

---

# **The AMODEUS Project**

## **ESPRIT Basic Research Action 7040**

---

### **The System Modelling Glossary**

**Salber D., Coutaz J. & Nigay L. (editors)**  
**Laboratoire de Génie Informatique, University of Grenoble**  
**Faconti G. & Paterno F.**  
**CNUCE**  
**Duke D. & Harrison M.**  
**University of York**

**28 June 1995**

**Amodeus Project Document: System Modelling/WP 26**

---

#### **AMODEUS Partners:**

MRC Applied Psychology Unit, Cambridge, UK (APU)  
Depts of Computer Science & Psychology, University of York, UK. (YORK)  
Laboratoire de Génie Informatique, University of Grenoble, France.(LGI)  
Department of Psychology, University of Copenhagen, Denmark. (CUP)  
Dept. of Computer & Information Science Linköping University, S. (IDA)  
Dept. of Mathematics, University of the Aegean Greece (UoA)  
Centre for Cognitive Informatics, Roskilde, Denmark (CCI)  
Rank Xerox EuroPARC, Cambridge, UK.(RXEP)  
CNR CNUCE, Pisa Italy (CNR,CNUCE)

---

## **Abstract**

The System Modelling Glossary gathers definitions for the vocabulary that is commonly used within the RP1 System Modelling Package. Its purpose is to serve as a support/companion for reading RP1 papers. It is also aimed at providing the Amodeus project with a reference vocabulary for discussing system modelling issues. An hypertext version is available on the World-Wide Web at the following address: <http://www-lgi.imag.fr/Les.Groupes/IHM/AMODEUSGlossary/CoverPage.html>

## **Instructions**

Entries in the glossary may be words or expressions (e.g., “level of abstraction”). They are listed in alphabetical order. Some of the definitions may draw upon or replicate widely-accepted meanings. In this case, the source reference is denoted as [ref] and is listed in the “References” section at the beginning of the document.

## **Updates**

As the vocabulary used within RP1 evolves, some entries and definitions may become obsolete. Such entries are marked {Obsolete} and may be amended with a newer definition. Every member of the Amodeus project is welcome to suggest revisions to the RP1 glossary.

## **Flags**

The three following entries are marked with an asterisk [\*]: Affordance, Real-world validity and Task domain concept. The asterisk indicates that we strongly feel the need for other packages' input and comments on these definitions.

## **References**

- [CCS] A.J.R.G.Milner, *A Calculus of Communicating Systems*, Lecture Notes in Computer Science, Vol. 92, Springer Verlag, 1980.
- [CGRM] Information technology - Computer graphics and image processing-*Computer Graphics Reference Model*, ISO 11072, International Organisation for Standardisation, 1992.
- [Dictionary] *The American Heritage Dictionary of the English Language*, William Morris, editor, Houghton Mifflin Company, 1969.
- [LOTOS] T. Bolognesi, H. Brinska, *Introduction to the ISO Language LOTOS*, Computer Networks and ISDN Systems 14, pp 25-59.

**Abstraction** [CGRM]

Processing element within a layer that transforms an information type. The transformed information is transferred to the next higher layer in a hierarchy or to another subsystem in the system when that layer is the highest one in that hierarchy.

Abstraction

- (a) Process that transforms information into information whose semantic content and scope are richer/higher than the content and scope of the initial information.
- (b) Result of the process of abstracting.

Abstraction function

Concept to denote the capacity of abstracting within a system.  
See Rendering function.

Action [CCS or LOTOS]

Fact happening within a system identified by a name and by an information. The information can be either explicit (an instance of an information type is communicated) or implicitly associated to the event name (an information may be inferred from the event).  
Synonym for Event [CCS or LOTOS].

Action

- (a) Transformation between two states of a system.
- (b) Specification structure that defines a state transition by means of such a predicate.  
eans of such a predicate.  
a predicate.
- (c) Indivisible operation performed by an agent whose effect can be perceived by another agent (in terms, for example, of events received through communication channels).  
See Physical action, User physical action, System physical action, System conceptual action as subclasses of actions.

**Action-tree** [CCS or LOTOS]

Tree-like representation of a behaviour expression.

**Affordance** [\*]

Property that the presentation of a system conveys information about the actions that can be performed by the user of that system. These properties relate to the "real world" understandings that the user has.

**Agent**

Object capable of initiating the performance of actions.

**Alphabet**

Union set of event names of a behaviour. Set of events in which some agent can engage.

**Autonomous System**

System capable of autonomous behaviour.

**Backward recoverability**

Property that the system provides the user with an undo facility to return to a previous state.

See Recovery, Recoverability.

**Behaviour**

- (a) Set of all possible traces of a process. [CCS or LOTOS]
- (b) An ordered set of states. See Trace.

**Behaviour expression**

Description of a behaviour within some formalism (e.g., action-tree, temporal logic, or process algebra).

**Bias**

Structure in a model that is not essential for characterising the behaviour or properties of a system.

**Browsability**

Property that the system provides the user with commands to make perceivable different portions of the system functional state. (By modifying the presentation state, the user may access different portions of the state of the functional core.)

**Channel**

- (a) Common path shared by two or more components of a system.
  - (b) System structure/component used for the communication of information.
- See Physical channel, Digital channel, Human channel.

**Collection**

Contains higher level descriptions of the output primitives associated with the interactor which will be interpreted for producing symbols and elements perceivable by the user.

**Command**

- (1) Set of physical actions transformed by the system into a system conceptual action.
  - (2) System conceptual action.
- See Inspection command, Task domain command.

**Communication [Dictionary]**

Act of transmitting information.

**Computation**

Ordered set of operations that transforms the state of a system.

**Component**

Part of a whole. Depending on the context, can be instantiated as a software module, a subsystem, an agent, an interactor, an abstraction, a device, etc.

**Conceptual action**

Action performed by the system at the highest level of abstraction, i.e., a primitive function in the functional core.  
See Level of abstraction.

**Conceptual unit**

Part of the state of the functional core that models a task domain concept at the highest level of abstraction.

**Conformance**

Property that the presentation of a system mirrors the underlying behaviour of the system.  
See Layer conformance, Observability, Honesty.

**Connectedness**

Property of presentation that requires that notions of continuity in the state are faithfully rendered in the presentation.

**Context**

Set of state vectors used by a system in a computation. Examples of state vectors: state vector about the user, state vector about the presentation, state vector about the dialogue, state vector about the functional core, state vector about the environment.

See Dynamic context, Static context.

### **Continuity**

The property that given a function  $f:A \rightarrow B$  between ordered sets  $A$  and  $B$ , the neighbourhood around any point in set  $B$  corresponds to some neighbourhood around the inverse image of that point under  $f$ . This is a topological definition that generalises the usual notion of continuity found for example in real analysis, and is useful in that it also applies in the case of discrete sets.

### **Deontic logic**

A logic for reasoning about permissible or obligatory actions.

### **Detour**

Interaction that is not directly involved in achieving a goal.

### **Device**

- (1) Physical artefact necessary to a system to acquire (input device) or deliver (output device) information. Examples include keyboard, loudspeaker, ears and mouth.
- (2) Lowest level component of a system whose state changes result from physical actions.

### **Device assignment**

Relation between a device over a state and a non empty subset of expressions of an interaction language. A device  $d$  is assigned in state  $s$  to a set  $E$  of expressions of a language  $l$ , if it does not exist any device equivalent to  $d$  over  $s$  and  $E$ . Assignment is permanent if the relation holds for any state. Assignment is total if the relation holds for  $E$  equals to the set of expressions that define  $l$ . For example, in Matis, the mouse is permanently assigned to the expression of window resizing in the direct manipulation interaction language.

### **Device equivalence**

Relation between a non empty set of devices over a state and a non empty set of expressions in an interaction language. Devices in a set  $D$  are equivalent over a state  $s$  and a non empty set  $E$  of expressions in an interaction language  $L$ , if all of the expressions of  $E$  can be elaborated using either one of the devices in  $D$ . Equivalence is permanent if the relation holds for any state. Equivalence is total if the relation holds for  $E$  equals to the set of expressions that define  $L$ . For example, in Matis, keyboard and microphone are totally and permanently equivalent over natural language.

### **Device redundancy**

Relation between a set of devices over a state and an expression of an interaction language. Devices of a set  $D$  are used redundantly in some state  $s$  for an expression  $e$  of a language  $l$ , if these devices are equivalent over  $s$  and  $e$ , and if they are used simultaneously to express  $e$ . For example, the user can spell a character using the microphone and type in the same character.

### **Device complementarity**

Relation between a set of devices over a state and a non empty subset of expressions of an interaction language. Devices of a set  $D$  are complementary over a state  $s$  and a non empty set  $E$  of expressions of a language  $l$ , if  $E$  can be partitioned such that for each partition  $E_p$  of  $E$ , it exists a device  $d$  of  $D$  assigned over  $s$  and  $E_p$ . Complementarity is permanent if the relation holds for any state. Complementarity is total if the relation holds for  $E$  equals to the set of expressions of  $l$ . Language complementarity is best illustrated by spoken natural languages where concept names must be typed in. For example, in MUNIX, a multimodal user interface for Unix, commands that involve a file name such as remove, can be expressed using the microphone for the command name and options while file names must be elaborated with the keyboard.

**Dialogue**

- (1) Same as interaction.
- (2) Restriction on the behaviour of the interactive system to the set of actions performed with the participation of both human and computer processes.

**Dialogue strategy**

Set of behaviour expressions enforcing constraints over a dialogue. Examples of dialogue strategies include: reactive strategy (the system supplies the missing parameters of a command = default values), cooperative strategy (the system engages in a subdialogue to propose the set of possible values for a missing parameter and the user disposes), directive strategy (the system engages in a subdialogue to ask for the value of a missing parameter), negotiated strategy (the system engages in a subdialogue to propose a value for the missing parameter and the user disposes), intentional strategy (the system engages in a subdialogue to check with the user that the value for the missing parameter is correct).

**Dynamic context**

Portion of the context that may vary over time, e.g., during a session. This variation can affect the values of the set of state variables, as well as the actual set of state variables (as in learning systems).

**Dynamic system**

See Autonomous system.

**Equivalence**

A relation that is reflexive, symmetric and transitive.  
See Language equivalence, Device equivalence.

**Event [CCS or LOTOS]**

see Action [CCS or LOTOS]. The two terms have the same meaning in the CCS or LOTOS theory.

**Event**

- (a) Data structure used to transfer information between components of a system.
- (b) The observation that some action has occurred.

**Event class**

Name of an event.

**Event instance**

An instance of an event of a particular class.

**Event occurrence**

Same as Event instance.

**Fan-in [CGRM]**

Gathering of information units from multiple source processes to a single process.

**Fan-out [CGRM]**

Distribution of information units from one source process to multiple processes.

**Feedback**

See Response.

**Fission**

- (a) Computation of a process abstracting/presenting an information type into a collection of different information types to be transferred to a set of processes.

(b) Decomposition of an information type at some level of abstraction into multiple information types of the same level of abstraction.

See Fan-out, Fusion.

**Formal Method**

Collection of mathematical structures, together with a precise syntax for defining instances of those structures and organising them into domain-specific abstractions. It should be associated with a method for eliciting these abstractions and transforming them.

**Frame**

A collection of variables that are referenced or modified by an action.

**Functional core**

Component of a system that implements task domain concepts.

**Functional state**

State vector of the functional core.

**Fusion**

(a) Computation of a process abstracting/concretizing a collection of information types received from distinct processes into a different information type to be transferred to another process.

(b) Composition of multiple information types at some level of abstraction into a single information type of the same level of abstraction.

See Fan-in, Fission.

**Goal**

Desired property of a system; usually expressed as a state or set of states that a user intends to achieve. Particularly for reactive or control systems, goals may also be expressed as predicates over the behaviour or trace of the system.

**Guard**

See Precondition.

**Homomorphism**

In general, structure preserving transformation where 'structure' can be interpreted within many fields of mathematics. For example, a homomorphism between ordered sets  $\langle S, R \rangle$  and  $\langle Q, T \rangle$  is a function  $f$  from  $S$  to  $Q$  such that for all  $x, y$  in  $S$ , if  $(x, y)$  in  $R$  then  $(f(x), f(y))$  is in  $T$ .

**Honesty**

Property that the presentation of the system renders its functional state appropriately (e.g., does not distort the functional state). Honesty is a necessary but not always sufficient condition for the user to be able to elaborate a correct mental representation of the functional state.

See Connectedness as a special case.

**Human channel**

Physical channel used by a human being to acquire (sensori channel) or deliver (motor channel) information.

**Initiative**

Denotes the agent that leads an interaction. In many cases the interactional initiative is clear. For example, in the case of a system like UNIX the initiative is with the user, whereas in many expert systems the system takes the initiative (the user simply has to respond to questions supplied by the expert system).

**Input physical channel**

A physical channel involved in the acquisition of information.

**Interaction**

Non-empty ordered set of events involving more than one agent.

**Interaction dissonance**

Occurs when an agent interacts incorrectly in the sense that an appropriate communication is directed to an inappropriate agent. This notion is intended to express the situations where two agents interact at cross purposes. A typical example of this occurs when a user interacts in a way that is appropriate through one window of a window based system with the wrong window selected.

**Interaction language**

Language used by the user or the system to exchange information. A language defines the set of all possible well-formed expressions, i.e., the conventional assembly of symbols, that convey meaning. The generation of a symbol or a set of symbols, results from a physical action. It is modelled as an event produced via some physical channel carrying a message which is this symbol or set of symbols.

**Interaction object**

An agent that participates in an abstraction and/or presentation transformation. It may interact directly with a human agent.

See also: Interactor.

**Interaction trajectory**

See Interaction.

**Interactional invariance**

Measure of the vulnerability of an interaction to the actions of other agents that are not taking part directly in the interaction. The issue is the way that interactive system goals are influenced by autonomous behaviour as would be a concern if we were talking of autonomous systems. An interactive system should be capable of supporting certain objectives regardless of the other activities that are going on autonomously within the system.

**Interactionally-rich**

(a) Qualifies interactive systems that support multiple input interaction languages and/or multiple output interaction languages and these languages can be conveyed via multiple input devices and/or multiple output devices.

(b) is used to refer to interactive systems which engage user or users in a "natural" interactive environment. Users can employ the multiple interaction languages or modalities to have a more natural interaction with the objects of the work domain.

In this sense, therefore, a hammer which uses few modalities and interaction languages is interactionally rich because of the texture and directness of the interaction.

**Interactive system**

System for which a subsystem is explicitly defined to be a human.

**Interactor**

See Interaction object.

**Language assignment**

Relation between an interaction language over a state and a non empty subset of conceptual units of a system. An interaction language  $l$  is assigned in state  $s$  to a set of conceptual units  $C$ , if it does not exist any interaction language equivalent to  $l$  over  $s$  and  $c$ . Assignment is permanent if the relation holds for any state. Assignment is total if the relation holds for  $C$  equals to the set of conceptual units of the system.



**Language complementarity**

Relation between a set of interaction languages over a state and a non empty subset of conceptual units. Interaction languages of a set  $L$  are complementary over a state  $s$  and a non empty set  $C$  of conceptual units of the system, if  $C$  can be partitioned such that for each partition  $C_p$  of  $C$ , it exists a language  $l$  of  $L$  assigned over  $s$  and  $C_p$ . Complementarity is permanent if the relation holds for any state. Complementarity is total if the relation holds for  $C$  equals to the set of conceptual units of the system. Language complementarity is best illustrated by coreferential expressions. For example, in Matis, natural language and direct manipulation are complementary over the conceptual unit “city” and any state where the specification of a city name is possible : “flights from this city” and selection of a city name through direct manipulation.

**Language equivalence**

Relation between a set of interaction languages over a state and a non empty subset of conceptual units of a system. Interaction languages of a set  $L$  are equivalent over a state  $s$  and a non empty set  $C$  of conceptual units of the system, if all of the conceptual units in  $C$  can be represented using either one of the language in  $L$ . Equivalence is permanent if the relation holds for any state. Equivalence is total if the relation holds for  $C$  equals to the set of conceptual units of the system. For example, in Matis, direct manipulation language and natural language are permanently equivalent for specifying requests.

**Language redundancy**

Relation between a set of interaction languages over a state and a conceptual unit of a system. Interaction languages of a set are used redundantly in some state  $s$  for a conceptual unit  $c$ , if these languages are equivalent over  $s$  and  $c$ , and if they are used simultaneously to represent  $c$ . For example, a wall is represented redundantly by the system via a red line (graphics interaction language) and the message “mind the red wall!” (natural language).

**Layer conformance**

- (a) Relation between the abstraction/presentation transformations and the information types bound to two adjacent layers with respect to some invariants.
- (b) Generalisation of conformance between any pair of layers.

**Layer**

Hierarchical partition of a subsystem into elements for which specific processing and/or transferring entities as well as information types are precisely defined.

**Level of abstraction**

- (a) Layer within a system whose information types are characterized by a given semantic content and scope. The lowest level of abstraction corresponds to the poorest information type with regard to scope and content. The highest level of abstraction corresponds to the richest information type with regard to scope and content. These levels as well as any level in-between depend on the perspective or the objective of the modeller and/or the modelling technique.
- (b) different perspectives in the design process of a system.

**Measure**

Produces higher level input data by applying its function to the input data.

**Media**

Same as physical device type or set of physical devices types used for communication.

**Message**

Any value transmitted on a channel.

**Metaphor**

Representation of one system model in terms of another that has some real world significance or analogy, compare affordance. The classic example is the metaphor by which the file and directory structure of an operating system is understood in terms of a desktop with windows (folders) and icons.

See Homomorphism.

**Migratability**

Ability of the interactive system to dynamically transfer performance between agents.

**Modal**

Having possible modes or worlds; modal logic is a system for reasoning about possible and necessary truth.

**Modality**

Method characterized by the type of usage of interaction languages and physical devices for communicating information at the user interface.

**Mode**

Same as Context.

**Model**

Any representation of a real or imagined system.

**Multidevice**

Property of the system that provides the user with more than one physical device (simultaneously or not) to communicate information to the system and/or uses more than one physical device (simultaneously or not) to communicate information to the user.

See Multimedia.

**Multilanguage**

Property of the system that provides the user with more than one interaction language (simultaneously or not) to communicate information to the system and/or uses more than one interaction language (simultaneously or not) to communicate information to the user.

**Multimedia**

(a) Use of more than one medium for communication.

(b) Same as multidevice.

**Multimediality**

Property of a system that supports multimedia interaction.

**Multimodal**

Which provides the user with more than one modality (simultaneously or not) to communicate information to the system and/or uses more than one modality (simultaneously or not) to communicate information to the user.

**Multimodality**

Property of a system that supports multimodal interaction.

**Objective**

In general we shall be interested in the extent to which a system supports a set of objectives. An objective is considered to be a pre- and post-condition on state which may be linked with the agent who owns the objective.

**Observability**

Property that the presentation of a system contains sufficient information to allow the user to determine the functional state of the system.

See Conformance, Browsability, Honesty.

**Ordered Set**

A set  $S$  and relation  $R$  over  $S$  such that  $R$  is reflexive, transitive, and maybe antisymmetric .

**Output physical channel**

Physical channel that delivers information.

**Parallelism**

Simultaneity in the activity of multiple agents. Two events performed by different agents engaging in parallel activities may occur in overlapping time intervals.

**Passive command**

Command whose effect does not modify the state of the functional core of a system.

**Performance**

Interaction trajectory that achieves an objective.

**Physical action**

Action performed either by the user or by the system on a physical device.

**Physical channel**

Channel involving a physical device.

See Input physical channel, Output physical channel, Channel, Human channel.

**Poset**

Ordered set  $\langle S,R \rangle$  where the relation  $R$  is a antisymmetric.

**Postcondition**

Predicate that describes the state of a system after the performance of some action. Actions are often specified by pre- andpost-condition pairs.

**Potential**

Scope that exists for exploiting the functionality of the interactive system at any stage in an interaction irrespective of any interactional initiative. "Event" potential is the user's scope for engaging in alternative actions yet continuing to achieve the same objective (for example filling a form by entering the fields in any possible permutation). "Objective" potential is the user's scope for achieving alternative objectives at a particular stage in the interaction. Hence low objective potential occurs when a user is locked into a single objective at a stage in the interaction.

**Precondition**

Predicate on the state of a system that determines those states from which an action can be carried out.

**Predicate**

A boolean-valued function of the state, behaviour, or trace of a system. A predicate may represent a property.

**Prefix**

(of an ordered set  $\langle S,R \rangle$ ) is a subset  $T$  of  $S$  such that for every  $x$  in  $S$  and  $y$  in  $T$ , if  $(x,y)$  is in  $R$  then  $x$  is in  $T$ . Note that the empty set  $\langle \emptyset,R \rangle$  and the full set  $\langle S,R \rangle$  are always prefixes of  $\langle S,R \rangle$ .

**Prefix Closure**

(of an ordered set  $\langle S,R \rangle$ ) is the set of all ordered sets  $\langle T,R \rangle$  such that  $\langle T,R \rangle$  is a prefix of  $\langle S,R \rangle$ .

**Presentation [CGRM]**

Processing element within a layer that transforms an information type. The transformed information is transferred to the next lower layer in a hierarchy or to another sub-system in the system, when that layer is the lowest one in that hierarchy.

**Presentation**

- (a) That part of the system or component of a system whose state can be perceived and potentially perceived by the user through browsing commands.
- (b) Perceivable state of the system or component of a system or potentially perceivable state of the system or component of a system through inspection commands.

**Presentation state**

State vector of the presentation component of the system.

**Process**

Same as Agent.

**Property**

An observable of a system that can be described by a predicate and measurable (if used to assess the usability of a system).

**Proportionality**

Property that holds when measurable properties of a presentation are proportional (in the mathematical sense) to some value or function in the functional state.

**Reachability**

Property that requires that some state or set of states can be reached from a given state through the actions defined on a system.

**Real world validity [\*]**

that a metaphor corresponds to the users intuitions about what the system will do, for example the mapping of left in (visual) space to an appropriate notion in the underlying model.

See Metaphor, Affordance.

**Recovery**

Performance of actions that take a system from some 'unsafe' or undesired state to one satisfying some safety property.

See Recoverability.

**Recoverability**

Property that the system provides the user with commands to undo the effect of some action.

See Recovery, Undoability.

**Refinement**

(1) A model C of a system that includes the observables of a model A and whose behaviour is consistent with that of A - the actual consistency requirements vary between refinement notions. C is called the concrete system and A is the abstract system. Usually C will introduce new observables.

(2) The process of developing a refinement (1) for some system.

**Rendering, a**

Some perceivable presentation.

**Render, to**

To make perceivable.

**Rendering function**

Concept to denote the concretization activity within a system.  
See Abstraction function.

**Rendering relation**

A relation between the functional state and presentation of a system or component.

**Repair**

See Recoverability.

**Response**

An event generated or caused by some agent.

**Sequential process**

Process whose behaviour is a totally ordered set of events, i.e., a process that receives at most one event at a time (e.g., if two events arrive at the same time, one is processed, one is lost or delayed).

**Semantic delegation**

Transfer of task domain concepts from the functional core into the user interface.  
See Semantic repair.

**Semantic repair**

Semantic delegation used to augment the functional core with concepts that were forgotten in a previous implementation of the functional core and that cannot be incorporated in the functional core for technical reasons (e.g., source code is not available).

**Session, user**

Step between the initial state and the terminal state used by the system for the user.

**Software architecture, a**

A particular set of software components satisfying a set of relationships and described at some level of refinement (does not necessarily derive from an architecture model).

**Software Architecture Model**

Guide (model) for developing a software architecture.

**Software Architecture Model, Conceptual**

Guide (model) for developing a software architecture whose components are described at a conceptual level.

**Software Component**

Software unit that encapsulates a set of logically connected operations and data.

**State**

Assignment of values to names representing the observables of a system.

**State vector, of a component, of a system**

Names that define the state of a component, of a system.

**Static context**

Portion of the context that does not vary over session but may be modified between sessions.

**Step**

Any pair of states in a behaviour. For action A, an A-step is a pair of states that are related by A.

**Stimulus**

Event received by some component of a system from another.

**Stuttering step**

Step that leaves some subset of the observables unchanged.

**Subsystem**

Component of a system that behaves like a system.

**Syndesis**

The expression of a cognitive theory and system model within a common (formal) representation to allow description of and reasoning about the interaction between human and computer agents.

**Syndetic modelling**

Syndesis using ICS (interacting cognitive subsystems) and FSM (formal system modelling).

**System**

A whole or collection of components capable of performing information processing and/or information transfer, and whose behaviour can be observed and possibly measured.

Sys

**tem physical action**

Action performed by the system whose effect is made perceivable through output physical channels and possibly captured by human input channels.

**System state**

Same as State.

See Functional state, Presentation state.

**Task**

(a) A goal, together with some procedure or ordered set of actions that will achieve the goal.

(b) An ordered set of actions that will achieve a given goal when performed from a state satisfying some precondition.

**Task domain concept** [\*]

Concept identified by task analysis as relevant to the user to accomplish the tasks in that domain.

**Task domain command**

Command whose effect modifies the functional core component of the system.

**Task-equivalent interactive systems**

The space of all the possible interactive systems able to support a given set of tasks.

**Template**

Subset of the observables of a system that are relevant to the performance of some task.

**Trace**

An ordered set of events.

See Trajectory.

**Transition**

A pair of states, usually defining the effect of an action on a system.

**Undoability**

Same as Backward recoverability.

**User interface system**

Component of a system that implements the abstraction and/or rendering process of information between the functional core and the physical devices of the system. It includes the presentation component.

**User interface**

Common shortcut for User interface system.

**User interface state**

State vector of the user interface subsystem.

**User physical action**

Physical action initiated by the user.