Elettra Sincrotrone Trieste

Elettra
Sincrotrone
Trieste

TANGO

School on TANGO Controls system

Human Computer Interaction

Giacomo Strangolino

IT programmer at Elettra – Sincrotrone Trieste

Assistant professor 2010-2014, University of Trieste,
Faculty of engineering, principles of computer science

mailto: giacomo.strangolino@elettra.eu
http://www.tango-controls.org

HCI^{Q}

Branch of knowledge dealing with:

- Design, evaluation and implementation of computer systems for humans
- Design computer systems supporting human activities so that they are ***safe***, ***simple***, ***user friendly***, ***intuitive***, ***productive***, ***reliable***.

A green "Q" superscript indicates a HCI characteristic integrated in the QTango software

HCI

Understand how humans use the systems

Design tools for developers

Users needn't change to adapt to machines

HCI is part of the whole design lifecycle of a project

Users' needs, skills, preferences determine HCI design

HCI

Determines the success of a product in market

50% of the lines of code of a modern software

Neglecting HCI aspects can prevent from success when introducing a new system

Less flexibility, less productivity, more errors, users disappointment

HCI

catastrophic mishaps due to HCI design overlooking:

- **Three Miles Island nuclear accident** (ambiguous control room indicators in the power plant's user interface, the light on the control panel did not indicate the position of the valve, only the status of the solenoid being powered or not);

- **Iran Air 655 flight shooting down** by the US warship *Vincennes;*

HCI

- In medical sector, **administration of wrong doses** of drugs / radiations;

- **Lousy cockpit design crashed an airbus**, killing 228 people (lack of pilot feedback from the cockpit controls led to the crash of Air France Flight 447)

Elettra
Sincrotrone
Trieste

TANGO

I. Cognitive perspective

Ia. Perceptual System

Sensory channels

- ✔ Sight
- ✔ Sense of smell
- ✔ Hearing
- ✔ Touch
- ✔ Taste

Ia. Perceptual System

✔ **Sight**

✗ User focused on the center of the screen can detect moving objects in the corners

✗ Study of *depth* in order to create plausible 3D interfaces (objects dimension, overlapping, contrast, brightness, contour, shadow, texture);

Ia. Perceptual System

✔ **Sight, *reading***

✗ Capital letters, bizarre fonts slow down reading (can be desired sometimes), except for codes (e.g. flight codes)

✗ Known words recognised by shape, not by single letters.

Ia. Perceptual System

Hearing

Humans can:

- Identify many sounds at the same time;
- Estimate the distance of the source (by intensity, delay);
- Identify the sources;
- Filter out unnecessary sounds;
- Familiar sounds are recognised without diverting the attention.

Ia. Perceptual System

Touch

Temperature and shape affects how we grab and handle an object

- Keyboard must give a precise sensation when a key is pressed (problem of *touch screen* keyboards…);
- Awareness of the hands position on the keyboard is relevant for typing speed;
- Awareness of body and hands position, comfort.

Ib. Cognitive System

- ✔ **Memory**

- ✗ Short term memory;
- ✗ Long term memory;
- ✗ Sensory memory *(taken in by sensory receptors and processed by the nervous system, short transferred to STM).*

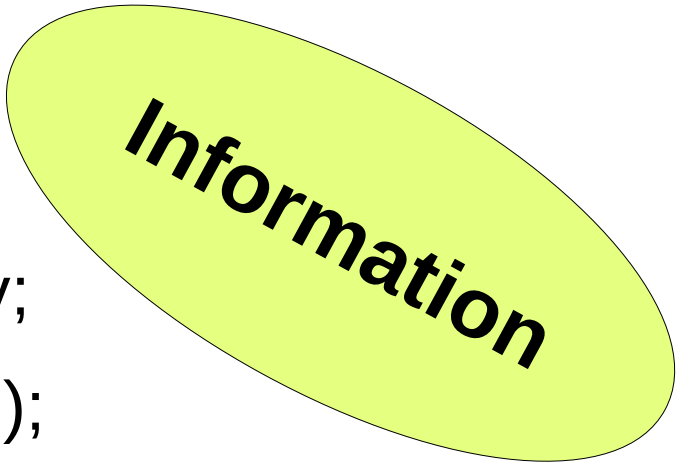
Design interfaces so that the user focuses only on relevant information

Ib. Cognitive System

Memory

Design interfaces so that the user focuses only on relevant information

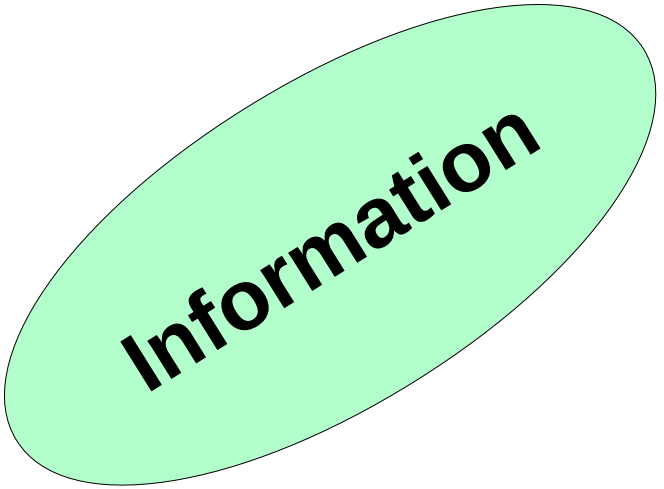
- Don’t put too much;
- Don’t put too little;
- Order and group semantically;
- Use alarms (graphical, sound);
- Dedicate the most important area of the screen to significant information.



Ib. Cognitive System

Memory, short term

- ✗ Quick access (~70ms);
- ✗ Quick decay;
- ✗ Low capacity (7 +/- 2 chunks)
- ✔ Example: these saturate STM: ***CER FUL SEC ETS SAU***



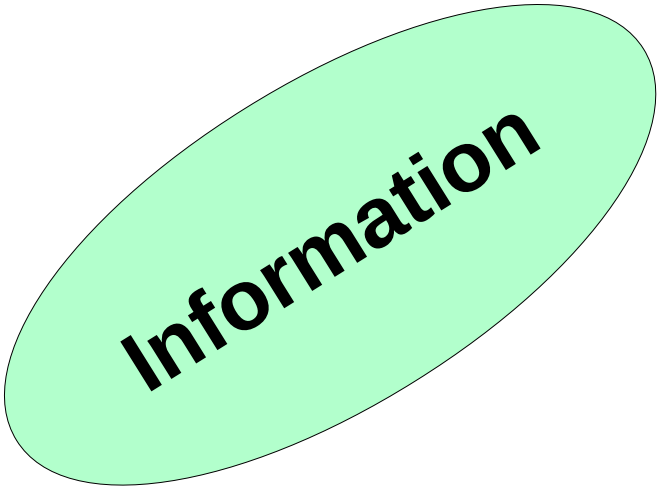
Saucerful ~~of~~ ***secrets***

Facilitate Chunking!!

Ib. Cognitive System

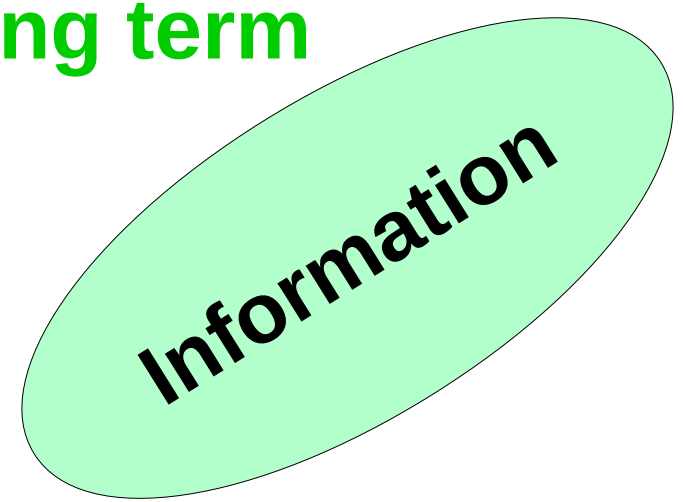
Memory, long term

- ✗ Slower access (~100ms);
- ✗ Very slow decay;
- ✔ Very high capacity



Ib. Cognitive System

Memory, long term



Fetch by

- Recall (must remember);
- Recognition (I identify a known information);

Use this!

- Meaningful;
- Familiar;
- Organised;
- Concrete

Ib. Cognitive System

Differences between individuals

- Expert/beginner;
- Culture;
- Intellectual skills;
- Age;
- Sex;
- Phisical, motor skills.

Avoid pushing the user to the limit of his possibilities

Ib. Cognitive System

Mental models

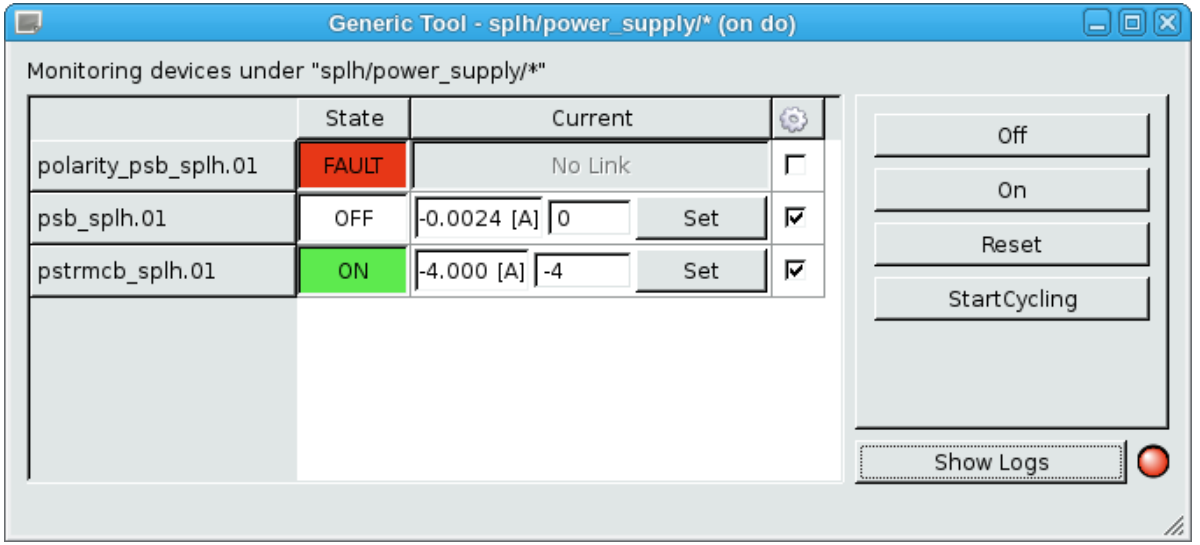
Explanation of someone's thought process about how something works in the real world

- ✗ Easy to learn;
- ✗ Unmet expectations;
- ✔ Integration (*composite* **metaphores**, like menus, windows, scroll bars)

Ib. Cognitive System

Mental models + sight + hearing

- Sound for alarms;
- Code colors[Q]: OK, Error/Alarm, Warning, as in the traffic light metaphor;



Ib. Cognitive System

Human errors

- Mistake (an incomplete or wrong mental model is applied)
- Slip (right model, but lack of attention, fatigue, stress);

...but user interfaces must not make it easy to make mistakes!

Ib. Cognitive System

Human errors

- Do not put buttons or actions performing opposite actions close to each other;
- Do not put close to each other common actions that in a previous version were well separate;
- Disable elements that must not be used in a certain *state*[Q] (**mode error**: I think I'm in a certain state, but it's not);
- Pop up warning and confirmation dialog windows before allowing a critical action on a system[Q].

Pt II

HCI and execution time

II. HCI and execution time

- Delays between *actions* and interface *feedback* produce unpredictable effects (*cursor tracking, icon wars*);
- On the other side, the interface must allow the user to read and understand messages, irrespective of the computational speed.

II. HCI and execution time

Factors:

- Calculation time;
- Memory access time;
- Graphics card speed;
- Network access;
- ...

II. HCI and execution time

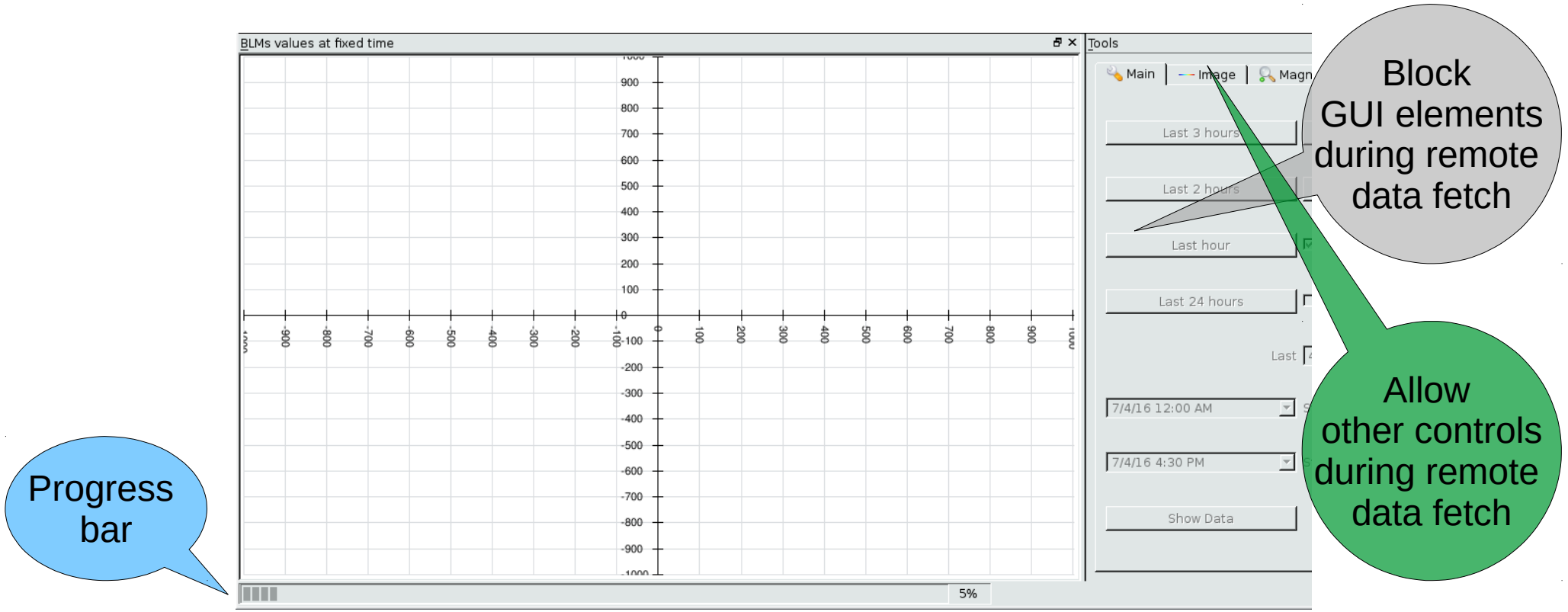
Guidelines:

- Inform the user about the operation taking time (progress bar)
- Put the interface in a *state* where actions depending on the slow operation result are blocked (*disable parts of the GUI*)
- Allow for actions that can be performed *while* a slow action is going on
- Cache on memory (or disk) information not supposed to change shortly instead of repeatedly fetch it from the network

II. HCI and execution time

Guidelines (II):

- Perform background operations in a *separate thread*, not in the graphical interface's one;



Pt III

Physical characteristics of the interaction

III. Physical characteristics of the interaction

- Logical and functional arrangement of the controls:

  1. by functionality,

  2, by logical or time sequence;

  3. by frequency of usage.

- User must keep on hand all the controls and devices;

- In modern interfaces, physical devices can be replaced by virtual controls (*glass interface*)

III. Physical characteristics of the interaction

- **Environment**: user's comfort, space to move, temperature, light, noise, utilization time, and so on...
- **Colors**: well defined, clear, don't use blue for critical information[Q].

1) Color is not a source of information, but a complement to it.

2) Avoid *color pollution*.

3) Useful to divide a display in regions with specific tasks

Pt IV

Interaction styles

IV. Interaction styles

- Command line (*powerful, flexible, based on recall*);
- Natural language (has limitations, restricted to specific areas);
- Menu driven (based on recognition, requires clear and organized options);
- Question/answer (limited in flexibility, specific to a task);
- Form filling (easy, good if supports correction and input validation);

IV. Interaction styles (II)

- **WIMP** interface (Windows, Icons, Menus and Pointers), windows must be managed: tiling, overlapping…);
- Point and Click, avoid keyboard and even mouse usage, if touch;
- 3D, realistic, immersive, effective if the paradigm reflects how we interact with the real world, in the real space.

IV. Interaction styles - WIMP

- Window (+ scroll bar, title bar, corners, layout)
- Pointer (cursors shape, actions: click, double click, move, drag, drop, right click…);
- Button;
- Slider;
- Toolbar (often useless because of small and obscure icons, they become useful with time if the user finds them also in menus)

IV. Interaction styles - WIMP

- Dialog box (modal ones compel the user to take care of them before going on with the main window);
- Menus (problems: grouping, reduce slip errors, consistency between applications, avoid putting close to each other actions with opposite effects, like *Save* and *Delete*)

IV. Interaction styles – Icons

Characteristics

- Depend on context (social, cultural, environmental, application);
- Function: label, state, warn, identify (files, directories), manipulate (zoom, shrink windows), contain (trash)…
- Based on **recognition**;
- Concrete objects or abstract symbols;
- Mapping between representation and concept (resemblance, exemplar, symbolic, arbitrary)

Pt V

Usability principles

V. Usability principles

- **Predictability**: determine future interaction according to past interaction[Q];
- **Honesty (“*observability*”)**: the internal state of the system is clearly represented by the interface[Q];
- **Familiarity**: how much former experience can help[Q];
- **Generalizability**: extend specific interaction patterns to akin situations[Q];
- **Consistency**: similar behavior in similar circumstances[Q].

Elettra Sincrotrone Trieste

TANGO

V. Usability principles (II)

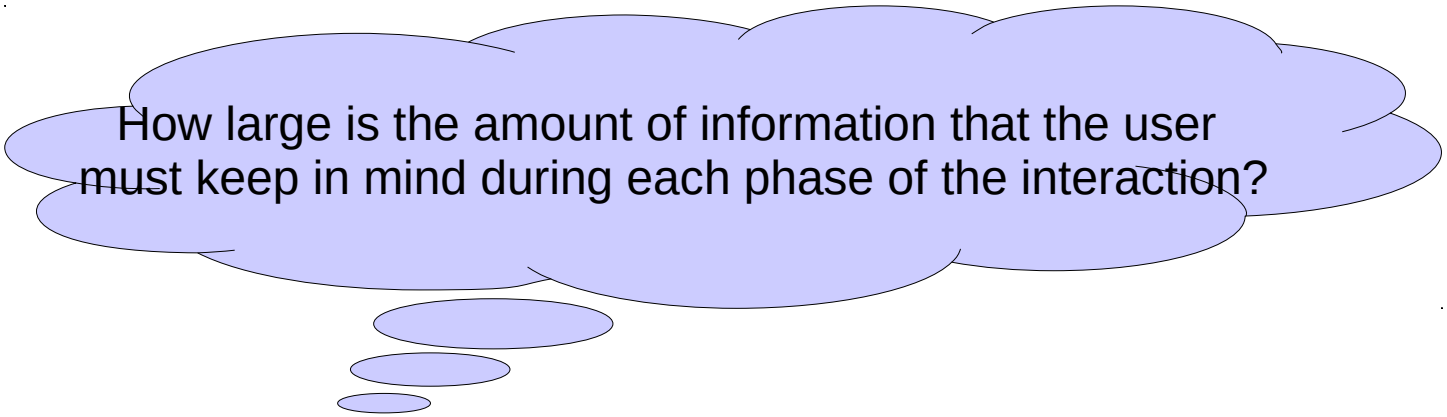
- **Dialog initiative**: who starts the action? The system or the user?
- **Multi threading**[Q]: concurrent tasks, GUI thread;
- "**Customizability"**: by the user (adaptability), "*adaptivity*": the system customizes itself becoming acquainted with the user;
- "**Recoverability"**: backward error recovery, move between states, avoid *dead* states;
- **Responsiveness**[Q]: the time perceived by the user to accomplish a task. Must be consistent across equivalent operations

Pt VI

HCI and requirements analysis

VI. HCI and requirements analysis

- **Task analysis**: determine the task characteristics
- **User analysis**: determine the kind and number of users of the system, together with their skills, culture, prior experiences;
- **Environment analysis**: where will the system be installed? Which kind of support will the environment provide to the user?



VI. HCI and requirements analysis

- Interviews, opinions;
- Existing documentation;
- Observation of the user, simply spending some time with him or actively ask him questions.
- *Predictive* evaluation: anticipate possible problems without directly testing;
- Interpretative evaluation: observe the user's environment and infer a typical behavior, a possible interaction.

Pt VII

User support

VII. User support

- **Help**, provides support for a *specific* problem;
- **Documentation**, system oriented assistance.

Common questions:

- Purpose: what can I do with the system?
- Definition: what is this? What is this for?
- Task execution: how can I do this?
- Diagnostics: how could this happen?
- State identification: where am I now?

VII. User support

Possible requirements (ideal world)

- **Availability**, always, without need to abandon what the user is doing;
- **Accuracy, completeness**;
- **Consistency**, across styles, terminology, presentation, …
- **Robustness**: the user is already in trouble…
- **Intrusiveness**: don’t hamper the user activity.

VII. User support

Approaches (I)

- **Command assistance**: upon user request on a specific command, *e.g.* Unix *man* pages;
- **Command prompts**: help in response to a wrong command;
- **Online tutorial**, move by steps and examples;
- **Online documentation**:
  - with a quick reference
  - With a concise, essential version
  - hypertextual

VII. User support

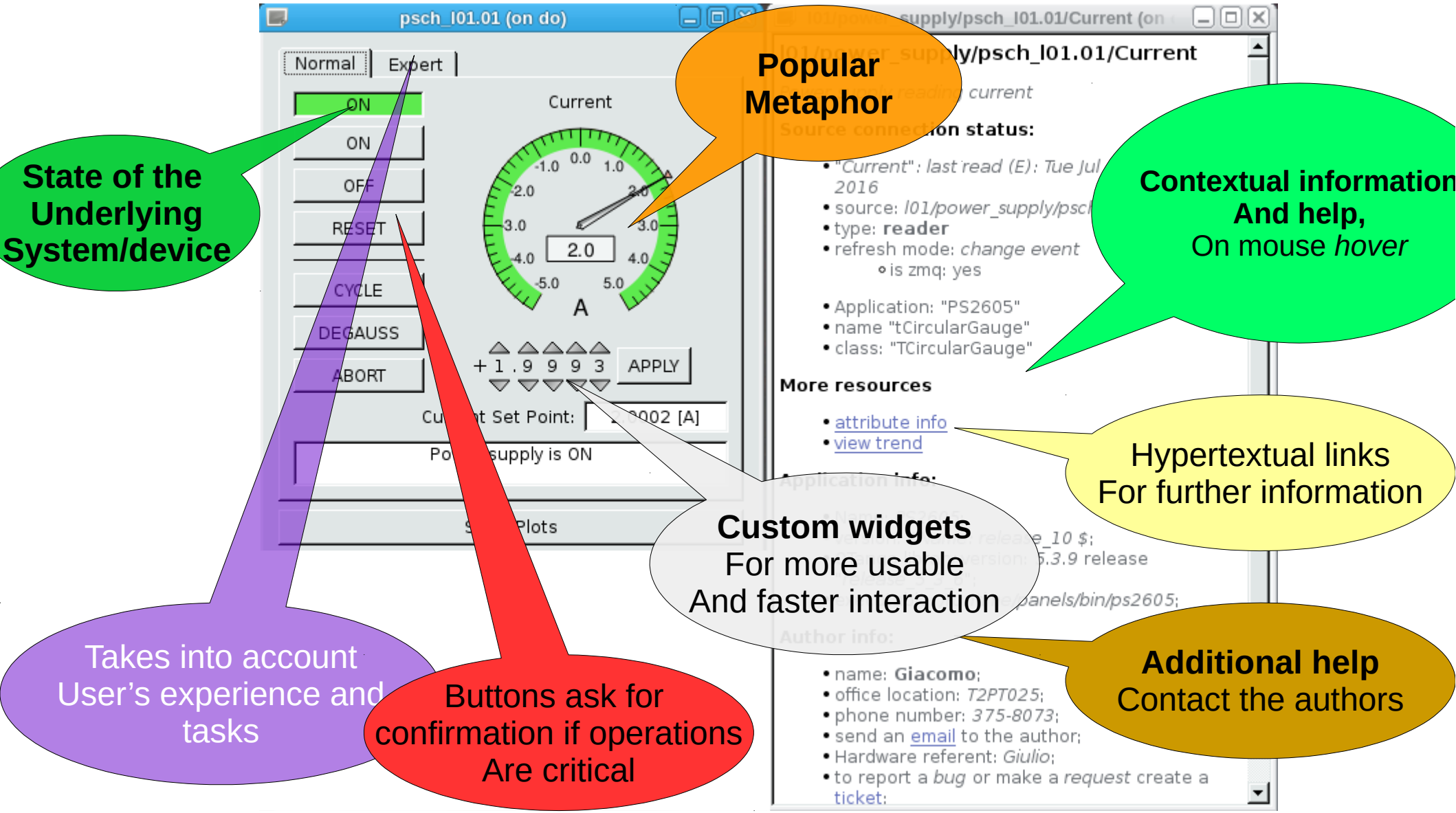
Approaches (II)

- **Context sensitive help**: state and object dependent help (**?**) button or contextual help on mouse *hover* event, *tooltips*;
- **Adaptive**: adapts to the user, suggests alternative ways to do a task, often monitoring the user's activity;
- **Adaptable**, the user configures the help system;

Pt VIII

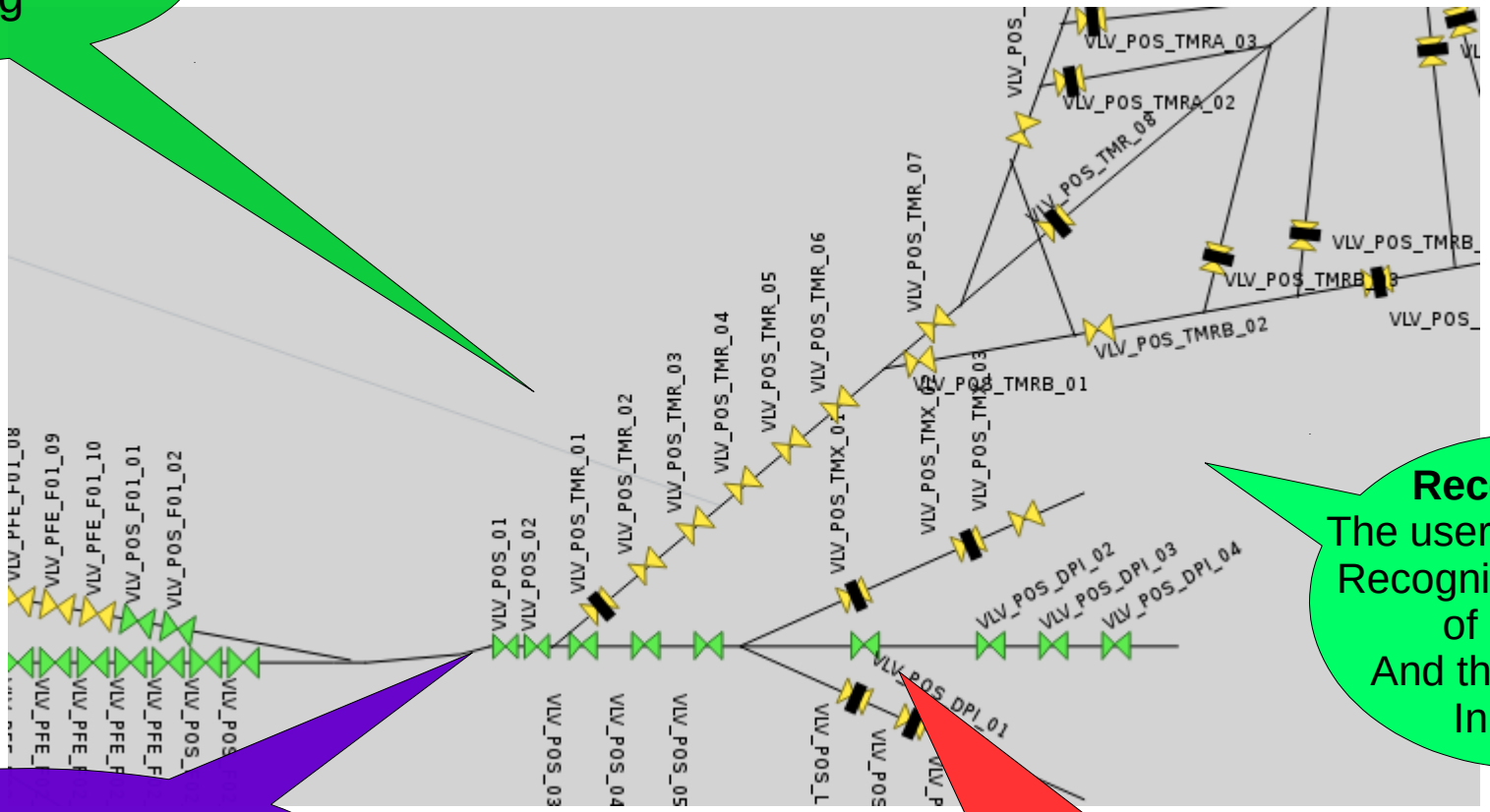
An example

VIII – An Example



VIII – An Example

Interaction models:
mapping



Recognition
The user immediately
Recognises the type
of objects
And their position
In space

Icons
Immediately reveal the
Kind of object (recognition)

Observability
User immediately recognize
The state of the system from
A visual representation

VIII – An Example

Elettra Sincrotrone Trieste

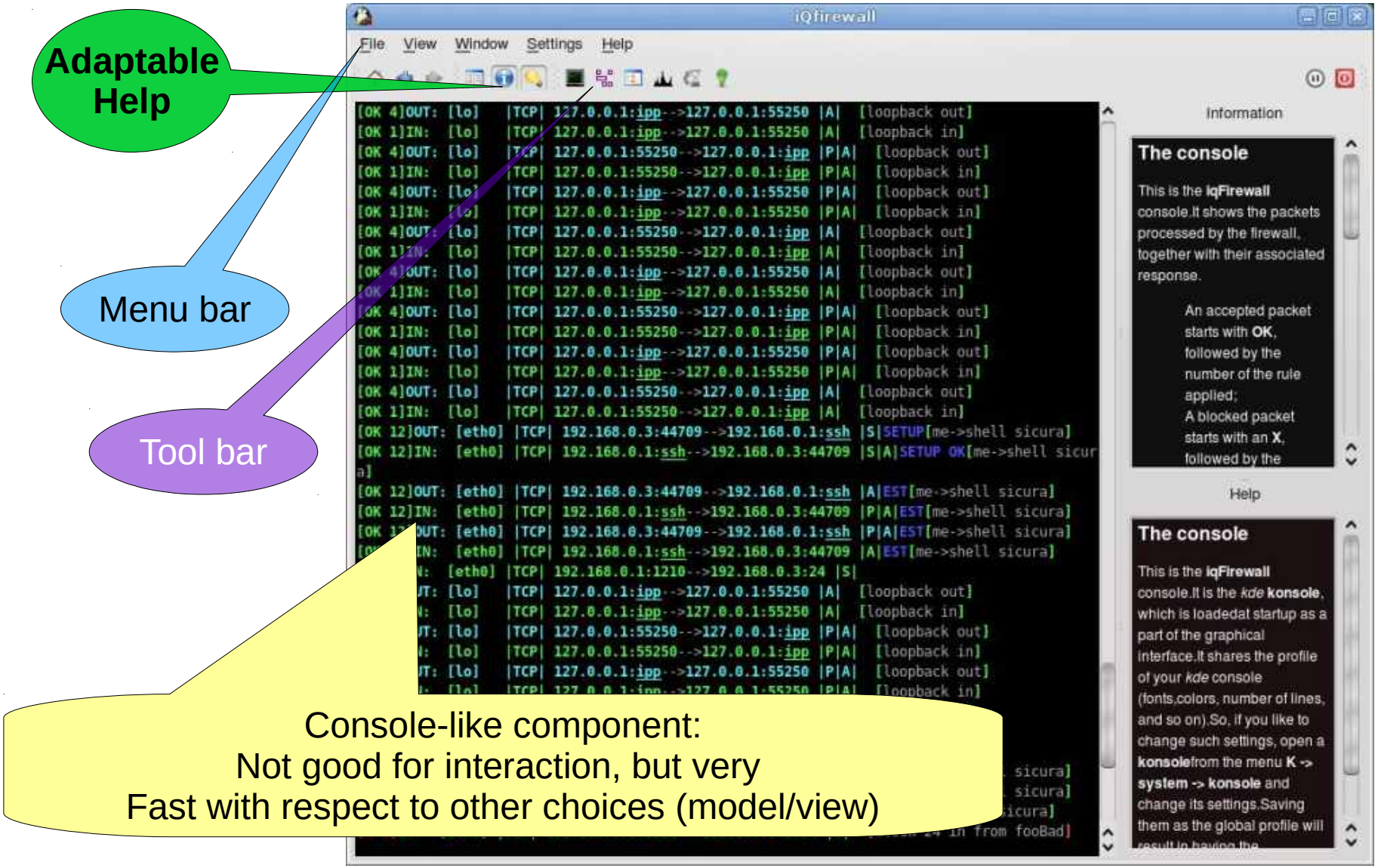
TANGO

Usability principles: Consistency

State* and *status
In the same position across different applications

Color
Colors integrate the information, without Being the source of it

VIII – An Example



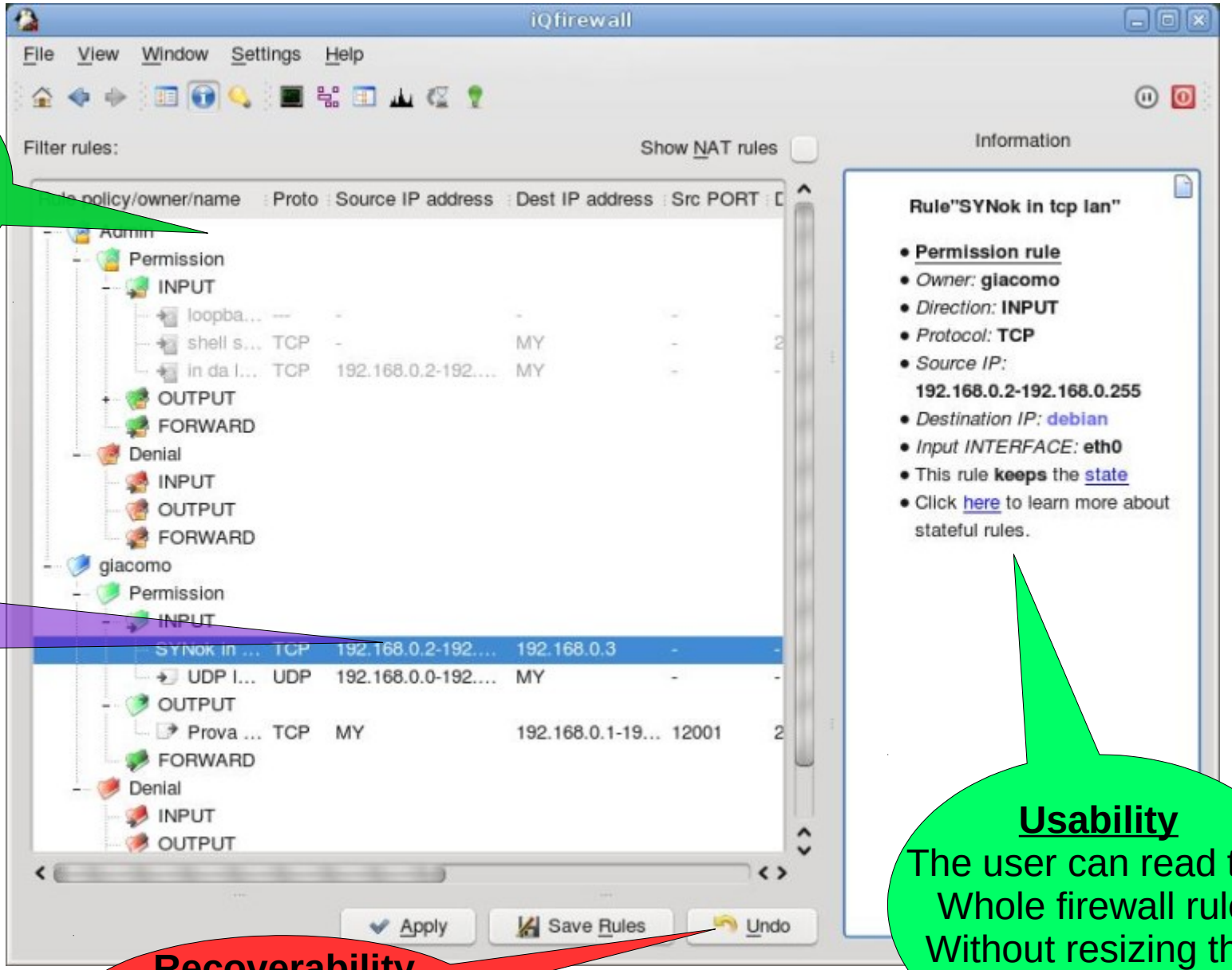
VIII – An Example

Elettra Sincrotrone Trieste

TANGO

Tree view, Summary, interaction

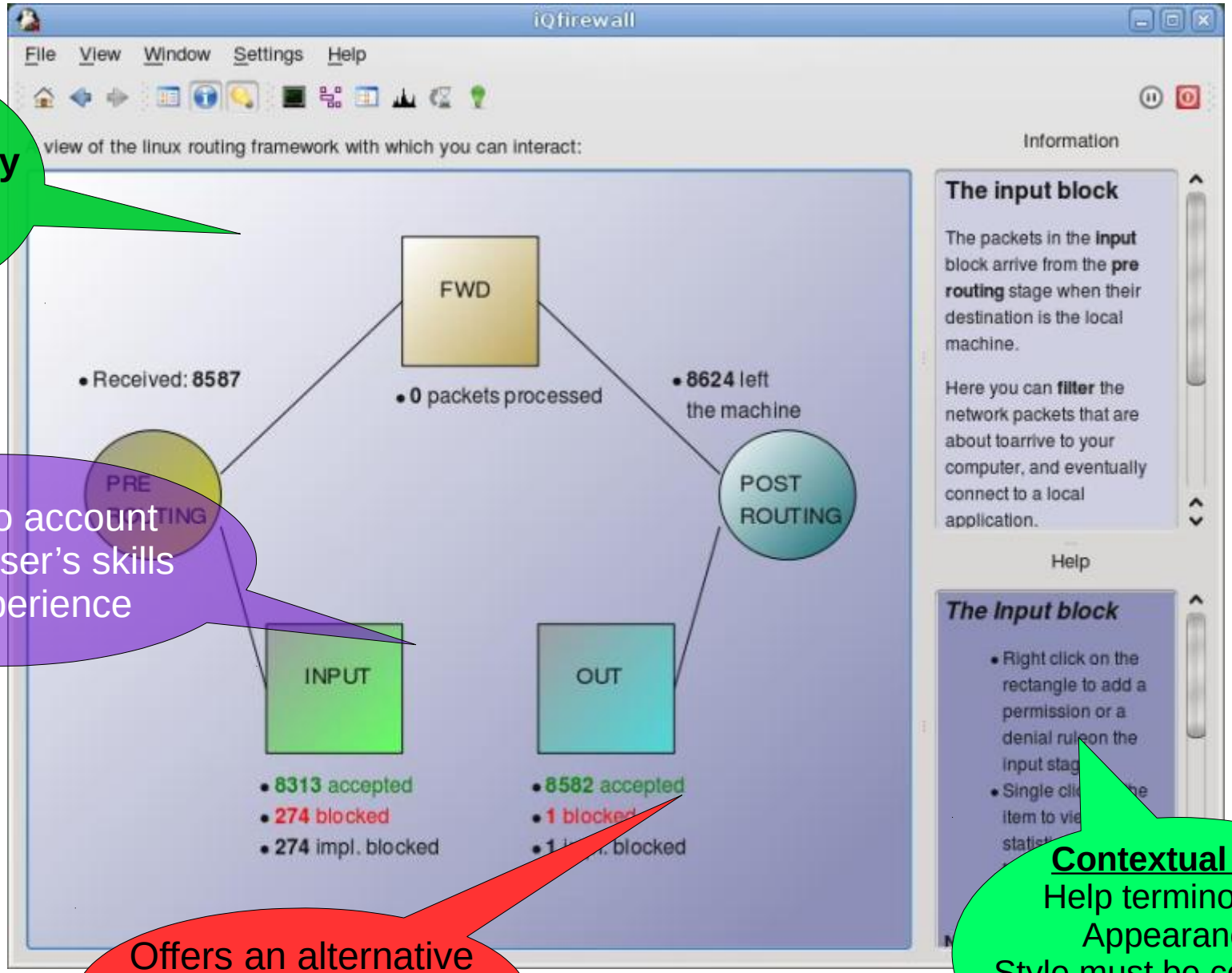
On mouse *Hover* event The information changes



Recoverability
Undo Pattern

Usability
The user can read the Whole firewall rule Without resizing the Tree columns

VIII – An Example



Customizability
Adaptability

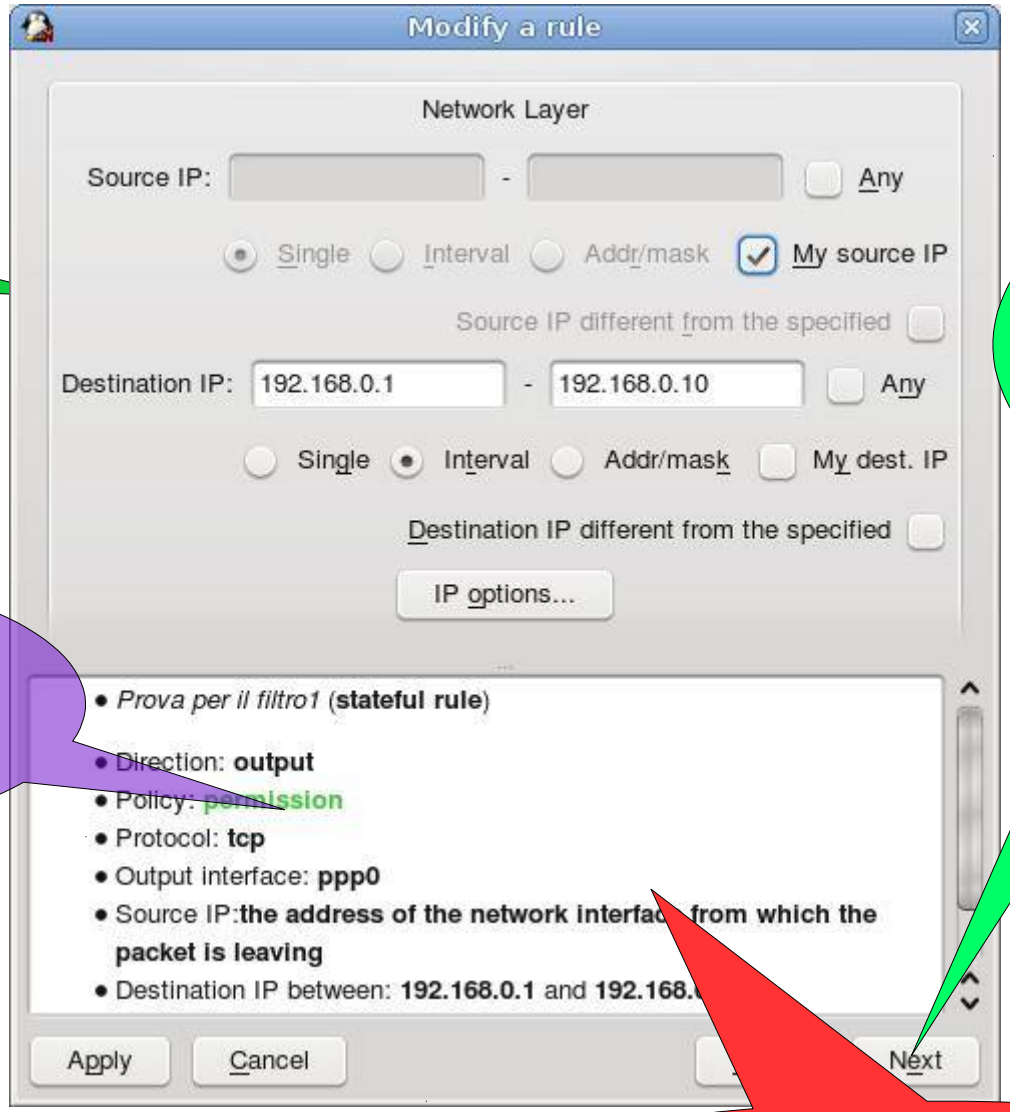
Takes into account
Different user's skills
And experience

Offers an alternative
Path towards the
Same target

Contextual help
Help terminology,
Appearance,
Style must be consistent
Across the application
elements

VIII – An Example

Customizability
Adaptability



Takes into account favorite user's Interaction style

Step by step approach
Offers a different way
To configuring a rule,

Summary of the rule,
Gives an overview of all the fields, also
Those not visible in the current step

Design patterns

Bibliography

- Prof. Luca Chittaro, University of Udine, Human Computer Interaction
- http://www.hcibook.com/hcibook/search/dosearch.php?query=interfaces&start=20

The End

- **Thanks for your attention**

mailto: giacomo.strangolino@elettra.trieste.it