MODELING RESOURCE MANAGEMENT IN INSTRUCTIONAL SYSTEMS DEVELOPMENT PROJECTS

Alexei V. Sioutine, Paal I. Davidsen, J. Michael Spector[1]
alexei@ifi.uib.no, davidsen@ifi.uib.no, mike@ifi.uib.no
Department of Information Science,
University of Bergen,
N-5020 Bergen,
Norway

ABSTRACT

This paper describes the modeling of project development and management in the field of instructional systems development (ISD), specifically in accordance with a recent ISD model (ISD^4 or Fourth Generation ISD) which takes into account the dynamic aspects of instructional planning and development (Tennyson, 1993). Instructional systems development is a reasonably structured and well-established process for developing education and training. Today instruction often involves technology-based learning materials and environments, often referred to as courseware systems. These systems involve significant and expensive software development and typically represent a level of complexity not encountered in more typical business-oriented software development projects. Large-scale courseware development projects often run behind schedule and over or beyond allocated budgets. They frequently use a lot of resources in terms of manpower, especially expertise that is already in scarce supply.

In ISD^4 projects are described as evolving phases (Analysis - Design - Production - Implementation - Maintenance). The first two phases are often the most crucial for a project in terms of ending up within time and budget constraints. Therefore, our modeling process has concentrated around the analysis and design phases, and specifically on the most common ways in which activities in these two ways are interrelated. The model consists of four basic sectors that describe key aspects of the structure of the project: (1) human resource management; (2) control; (3) plans; and, (4) development. Typical complexities of ISD^4 projects are represented in the model, such as lack of transparency and delays in progress reporting, task processing failures and feedback, requirement for and allocation of experts and novices, expert-novice interactions, etc.

INSTRUCTIONAL SYSTEMS DEVELOPMENT - ISD

This paper describes the modeling of project development and management in the field of instructional systems development (ISD), specifically in accordance with a recent ISD model (ISD^4 or Fourth Generation ISD) which takes into account the dynamic aspects of instructional planning and development (Tennyson, 1993). Instructional systems development is a reasonably structured and well-established process for developing education and training. Today instruction often involves technology-based learning materials and environments, often referred to as courseware systems. These systems involve significant and expensive software development and typically represent a level of complexity not encountered in more typical business-oriented software development projects. Large-scale courseware development projects often run behind schedule and over or beyond allocated budgets. They frequently use a lot of resources in terms of manpower, especially expertise that is already in scarce supply.
System dynamics has already been successfully applied in modeling and describing other types of software projects (Abdel-Hamid and Madnick, 1991). The study of Instructional System Development problems using methods of system dynamics (SD) was initiated by the

[1] The authors would like to thank Maria Poutilova for the input she gave to the model and for her critics and revisions of the model.

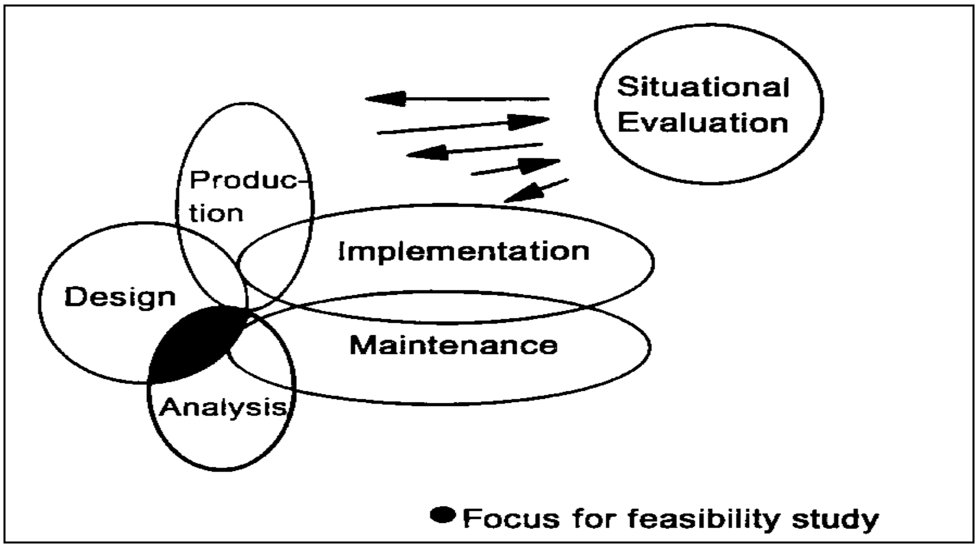

Figure 1. The ISD[4] development model developed by Tennyson (1993).

Grimstad Group[1]. The Grimstad Group collaborated in a number of conferences sponsored by NATO in the early 1990s. An initial pilot project determined that some of these results could be applicable and transferred to ISD projects (Grimstad Group, 1994).

ISD projects involve several phases of development, similar to what is commonly encountered and described in various software-engineering models. Originally, ISD was described in simple, linear terms. In ISD4 these projects are described as a sequence of phases (Analysis - Design - Production - Implementation - Maintenance regularly subject to a Situational Evaluation) which may involve internal and external iterations and which vary dramatically depending on local circumstances and constraints. The structure of the ISD development is represented in Figure 1.

Analysis and design tasks can be described as passing through two stages: Identification and Processing. Tasks can be completed correctly (successfully) or incorrectly (unsuccessfully). Failed tasks must be recognized as such and reworked. The tasks to be reworked can circulate inside a phase as well as between phases. The failure rate depends on the experience of the team, the project know-how.

There is clearly a delay associated with tasks having to be reworked, and failure to anticipate such a delay typically causes unexpected schedule overruns. There is also a relationship between how many tasks that need to be reworked and the composition of the workforce (percent of experts and novices working on tasks). Experts typically provide better initial estimates of how large a project is, create fewer tasks requiring rework and typically work more efficiently. Experts also cost more to keep on a project and are often in high demand and short supply. Such complex and dynamic factors as these make it difficult to determine optimal resource allocation for projects.

[1]Grimstad Group consisted of the following persons: Pål. Davidsen (University of Bergen, Norway) - SD modeling; Jose J. Gonzales (Agder College, Norway) – SD software; Daniel J. Muraida (US Air Force Research Laboratory, San Antonio, TX) – experimental psychology; J. Michael. Spector US Air Force Research Laboratory, San Antonio, TX) – instructional design; Robert D. Tennyson (University of Minnesota) –ISD.

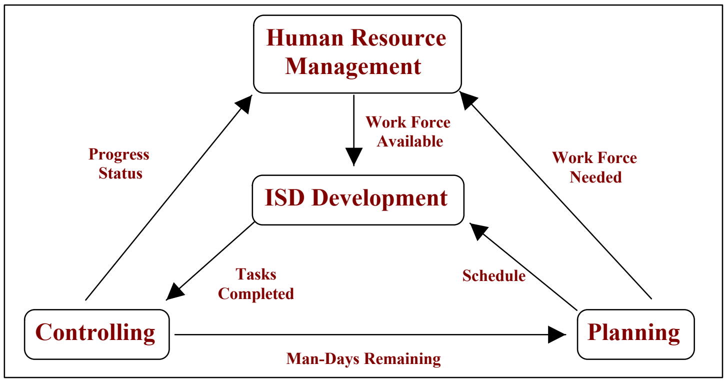

Figure 2. Sector s structure of the ISD model

(Abdel-Hamid & Madnick, 1990).

THE STRUCTURE OF THE MODEL

The first two phases are often the most crucial ones for a project in terms of ending up within the time and budget constraints. Therefore, our modeling process has been concentrated around the analysis and design phases. Specifically, our attention is focusing on the most common ways in which activities in these two phases are interrelated. They appear critical to resource allocation planning and to getting a project started on a productive path. Project phases are not entirely discrete, but they can be generally described in terms of activities or work carried out by various project personnel. These activities are often described as tasks and considered a concrete piece of work or a specific activity with an associated work effort required for completion. Tasks, then, are identifiable and describable, having more or less well-defined scope, and are usually measured in terms of person-days from the point of view of resource allocation.

The model consists of four basic sectors that describe key aspects of the structure of the project: (1) human resource management; (2) control; (3) plans; and, (4) development. A large part of the model is based on the work by Abdel-Hamid and Madnick (1990) applied to the modeling of the software project development. The structures that have been used in the ISD model are those of the team productivity, motivation and exhaustion of work force and of the control of the project. Figure 2 depicts the ISD model's major sectors with mane interrelationships between each of the sectors.

ISD development can be described as the accomplishment of analysis and design phases of the project. Tasks can be completed correctly (successfully) or incorrectly (unsuccessfully). Unlike the model by Abdel-Hamid and Madnick, the identification process of tasks is being modeled here explicitly. Identification of new tasks is a non-linear process dependent on the progress of the project as tasks are interdependent (concurrent) (Ford, 1995). The diagram portrayed in Figure 3 represents the ISD development process (workflow) inside and between the analysis and design phases.

Failed tasks are typically recognized as such and reworked. The tasks to be reworked can circulate inside a phase as well as between phases. The failure rate depends on the experience of the team, workforce composition and the project know-how. The failure of a task means the failure of a concrete piece of work or activity and means that it must be completed again. This is another difference from the Abdel-Hamid & Madnick software project model where required rework was described by the errors in software code and rework was dependent on the density of these errors.

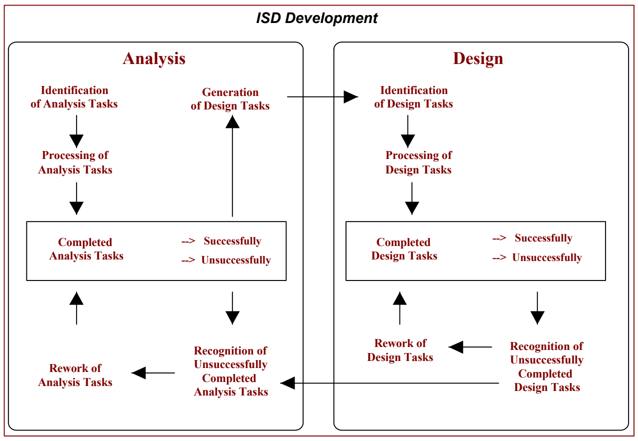

Figure 3. The development of instructional system (ISD). Interrelation of analysis and design phases.

The success of rework in our formulation is non-linearly dependent on the iteration process of tasks and the project progress. That is, the more number of times tasks are repeated (reworked) and/or the further the project has moved towards its completion, the more likely is it that such tasks will be completed successfully.

The sector that models the human resource management consists of two sub-sectors: (1) the allocation of people; and, (2) the sector of hiring and transfer of people. The first sub-sector simulates the allocation of people on different types of job based on the availability of workforce and resource requirements in different phases. The second sub-sector simulates the necessary hiring or transfer of people into or out of the project. A simplified structure of human resource management is portrayed in Figure 4. The decisions for hiring or transfers are based on the workforce sought in order to complete the project on time. That information comes from the control and planning sectors. The workforce is differentiated in two types: experts and novices. As a consequence of different types of workforce, the productivity of an average person in the project team will be lower than the potential productivity of the team (Steiner, 1972; Abdel-Hamid and Madnick, 1989).

In the planning sector, the schedule for the project is made. Reports about work force needed for human resource management are completed. Shifts of schedule can occur when a project is perceived to be behind or ahead of schedule. Another way to try to put projects back on schedule is to add more people to the team. But the side effects of that policy is that new people cause the average productivity to fall due to their lack of experience, and increased requirements for communication and training.

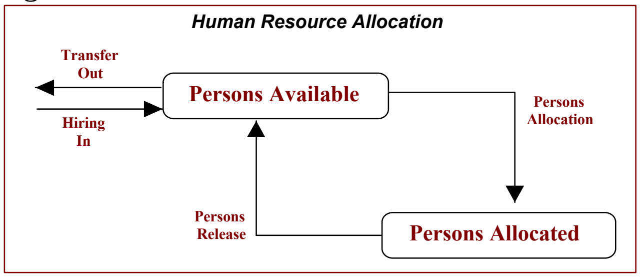

Figure 4. Human resource management: hiring in, transfer out, allocation and release of workforce.

IDENTIFICATION OF TASKS

Tasks for processing are being identified from the number of unidentified tasks. After identification, they flow into the tasks identified for processing. The identifiable tasks remaining and the associated identification time determine the rate of identification of analysis tasks. The total number of tasks is an exogenous input – a parameter that characterizes a particular project. This number is unknown to the manager of the project, who has only some estimate of how many tasks there will be in the project. The structure of the identification of tasks for analysis is portrayed in Figure 5.

The tasks identified for processing are being assigned to the workforce upon the initiation of their processing. This process is dependent on the workforce allocation. The allocation of people is determined by the resource requirements, which are based on the number of tasks identified pending processing and the number of tasks in processing. The actual allocation of workforce can be constrained by the availability of people in the resource pool. Therefore, the initiation process is dependent on the way the resources are being managed in the project and how the tasks are prioritized. A similar structure represents the identification of tasks in design taking into account tasks that are being turned back for re-analysis.

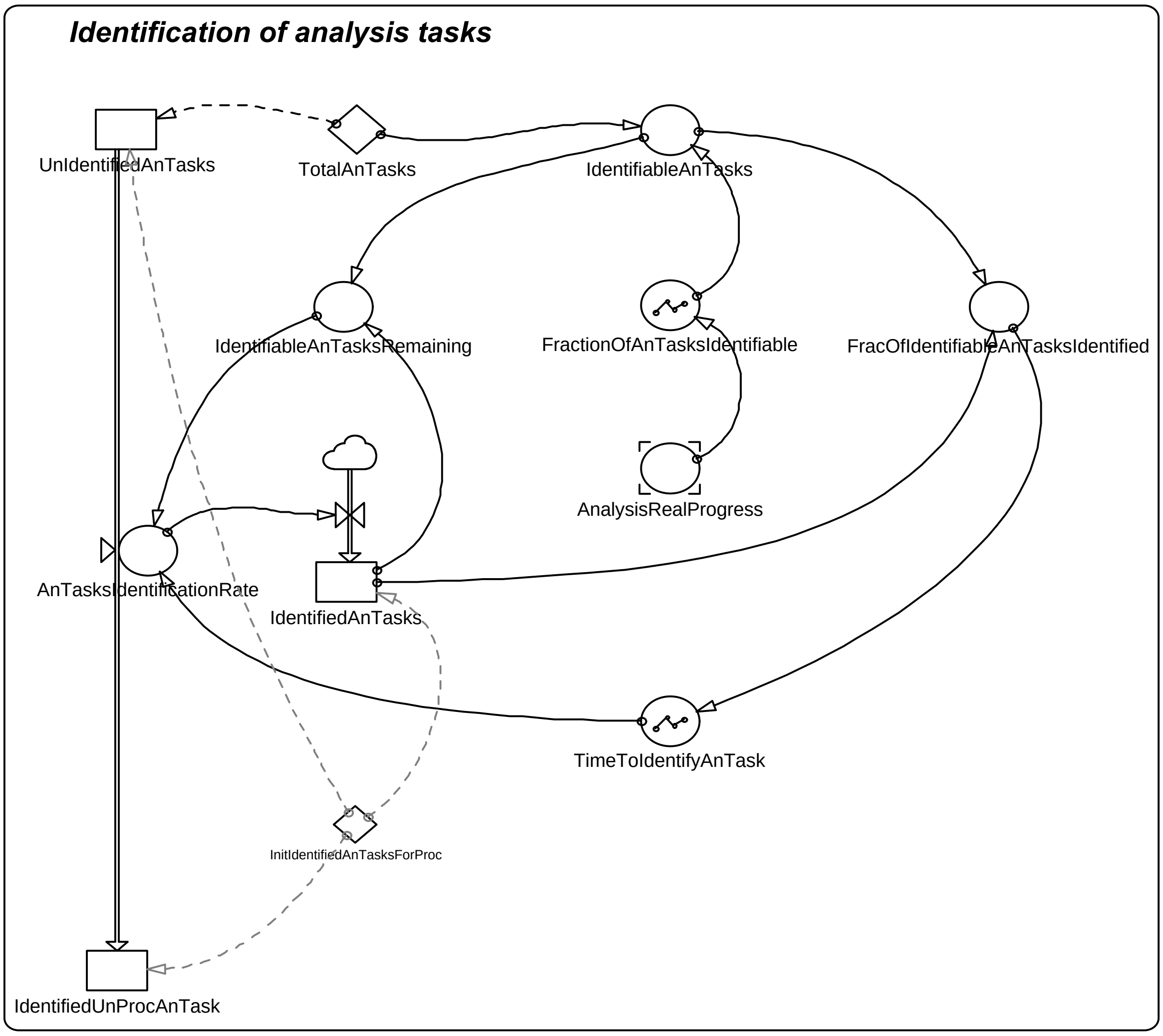

Figure 5. Identification of analysis tasks.

SIMULATION RESULTS

We use an "example" of a project for the demonstration of model behavior. The "example" project is described by 1000 tasks in analysis and 1000 tasks in design. The human resource requirements for analysis and design tasks are chosen to be 10 person-days/task, which translates into the 0.1 task/person-day of nominal productivity, by an expert.

At the start of the project, management has to decide how much labor and time to allocate. In ISD projects there are no established techniques for the estimation of the job size, and, therefore, the estimation is based on experience. Usually the size of the project is underestimated. In the model, the underestimation is chosen to be 40%. That is, the job size is estimated to 600 tasks in analysis and 600 tasks in design (as opposed to 1000 tasks in analysis and 1000 design tasks). The estimation of the job size of the project in person-days is then based on this number of tasks and the assumed productivity of an average employee. Initially, the project is scheduled to be completed on day 500 after the start of the project. This schedule is overrun for two main reasons. As the project progresses, undiscovered tasks become discovered and the size of the project is adjusted. Moreover the value of the assumed productivity is adjusted according to perceived results from processing.

Figure 6 depicts the model's base case exhibiting the key measures of the progress: (1) perceived job size in person-days, (2) total person-days expended in the project, (3) scheduled completion date of the project, and (3) total team size on the project. As the project progresses, the scope of the effort becomes more visible; the perceived job size is adjusted, leading to revisions of the schedule. The overestimation of the real size is caused by underestimation of the perception of the development productivity. That is, management looking back at what has been done and at the productivity that was experienced tends not to believe that people can in fact work efficiently.

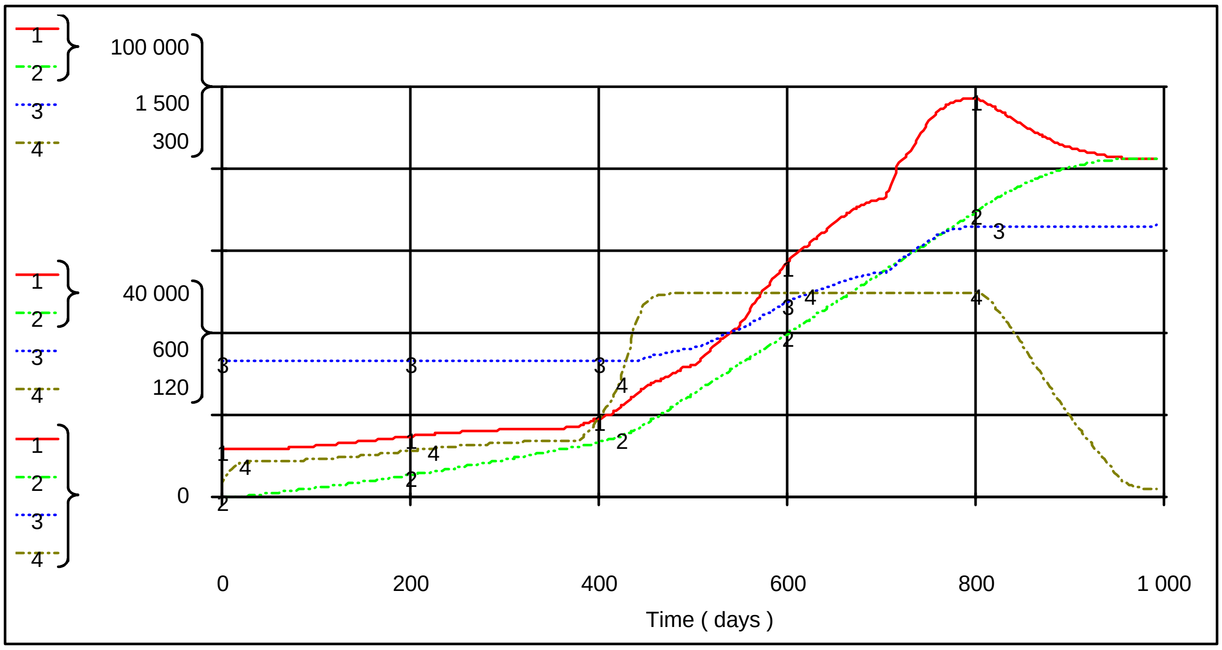

Figure 6. Simulated example of a project: an overview. 1) Perceived job size in person-days; 2) Total person-days expended; 3) Scheduled completion date; 4) Team size on the project.

When the project is perceived to be late, the first attempt to bring it back on schedule is made by adding more people to the project. The team size increases until some constraint is reached (in the model it is set to 150 persons). When that is the case, the schedule is adjusted upwards.

The overall development of the project is characterized by three major stages. In the first stage, at the very beginning of the project everything goes according to the plan. The cumulative person-days are expended approximately linearly. The schedule remains stable and only adjustments due to the discovery of new tasks, not perceived initially, are made. The second stage starts around day 400, when the project approaches the initially scheduled completion date. It is realized that the work to be done is still very high and more people are being hired into the project. The schedule is adjusted only 30 days later after the increase of the team size of the project. The slope of the expenditures curve increases, reflecting the increase in total workforce from approximately 60 persons at the initial stage up to 150 persons. The third stage of the project starts around day 800. At that moment, project schedule becomes stable again. The perceived job size, overestimated at that moment due to the perceived losses in productivity, is now being revised and adjusted downwards towards the real size. The total workforce is being lowered as people are transferred out of the project.

Figure 7 depicts the dynamic behavior of the tasks that require processing or rework in both phases. At the initial stage of the project, the analysis has almost no tasks that need to be reworked. The rework is represented by dark gray colors in both phases. As the design starts to process tasks, the failed tasks are being recognized and those that failed in analysis are turned back for re-analysis. These rework tasks appear in analysis towards the end of the project.

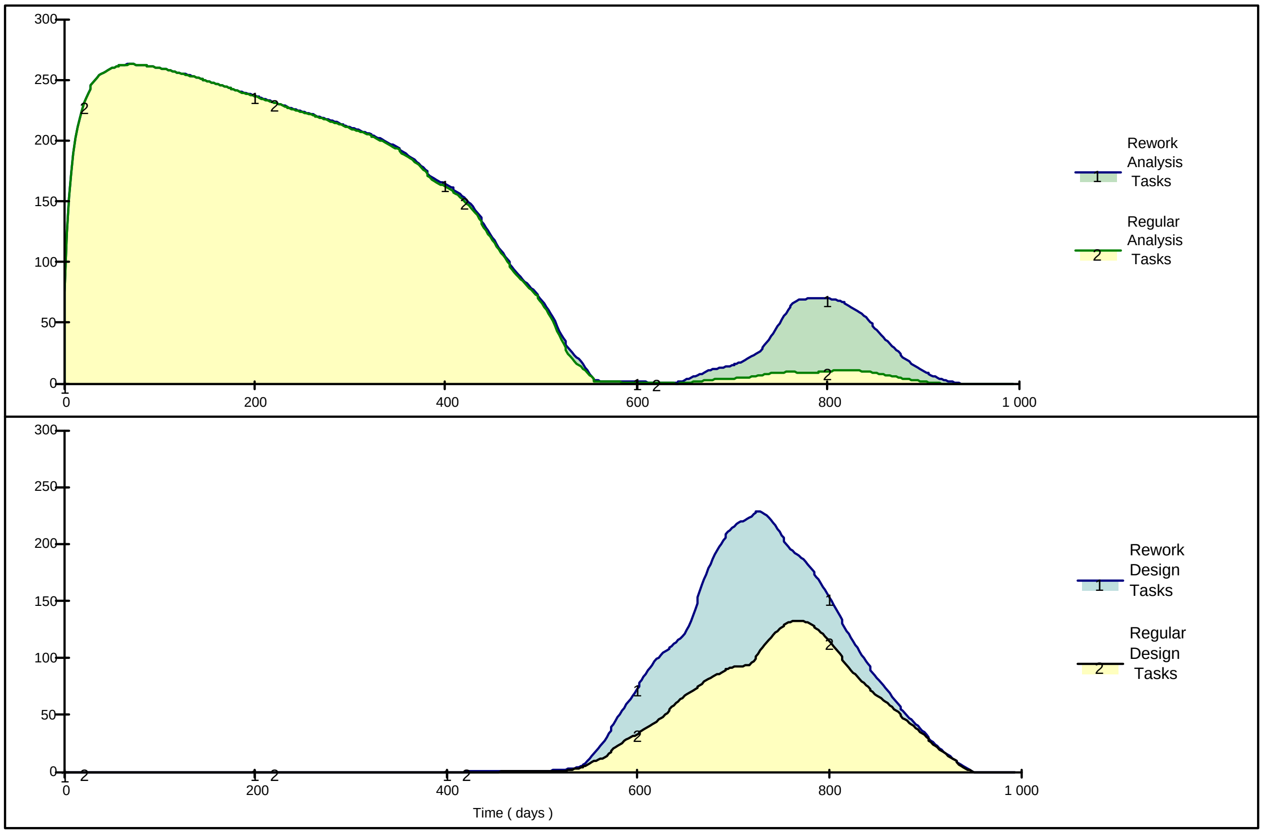

Figure 7. Tasks in analysis that require processing or rework upper panel. Tasks in design that require processing or rework lower panel.

The design contains a large percentage of rework to be done almost throughout the whole phase. Rework dominates at the very beginning and in the middle of the design process, and only when the analysis has fixed its failures, the regular design tasks take over leading to the completion of the design part of the project. The more resources can be allocated to design, the faster rework in analysis and design is being discovered.

Figure 8 depicts the overall task dynamics including unsuccessful processing and delays in the process. In analysis darker colors represent rework. In design, darker colors correspond to the rework and failures caused by analysis.

The failed tasks in analysis accumulate as unrecognized analysis tasks unsuccessfully processed. Most of these tasks flow into design and generate design tasks that eventually will lead to the flaws in analysis being recognized and returned as analysis tasks unsuccessfully processed.

At the beginning of the project, identified analysis tasks are initiated for processing fast enough so that almost none of the tasks stack up. In the middle of the development, they peak twice because of the inability to allocate manpower due to the lack of workforce, as the need for manpower was underestimated because of the biased perception of the real productivity.

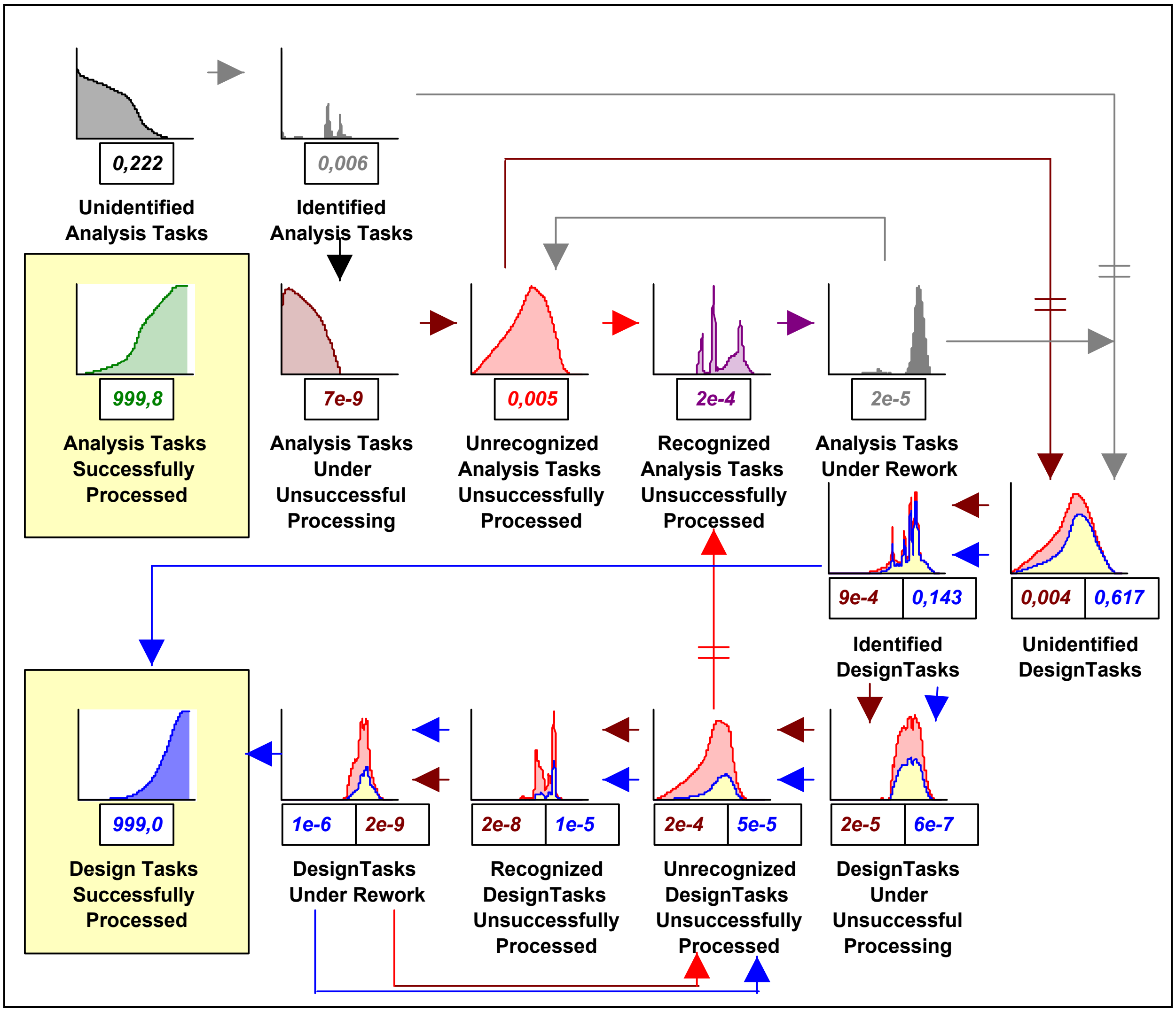

Figure 8. The overview of the tasks dynamics indicating major flaws and delays in the process.

The completed analysis tasks generate design tasks much faster than it is possible to identify these design tasks. The number of unidentified design tasks determines the rate of identification of these tasks. The more tasks there are, the easier it is to identify each new one. Therefore, the unidentified design tasks build up until there is a significant amount of them, so that they can be identified and thereafter assigned to the workforce for processing. As indicated, there are a lot of tasks that fail in analysis so that design cannot progress until these flaws are fixed in analysis rework.

The major source of rework in both phases and the delay in recognizing this rework is caused by the inability of analysis manpower to recognize their own failures fast enough so that many of these analyzed tasks generate design tasks deemed to fail. Therefore, one of the points where improvements can be made is by increasing the fraction of recognized unsuccessfully processed analysis tasks. Figure 9 depicts an example of a run where we assume that analysis failures are completely recognized by analysis itself before continuing into design. Now the rework in analysis occurs much earlier. The project completion time is better than in the base example, and the amount of rework in design is much less.

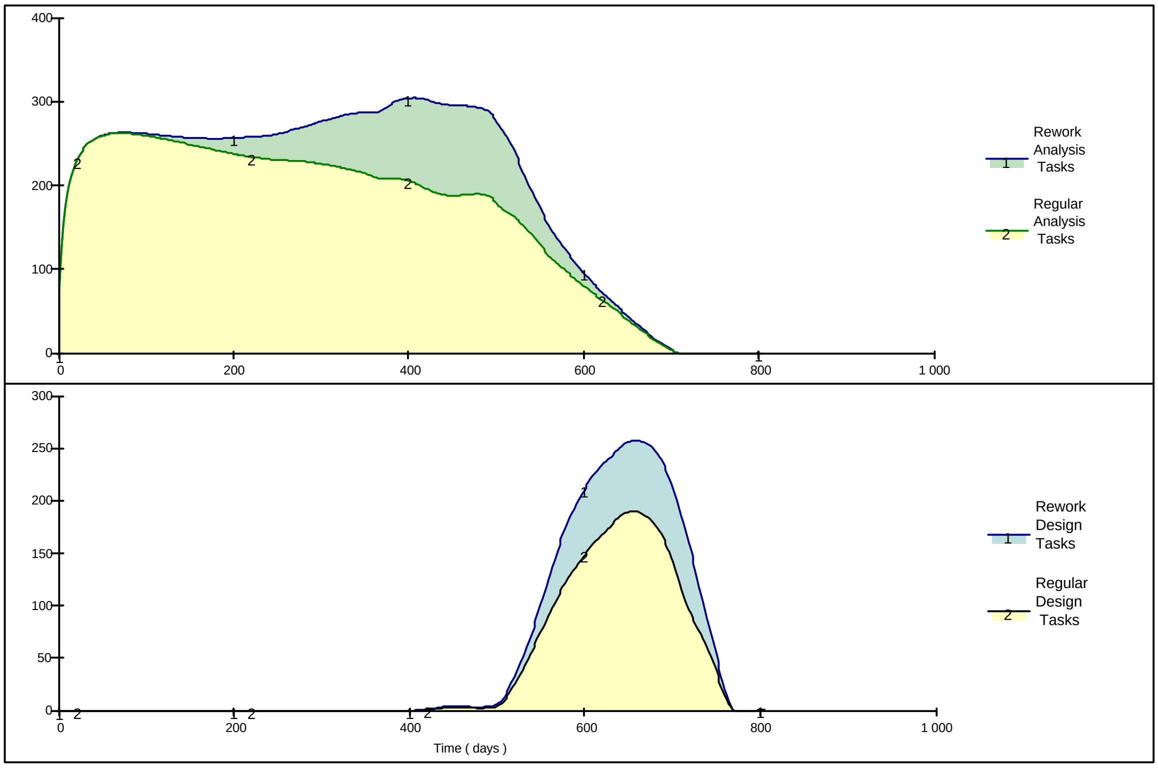

Figure 9. The simulation results under the assumptions that unsuccessfully processed analysis tasks are being recognized by analysis itself without flowing into the design phase.

CONCLUSIONS AND FUTURE INSIGHTS

Due to the widespread use of instructional systems and training programs, and their increased significance as well as the efforts associated with the development of such systems, such as distance learning, computer based training etc., the management of complex instructional design and development projects has become a critically important issue.

The objective of this paper has been to present a system dynamics model representing the structure of a generic ISD project development and management. Our model incorporates findings about the instructional development processes, and management. The situation has been simplified and our model does not take into account quality issues. In reality, the situation is more complicated since quality is a highly critical factor and would also have to be taken into consideration, minimizing costs while still delivering an acceptable end product on schedule. A more sophisticated goal could be to incorporate the quality assurance in the project structure.

The model can be used as an experimental simulation tool for understanding the complexities of such projects. The overall aim of the research in the field of instructional development is to educate the managers and members of ISD projects. The model, applied in an interactive learning environment, could provide a safe tool, which managers can use to experiment with and try out different policies and strategies in resource allocation.

The first step in that direction has been completed the spring of 98. A prototype for such a game has been developed, involving three actors. The roles to be played are Project Manager, Analysis Task Leader and Design Task Leader. Each of the actors is responsible for human resource allocation to the project and in different phases of the project. Further effort involves the completion of such a learning environment.

REFERENCES

1. Abdel-Hamid, T. K. and Madnick, S. E., "Software Project Dynamics, *an integrated approach*", Englewood Cliffs, N.J.: Prentice Hall, Inc., 1990.
2. Abdel-Hamid, T. K. and Madnick, S. E., "Software Productivity: Potential, Actual, and Perceived." *System Dynamics Review*, (Summer 1989).
3. Anthony , R. N. and Dearden, J., *Management Control Systems*. Chicago, IL: Richard D. Irwin, Inc., 1980.
4. Aron, J. D. Estimating Reasons for Large Programming Systems, In *Software Engineering: Concepts and Techniques*, ed. J. M. Buxton, P. Naur, and B. Randell Litton Educational Publishing, 1976.
5. Barber, R. L. *Software Reflected*, New York: North Holland, 1982.
6. Brooks, F. P., *The Mythical Man-Month*. Reading, Mass.: Addison Wesley, 1978.
7. Boehm, B.W., *Software Engineering Economics*. Englewood Cliffs, N.J.: Prentice Hall, Inc., 1981.
8. Davidsen, P. I. "Implementing Elements of system dynamics approach to organizational learning", *Proceedings of ESS 94, SCS*. Ghent, Belgium (March 1994).
9. Ferdinand, A. E., *Systems, Software, and Quality Engineering, Applying Defect Behaviour Theory to Programming* Van Nostrand Reinhold, N.Y., 1993.
10. Ford. David N. The Dynamics of Project Management: An Investigation of the Impacts of Project Process and Coordination on Performance. Doctoral thesis, Massachusetts Institute of Technology, Cambridge, MA, 1995.
11. Daly, E.B., "Management Of Software Development." *IEEE Transactions on Software Engineering*, (May 1977).
12. DePree, R.W. The Long and Short of Schedules, Datamation, June 15, 1984.
13. Forrester, J. W., *Industrial Dynamics*. Cambridge, MA: The MIT Press, 1961.
14. Grimstad Group: Report of the Grimstad Group. In R. D. Tennyson (Ed.), *Automating Instructional Design: Computer based development and delivery tools*. Berlin: Springer-Verlag, (1995).
15. Mills, H. D., Software Development, *IEEE Transactions on Software Engineering*, SE-2 Dec. 1976, pp 265-273.
16. Reigeluth, C. M., *Instructional Design Theories and models: an Overview of their Current Status*, Lawrence Erlbaum Associates, Publishers, Hillsdale, New Jersey, London, 1983.
17. Richardson, G. P. and Pugh, G. L. III., *Introduction to System Dynamics Modelling and DYNAMO*. Cambridge, MA: The M. I. T. Press, 1981.
18. Senge, P.M. "*The Fifth Discipline*", Doubleday Currency, N.Y., 1990.
19. Steiner, I.D., 1972. "Group Process and Productivity." New York: Academic Press.
20. Tennyson, R. D. A framework for automating instructional design. In J.M. Spector, M.C. Polson, & D. J. Muraida (Eds.), *Automating Instructional Design: Concepts and Issues*. Englewood Cliffs, NJ: Educational Technology,1993.
21. Weinberg, G. M., *Understanding the Professional Programmer*, Boston: Little, Brown. 1982.