# Interaction Models

Human Computer Interaction involves interaction between two complex systems - the human and the computer - each of which has a different view of the domain, task, etc..

The interface should translate between the two, allowing them to communicate efficiently.

This communication often fails.

Interaction models can help us understand why communication failures occur, and identify the root causes.

They also enable us to investigate and compare different systems and interaction styles, and to consider basic interaction problems.

## The Execution-Evaluation Model

One of the most influential interaction models is the Execution-Evaluation Model, proposed by Donald Norman in 1988.

According to this model, the purpose of an interactive system is to aid the user in accomplishing *goals* within a specific *domain*.

The following terminology is used:

* **Domain** - an area of expertise and knowledge in some real-world activity
* **Tasks** - operations that manipulate concepts related to the domain
* **Goals** - the desired output from a task.
* **Intention** - a specific action required to meet a goal.

The system and the user each use their own language:

* The *system* uses the *core language*

The core language describes computational aspects of the domain relevant to the *system state*.

* The *user* uses the *task language*

The task language describes psychological aspects of the domain relevant to the *user state*.

The model assumes that the *interactive cycle* can be broken down into two stages:

* The user has a plan which is carried out (*execution*)
* The user observes the results in order to formulate a plan the next stage (*evaluation*).

These two stages - execution and evaluation - can be further sub-divided as follows:

* establishing the goal
* forming the intention
* specifying the action sequence
* executing the action
* perceiving the system state
* interpreting the system state
* evaluating the system state with respect to the goals and intentions

Note that each of these stages is viewed from the point-of-view of the user.

Norman uses this model to explain problems in interfaces.

* A *gulf of execution*.

This occurs when the user's formulation of the tasks required to achieve a specific goal (formulated in the task language) conflict with the tasks allowed by the system.

* A *gulf of evaluation*.

occurs when the user attempts to interpret the system state following an action (using the task language), but cannot fully translate the output presented by the system (using the core language)

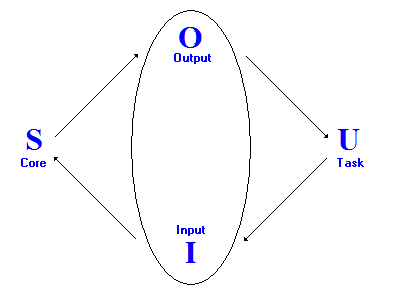
## The Interaction Framework

Abowd and Beale (1991) proposed an extension to Norman's model which they called the Interaction Framework

This model explicitly includes the system as a component.

The interaction is divided into four major components:

* the user (U)
* the system (S)
* the input (I)
* the output (O)



The Input and the Output together form the Interface, which sits between the User and the System.

Each component uses its own language.

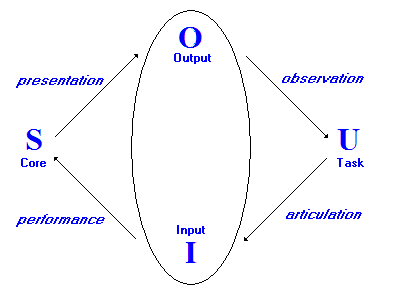
As in Norman's model, the system uses the core language and the user uses the task language.

In addition, the input uses an *input language* and the output uses an *output language*.

These languages - especially the core, input and output languages - may overlap or be nearly identical.

However, the ability to identify different languages for each component makes this model very flexible.

There are four stages to the interactive cycle, represented by the arrows in the diagram:



* **Presentation** - the system changes state, and presents this changed state via the output.
* **Observation** - the user assesses the results by observing the output.
* **Performance** - the task is translated into operations to be performed by the system.
* **Articulation** - the user formulates a goal, and a task to achieve that goal.

Note that each phase involves translation:

* From the Output language to the Task language
* From the Task language to the Input language
* From the Input language to the Core language
* From the Core language into the Output language.

The first two stages - Articulation and Performance - correspond to the Execution phase of Norman's model.

The latter two stages - Presentation and Observation - correspond to the Evaluation phase of Norman's model.

The model can also be used - in conjunction with suitable metrics - to determine the amount of translation required in each phase.