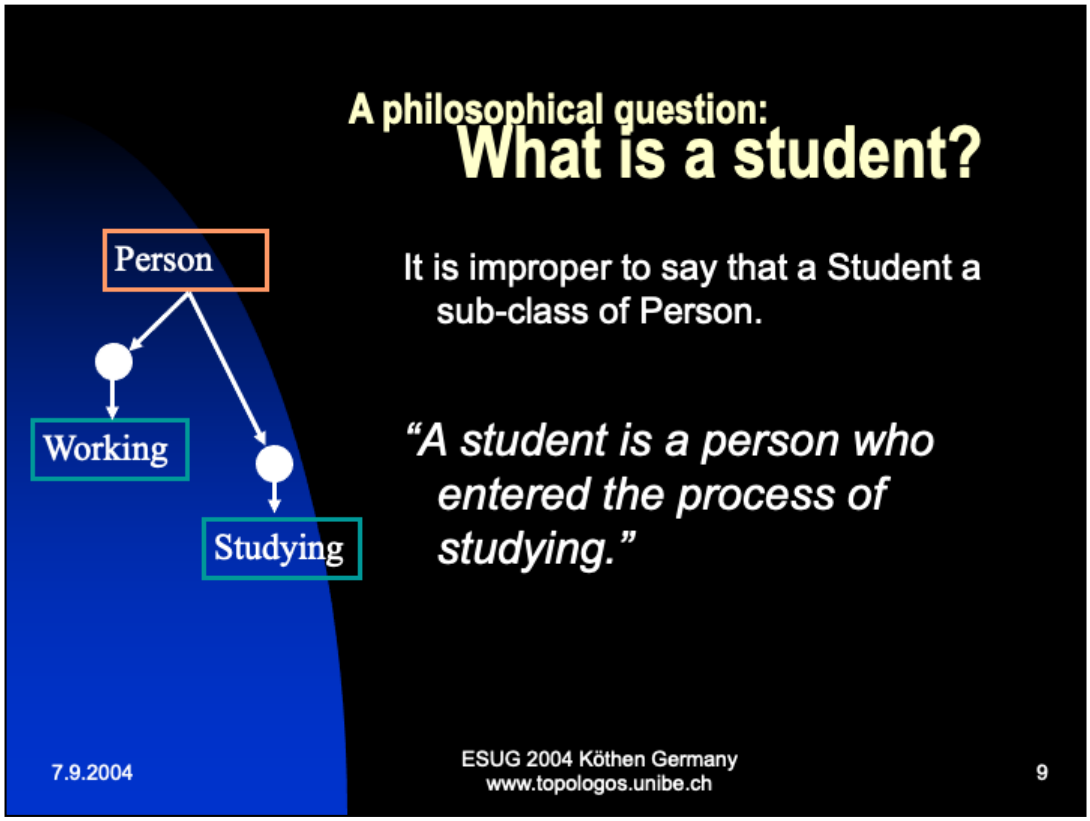
Or should we instantiate two objects, one object employee and one object student?

Doing this we will lose the information that these objects represent the same person.

→ 3 next

Therefore, it is improper to say that a student is a sub-class of Person.

We should say that a student is a person who entered the process of studying.

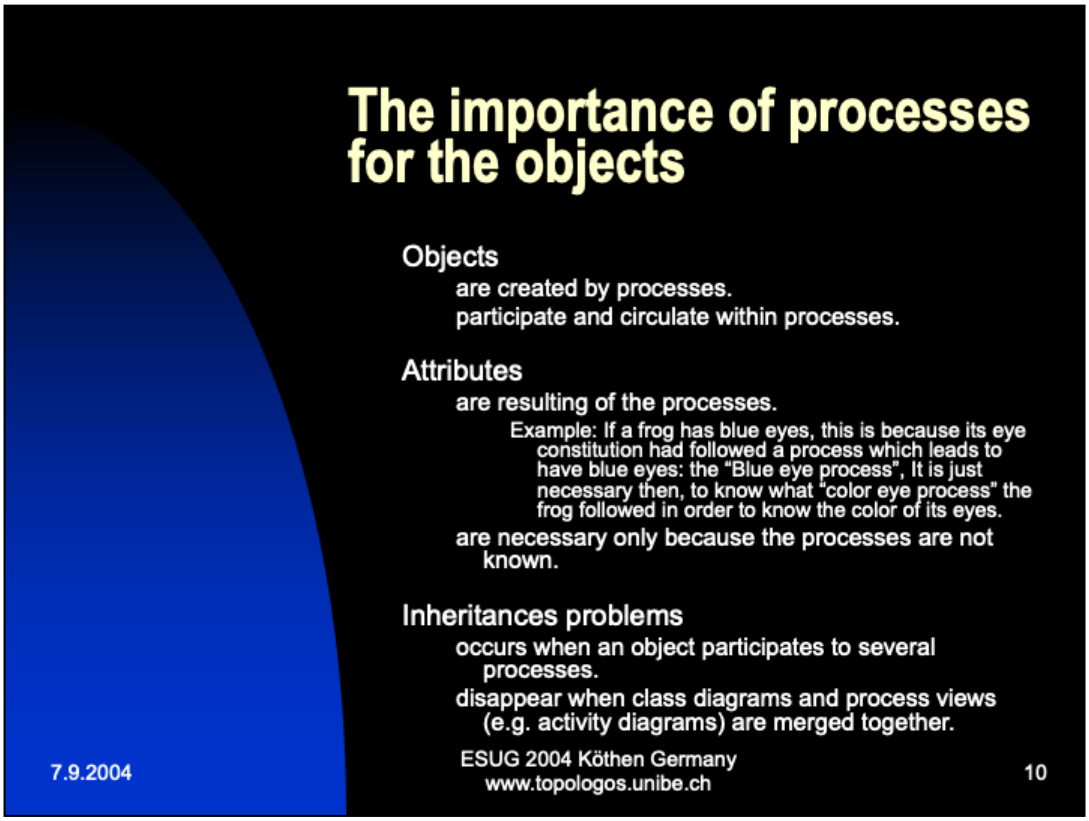


A philosophical question: What is a student? (2/2)
The conceptual problem can be illustrated with two simple questions:
-What is a student?
-Is a student a sub class of a person?

→ 2
Consider a person who is both employee and student.
Should we instantiate one object person?
Then loose the specific attributes of both employee and student.
Or should we instantiate two objects, one object employee and one object student?
Doing this we will lose the information that these objects represent the same person.

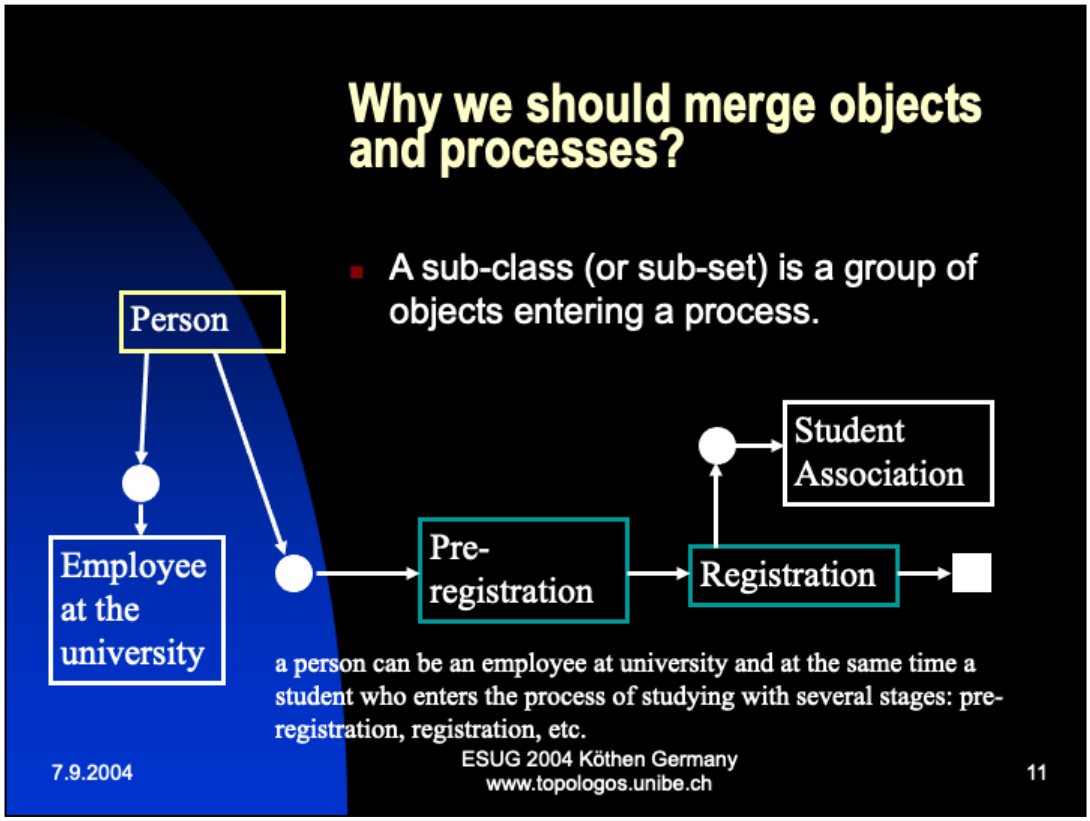
→ 3
Therefore, it is improper to say that a student is a sub-class of Person.
We should say that a student is a person who entered the process of studying.
A person can enter a multitude of process and inherit there from several attributes.

*** Because objects are created by processes, and objects circulate within processes,**
many inheritances problems disappear when class diagrams (object modelling) and process /activity diagrams) are merged.
Attributes are often there because the process which is the cause of the attribute is not known.
For example, attribute isStudent of a person is required only if the process student is not explicitly described.
I will explain why.



More …

Because objects are created by processes and circulate within processes, many inheritances problems disappear when class diagrams and process view or activity diagrams are merged.



A sub-class (or sub-set) is a group of objects entering a process.

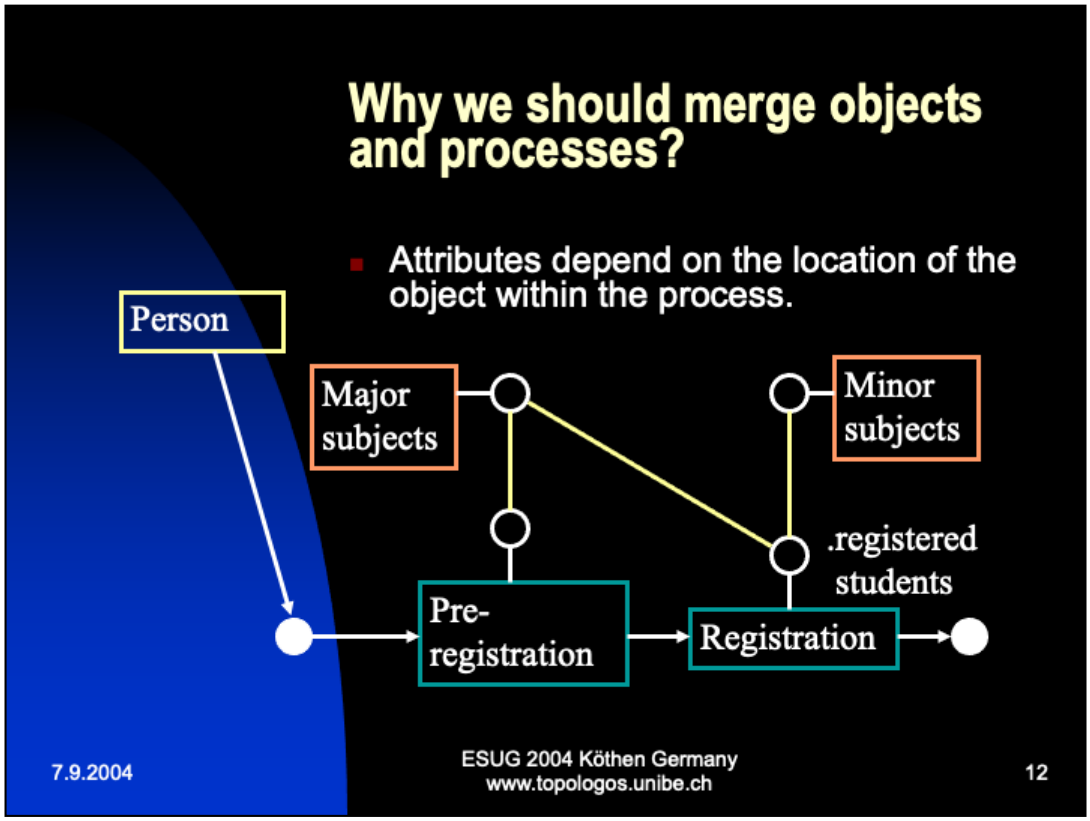
So, if we define the process activities where a class of objects can enter, we defined automatically the sub-classes of this class.

For example, a person can be an employee at university and at the same time a student who enters the process of studying with several stages: pre-registration, registration, etc..

A person can participate in as many processes he or she wants.

The person will inherit the attributes of all the processes he or she is entering.

This is a simple solution for the problem of multiple inheritance.



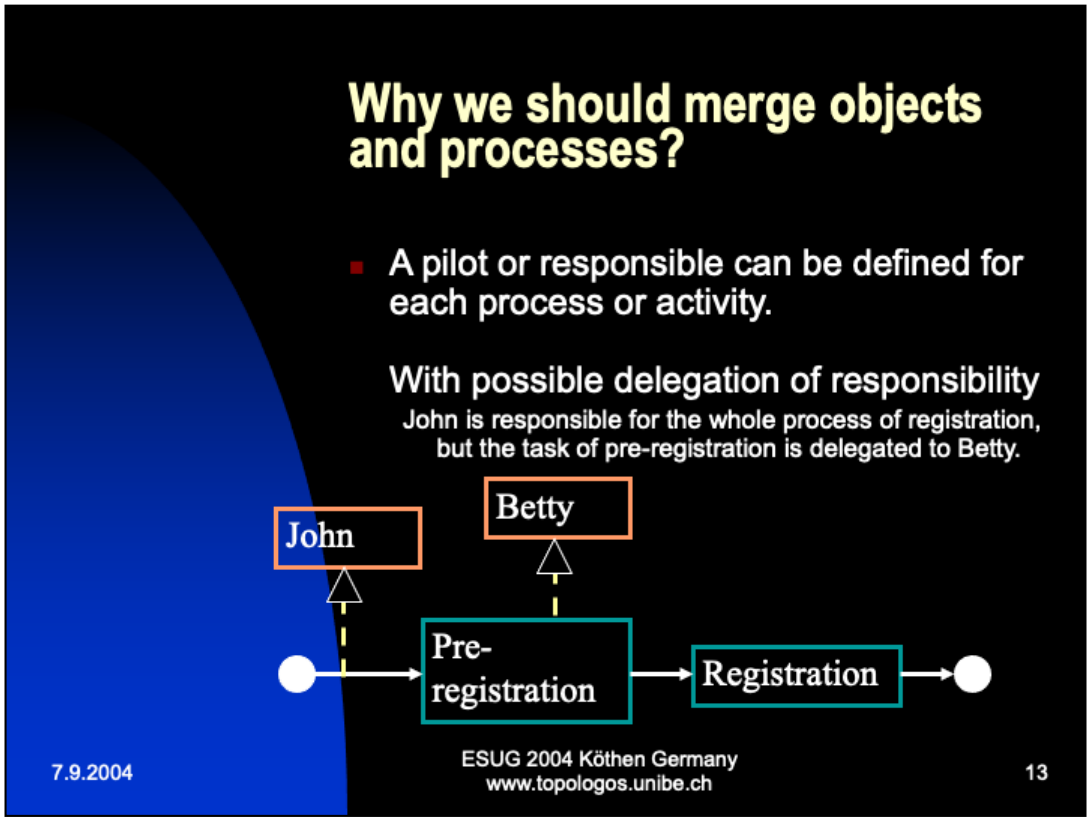
Attributes depend on the location of the object within the process.

For instance, in the pre-registration stage a student has only major subjects.

But a registered student may have both major subjects and minor subjects.

That is the reason why it is difficult, almost impossible, to merge several databases from different administrations or organisations.

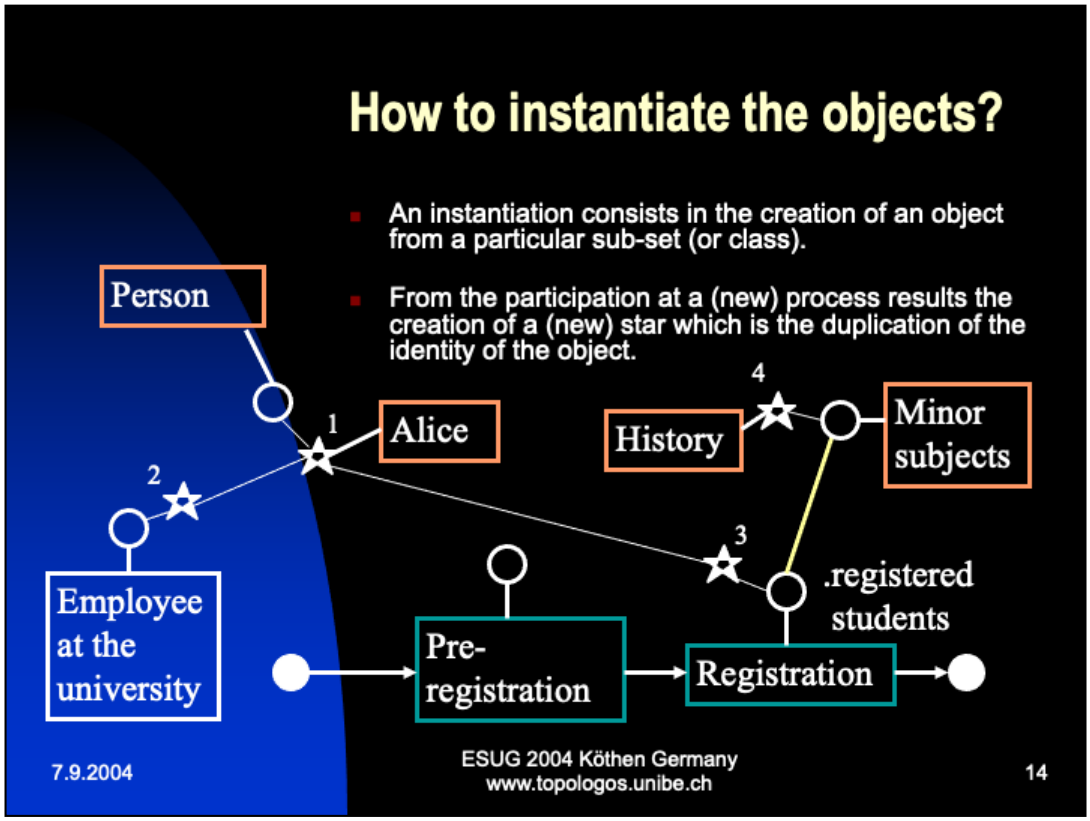
Each organization has its own process, and the data or attributes of the objects are specific to the process and therefore specific to the organisation. Some attributes of an object may have no signification or a different meaning outside the process or outside the organisation.



A pilot or responsible can be defined for each process or activity. Delegation of responsibility is possible.

For instance, John is responsible for the whole process of registration, but the task of pre-registration is delegated to Betty.

John and Betty belong to the object view. Pre-registration and registration belongs to process view.



An instantiation consists in a duplication of the identity.

For instance, Alice is a person.

She is an employee at university and at the same time she is a registered student who has major subjects but also a minor subject as for example history.

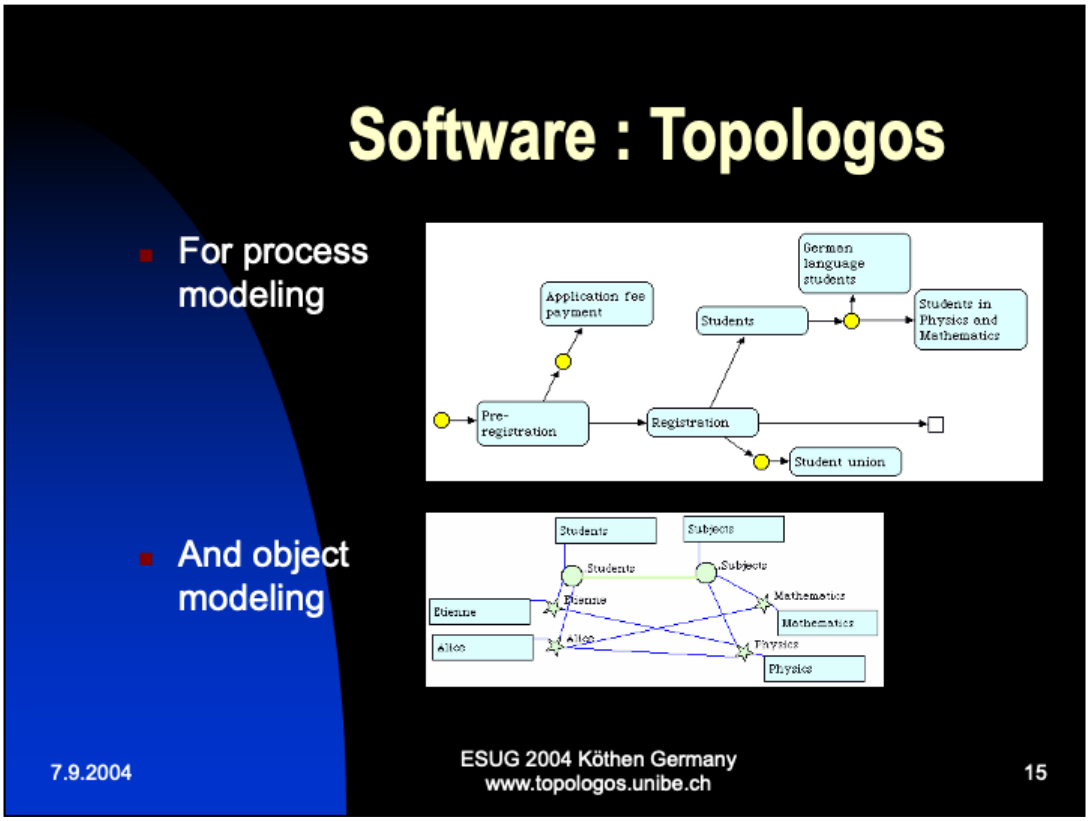
Each instantiation of the identity is represented with a star.

The stars are contained or linked to a container which is resented with a circle. A circle is a place or a set of individuals.

In a conventional queuing network, used in a context of an event driven simulation for instance, the objects are in one (and only one) queue.

It is impossible for instance to consider a wheel of a car which contributes to the fabrication of two cars.

In our case an object may contributes to several processes, hence we must consider not only the creation of the object #1 (instantiation) but also the duplication of its identity #2 3 4.



How does the Software work?

A short Demo.

-OUniverse new

-Creation Menu [MR]

-Create default model [default]

-Move elements

-Give a name and save

-Quit and reload

-Show step by step

- Entities and processes NODES
- Places (Classes) CIRCLES
- Instantiated objects STARS

-Add NODE, DOT, STAR, CIRCLE (Topologos 's basic elements)

-Make a process

-Define a PILOT

-Show Normal View and UML Like view

-Make examples from slides

-Note that STARS will be automatically created / Web Services

-Load complex model (m-esugUni.XML)

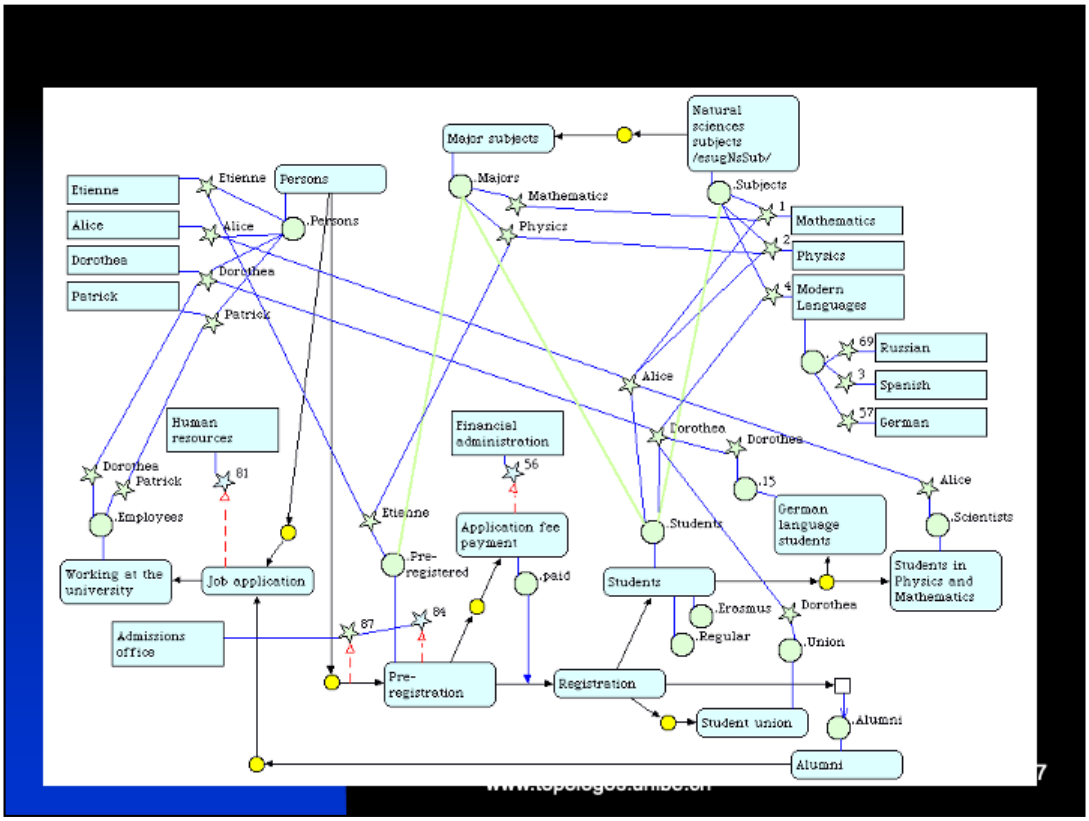
-select Star 'Alice' Etienne Dorothea Patrick

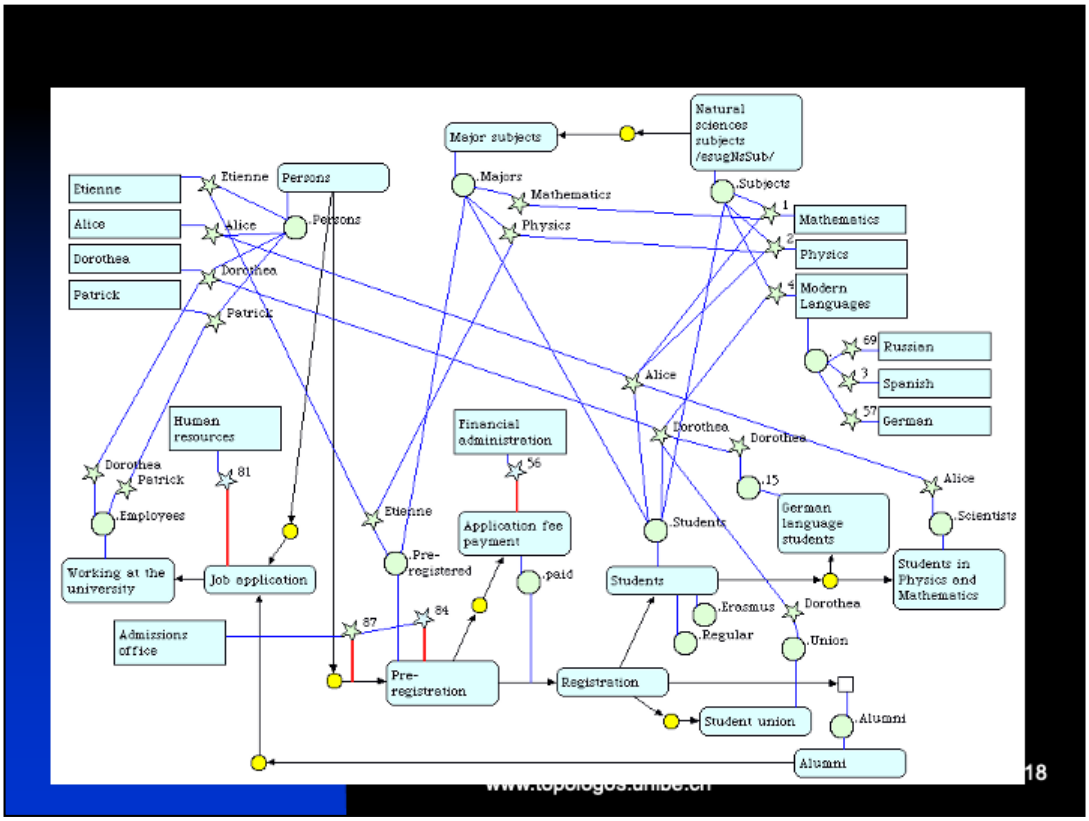
-Load sub-model (m-esugNsSubjects.XML)

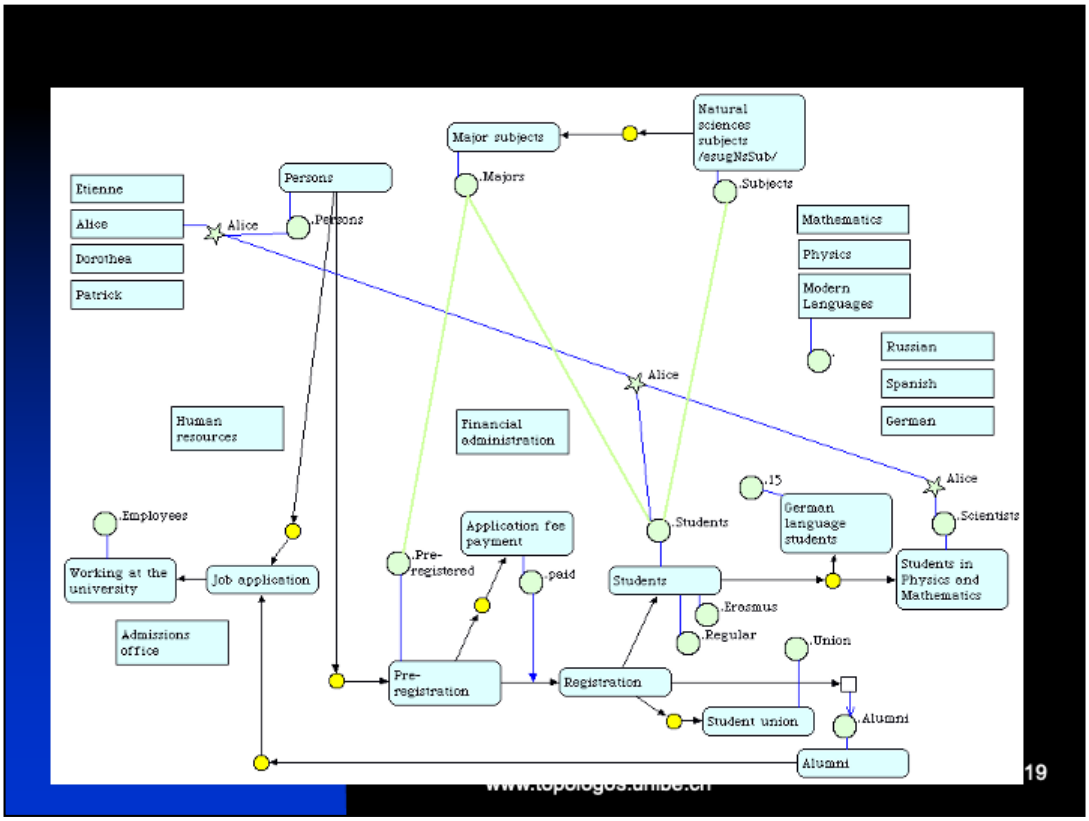
Software Elements

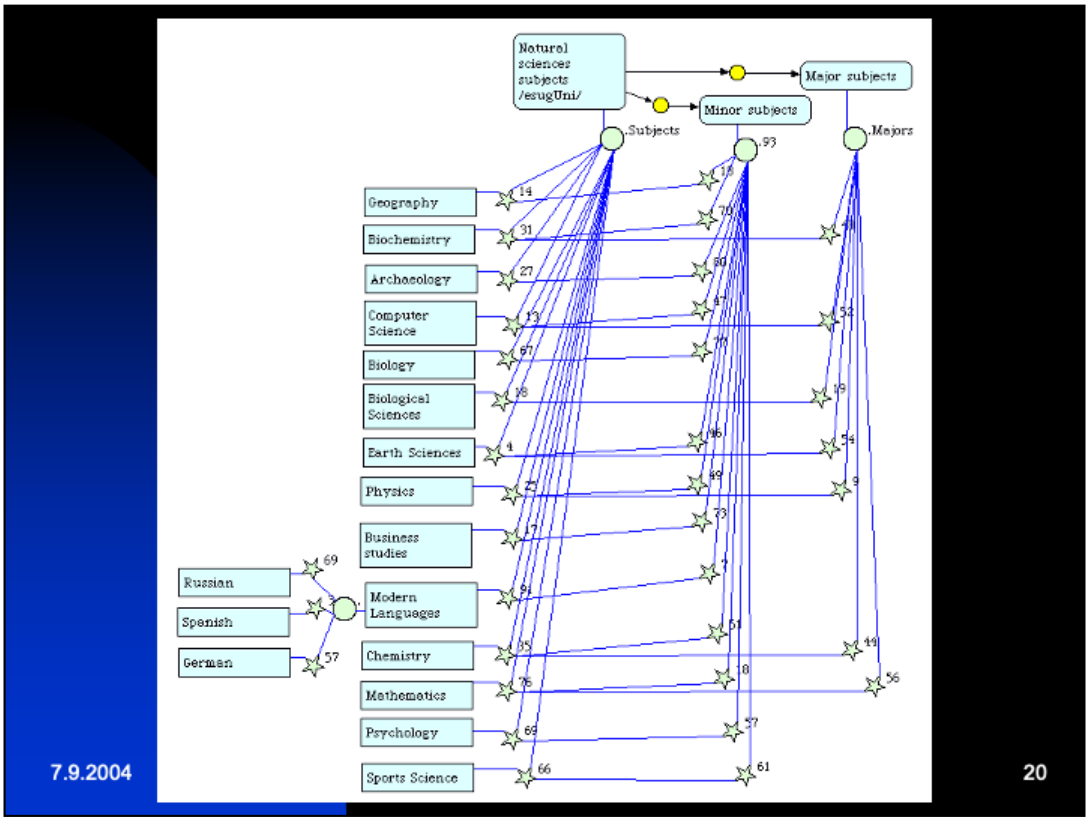
	**NODE**	**Entity** **Class or Object**
	**NODE**	**Activity** **Part of a process** **Process**
	NODE DOT	**Entity enters a process** **with duplication of identity**
	NODE GATE	**Gate of life** **Out coming of a process** **with lost of identity**
	**STAR**	**Identity**
	**STAR CIRCLE**	**Place** **Collection of objects** **Class**
	ARC	**Specify the process flow**
	RELATION	**Relationship between stars,** **Or between node and star.**

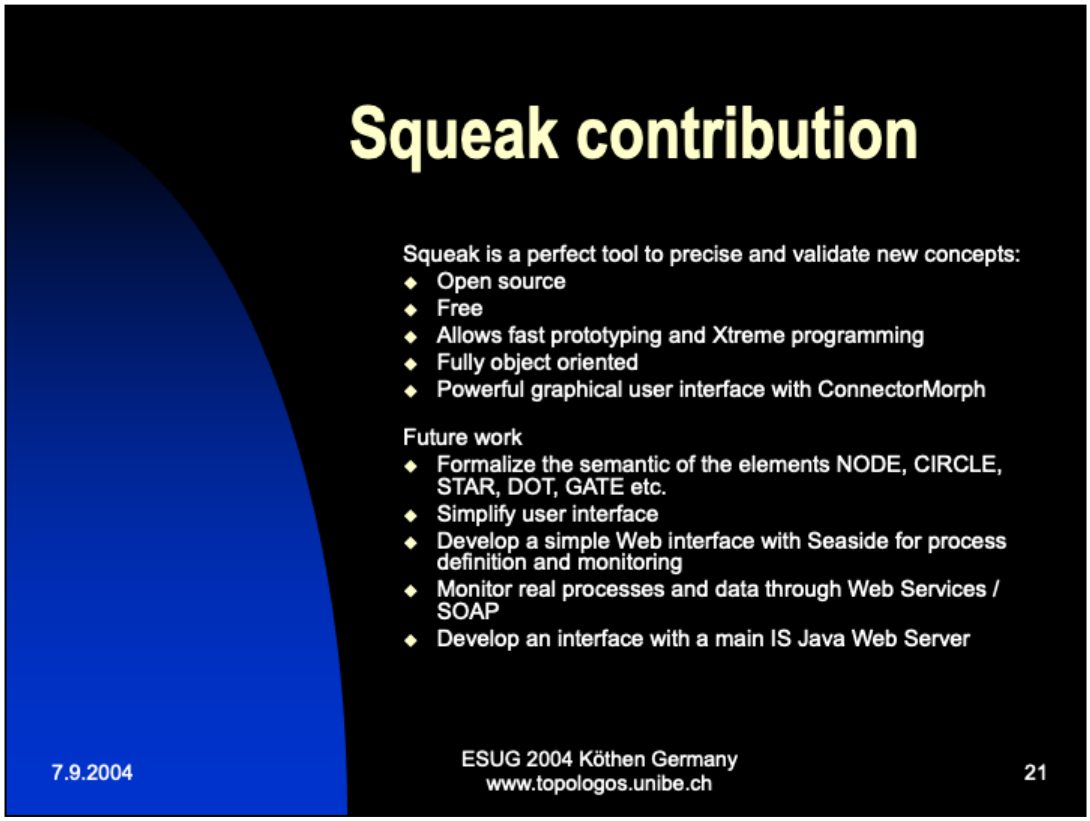
A DOT is a NODE without name. A GATE is a particular DOT.
A CIRCLE is a STAR whose name begins with a . (point)


Natural sciences subjects /esugNaSub/
Major subjects
Persons
Etienne
Alice
Dorothea
Patrick
Majors
Mathematics
Physics
Subjects
1
Mathematics
2
Physics
4
Modern Languages
69
Russian
3
Spanish
57
German
Human resources
Financial administration
Dorothea
Patrick
81
56
15
Alice
Employees
Etienne
Application fee payment
Students
German language students
Scientists
Working at the university
Job application
Pre-registered
paid
Students
Students in Physics and Mathematics
Admissions office
87
84
Erasmus
Regular
Union
Pre-registration
Registration
Student union
Alumni
Alumni


Natural sciences subjects /esugNsSub/
Major subjects
Etienne
Alice
Dorothea
Patrick
Persons
Majors
Mathematics
Physics
Subjects
Mathematics
Physics
Modern Languages
Russian
Spanish
German
Human resources
Financial administration
Employees
Working at the university
Job application
Admissions office
Application fee payment
Pre-registered
paid
Pre-registration
Registration
Student union
Alumni
Students
German language students
Scientists
Students in Physics and Mathematics
Erasmus
Regular
Union


Natural sciences subjects /esugNsSub/
Major subjects
Persons
Majors
Subjects
Etienne
Alice
Dorothea
Patrick
Alice
Persons
Mathematics
Physics
Modern Languages
Russian
Spanish
German
Alice
Human resources
Financial administration
Alice
15
German language students
Employees
Application fee payment
Students
Scientists
Working at the university
Job application
Pre-registered
paid
Students
Students in Physics and Mathematics
Erasmus
Regular
Admissions office
Union
Pre-registration
Registration
Student union
Alumni
Alumni
www.topologos.unibe.ch
19


Natural sciences subjects /esugUni/
Major subjects
Minor subjects
Subjects
Majors
93
Geography
Biochemistry
Archaeology
Computer Science
Biology
Biological Sciences
Earth Sciences
Physics
Business studies
Modern Languages
Chemistry
Mathematics
Psychology
Sports Science
Russian
Spanish
German
7.9.2004
20



What is the contribution of squeak to our project?

"A student is a person who entered the process of studying."

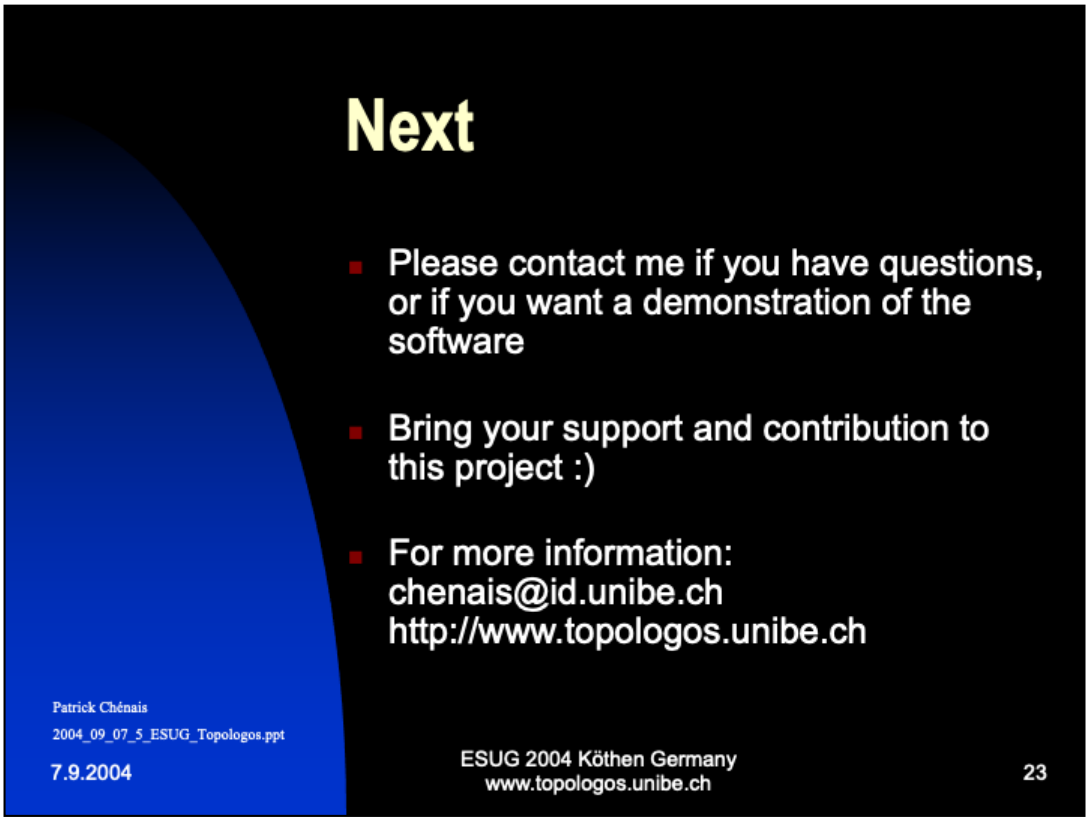
I needed a tool to validate this new concept. As Stéphane Ducasse introduced me to squeak in November, I noticed that squeak could be the perfect tool to precise and validate the concepts, because it is fully object oriented and has a powerful graphical user interface. Hence, I used squeak to develop a software called Topologos which has some main features we will see now.

What is the future work planed?

Topologos is still at the beginning of the development, and I may have some acceptance problems for Squeak which is so much different than others software. Hence, I surely need your support for the future work. We must show that Squeak is an efficient platform that can be used by administrative services. The first task is simplifying the user interface, especially for process modelling. Then a web interface has to be developed to enable users to define and later to monitor their own processes. The web interface will be developed using Seaside or Squeaklets. The final goal is to perform monitoring of real processes and data of the university with a live connection to production data bases through Web Services (and SOAP).

Conclusions

- Objects and processes must be tightly coupled in one modelling tool in order to obtain transparency of data and processes.
- The questions of the identity and the instantiation of objects 'What is a student?' lead to a new definition which allows integration of processes and objects within 'Object Flow Diagrams'.
- The software supports the definition and monitoring of processes and data (classes and objects).



Please contact me if you have questions, or if you want a demonstration of the software.

if you want to support or contribute to this project, you can obtain more information at the following address http://www.topologos.unibe.ch

I thank you for your attention.