AUTOMATING THE TRAINING DEVELOPMENT PROCESS

Carol J. Scott
Member of Technical Staff
Jet Propulsion Laboratory, California Institute of Technology
Pasadena, California

Abstract

Automating the task of [raining development al ways sounded like. a great idea, and recentl y, a system was developed that actually dots just that. Antiquated course development methods may begin to fade with the introduct ion of the Automated Training Development System (ATDS) Copyright ©1993, California Institute of Technology. Not only dots it automate curriculum development and generate instructional materials, it nourishes the information gathering process and makes a case for cl im inating the need for preliminary training requirements.

1. Introduction

Most jobs require some kind of training to bring ncw employees to an acceptable stale of productivity; however, many of these jobs do not require formal training. On-the-job instruction by experienced personnel is often as productive and less costly. Formal training is likely to be available only where large sy.stems or critical operations arc involved, and even then, on-the-job instruction still plays an important role. in the learning experience. So, in looking at the role of formal training, wc have alread y established some parameters for development: training has a scope and training is usually supplemented by on-the-job instruction.

Training priori ties arc driven by a training developer's perception of customer need balanced by the ability to respond. Customers arc a diverse set, ranging from upper management to the newest recruits awaiting "basic" training. Although customer training needs arc traditionally expressed as a set of training requirements, they frequentl y do not reflect the scope or complexity of their operational environment. When used to develop training courses, the results can be less than adequate.

In the past, trainers have dealt with the problem of incomplete requirements by accepting them, then enhancing thcm within the development process in hopes of producing an acceptable product, With a diverse clientele it is difficult to determine course goals that satisfy the needs of each employee, consequently course material is often modified to adapt to the occasion or group of students. Requirements remain the primary method in determining criteria for course development.

2. The Legend of Training Requirements

The trainer's primary responsibility is to the customer, i.e.., the learning community and its management. Often the representative tasked to produce training requirements for the customer has difficulty obtaining them. There arc cases where the customer is not familiar enough with the system for which training is to occur, to describe its functionality, much less identify the training needs required in learning to operate it. For example, a large project may hire an entire team of system operators in preparation for a system delivery.

Trainers can't refuse to offer training to customers simpl y because they arc unable to provide training requirements, and trainers arc often faced with making their own assumptions regarding course content, bawd upon personal observations and previous experiences. '1'here's nothing wrong with doing that, except that it perpetuates them yth that usable requirements have actually been received and that they arc of value in determining ultimate course development goals.

Keep in mind that developing a set of training courses to satisfy existing customer requirements is not necessarily the same thing as developing training to meet the customer's needs. In fact, the moment training developers commit to developing training around customer requirements they arc building obsolescence into the training products and limiting their ability to respond to changing needs.

3. An Innovative Alternative

An alternative would be to have pre-developed training available, designed at the task level to meet nearly every operational scenario. A database of training opportunities would provide a catalog of training options from which the customer could select individual tasks, complex procedures, or end-to-end operations according to current need.

Predeveloped training would require early access to the system the customer is using, but once developed, would only require updates for ncw capabilities or system modifications,

4. ATDS: A Development Tool

The operations System Training Group at the Jet Propulsion 1 aboratory (JPL) recentl y completed a study to assess the adaptability of the Automated Training Development System (ATDS) for usc in its training environment. ATDS was developed as a training tool for the. JPL's All Source Analysis System effort and is based

on the standard for military training programs (MIL-STD-1379D.) This tool takes a new look at the training development process and promotes the idea of developing training from a system perspective, focusing on components in terms of the whole process.

Former training philosophies focused on the individual course as the development product, It did not provide motivation for a trainer to become a system expert, in fact, the trainer wasn't expected to know anything about system functionality except what was in the customer's requirements.

ATDS is a building process that allows a trainer to dissect a process, analyze all the components, and put it back together step-by-slcp in the form of a database. One significant aspect of ATDS is that the resulting database is the primary product of training development. Customer input is not required up-front. In fact, the training developer can learn the system, define all the tasks of system functionality, create and populate an entire database before i nterfacing with the customer. When the time. for interaction does arrive, the training developer is already familiar with all aspects of system functionality and can assist the customer in making informed decisions about training course content.

A Paradigm Shift

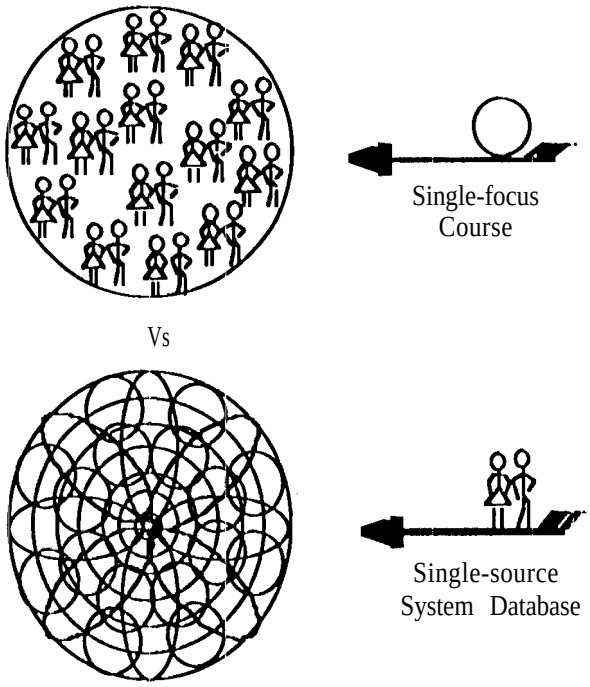


Figure 1. linking users to appropriate training material.

Figure 1 compares the effectiveness of developing a training database of all system tasks as opposed to developing single-focus courses. The training database contains everything nccdcd to generate al I courses, whereas single-focus courses may require massive cut-and-paste efforts, or total redevelopment in order to upgrade them or convert thcm for mull iple users.

ATDS attracted our attention because of its automated material generation capabilities and a promise of efficient course material maintenance, The system provides a means for establishing a single-source, structured training system baseline, where thousands of pieces of information can be correlated. ATDS provides manageability, as well as maintainability, and will be especiall y significant to those who have already agonized over the training development process, course. by course.

4.1 ATDS: An Instructional System Development Methodology

ATDS provides a repository for the building blocks of a process, as well as all details as they become available. In learning ATDS, you clear your mind of thoughts about course content and concentrate on the information gathering process from the global perspective, as in a picture puzzle, where pieces fit to transition each pattern seamlessly to the next.

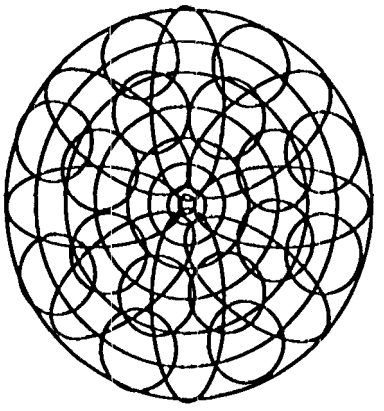
'1'here arc five major phases to Instructional System Development: Analysis, Development, Design, Presentation, and Reporting. Each phase is functionally represented by ATDS.

4.2 Analysis

A "task" is a subset of a job. It is composed of individual skills, supported by knowledge elements and sub-skills that must be completed in order to accomplish the task. Even though several employees may perform some of the same tasks, it is the unique nature of a set of tasks that defines each job. Figure 2 represents all the tasks for which you could possibly develop training material.

You begin the Analysis Phase by listing all possible tasks that could be performed within the scope of the system. All operational system elements can be pursued as potential candidates for training and this list of candidate tasks is the starling point for building your database. This is an excellent opportunity to accomplish a dual purpose: 1) become proficient in system operation, and 2) analyze each task in terms of skill and knowledge for entry into the database. Each of the candidate tasks may have individual prerequisites of competency that one must have prior to beginning the task; these prerequisites arc called "entry skills and knowledge."

ANALYSIS



Everything you could ever want to know about your subject

a.k.a.

(Capturing your own learning experiences for posterity)

*Figure 2. **Analysis Phase in the ATDS instructional Design Method***

4.3 Development

ATDS is a building process within each defined task that describes every competency of skill and knowledge required to complete the task. These competencies arc called "enabling skills and knowledge." It is the level of detail in this skill and knowledge hierarchy within each task that determines the overall quality of your development efforts. Figure 3 represents the development of competencies within each task.

A training developer can enter any number of additional pieces of information relating to each skill or knowledge as required, such as:

- narratives describing each skill or knowledge,
- graphics associated with selected skills or knowledge
- instructor notes to guide the trainer through each skill or knowledge
- instruction sheets including
  - information sheets
  - assignment sheets
  - problem sheets
  - job sheets
  - outline sheets
  - diagram sheets
- instructor references 10 additional documentation
- equipment require.mcnls
- test questions
- performance exercises

Other information may bc entered at the task level:

- learning objectives
- training objectives
- course run time and other statistical information

DEVELOPMENT

task
S, S, S, K,
K, K, S, K

task
S, K, K, S,
K, K, S, K

task
K, K, K, S,
K, S, S, K

task
K, S, K, S,
S, S, K, K

task
S, K, S, S,
S, K, S, K

task
S, S, K, S,
K, K, K, K

task
S, S, S, S,
S, K, K, S

task
S, K, S, K,
S, K, S, K

task
K, K, K, K,
K, S, S, K

•
•
•

Comprehensive Database Construction

Figure 3. Development Phase in the ATDS Instructional Design Method

Still more information can bc addressed during the course. generation process.

The result of the Development Phase is a complete baseline, of training material. Once the baseline is complete, maintenance is simple. Maintenance is always performed on the current working version of the database.

4.4 Design

The Design Phase is the point in the process that welcomes customer requirements. Training material has been developed and stored in the database; the training developer can now select appropriate tasks to included in each course as indicated by user survey or customer requirements.

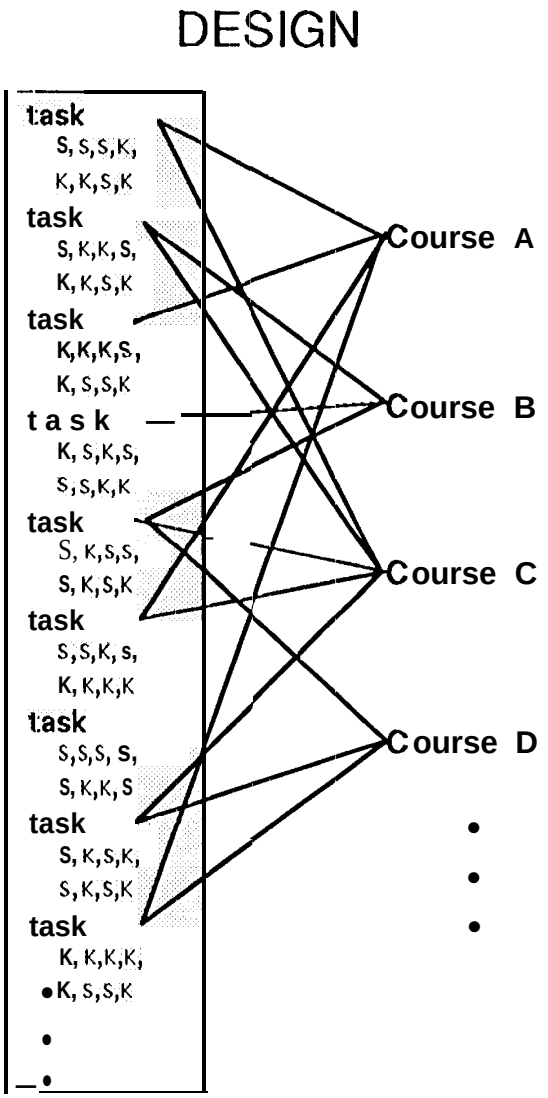


Figure 4. Design Phase in the ATDS Instructional Design Method

To enable the customer to submit valid requirements, a published catalog of tasks by subject as well as by position, function, team, etc., will allow informed selection and communicate their needs in common terms. As you build courses, you can save lists of tasks that define specific positions and publish them in the catalog as examples of training requirements by position. This will eliminate many grueling hours of labor on the part of future customers. Being able to check off subject areas of known need is indeed a humane way to deal with an often disagreeable process.

An ATDS course is simply a collection of those tasks within your subject database that are critical to a specific job or group of jobs. In other words, a course is made up of all the critical tasks required 10 perform a job. Figure 4 represents the Design Phase of the process.

For positions where subject matter is common to large groups of learners, ATDS provides a Task Selection Tool for identifying course requirements. The training developer surveys a large group of customers, and a weighted model assesses their responses and selects tasks identified as critical.

Once the subject matter has been established, course criteria can be entered into ATDS and the document generator will produce output products, in file format. These files can be verified prior to printing, and where necessary, minor modifications may be made using an editor. Each file can be sent directly to a printer.

4.5 Presentation

The Presentation Phase of ATDS focuses on the material aides generated. Material is provided for both instructor-led and self-paced training. Options for course generation allow the training developer to select the medium. Figure 5 describes some of the ATDS-generated training materials.

With ATDS products, the level of detail is consistent throughout the material. Every facet of the process being trained is documented and scripted for instructor USC. It is easy to remain focused on the material being presented.

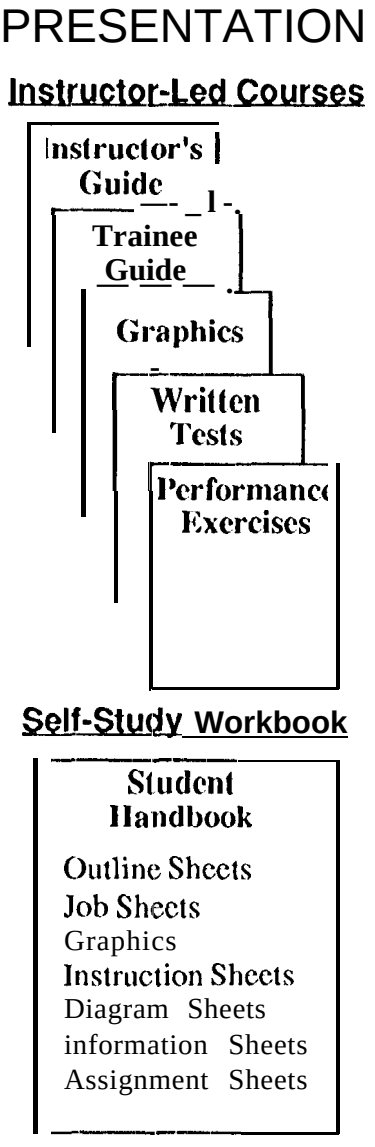


Figure S, Presentation Phase in the ATDS in.wruclional Design Method.

4.6 Reporting

A student record function stores information on tasks completed by each student. This information can be compiled for groups of students and track individual performance.

5. Catching Up to the Power Curve

One problem for training developers in a custom software environment is that we are generally funded at the end of the software development cycle. When you 're not in the starting gale when the race begins, you can't be expected to cross the finish line with the system. [1]

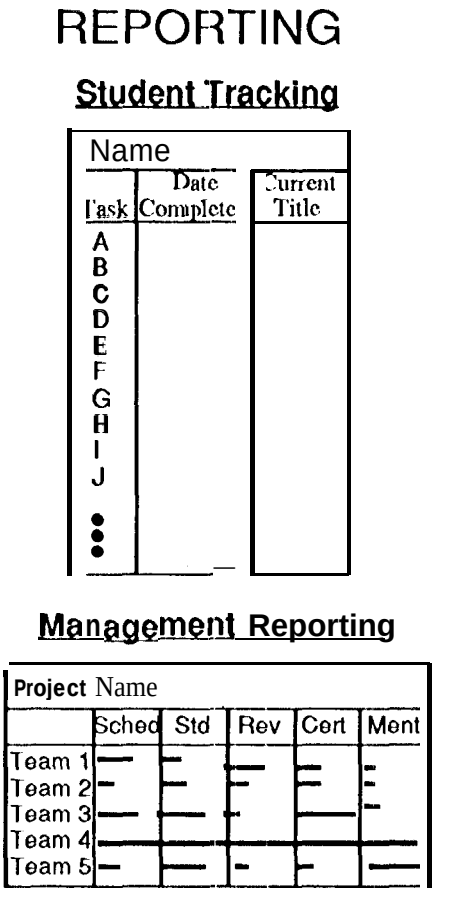


Figure 6. Reporting Phase in the ATDS Instructional Design Method.

Now imagine developing training routines for every aspect of a system, as a system is being built. One argument against this idea might be, "there are many tasks that may never need to be addressed in a training class so why waste lime generating the material." Good point! However, what happens when the development team moves on and second or third generation developers are hired 10 maintain and upgrade the system? In the case of software development, there would be a definite advantage to preserving the global knowledge of system functionality. It is not unheard of to throw out a software system and start over because no one can figure out what went into the original design or how it was intended to grow, and a new development effort forges ahead. This could be a case for preserving the system's design heritage through careful, up-front training development of all aspects of its functionality. Information would no longer be lied to individuals, or lost over time as personnel movement takes place.

A realistic goal would be for training developers to capture their own learning process using ATDS. ATDS will preserve every piece of information associated with a task, whether or not it is destined for one of the output products. Once every aspect of system functionality has been captured in tbc ATDS database, updating it as the

system changes is a simple task. The goal, then, is to accumulate a baseline set of training tasks that represents total system functionality. Generating course material is simply a matter of selecting the right set of tasks for specific users. If the first course lacks something, simply select a revised list of tasks.

6. Conclusion

One of the most important features of ATDS is database maintainability. Once you have your material baselined, maintaining it becomes a trivial task. Everything is contained and maintained within the database, and if properly developed, could be a training component of a software delivery and provided to CM as a controlled item.

The ATDS program is a tedious collection process, but its output products will reflect the diligence used in analyzing the subject matter and filling the database with adequate, detail. ATDS requires a significant commitment of time and effort on the part of a training developer; the reward is knowing it only has to be done once.

ATDS will be most cost effective when evaluated over time. Initial training development will probably be a lengthy process; its value will be realized in its quality, flexibility, and maintainability.

The research described in this paper was carried out by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

[1] Scott, C. J., 1992. Training: Plugging the Information Gaps. Proceedings of the Second International Symposium on Ground Data Systems for Space. Mission Operations (SPACEOPS '92), November 16-20, Pasadena, CA,