

PANGLOSS

A DIGITAL LIBRARY OF COMPUTER ASSISTED MULTILINGUAL AND MULTIMEDIA LEARNING AND TRAINING TOOLS FOR LANGUAGES

A PROJECT OUTLINE

**PETER STOCKINGER
ELISABETH DE PABLO
HAMMOU FADILI**

(MSH – ESCOM 1995)

Contents

1) INTRODUCTION	3
2) THE CONTEXTUAL ADAPTATION REQUIREMENT FOR COMPUTER-AIDED LEARNING AND TRAINING SYSTEMS.....	4
3) THE SCRIPT UNIT	9
4) THE GENERAL FRAMEWORK OF A CONTEXT-SENSITIVE SPECIAL- LANGUAGE CAL-SYSTEM	11
I) The Domain-Module.....	12
II) The Instruction-Module.....	15
III) The Student (or User)-Module.....	18
IV The Interaction Patterns between the Three Modules	20
6) FINAL REMARKS	22

1) INTRODUCTION

The Pangloss project pursues the following objectives:

- conception and development of generic components (i.e., of a library of re usable modules) for the computer-assisted, interactive authoring of multilingual and multimedia learning tools for special languages (such as in finance, law, insurance, health care and medicine, science and technology, ...);

- conception of a streamlined design methodology for the cooperative and remote authoring of multilingual and multimedia learning tools for special languages;

- development of specific multilingual and multimedia special language CAL-systems in several pilot domains (finance, law, and insurance).

2) THE CONTEXTUAL ADAPTATION REQUIREMENT FOR COMPUTER-AIDED LEARNING AND TRAINING SYSTEMS

It is a principal fact that information, communication, as well as learning systems in general, should possess the "quality" of *context-sensitivity*.

In other words: they should be conformable to particular situations of information and knowledge transfer. They should be able to fulfill of what is called not only in communication sciences or semiotics, but also in distributed artificial intelligence, a *communication contract*, that is, loosely speaking, the accomplishment of a proposed or required information or knowledge objective, viz. goal, whereas the proposed or required goal is defined by the user and the realisation of this goal is undertaken by specific information- and knowledge-actions executed by a (communication, information or again learning) system.

With respect to such a "*contractual view*", there are several *principal requirements* which have to be taken into account in a conceptual framework governing the specification and development of a multimedia (and multilingual) information and communication system in general and of a multimedia (and multilingual) special language CAL-system in particular:

a) Knowledge and information that are handled by a particular system are, principally, *negociable*, i.e., *adaptable and modifiable*, not only following the needs, objectives, and epistemical states of a user, but also because of their intrinsical epistemical and temporal limitations.

A practical consequence of this requirement is the conception of a highly modularised "*contents library*" of so-called thematic scripts or again thematic configurations that refer to a given domain of expertise such as, for instance, to the (lexical, terminological, grammatical or again phonetic) domain of special languages.

b) A particular system must be able, too, to adapt its strategies to guide the user adequately during the exploration and "manipulation" of the thematic worlds of an application, such as, for instance, the lexical data of a special language.

This requirement concerns a specific type of H/S-interactions called *addresser-oriented interactions* which are differentiable in several more particular functions of guide, help, warning, advise, ...);

c) The architecture of a system must be designed such in a way that it can take into account the fact that a user can, within the context of a given learning or training strategy, access, explore, learn, and manipulate relevant data following his own habits, desires or objectives.

This requirement concerns a specific type of H/S-interactions, called *addressee-oriented interactions* which are restricted to circumscribed navigation-liberties, that is, in general, to knowledge-valuated or conditional links between linguistic or other data.

d) Finally, the interface(s) and interface displays of a particular system must be organized and codified following standards of languages of expression that are easily understandable by a user.

This requirement concerns also, the constitution of a library of more generic interface components which can be assembled into several concrete or real interfaces.

Given these general requirements, a multilingual and multimedia special language CAL-system in the Pangloss project is principally supposed to be characterized by the following characteristic features:

a) Its *content level* (i.e. the lexical, terminological, and grammatical material, the definitions, explanations and examples as well as the phonetic precision) *must be adapted* not only to the given knowledge-level of the user but also to his/her needs and desires. It has to take into account, too, the more or less unanimous, viz. conflicting nature of a given linguistic data (such as, for instance, competing definitions of a term).

b) The *organization* of the linguistic and phonetic material in the form of a "courseware" must be *motivated and developed* in a rather "natural" way following typical language-acquisition strategies (for instance, a courseware should be designed according case-based, example-driven vs model- or grammar-based maxims, or again according more specific maxims such as "first the more recent and more commonly accepted definitions of terms, then the older and/or more specific ones, ...).

c) The system has to support special and user-attuned (i.e. conditional) *hypermedia-links* to *other internal* as well as *external information resources* (lexical, textual, and multimedia databases, dictionaries, encyclopedias, ...).

d) The *testing and evaluation procedures* do not only have to calculate "blindly" the results of a "lesson" but they have - metaphorically speaking - to *explain* the obtained results and to provide a user with *efficient help* for overcoming his/her particular linguistic difficulties.

e) The *test-methods* themselves have to be *diversified*. Besides classical types of tests, they have to also included games, puzzles, and simplified simulations of real situations or linguistic case-studies;

f) The *user-interfaces* not only have to be multilingual but have to be *organized and designed* in such a way that they could be rather easily adapted to specific user habits.

Other characteristic features of a multimedia and multilingual special language CAL-system in the Pangloss project concern the notion of a *CAL-system itself*.

g) The *system itself* has to be designed both as an *off-line* or as an *on-line product* in order to preserve the possibility of its use as a product in *distance learning*;

f) The *system itself* should be designed in a *highly modularized way* in order to allow the re use of its different components in the design and development of other special language CAL-systems;

g) Finally, the *system itself* should be designed as a *generic* as well as *concrete* or specific product in order to allow the rapid development of *different versions* of it (i.e., basic versions or more or less highly specialized versions).

These seven characteristic features define basically the *context-sensitivity* of a multilingual and multimedia special-language CAL-system in the Pangloss project.

They assess, too, a *more sophisticated theoretical and methodological framework* for the specification, conception, and development of a context-sensitive computer-aided learning system (cf. chapter 5).

The central notion in this theoretical and methodological framework is the *script unit*.

3) THE SCRIPT UNIT

The script unit is a *classificational, functional, interactional, and expression entity* of which a whole script or scenario is composed: by analogy, it is possible to consider a script as a "text (or a document) in a context" and a script unit as a "phrase in a context".

In the most simple case a script unit corresponds to a whole script, but, in general, a script is composed of a more or less important ordered set of script units.

As a *classificational entity*, a script unit identifies, in the proposed theoretical and methodological framework:

- the types of linguistic or other data which are relevant for a given special-language CAL-system,
- the types of addressee-bounded actions,
- the types of addresser-bounded actions.

As a *functional entity*, a script unit defines *dynamic constellations* between the three classificational types of a script mentioned above.

In other words, it determines for as well as between a set of linguistic situations, sets of addressee-bounded and addresser-bounded actions by the means of which a user can explore and manipulate linguistic (or other) data with the help of the system.

Finally, it determines, too, a set of interfaces (or interface components) which may be activated (and assembled) following the specific habits of a user (cf. below).

As an *interactional entity*, a script unit defines dynamic constellations between functional script units, that is: between script unities which encapsulate a dynamic constellation between the three classificational types of scripts mentioned above.

In other words, as an interactional entity, a script unit regroups several script units in one more complexe module corresponding, for instance, to a specified "lesson" or "course".

Finally, as an *expression entity*, a script unit defines the formal and proper physical organization of interfaces and interface-displays on the classificational, functional and interactional level.

The script unit as an expression entity is based on a canonical structure that guides the formal and physical organization of interfaces and interface-displays. The canonical structure recovers:

- an entry for the linguistic (or non-linguistic) data to which are dedicated one or more specific regions of the interface,
- an entry for the specific addresser-bounded actions to which are dedicated one or more specific regions of the interface as well as one or more H/S-interaction objects,
- an entry for the specific addressee-bounded actions to which are dedicated one or more specific regions of the interface as well as one or more H/S-interaction objects;
- an entry for the expression codes (written texts, graphic objects, pictures, movies, sound, ...) used in the interface,
- an entry for several timing and synchronization measures.

4) THE GENERAL FRAMEWORK OF A CONTEXT-SENSITIVE SPECIAL-LANGUAGE CAL-SYSTEM

Generally, a context-sensitive multilingual and multimedia courseware, and in particular a courseware pursuing the acquisition of a special language can be characterized with the help of three principal modules that are interconnected:

- the domain module,
- the instruction module, and
- the student (or user) module.

The *domain module* stores the knowledge of a given domain of reference as well as the computer-aided exploration of that knowledge by an user.

The *instruction module* deals more particularly with the "control" of the knowledge that a user has acquired in exploring the stored knowledge in the domain module.

The *student module*, finally, deals with the comparison and evaluation of the knowledge of a user with respect to the stored knowledge in the domain module.

With respect to the previously introduced definition of the script-unit as a classification entity, each of these three principal modules consist of four basic types of script-units:

- the type concerning the selected and stored linguistic information or knowledge data that constitute a courseware;
- the type concerning the addressee-bounded actions (i.e., the actions of the "student" or, more generally, the user),

- the type concerning the addresser-bounded actions (i.e., the actions of the "system"),
- the type concerning the interfaces by the means of which a stored knowledge will be explored and learned and by the means of which the communication modalities between the "student" and the "system" are defined.

According to the four basic types of script units and with respect to the necessary restrictions to the context of the Pangloss project, the quoted above modules can be globally specified as follows.

I) The Domain-Module

1.1) Script units concerning the stored linguistic data (i.e., the linguistic knowledge of a domain of reference that should be learned or assimilated by an user).

They are specified by the following components:

a) selected relevant terminological, lexical, grammatical and phonetic data of a given special language (the selection itself can be done "manually" or interactively with the help of lexical parsers, electronic lexical databases and lexical-grammatical dictionaries, thesauri, phonetic dictionaries, and so on);

b) "manual" or computer-aided organization of these data in basic and more specialized entities (the notion of "specialization" is intended, here, in a rather broad sense, by among others, domain-dependent specializations, regional specializations, historically defined

specializations, ...);

c) generic structures by the means of which are organized dictionary-like "articles" (including the formation of definitions, descriptions, and/or examples) as well as their necessary contextual attunements;

d) generic structures for the development of case-based and/or model-(i.e., language-) based situations (as well as their contextual attunements) by the means of which the meaning of a term or a set of terms can be explained;

e) a set of specified links by the means of which articles and situations will be interconnected and accessible by a user if he/she satisfies given initial conditions;

f) a set of standard acquisition paths along which articles and situations will be regrouped and sequentially organized (the notion of "sequentiality" should be understood, in this context, as a general one specifiable in linear, alternative, parallel, ... paths)

1.2) Script-units concerning the addressee-bounded actions (i.e., the actions of query, navigation or exploration of a selected special language knowledge-data by a user (a "student"))

They are specified by the following components:

a) a set of major addressee-bounded action-types such as the query ("content browsing", "multiple choice", "NL-query", ...), the query-specification, the navigation ("contextualized content browsing", "contextualized multiple choice", "heuristic-based exploration", ...) as

well as of different other question- and help-asking actions

b) a set of types of initial conditions that a user has to satisfy in order to be free to navigate via restricted navigation-paths

c) a set of imposed or obligatory navigation-paths that a user has to follow during a "lesson"

d) a set of free or restricted navigation-paths linking the courseware to external information or knowledge bases

1.3) Script-units concerning the addresser-bounded actions (i.e., the system specific actions of guidance, help, relevant information search, and so on)

They are specified by the following components:

a) a set of action-types concerning more particularly thematic or linguistic research strategies of a queried information or knowledge data in internal or external databases

b) a set of action-types that are based on specific user- or domain dependent heuristic rules (for example: if a user selects iteratively example-based presentations of terminological data, the system should be able to recognize this and to ask him if he/she wants to continue primarily in this way, ...)

c) a set of major system-action types of which the principal function is to support the user, such as the action type "assistance in the user's navigation activities" ("memorizing" of the user's path", "return to a solicited point in the navigation path", ...), contextually appropriate

"answers" to user's requests, contextually appropriate "warning actions", contextually appropriated "actions of advice", ...

d) a set of major system action-types of which the principal function is to request some specific action by the user (cf. especially these action types by the means of which a user is requested to follow an obligatory navigation path or again to demonstrate that he/she fulfills initial conditions in order to be able to enter in a restricted navigation path).

1.4) Script-units concerning the multilingual and multimedia interfaces by the means of which the previously specified script-units of the domain-module will be represented (for further details, cf. the paragraph in chapter 4 concerning the script unit as an expression entity)

II) The Instruction-Module

II.1) Script units concerning the "problematization" of linguistic data that include a library of tests, games, and so on by the means of which the acquisition of the linguistic knowledge by an user can be controlled.

They are specified by the following components:

a) a set of types of tests (such as simple question-answer tests, the correct reconstruction an incorrect presentation of linguistic material, the search of correct solutions to a more complex linguistic problem, ...) but also of more sophisticated linguistically motivated

games and competitions between several users

b) a set of linguistically motivated "cascades" (i.e., successions) of tests and/or games adapted to the user and to his/her learning preferences

c) particular test and game periods (i.e., after each "lesson", after a given period of consultation, randomly, ...)

II.2) Script-units concerning the addressee-bounded actions that concern especially the actions of answering tests, resolving problems and playing games.

They are specified by the following two components:

a) a set of the major action types that must be provided by the user in order to realize a test or to play a linguistically motivated game (cf. the already generally recognized and available action types in more advanced multimedia educative software systems like Director, Authorware, Icon Author, Toolbook...)

b) a set of more particular action types concerning, for instance, the request of help for solving a problem, the request to be guided to relevant "articles" or linguistic situations in the domain module, or again to external information and knowledge data bases, the request to compare a solution with predefined solutions, the request to interrupt a test or a game, ...

II.3) Script-units concerning the addresser-bounded actions concerning firstly actions of tutoring, assistance, or coaching provided by the system when a user answers tests or plays linguistically motivated games, and secondly actions of evaluation of given tests or game results.

They are specified by the following components:

a) a set of system action types of which the principal function is to intervene in a solution process by the means, for instance, of an interactive assistance, a partial problem solving activity, references to predefined solutions,

b) a set of system action types of which the principal function is to request some specific action by the user (cf. especially these action types by the means of which a user is requested to perform certain actions or again to perform certain actions in a given order, ...)

c) a set of system action types of which the principal function is to provide a user with requested information during the solution of a problem or again to connect him/her with relevant libraries in the domain module

d) a set of system action types of which the principal function is to evaluate the test or game results of a user (i.e. graduation of the importance of correct and wrong responses; comparison of given results to some normative standard of already realized scores, "notation", ...)

II.4) Script-units concerning the multilingual and multimedia interfaces by the means of which the previously specified script-units of the domain-module will be represented (for further details, cf. the paragraph in chapter 4 concerning the script unit as an expression entity).

III) The Student (or User)-Module

III.1) Script units concerning the preliminary or acquired linguistic knowledge of a user

They are specified by the following two components:

a) generic structures by the means of which is constituted a library that contains the tested preliminary special-language knowledge of a user

b) generic structures by the means of which is constituted a library that contains pieces of the progressively acquired special language-knowledge of a user, including especially false conceptions, bugs, etc.

III.2) Script-units of addresser-bounded actions concerning the explanation activity in a learning process (based on the results of previous tests or games in the instruction-module)

They are specified by the following three components:

a) a set of diagnose-action types (i.e., simple recognition of incorrect responses, comparison with previous responses or responses furnished by other users, ...)

b) a set of explanation strategies (i.e. simulation of the consequences of a incorrect response, situation/example-driven explanations, model- (i.e., language-) driven explanations, interactively processed explanations (i.e., in helping the user to recognize an incorrect response), ...

c) a set of more specific consequence-action types (i.e., retribution of the user's effort, organization of further "lessons" with respect to the results of a test or a game, ...);

III.3) Script-units of addressee-bounded actions concerning more specific request activities

They are generally specified by request-action types by the means of which the user aims at either one or more explanations concerning his/her test results or again the comparison of an actual result with already existing ones.

III.4) Script-units concerning the multilingual and multimedia interfaces by the means of which the previously specified script-units of the domain-module will be represented (for further details, cf. the paragraph in chapter 4 concerning the script unit as an expression entity).

IV The Interaction Patterns between the Three Modules

In more advanced computer-assisted learning or tutoring systems, there are typically two major structural patterns that coordinate the three quoted modules:

a) first: "domain-module", then: "instruction-module", finally: "student-module"

b) first: "instruction-module", then: "student-module", finally: "domain-module".

The *first* structural pattern describes the *standard learning situation* that, after having explored some linguistic (or other) knowledge, the user is submitted (or submits himself) to a cycle of tests or "educative" games by the means of which, finally, not only his acquired knowledge but also several "bugs" or "misconceptions" that characterize his knowledge can be discovered and evaluated.

After the phase of evaluation, more advanced computer-assisted learning or tutoring systems proposes the user to explore parts of the knowledge domain that are appropriate to his level of comprehension (which is, from a formal point of view, nothing else than the reiteration of the first interaction pattern).

The *second* interaction pattern appears, generally, at the beginning of a consultation or a learning session. Its purpose is to elicit the level of knowledge of a user as well as possible misconceptions that characterize his knowledge in order to adapt the first interaction pattern

to the context of the user. The second structural pattern is sometimes - quite metaphorically - called a "tutorial strategy".

The both quoted structural patterns are examples of what is called, in the proposed framework, *interactional script-unity*, that is: a script unit which consists of an two or more functional script units as well as by more specific relationships between them.

The multilingual and multimedia special-language CAL-systems in the Pangloss project will be based on these interactional patterns.

In this sense, they may be used not only as computer-assisted learning systems but also as user-sensitive information query and navigation systems (as a kind of "personalized" electronic encyclopedias) in the field of special languages.

6) FINAL REMARKS

1) It is rather obvious that the specified general framework of a CAL-system in the Pangloss project possesses a more general conceptual structure which can be (at least partially) re used in the conception and development of CAL-systems of which the reference domain is not necessarily a linguistic one.

2) Even if each CAL-system in the Pangloss project is designed and developed within the general framework presented in the 5th chapter, not every particular system has to be provided with the whole potentiality of this framework.

In other words, the Pangloss project can be designed and developed to include more or less sophisticated, more or less specialized CAL-systems - the important point here is, that they all are designed and realized within a common conceptual framework.

3) The conceptual framework constitutes, in this sense, a common standard for the elaboration of a streamlined design and development methodology as well for the conception of useful procedures for the remote cooperative authoring of CAL-systems.

4) The specifications of each script-unit that constitutes one of the three modules seem to authorize rather precise conceptual definitions of data-types by the means of which it should be possible to identify and to retrieve, in a given database, (potentially) relevant data for a learning program.