

User interface

An interface is what relates two elements or systems, it is the environment inserted between them.

In general, when we talk just about the **interface**, we mean the interface between the computer system and the user, and it is simply the communication mechanism between both elements. It is very important in interactive systems. An interface must guarantee:

- **Functionality:** it must be the suitable one (one of the usual vices is overloading it and putting more functionality than necessary in it).
- **Reliability:** it is demanded at 100%.
- **Habitability or ergonomics:** easy learning and retention, and satisfaction of the user at 100%.

Interface is separated from the system functionality itself (independence) because this favours the partition of the problem (3-tier architecture) and makes maintenance, reusing and the use of specialized tools (interface generators) easier.

The first **element** to take into account is the human users, it is necessary to know very well the tasks each one of them must carry out with the system. It is necessary to:

Human user	<ul style="list-style-type: none"> - Model him/her (type) - Adapt to his/her knowledge, goals and attitudes. - Match his/her goals. 	and also	<ul style="list-style-type: none"> - Usual - Occasional 	-	<ul style="list-style-type: none"> - Beginner - Average - Advanced
------------	--	----------	---	---	---

The syntactic/semantic model must be taken into account:

- **Syntactic knowledge:** it defines *how it is used* and it is easily forgotten.
- **Semantic knowledge:** it defines *what it does* and it is more stable (it is remembered).

The interaction can also be classified according to its typology:

Type of interaction	<ul style="list-style-type: none"> - Dialogue - Direct manipulation (on a simulation of the result) 	-	<ul style="list-style-type: none"> - Menus - Forms - Commands 	-	<ul style="list-style-type: none"> - Formalized - Natural language
---------------------	---	---	--	---	--

(A formal command language should be easy, consistent, structured and complete.)

Finally, dialogue can be:

- Sequential (menus, forms, etc.)
- Asynchronous (events, commands)

According to who takes the initiative:

- The system: with menus and messages.
- The user: with commands, events, non-sequential forms.

Beginners and occasional users will prefer a system that guides them a lot with a sequential dialogue (by means of menus, messages and sequential forms). On the other hand, the usual and advanced (or expert) users will prefer much more freedom

to move and be able to quickly access any option of the system by means of asynchronous dialogue.

The **GUI** (*graphical user interfaces*) of the current systems work with event orientation. Any action of the user triggers an event.

They include elements of all types (menus, forms, commands through shortcuts or intelligent icons, and direct manipulation). This way it is easier that only one interface can adapt to all the expertise levels of the users.

They also permit to disable options, either hiding them or not, so that only one interface can be utilized by all the users of the system, and besides it can be easily adapted to any state of the execution.

In all cases it is necessary:

- Consistency (in actions and terminology) || Coherence
- Simple management and recovery of errors
- Possibility to undo
- Not requiring the user's short term memory
- Providing shortcuts for the expert or advanced user

Three final recommendations:

1. It is advisable that the interface can adapt to each and every type of user.
2. Do not ask for more information than absolutely essential. Reduce repetitive operations and ask for confirmation only of those dangerous, complex to undo or irreversible operations, making the usual behaviours of the user easier.
3. The basics are that the interface works fine (suitable functionality and reliability), start to think about making it nicer only afterwards.