CADMOS:

An Approach to Developing Web-based Instructional Systems

SYMEON RETALIS AND EMANUEL SKORDALAKIS

{RETAL, SKORDALA}@SOFTLAB.NTUA.GR

NATIONAL TECHNICAL UNIVERSITY OF ATHENS

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

TEL: ++301 7722487, FAX: ++301 7722519

Abstract

The educational use of the Internet and the World Wide Web has grown enormously over the last few years and continues to grow at a tremendous rate. Instructional systems, based on these networked technologies, have been constructed to solve instructional problems. There is evidence that these systems efficiently support the instructional process. Instructional developers should follow effective and quality driven development methodologies, which are specifically developed to support the development of network based instructional systems. This article presents a methodology, CADMOS, which accords to the principles of the third and partially the fourth generation of Instructional Systems Development (ISD). The main innovation of CADMOS is its advocacy on splitting an instructional system into four sub-systems: Human, webware, other learning resources, and specific infrastructure sub-systems.

Keywords: Instructional Systems Development (ISD) methodology, World Wide Web

CADMOS: An Approach to Developing Web-based Instructional Systems

With the dramatic advances in networked technologies, especially the Internet and the World Wide Web (WWW), new kinds of instructional systems have arisen that emphasize the interactivity in learning. The added value of networked technology is that it enables the enrichment of the learning paradigm in order to:

a) Support open, flexible and learner-centered patterns of study;

b) provide new ways for learners to work collaboratively;

c) facilitate the development of communication and co-ordination skills, and

d) encourage the development of technological skills (ALTP, 1997).

A question that should be addressed to instructional developers, is how to develop such instructional systems capitalizing on the strengths of the added value of networked technologies?

Existing answers

Instructional systems aim to support and partially automate the instructional process. For example, an instructional system may support a course, a seminar or even a series of lectures on a particular subject matter domain. These systems also intend to satisfy instructional needs for a domain which have been created mainly because of the advances in research and technology, the information society, and the globalization of the market (Hodgson, 1997).

The Instructional Systems Development (ISD) process has evolved from an intuitive to a very sophisticated process, which depends on theories and practice from research fields such as pedagogy, courseware engineering, and software engineering. Four generations of ISD models can be found in the literature (Tennyson & Morrison, 1997). The most commonly used is the third generation, called ISD^3. Over the last five years the fourth generation, called ISD^4, has been emerging.

The two latest generations advocate that the development process should follow the principles of a problem solving approach. They both agree that the ISD process is staged and highly iterative. The main difference is that ISD^4, in contrast to ISD^3, advocates that each instructional development project is a unique situation requiring that a methodological solution and a development process should be constructed based upon the given problem or need (Tennyson & Morrison, 1997). So, for each instructional problem a specific instructional system should be designed based on the findings of the problem assessment. Moreover, ISD^4 developers should follow a systems dynamics process model.

Nowadays, instructional systems make extensive use of networked technologies. A consequence of this trend is that developers build complex instructional systems that incorporate a variety of organizational, administrative, instructional, and technological components (Carlson, 1998). ISD^3 and ISD^4 do not propose specific ways on how to separate the aforementioned components.

In this article we suggest a methodology for developing instructional systems which is based on an explicit division of the instructional problem solution into components or parts. Each one of these parts explains the way that (a) the human agents, (b) the web-based learning resources, (c) the other resources (e.g., materials in printed, analog format), and (d) the specific technological infrastructure should support the instructional process, accordingly.

The proposed answer

Methodologies for the development of instructional systems that make use of networked technologies have recently arisen. In this article, a new methodology called CADMOS is described. It mainly belongs to the ISD^3 generation. But it emphasizes the division of an instructional problem solution into explicit parts. This differentiation is a result of the adoption of the philosophy of YASM, a development methodology for artificial systems. The methodology emphasizes the development of software systems that are part of the artificial system.

With the artificial systems we automate, either totally or partially, the execution of processes that are mandatory for satisfying human needs (Skordalakis, 1998). Without such automation their execution, in many cases, is impossible, at least with the speed and accuracy required in real situations. In all cases, executing such processes is tedious, tiresome, and monotonous. Therefore, it should be avoided when possible. The human needs create a specific problem whose solution is the construction of an artificial system. One specific type of artificial system is an instructional system that is created to support the instructional process, satisfying teachers' and learners' needs.

The methodology that is presented in this article is the outcome of action research performed at the National Technical University of Athens. It results from a project entitled "undergraduate studies by distance" which was funded by the Greek Ministry of Education (Koutoumanos et al., 1996) as well as two other university funded development projects. The aim of these projects was to enrich the classroom-based teaching mode with open learning techniques using new technologies. Thus, three web-based instructional systems have been created for (1) an undergraduate course in the subject matter domain of Compilers, (2) a postgraduate course called "Internet Publishing", and (3) a course for continuing education concerning the XML language.

In this article, we will present the philosophy and an overview of the sub-processes of the instructional development process according to CADMOS. Emphasis will be given to the problem solving sub-process and the explicit division of this instructional problem solution into parts.

The structure of the article is the following. First, we will briefly present the YASM and the ISD^3 methodologies whose merger resulted in the CADMOS methodology. Then each development sub-process (problem solving, construction, and utilization) will be analyzed. Finally, concluding remarks about CADMOS will be made including comments on the future plans.

Merging Artificial Systems Development Methodology and ISD^3

YASM methodology

YASM, as a software development methodology, is a collection of process models, methods, tools and standards that aim to help developers of software systems with constructing good quality end products in an efficient way. According to its philosophy, the software systems are only one part of artificial systems.

The artificial system consists of four interrelated sub-systems: The human sub-system, the software sub-system, the hardware sub-system, and the specific infrastructure sub-system. Each one of these sub-systems needs a specific methodology for its development. YASM focuses on the development of the software sub-system. The development process model of an artificial system according to YASM is shown in Figure 1.

Insert Figure 1 About Here

The development of an artificial system is comprised of three basic processes: The solution finding process, the system construction process, and the system utilization process. The solution finding process consists of the problem formulation sub-process and the problem solving sub-process. The latter sub-process gives as output the problem solution. Finding the problem solution needs prior assessment of the factors that will influence the problem solving (e.g., the availability of resources), but it does not take into consideration the processors (or the technological infrastructure) needed for the automation of the execution of processes. This solution is the desired one in an abstract form and described in natural language.

During the system construction process the problem solution is divided into four parts:

1. The human part which involves all procedures requiring human interaction (e.g., recording constant data which will be used as input to a software program);
2. the soft part which automates the information processes;
3. the hard part which concerns the development of the hardware infrastructure that is required for automating the information processing (e.g., data loggers or counters), and
4. the specific infrastructure part which deals with the infrastructure in terms of software and hardware or other devices that are needed for the utilization, or maintenance of the artificial systems (e.g., a version control system for the registration of new evolutions of the system).

The system construction process is highly dynamic because of the decisions made, which are bound to the particular implementation and the run time environment. It focuses on creating the solution architectural plan and the actual implementation of this plan. It is obvious that for a given problem solution a number of possible implementations may exist.

Finally, the system utilization process is concerned with the delivery, use, validation and maintenance of the artificial system. The input materials to the artificial system are being processed and checked for their quality. The quality control gives feedback to the artificial system for its maintenance (this may be corrective or perfective).

ISD^3

The ISD^3 model also proposes a problem solving approach to instructional development comprised by four dynamically interrelated stages (Tennyson & Morrision, 1997):

Stage 1: assessment, which deals with the identification of the instructional problem; assessment of the factors that affect problem solution finding (resources, learner population, etc.), and preparation of the problem solution plan as well as the managerial plan.

Stage 2: design, which deals with the production of prototypes for testing and evaluating design alternatives before committing to the actual production of one of them.

Stage 3: production, which is concerned with the production of instructional material, documentation reports and the preparation of evaluation and dissemination plans.

Stage 4: implementation, which is the delivery of the instructional system to the learner population, its summative evaluation and its maintenance.

Although the ISD^3 and the artificial systems development methodology have adopted the problem solving approach and the dynamic interrelation of the problem solving steps, their philosophies differ with regard to one issue. The artificial systems development methodology proposes that first a desired solution to the problem must be found, which should not take into consideration any constraints or factors stemming from the run-time environment (unless specifically declared). Only then the architectural solution plan properly fit to the run-time environment should be designed and implemented.

It is obvious that for some given instructional needs that compose an instructional problem, there are a limited number of ideal solutions that fit to specific, well-assessed and defined circumstances. However, there are a variety of ways for implementing one desired solution following a procedure consisting of mix-and-match and fine tuning of elements taking into consideration the risks and constraints involved in each implementation.

CADMOS: The merger

CADMOS advocates that the YASM philosophy should basically remain unchanged in order to solve an instructional problem. The instructional development should contain three main processes, each one subdivided into sub-processes as illustrated in Figure 2: The instructional problem solving process, the instructional system construction process, and the instructional system utilization process.

Insert Figure 2 About Here

The Instructional Problem Solving Process

The purpose of the instructional problem solving process is twofold: (a) formulation of the instructional problem and the assessment of constraints for that problem, and (b) instructional problem solution finding, that is, the construction of an ideal solution (Tennyson & Morrison, 1997).

The instructional problem formulation

Influenced by the philosophy of ISD[3] and ISD[4] models, and more specifically by the situational evaluation process of ISD[4], the problem formulation sub-process is a highly iterative and dynamic set of activities, during which the instructional developer or development team should:

- Assess learning needs.
- Assess the user/target population.
- Determine ID competence of the developer or team.
- Define constraints and restrictions imposed to the solution finding process, such as specific educational philosophy, available resources, et cetera.
- Define the instructional problem, that is, state the scope of the instructional system.

<u>The instructional problem solution finding</u>

The aim of the problem solution finding sub-process is to construct a desired solution to a given instructional problem already defined by the situational evaluation sub-process.

The instructional problem solution should not entail technicalities on how it will be implemented in a specific run-time learning environment. The term "learning environment" means "a community with its own culture and values providing a variety of learn places that support student learning. The culture of a learning environment shapes the attitudes of both staff and students. It will incorporate traditional values such as "... respect for different opinions, integrity in scholarship, and the pursuit of knowledge ..." (Ford et al., 1996).

Insert Figure 3 About Here

The solution to the instructional problem emerges from <u>blending</u> four interrelated sets of learning elements: The syllabus, the learning objectives, the didactic events and other issues like assessment methods, prerequisites, fees, and so forth. This solution is illustrated graphically in Figure 3 and is the main product of this process. It must be a well written, highly detailed document following specific guidelines on how to formulate the learning objectives, the syllabus, the didactic events, etcetera (for an overview, see Gagné, Briggs, & Wager, 1994; Rowntree, 1994; Tennyson & Breuer, 1997).

The Instructional System Construction Process

The construction process is the actual implementation of an instructional problem solution for a specific learning environment. The construction phase receives as input the solution from the problem finding process. Then, this solution is divided into four parts: (1) human part, (2) webware part, (3) other resources part, and (4) specific infrastructure part. This division is done within the systems engineering sub-process. Each part of the solution specifies how the solution will be realized and supported into a real learning environment. Having specified the solution parts, each one of them is developed according to related methodologies. The result is four subsystems that are all integrated into one "whole".

Instructional system engineering sub-process

The construction process begins with a sub-process that is dedicated to the creation of the architectural plan of the ideal solution according to the use case technique, which should fit into the specific run-time learning environment. A use case is defined as "a set of sequences of actions a system performs that yield an observable result or value to a particular actor" (Eriksson & Penker, 1998). As a complement to a use-case description (with the aid of use case diagrams according to Unified Modeling Language notation) a number of actual scenarios is provided to illustrate what happens when the use case is instantiated.

According to Verreck and Weges (1994), the general meaning of scenario is a sequence of events, showing interactions, information exchanges between agents, etc. By developing a scenario one gets a feeling of the main objects, processes, resources and their interdependencies in a particular existing or future situation. A scenario can thus be conceived as an internally consistent vision about a situation to come.

The resulting architectural blueprint of an instructional system includes details about each one of the subsystems. For the human part, for each kind of human actor involved in the instructional process his/her roles are described in as much detail as possible.

The webware should be regarded as a mosaic of resources. For each of the resources, the instructional engineer (designer) should specify details about its type (expositive, active) as well as its interrelationships with the others. For specifying the other resources, their format (print, audio, video, etc.) as well as a general description of them should be made. For both the webware and the other learning resources, specific tables should be created. These tables will show the apportionment of the elements of the solution (learning objectives, didactic events, and syllabus) to each one of the resources under development.

Finally, after reviewing the services provided by the common infrastructure, the specific infrastructure that needs to be developed is designed and documented. An instructional system makes use of a common infrastructure, which is a set of resources that provides services that are independent of the instructional problem (e.g., laboratories, networking facilities, etc.). However, in order to best support the instructional process, a specific infrastructure should be created (e.g., multimedia conferencing system, special hardware components, etc.) which will provide services unique to the particular instructional problem. CADMOS has adopted the Unified Modeling Language (UML) for designing the specific infrastructure. UML offers a lot of design facilities, and it tends to be an industrial standard in Object Oriented design for software systems.

Implementation sub-process

Having created the solution architectural plan, the construction phase continues with the implementation of each one of the solution parts, that is, the human part, the technological part, and the learning resources part. Methodologies stemming from cognitive engineering (in order to best train the human agents on how to play their roles effectively), courseware engineering (to develop high quality learning resources in time and within certain cots limits), and software as well as hardware engineering (for the construction of the proper software tools and the specific settings that will support the instructional process) will be followed during their implementation, accordingly. The development approach presented in this article does not propose any specific methodology for the construction of the human part, the other resources, or the specific infrastructure technological part. Nevertheless, CADMOS suggests a methodology at the micro-level for the development of the webware part. According to CADMOS the construction process of the webware should follow an evolutionary development process model, specific development methods and guidelines of best practice as well as templates for documenting the activities performed within each subprocess.

Webware construction using CADMOS

Webware consists of digital learning resources that incorporate a variety of media and a mix of instructional learning or pedagogical methods. Typically, webware looks like a mosaic containing a variety of components (Forte, Wentland, & Duval, 1997):

- Expositive components: these are typically hypermedia documents viewed by a user without entailing any degree of involvement, other than navigation.

- <u>Active components</u>: these are multiple-choice questionnaires, simulations, auto-evaluation exercises, self-assessment materials, and so forth.

Webware development is not an easy task. In general, courseware development methodologies follow the philosophy of the software development methodologies in terms of variations of the process models they have adopted or adapted. CADMOS proposes a specific micro-level methodology for ensuring the development of a high quality webware end product following a standardized development process.

It is outside the scope of this article to extensively give the microscopic view of CADMOS. However, its bases are the following:

- The webware process model is a variation of an evolutionary process model used in software engineering (Schash, 1990). This model, shown in Figure 4, involves developing and delivering the webware in an evolutionary fashion. The use of the evolutionary development process model suggests that the webware be divided into components, i.e. increments, which cover part of the functionality of the overall product and part of the overall requirements. The developer designs, builds, formatively evaluates and delivers the webware increment by increment. The learner's perception of the overall webware end product is a moving target. As increments are built and delivered, learners' and teachers' opinion regarding the values of the remaining increments may change. Features of them previously felt to be essential to the overall product may be deleted or new increments may be added to the list of increments to be incorporated into the overall product during some future phase.
- During the design phase the architectural blueprint of the webware is transformed into a detailed design according to the principles of an object-oriented webware design method, called CADMOS-D. The detailed design includes structural details of the webware

resources (the WebPages, the multimedia elements of each Webpage, the active elements, etc.), navigational issues, and abstract graphical user interface specifications.

- The formative evaluation method that is used, and which acts as an umbrella sub-process for quality checking of the increments that are being developed, is a slight variation of Tessmer's method (1995).

More details about CADMOS micro-level methodology can be found in Retalis and Skordalakis (1998).

The Instructional System Utilization Process

The last stage of an ISD process is concerned with the utilization of an instructional system. It is within this process that the developed system will be summatively evaluated and reviewed. The utilization process consists of:

- The instructional process when the instructional system developed is being used, tested and validated in a real learning environment.
- The summative evaluation of the instructional problem solution sub-process which provides feedback from both learners and teachers.
- The customization and distillation of the outputs of the instructional process which can be used for the evolution of the instructional system.

The key outputs of the utilization process are:

- A stream of learners motivated to be lifelong learners with qualifications and increased prospects of employment.
- Teachers/tutors who have improved their skills and knowledge when taking part in the learning process.

- Learning products (e.g., projects, assignments, archives of the communication between human agents, etc.) which can be used when properly distilled and customized as learning resources.
- Results of a quantitative and qualitative analysis of data provided by the human actors involved in the learning process (i.e., feedback). These results are input data for the maintenance sub-process that gives as output a new version of an instructional system.

CADMOS suggests the use of a specific summative evaluation method, entitled CADMOS-E. This is a pretest–posttest method incorporating some aspects of the illuminative evaluation approach (Retalis et al., 1998). It is a stepwise method supported by specially developed pretest and posttest questionnaires, which provide data for both quantitative and qualitative analysis. The focus of this evaluation is on the learning effectiveness of the instructional system as well as the identification of extensions and revisions that need to be made to the system. The learning effectiveness was conceptualized as being related to a multiple measurement index consisting of cognitive and attitudinal outcomes (Makrakis et al., 1998). The effectiveness of an instructional system is influenced by a number of variables such as:

- Quality of the webware and other learning resources;
- changes in learning styles;
- interactions with peers, instructors, and means of communication, and
- the quality of services that the specific and common infrastructures provide.

These variables are composite and are measured by multiple items, each measuring a slightly different aspect of the main variable. Moreover, the results from learners' assessment after the completion of the course are analyzed and combined with the analysis results of the evaluation study in order to specify the effectiveness of the whole instructional process.

Evaluation and Concluding Remarks

The explosive growth of the Internet and the World Wide Web combined with the technological availability for information publishing and for collaboration across distances are just building blocks of the provision of effective solutions to learning problems. Learning needs to be strongly motivated and systematically designed and structured to be efficient (Forte et al., 1997). In this article, CADMOS, a web-based ISD methodology has been presented. It is a merge of artificial systems development methodology and ISD^3. It also extends ISD^3 in the sense that the solution plan is explicitly divided into four parts, each one representing a sub-system of an instructional problem solution which needs a special development approach. The idea of splitting the solution into four parts came from YASM, an artificial systems development methodology, and the specific characteristics and idioms of an instructional problem solution based on the networked technology.

CADMOS is a combination of evolved practice with applied theory. Only three instructional systems have been developed following closely this approach, as yet. The development effort was about 8-15 person-months and the development teams were comprised of several people with interleaved specialties. The three instructional systems produced were in the subject matter domains of Compilers, Internet Publishing, and XML. Their objective was the modernization of the conventional curriculum of under and postgraduate studies with the introduction of open learning delivery vehicles supplementary to the traditional ex-cathedra methods.

CADMOS seems to be promising. The evaluation of the CADMOS methodology was based on the following criteria (Avison & Fitzgerald, 1995, Jayratna, 1997):

1. The quality of its end products;
2. the quality of the development process, and
3. the standardization of the development process.

<u>The quality of the end products</u>

All three instructional systems developed used the WebCT learning environment (see www.webct.com) as a specific technological infrastructure. The Compilers and the Internet Publishing courses were summatively evaluated using the CADMOS-E evaluation method, whereas the XML course is currently being evaluated.

About 80% of the 12 students who attended the Compilers course passed the final exam and took grades higher than 8 (maximum grade = 10). The students were very enthusiastic with the on-line learning material, especially with the case studies, which incorporated active elements (e.g., real time lexical analysis or compilation of source code), and the self-assessment questionnaires, as well as the interesting discussion topics that triggered asynchronous communication. However, they did not appreciate some parts of the course notes because they contained a lot of textual information.

The analysis of the evaluation study of the course on Internet Publishing showed that the students liked the course a lot and they were stimulated by being able to apply the acquired knowledge in real practice (small scale project assignments). The course content achieved a high quality rating and the students appreciated the structure of the course, the graphical interface, and the writing style of the course notes. They also mentioned that they found the use of the peer-review pedagogical method (concerning the project assignments) helpful.

Lack of a tool supporting CADMOS

The interview of members of the three development teams revealed that they found the CADMOS methodology useful and user friendly. The fact that CADMOS provided templates of the documentation that needs to be delivered during the construction of an instructional system as well as its use of specific tools like the Rational Rose (for the design) (see www.rational.com) in addition to the use of well documented methods such as CADMOS-D, formative evaluation, and CADMOS-E were very much appreciated and contributed to the quality control and management of the project.

A problem faced and admitted by all development teams was the lack of sufficient documentation and specific tools for creating the use case scenarios. The teams were helped in this task by the literature on UML.

The benefits of having a methodology

Having a common development approach throughout an organization is beneficial. Staff can move from one project to another and maintenance of the end products is easier. The Internet Publishing and Compilers instructional systems, the development of which lasted 14 and 15 person months respectively, were created in many increments with the involvement of various specialists. CADMOS proved to offer guidance and standardization for the accomplishment of the development tasks.

In conclusion, CADMOS attempts to reduce the degrees of freedom ensuring the quality of both end product and development process. Research work is under way for testing and validating CADMOS on two recently started small-scale projects. Moreover, research is underway to create a tool for supporting the whole CADMOS-D method.

References

ALTP (1997). Networked open learning: Course material. Lancaster, UK: University of Lancaster: Advanced Learning Technology Programme.

Avison, D., & Fitzgerald, G. (1995). Information systems development: Methodologies, techniques and tools. Columbus, OH: McGraw-Hill.

Carlson, P.A. (1998). Advanced educational technologies – Promise and puzzlement. Journal of Universal Computer Science, 4(3), pp. 210-216.

Eriksson, H.E., & Penker, M. (1998). UML toolkit. New York: Wiley Computer Publishing.

Ford, P., Goodyear, P., Heseltine, R., Lewis, R., Darby, J., Graves, J., Sartorius, P., Harwood, D., & King, T. (1996). Managing change in higher education: A learning environment architecture. London, UK: Open University Press.

Forte, E., Wentland, M., & Duval, E. (1997). The ARIADNE project (Part 1): Knowledge pools for computer-based and telematics-supported classical, open and distance education. European Journal of Engineering Education, 22(1), 46-61

Gagné, R., Briggs, L., & Wager, L. (1994). Principles of instructional design. Fort Worth, TX: HBJ College Publishers.

Hodgson, V. (1997). New technologies and learning: Accepting the challenge. In J. Burgoyne & M. Reynolds (Eds.), Management learning. 117-136, London, UK: Sage Publications.

Jyaratna, N. (1997). Understanding and evaluating methodologies: NIMSAD A systemic framework. Columbus, OH: McGraw Hill.

de Jong T., Sarti L. (eds.) (1994). Design and production of multimedia and simulation-based learning material, 253. Dordrecht, The Netherlands: Kluwer Academic Publishers.

Koutoumanos, A., Papaspyrou, N., Retalis, S., Maurer, H., & Skordalakis, E. (1996). Towards a Novel networked learning environment. Proceedings of WebNet'96, 267-272, November, San Francisco, USA.

Makrakis, V., Retalis, S., Koutoumanos, A., Papaspyrou, N., & Skordalakis, M. (1998). Evaluating the effectiveness of an ODL hypermedia system and courseware at the National Technical University of Athens: A Case Study. Journal of Universal Computer Science, 4(3), 259-275.

Retalis, S., & Skordalakis, E. (1998). Building quality Web based tele-learning systems, Proceedings of the XV IFIP World Congress–Teleteaching'98, Distance Learning, Training and Education, Part II, 835-844, 31 August–4 September 1998, Vienna-Austria.

Retalis, S., Makrakis, V., Papaspyrou, N., & Skordalakis, E. (1998). An enriched classroom model based on the World Wide Web technology: A case study. Active Learning, 8, 15-18.

Rowntree, D. (1994). Preparing materials for open, distance and flexible learning: An action guide for teachers and trainers (Open and Distance Learning Series). London, UK: Kogan Page.

Schash, S.R. (1990). Chapter 3: Software Life Cycle Models. In Software Engineering (pp. 20-40). Homewood, IL: Aksen Associates.

Skordalakis, E. (1998). YASM: Yet another software development methodology (in Greek). Technical Report. Athens, Greece: Software Engineering Laboratory, National Technical University of Athens. [see also: ftp://ftp.softlab.ntua.gr/pub/yasm/yeasm.doc]

Tennyson, R.D., & Morrison, G.R. (1997). Instructional development: A problem-oriented approach. Englewood Cliffs, NJ: Merrill Prentice-Hall.

Tennyson, R.D., & Breuer, K. (1997). Instructional theory: Psychological perspectives. In R. D. Tennyson, F. Schott, N. Seel, & S. Dijkstra (Eds.), Instructional design: International perspectives (Vol. I: Theory and research), 113-134 ,Hillsdale, NJ: Erlbaum.

Tessmer, M. (1995). Formative multimedia evaluation. Training Research Journal, 1, 127-149.

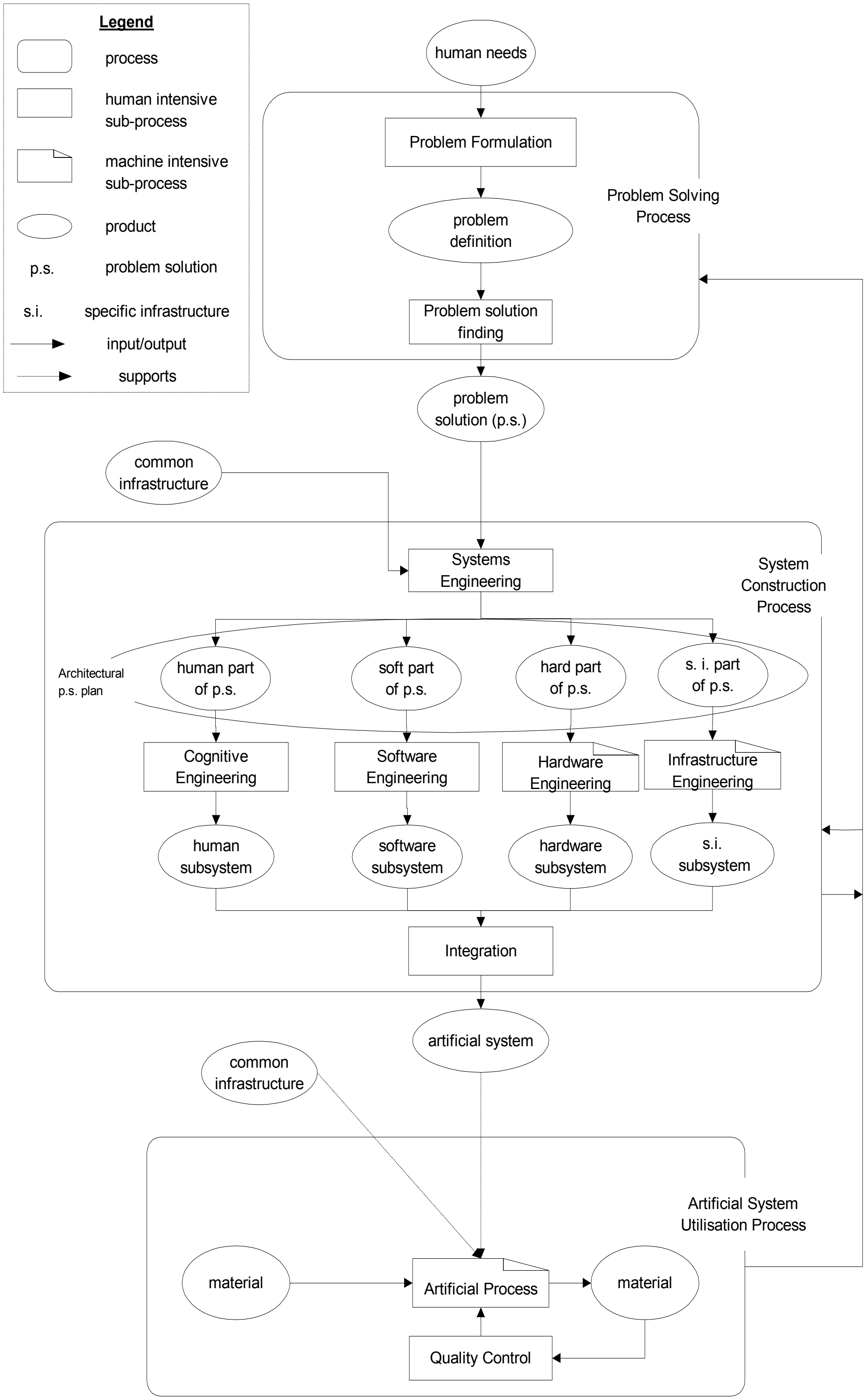

Figure 1. Development stages of an artificial system according to YASM

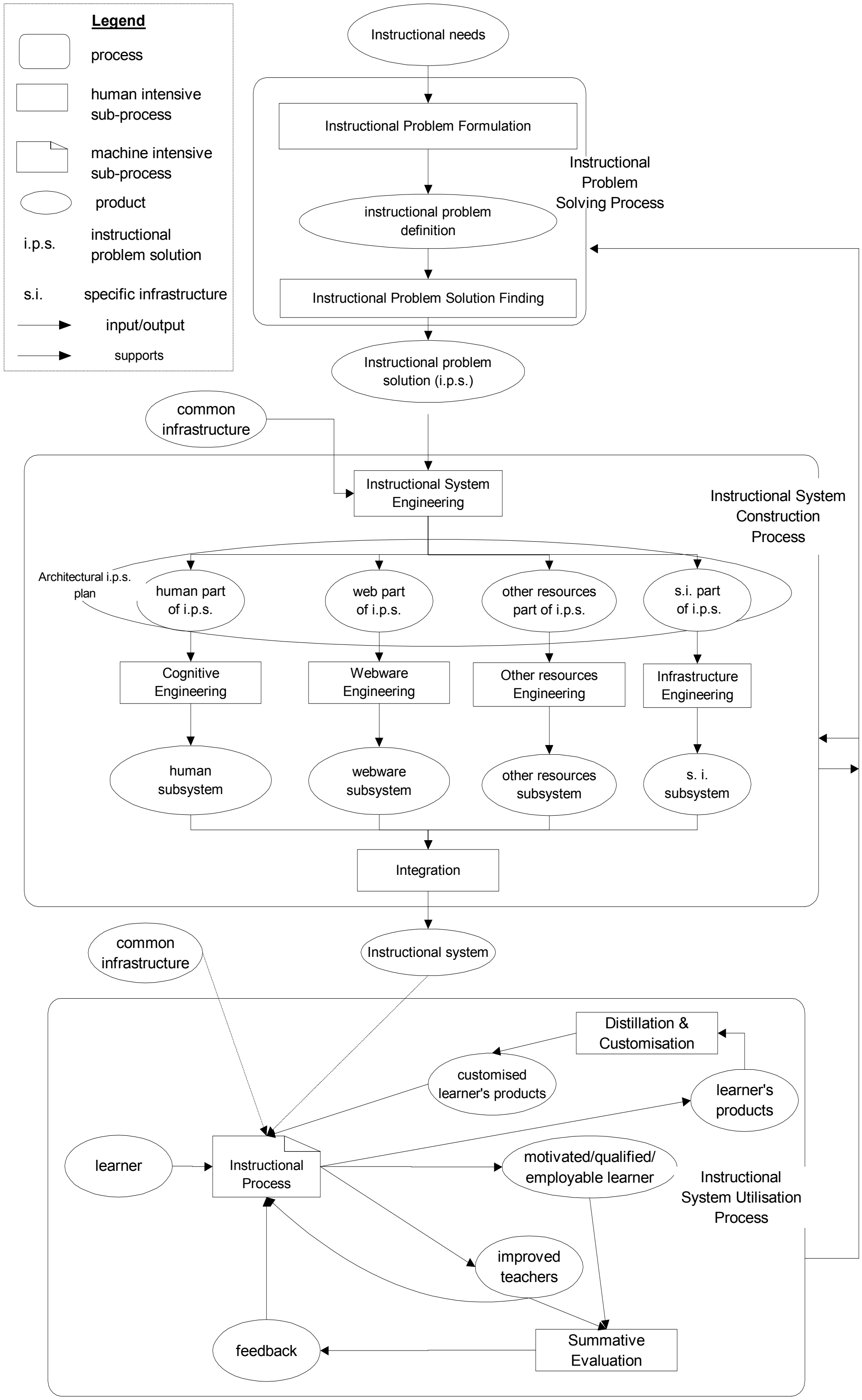

Figure 2. Development stages of an Instructional System according to CADMOS

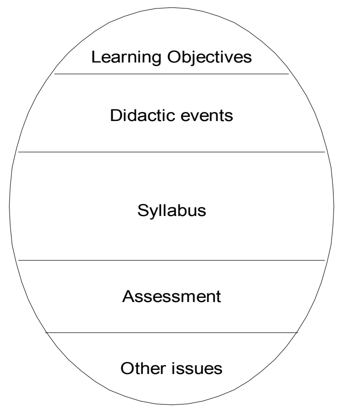

Figure 3. An ideal solution of an instructional problem

Figure 4. Webware development process model