A PROPOSED RESOURCE MODEL FOR SOFTWARE PROJECT MEASUREMENT ENVIRONMENTS*

D. Ross Jeffery[1] & Victor R. Basili[2]

[1] Department of Information Systems
University of New South Wales
P.O. Box 1, Kensington NSW 2033

[2] Department of Computer Science
University of Maryland
College Park, MD. 20742

ABSTRACT

This paper presents conceptual process and data models of resource planning and consumption for the software development process. Such conceptual models are a prerequisite to the development of process management oriented software engineering environments. The data model is derived from a generalized software process model and consists of a four dimensional view of resource data. This view aims to encompass the required data for resource planning, evaluation and control. The conceptual data model can be used as a basis for an integrated software engineering environment or as a basis for informal management of the software process.

* This research was funded in part by the National Aeronautical and Space Administration Grant NSG-5123 to The University of Maryland, and the Special Studies Program of The University of New South Wales.

1. INTRODUCTION

This paper presents a conceptual process model of resource consumption during software development, and a conceptual data model of the resources consumed. These models, particularly the data model, are a pre-requisite to the development of an integrated project support or measurement environment which aims to capture resource data for the purposes of project planning, evaluation, or control. The models presented in the paper are a part of the TAME (Tailoring A Measurement Environment) project [see Basili, Rombach 87] being carried out at The University of Maryland. The TAME project aims to develop an Integrated Project Support Environment concerned with measurement of the software process and product.

The conceptual data model proposed is a four dimensional view of resource data which has been validated against the published literature on software resource databases [see Jeffery, Basili 88]. The resource dimensions embodied in the model are:

1. Resource type - the nature of the resource consumed
2. Resource use - descriptors and measurement of the resource usage
3. Resource Incurrence - whether estimated or actual values, and
4. Resource availability - a measure of the viewpoint adopted in considering the use of the resource

The paper presents an overview of both the resource process and data models along with guidelines for their use in the project management environment.

2. RESOURCE DATA AND PROCESS MODELS

The most extensive work carried out to date on the establishment of the data which needs to be collected to manage the software process and product is the STARS project Proposed Baseline Software Measurement Data Item Descriptions [STARS84]. The approach adopted in the STARS project has been a bottom-up process of defining data capture forms which can be used during the software process. The data deemed desirable is embodied in the many forms designed. This approach contrasts with that at TRW Inc. where they have defined a database schema for process and product data (see [Penedo, Stuckle 85]). This schema was developed basically top down and does omit some data items which many would consider necessary for effective project management (see [Jeffery, Basili 88]). Another project which impacts the definition of a project database is the work of Abdel-Hamid [Abdel-Hamid, Madnick 86]. In this work the dynamic nature of software projects is emphasized and the complex data interrelationships which result is outlined. For example they make the very valid point that the process of estimating project time will have an impact on the software process that results, and that we need to consider this interrelationship in project control. Tausworthe [Tausworthe 79] also defines some of the necessary components of a software project database in his work, but as in the case of the STARS project, the data items are detailed in data capture forms rather than in any conceptual or pyhsical database model. It is significant that in the development of these prior models that the ANSI/SPARC model for the development of database models has not been applied. In all cases, with the possible exception of the TRW work [Penedo, Stuckle 85], the conceptual data and process models which lie behind the data item definitions and lists have not been reported.

This research has taken a top-down approach to the development of the database schema by defining a conceptual project process model which describes the nature of the software project, and based on this a conceptual project data model which can now be used (1) to develop physical data models in formal project tracking and reporting systems, and (2) as a basis for informal project management systems.

3. THE PROJECT ENVIRONMENT

Resources are consumed during the software process in order to deliver a software product. The software process has overall characteristics which need to be established in order to place the resource consumption in context. This characterization includes data on factors such as:

- project type
- organizational development conventions
- project manager preferences
- target computer system
- development computer system
- project schedules or milestones
- project budget
- project deliverables

In this data the broad project characteristics and the environment of the project are established. For example, is the process using evolutionary development or a waterfall method? Is the project to be developed by in-house staff or external contractors? What organizational constraints are being imposed on the project development time? What management constraints are being imposed on staffing levels?

These factors form the environment in which the software process will occur and determine in many ways, the nature of that software process. A simple example of this is the question of the process model - evolutionary or waterfall. This constraint establishes milestones and the pattern of resource use, and therefore partially determines the interpretation of the resource data collected.

An overall model of the software project is shown in Figure 1. In this figure the entity project is decomposed into a number of tasks or contracts, each task consuming the entity resource and producing the entity product. In the implementation of this model each of these entities will require many entities to characterize them. Thus the project and its environment have characteristics, as do the tasks and subtasks, the resources, and the products.

4. DEFINITION OF TERMS

In this paper several terms are introduced to describe aspects of resource consumption. These terms are:

- . desirable resources: Those resources deemed to be "ideal" for the project.
- . accessible resources: Those resources potentially available for allocation to the project
- . utilized resources: Those resources actually used or expected to be actually used on the project.

5. THE RESOURCE PROCESS MODEL

The software process can be described from a resource perspective as an interacting three-stage process involving the sub-processes of:

1. PLANNING
2. ACTUALIZATION, AND
3. REVIEW.

The planning process establishes and records the resource expectations or estimates before and during the software project. The actualization process tracks and records the actual use of resources during the software project, and the review process compares actuals with estimates for the purposes of (1) modifying the estimates and (2) learning from the divergence of actual from plan. This review process may also lead to a change in the actual data captured where it is learnt that additional or changed measures would be beneficial.

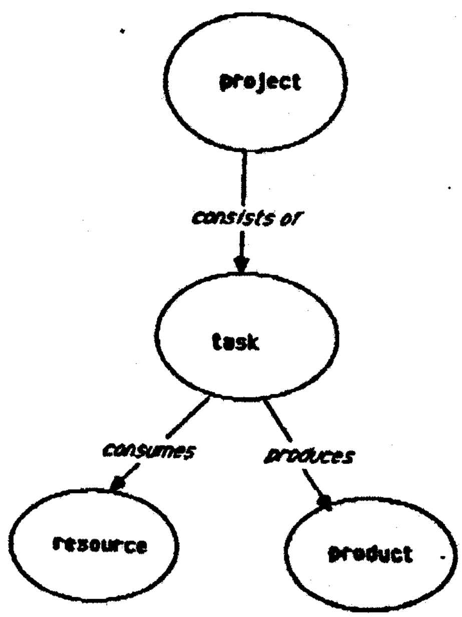

FIGURE 1. AN OVERVIEW MODEL OF THE SOFTWARE PROJECT

5.1 THE PROJECT CYCLE

Figure 2 shows the data flows, logical data stores, and processes involved in planning, actualization, and review. This data flow diagram shows three types of estimates being made; desirable, accessible, and utilized. The desirable resources are estimated (in process 1) by the project estimator on the basis of information concerning the project and the environment in which the project is to occur along with any project histories and/or knowledgebase which may be available. The accessible resources for this project are estimated in process 2, again by the estimator, using the desirable resources and the corporate resource database as input along with any history and/or knowledgebase information. The corporate resource database lists and describes the resources which can be called on by the organization along with any commitments for those resources.

The differences between the desirable and accessible resources form a significant database in the process of risk management, since this database reveals those aspects of risk resulting from decisions to develop the system with something less than, or different to the resources considered desirable.

The process of detailed project planning then continues in process 3, using the project accessible resource database as one of the inputs to this process, and generating the project resource plan which contains details of the estimated utilized resources. "Estimated utilized" resources are those which it is anticipated actually will be used on the project. Once again the difference between the outputs has meaning. Just as the difference between desirable and accessible represents a database resource

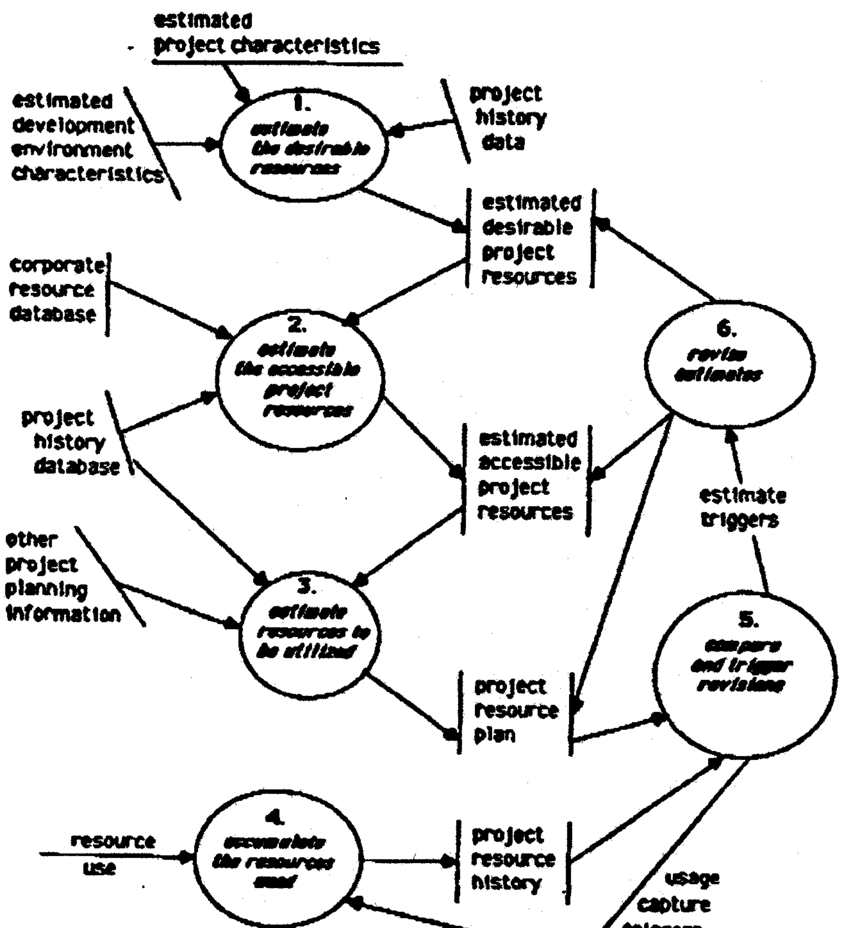

FIGURE 2. THE PROJECT CYCLE

for risk management, the difference between accessible and utilized forms a basis for contingency planning. Resources can be "committed" in two ways. The first is when an available corporate resource is both accessible and utilized. In this case the resource can be considered as a hard commitment to the particular project. Contingency plans are also permitted in this system where corporate resources are available for the project but rather than entering the utilized list they are entered as a contingent commitment to a particular project. In that way the planning process can allow contingency planning for individual projects and for corporate resources as a whole.

As the software process continues, resources actually utilized are accumulated via process 4, resulting in the project resource history data store.

The actual utilized data is used in process 5 to monitor the estimates and the usage data capture and thereby facilitate project control. If major divergences between estimates and actuals occur, this may trigger re-estimation of the resources.

5.2 THE POST-PROJECT REVIEW CYCLE

Figure 3 shows an overview of the use of the proposed structure in project reviews. The data accumulated during the project are used to review the project and generate learning based on the experience with the project. This new data consists of:

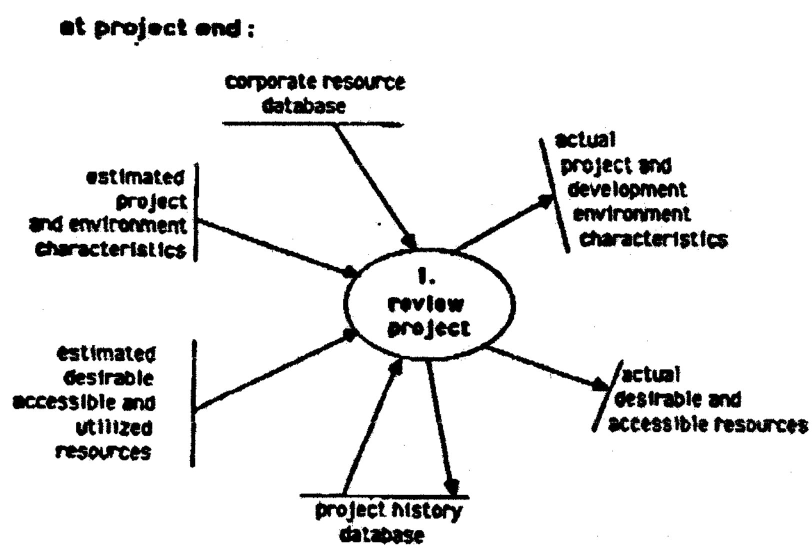

FIGURE 3. THE PROJECT REVIEW CYCLE

1. additions to the project history and knowledge databases

2. changes to the perceptions of the project and its environment, based on comparisons between estimates and actuals. For example, the comparison between actual and estimated milestones may reveal that shorter milestones were possible, or that desirable milestones should have been set longer. By making these comparisons it is possible to establish a project learning environment.

3. changes to the perceptions of desirable and accessible resources. Again learning is facilitated by the comparison of estimated and actual. In this way it may be learnt, for example, that aspects of the algorithmic effort estimation equations consistently over-estimated for this project. The reasons for this can be explored and, if necessary, adjustments made to the algorithm.

5.3 APPLICATION OF THE PLANNING AND REVIEW CYCLES

In any particular organization, it may be deemed sufficient to use only a part of the planning and review processes outlined here, and therefore only a part of the structure presented in this paper.

For example organizations may not wish to use project reviews, or they may not consider it appropriate to carry out formal contingency planning or risk management. At the simplest level only the estimated utilized and the actual utilized may be used, perhaps providing input to an informal project learning process which occurs at the individual level.

Specifically, it is most likely that in software environments with very little uncertainty (say an implementation of the nth slightly different version of a well known system in a stable development environment) there may be no need to explicitly consider the desirable or even accessible dimensions of the resource model. If uncertainty is very low, the utilized level of the model may capture all the necessary data. The advantage of the model in this case is that the data excluded from consideration is done so in the knowledge that there is no real information in this data.

In higher uncertainty environments, the model prompts the estimator to think explicitly of the resource risks and uncertainty of the development process, and to quantify or express that risk as a part of the resource database.

6. THE RESOURCE DATA MODEL

At the level below the characterization of the project and its environment we are interested in classifying the resources consumed in the generation of the software product. In this section of the paper we present a structure for that classification. This structure covers only the resource aspect of the project and is therefore only concerned with the software process and the resources consumed or used in the process. The model is not concerned with the software product.

Based on the software resource process model presented in section 5, the structure of the data model is four dimensional. These four dimensions are divided into two segments:

1. resource type, and
2. resource use

The two segments being separated are (1) the nature and characteristics of the resource, and (2) the manner in which we look at or consider the consumption of that resource.

6.1 RESOURCE TYPE

In the first segment we are concerned with classifying the nature of the resource; is it someone's time, dollars, or a physical object such as a computer, or a logical object such as a piece of software? We are also interested in describing the properties of those resources such as description, model number, and cost per unit of consumption.

By decomposing the resources into different types different views of the resources can be provided. For example, it may be important for operations personnel to know a breakdown of the hardware resources used on a project according to the different physical machines being used, whereas from a project manager's perspective at a point in time, the specific machine may not be of interest, but the availability of a certain class of machine may be critical. Resource managers will be interested in the types of resources available (for example, people) and the characteristics of those resources for project planning purposes. Thus the categorization provided here is the basis of the resource management environment, in that it is in this segment of the model that the resources are listed and described.

The resources of a software project can be classified as:

.hardware
.software
.human
.support (for example: supplies, materials, communications facility costs.)

These categories are intended to be mutually exclusive and exhaustive and therefore are able to contain each instance of resource data in one or other of the categories.

Hardware resources encompass all equipment used or potentially able to be used in the environment under consideration. (For example, target and development machines, terminals, work stations).

Software resources encompass all previously existing programs and software systems used or potentially able to be used in the environment under consideration. (For example, compilers, operating systems, utility routines, previously existing application software). This also includes previous project documentation, such as requirements, design and code which is suitable for reuse in future projects.

Human resources encompass all the people used or potentially able to be used for development, operations, and maintenance in the environment under consideration.

Support resources encompass all of the additional facilities such as materials, communications, and supplies which are used or potentially able to be used in the environment under consideration.

The values associated with these resources may be stored in both price and volume measures, where volume means, for example, hours of use or availability, or the number of times a resource is needed, and price refers to the $ values associated with that resource. This may be a cost per unit measure or a cost per period of time.

6.2 RESOURCE USE

The categorization of use within this dimension allows the resources consumed to be associated with different perspectives of the software process. For example, it is through this use structure that we are able to distinguish:

- between prior-project expectations of consumption and resources actually consumed, or
- between resources consumed in each phase of the project, or
- between the utilization of a resource and the availability of that resource, or
- between an ideal view of resource planning and the resources actually available

The use structure proposed is based on the process model of section 3 and consists of:

1. INCURRENCE
   - 1.1 Estimated
   - 1.2 Actual

2. AVAILABILITY
   - 2.1 Desirable
   - 2.2 Accessible
   - 2.3 Utilized

3. USE DESCRIPTORS
   - 3.1 Work type
   - 3.2 Point in Time
   - 3.3 Resources Utilized

6.2.1 INCURRENCE

This category differentiates between estimated and actual to allow the resource information to be gathered and used in a manner suitable to the management of the resource. It is necessary, for example, to store data on estimated resource usage, resource requirements, and resource availability. This data is distinguished from the actual resource incurrence or use, which is stored via the actual category.

These two categories then permit process tracking via comparisons between them and extrapolation from the actual data. At the project summary points, explanations and defined data accumulations on estimated and actual resource use provide feedback on the process. This feedback should contain reasons for variance between the estimated and actual so that a facility for corporate memory can be established and the necessary data stored to facilitate and explain any updates of the current resource values. It needs to be noted that the model proposed allows for different estimates and actuals at different points in time.

This structure requires that process data, as it changes in value during the project, will not be lost but will be stored in an accessible manner so that meaningful analysis of projects can be carried out using a database that provides complete details of the project history.

This philosophy specifically addresses the need for a corporate memory concerning past projects. By implementing such a structured project log the basic data for such a memory is available in numeric and text format.

6.2.2 AVAILABILITY

This category allows storage of a resource use by:

.desirable
.accessible
.utilized

These terms are defined in section 5 above. This categorization provides further refinement of the resource data. Through this, and say the incurrence category, it is possible to compare the actual resources utilized with the estimated utilization, and then trace possible reasons for variance through the desirable and accessible dimensions. For example, differences between planned availability and actual availability of a resource may be significant in understanding the software resource utilization that occurred during the process.

As outlined in section 5, the difference between desirable and accessible is those resources seen as desirable for the project but which were not available for use during the project. This difference may occur, for example, because of budget constraints or inability to recruit staff. The desirable resource list permits an "ideal" planning view. When compared with accessible it allows management to see any compromises that were made in establishing the project, thus facilitating a very explicit basis for risk management within the resource database. The database is thereby able to hold views of not only the resources actually applied to the project but also those resources which were considered to be desirable along with the reasons for their use or non-use. In this way the resource trade-offs are made explicit.

As in section 5, the difference between accessible and utilized represents those resources available for the project but not used. This difference will arise because of three possible reasons:

1. The resources prove to be inappropriate for the project under consideration, or
2. The resources are appropriate but they are excess to those needed
3. The resources are appropriate, and their use is contingent on an uncertain future event.

The use of these storage categories is explained in detail further below in section 6.3. Through this availability category we are able to distinguish between:

(1) the resources which are reasonably expected to be beneficial to the process (desirable),

(2) the resources which exist in the organization and are able to be used if needed (accessible), and

(3) the resources which are used in a project (utilized)

Through this categorization it is then possible to track resource usage and to pinpoint their use or non-use and to ascribe reasons particularly to their non-use as in the case of non-accessibility. As in the INCURRENCE category, the reasons for divergence between desirable, accessible, and utilized are stored in a feedback facility.

6.2.3 USE DESCRIPTORS

This category provides a description of the consumption of the resource item in terms of three essential characteristics of the consumption of that item:

1. The Nature of the Work being done by the resource: (e.g. coding, inspecting, or designing) This category can be used in conjunction with other views to distinguish between process activities, such as human resources estimated to be desirable in design work, or machine resources actually utilized in testing, or elapsed time implications of inspections.

2. Point in Calendar Time: This category pinpoints the resource item by calendar time. In this way resource items (estimated or actual; desirable, accessible, or utilized) are associated with a specific point in time or period of time. This facilitates tracing of time dependent relationships and the comparison of resource values over time.

3. Resources Utilized: This category measures the extent of resource consumption in terms of hours, dollars, units, or whatever is the appropriate measure of use.

The Use Descriptors also provide the link to the work breakdown structure which is commonly embodied in process models. This link is established through the association of a particular piece of work being done at a point in time with the work package described in the work breakdown structure. Thus broadly speaking, in a project database environment the resource is associated with a task which produces a product. A detailed treatment of the work breakdown structure is contained in [Tausworthe 79] and the resource relationships with tasks are outlined in [Penedo, Stuckle 85].

In validating the model developed here both [Tausworthe 79] and [Penedo, Stuckle 85] have been used to ensure that the conceptual data model is able to encompass all of the resource data suggested as necessary by these authors. In both cases it was found [Jeffery, Basili 88] that the model proposed satisfied in this respect, but also included the possibility of capturing data additional to that suggested by them.

6.3 COMBINING THE VIEWS

The structure suggested here can be viewed as a hierarchy for the purpose of explanation. Such a hierarchy is shown in Figure 4. In this figure we see that the proposed structure views the software project (which has attributes describing that project) consuming resources. The resources are characterized as having four dimensions of interest (type, use, incurrence, and availability). At the resource type level we describe each resource as being one of hardware, software, human, or support, and having various attributes. The attributes for each of these four types will be different in nature. For example, the human attributes might include name, address, organizational unit, skills, pay rate, unit cost, age, and so forth. The attributes for hardware will be quite different, describing manufacturer, purchase date, memory capacity, network connections, or similar types of characteristics.

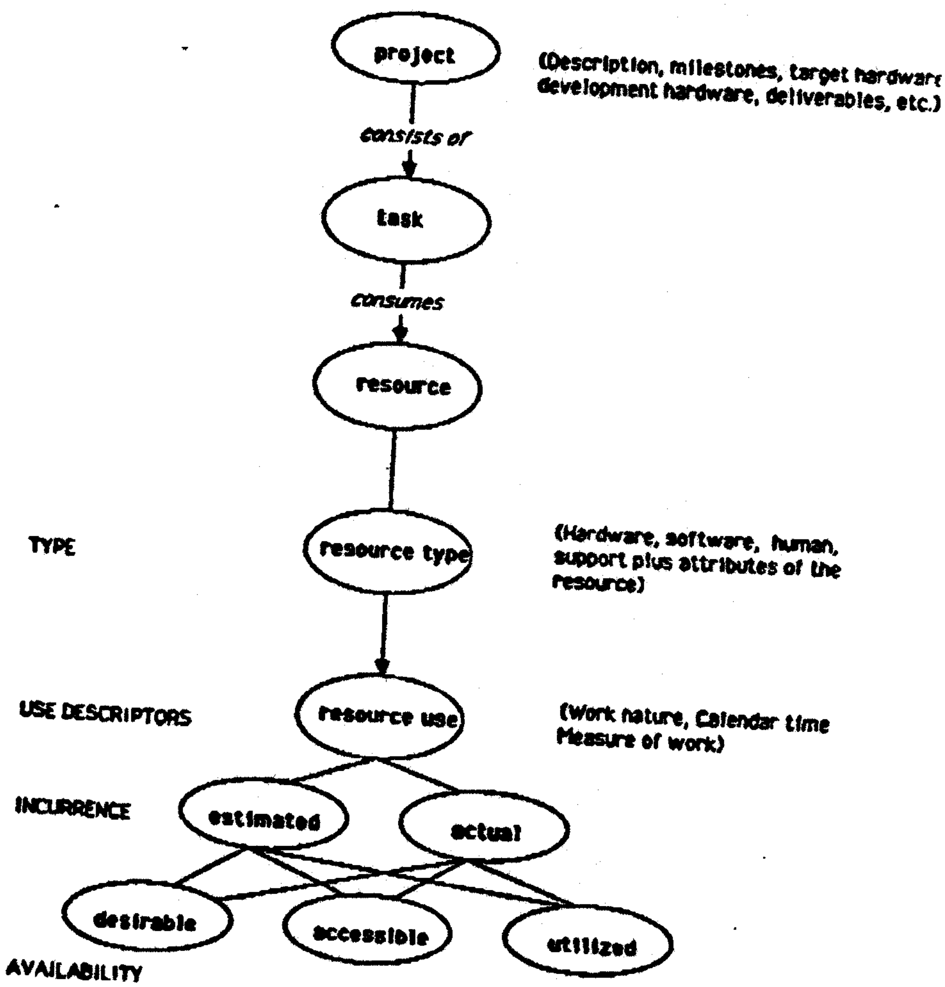

FIGURE 4. THE STRUCTURE OF THE TDC MODEL

At the next level in the diagram we model the use of the resource. In the first instance this involves the type of work that the resource is performing, the point (or span) in calendar time at which the work is being done, and the measure of the amount of work done. This last measure (amount of work) might be expressed in person-time, execution-time, connect-time, or whatever is the relevant measure of work for the resource instance.

The use of the resource is then described as being either estimated or actual, and both of these may be desirable, accessible, or utilized. In this way the following concepts are supported:

1. Estimated Desirable: The resources considered "ideal" at various stages of the planning process.

2. Estimated Accessible: The resources which are expected to be available for use in the process, given the constraints imposed on the software process (a contingency plan).

3. Estimated Utilized: The resources which it is anticipated will be used in the software process.

4. Actual Desirable: With hindsight, the resources which proved to be the "ideal" considering the events that occurred in the software process (a part of the learning process).

5. Actual Accessible: Again with hindsight, the resources which were actually available and could have been utilized (a part of the learning process).

6. Actual Utilized: The resources actually used in the software process.

Categories one through three are used initially for planning purposes. The numeric and text values associated with each of these three categories may be derived from:

a. individual or group knowledge
b. algorithmic models
c. a database of prior projects, and/or
d. a knowledge base

At the very simplest level, the planning process might establish only numeric values in the estimated utilized category based on individual knowledge alone. In essence, this is the only form of estimation used in many organizations, wherein project schedules and budgets are established by an individual, based on that individuals experience. These estimates represent the expected project and resource characteristics for the duration of the project.

In a large number of organizations algorithmic models are also used for estimation of resource utilization. In some cases the algorithms include uncertainty parameters allowing for contingency planning.

The use of databases of prior projects is less often used, although more common in larger organizations than smaller. These databases contain both numeric and nonnumeric data, although the full facilities envisaged here for corporate memory are not supported.

The use of knowledge bases is the least common, TAME incorporating some aspects, but it is suggested that through this facility the ISEE will be able to provide significant benefit, particularly in higher uncertainty projects.

The extensions suggested here allow these estimates to be enlarged in the following dimensions:

The nature of the estimate
The source of the estimates
The timing of the estimates

1. The nature of the estimate. The model allows project and resource managers to distinguish between desirable, accessible, and utilized estimates as discussed above. The estimated desirable dimension would be used at a fairly high level in the project planning process to outline the hardware, software, people, and support resources that are considered to be desirable for the project. This may list specific pieces of hardware and software which are desirable at certain points in time. It might also be used to list characteristics of the people (such as skills) that would be ideal on the project. The accessible dimension would then reflect the expected resources that will actually be available to be used. Again this could be at a fairly high level, indicating the resources available, the differences between these and those desirable, and the reasons why the two categories do not agree; reflecting cost constraints, or risk attitudes which have been adopted as part of the project management profile. The utilized category would normally extend to a lower level in terms of the project plan, detailing estimated resources perhaps down to the work package level and short periods of time.

2. The source of the estimates. It was suggested above that there are four major possible sources for these estimates; individuals or groups of people, a knowledge base, a database of prior projects, and algorithmic models of the process. Each of these should be supported in a measurement environment, and each has significant implications with respect to the design of such an environment. The current state of the art appears well equipped to support algorithmic models of some parts of the estimation process (for example, estimates of project effort based on one of the many available estimation packages such as COCOMO [Boehm 81], SLIM [Putnam 81], SPQR [Jones 86]). Similarly the tools available in the database environment allow the storage and retrieval of numeric data on past projects. However the storage and searching of text data on prior projects, the use of a knowledge base, and the support of group decision support processes are all the subject of current research (see for example, [Bernstein 87], [Nunamaker, et.al. 86], [Barstow 87], [Valett 87]).

3. The timing of the estimates. In the structure suggested, all estimates may be made before the commencement of the software process and also at any point in time during the process. However there are certain points in time during the process at which estimates are more likely to be updated. These are:

1. at project milestones
2. at manager initiated points in time at which major divergence between estimate and actual is recognized by the manager

3. at system initiated points in time at which the measurement system recognizes a potentially significant divergence between estimate and actual

The third possibility implies that the measurement system is able to intelligently recognize the existence of a problem with respect to the comparison of actual and estimate.

Categories four (actual desirable) and five (actual accessible) of the structure exist to provide a feedback and learning dimension to the project database. These values would be determined after the project is complete. And in the comparison of the estimates made at various stages of the process and these two categories, a process is facilitated in which the organization can learn based on the variance of expectations and actual which have occurred in the past projects. As with the estimates, the categories of desirable and accessible are used in order to allow the comparison of "actual ideal" with "actual available" so that an ex-post view of the management of the process can be captured. The question being asked here is: "How could we have handled resources better?" It is a learning mechanism to generate explicit new knowledge for the knowledge and data bases, and also to improve individual and group knowledge.

7. CONCLUSIONS

The research has proposed an intuitive resource data model which is intended to support software project management. This support may be in the form of an integrated software engineering environment, or as the basis for informal process management.

The resource data model proposed consists of four dimensions:

1. resource type
2. resource use
3. incurrence, and
4. availability

8. REFERENCES

[ABDEL-HAMID, MADNICK 86]
T.K.ABDEL-HAMID, S.E.MADNICK, "IMPACT OF SCHEDULE ESTIMATION ON SOFTWARE PROJECT BEHAVIOR," IEEE SOFTWARE, 3,4, JUNE 1986, PP 70-75

[BARSTOW 87]
D. BARSTOW, "ARTIFICIAL INTELLIGENCE AND SOFTWARE ENGINEERING," PROC. 9TH INTN'L. CONF. ON S'WARE.ENG. IEEE, MONTEREY, APRIL, 1987, PP.200-211.

[BASILI, ROMBACH 87]
V.R.BASILI, H.D.ROMBACH, "TAILORING THE SOFTWARE PROCESS TO PROJECT GOALS AND ENVIRONMENTS," PROC. 9TH INTN'L. CONF. ON S'WARE ENG. MONTEREY, APRIL, 1987, PP. 345-357.

[BERNSTEIN 87]
P.A.BERNSTEIN, "DATABASE SYSTEM SUPPORT FOR SOFTWARE ENGINEERING," PROC. 9TH INTN'L. CONF. ON S'WARE. ENG., MONTEREY, APRIL, 1987, PP. 166-178.

[BOEHM 81]
B.W.BOEHM, SOFTWARE ENGINEERING ECONOMICS, PRENTICE-HALL ENGLEWOOD CLIFFS, NEW JERSEY, 1981.

[JEFFERY, BASILI 88]
D.R.JEFFERY, V.R.BASILI, "VALIDATING THE TAME RESOURCE DATA MODEL", PROC. 10TH INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING, SINGAPORE, APRIL, 1988.

[JONES 86]
T.C.JONES, SPQR/20 USER GUIDE V1.1, SOFTWARE PRODUCTIVITY RESEARCH INC. JANUARY, 1986.

[NUNAMAKER, APPLEGATE, KONSYNSKI 86]
J.F.NUNAMAKER, L.M.APPLEGATE, B.R.KONSYNSKI, "FACILITATING GROUP CREATIVITY: EXPERIENCE WITH A GROUP DECISION SUPPORT SYSTEM," PROC. 20TH ANNUAL HAWAII INTN'L. CONF. ON SYSTEM SCIENCES, HAWAII, JANUARY, 1987, PP.422-430.

[PENEDO, STUCKLE 85]
M.H.PENEDO, E.D.STUCKLE, "PMDB - A PROJECT MASTER DATABASE FOR SOFTWARE ENGINEERING ENVIRONMENTS," PROC. 8TH INTN'L. CONF. ON S'WARE. ENG., LONDON, AUGUST, 1985, PP. 150-157.

[PUTNAM 81]
L.H.PUTNAM, "SLIM A QUANTITATIVE TOOL FOR SOFTWARE COST AND SCHEDULE ESTIMATION," PROC. NBS/IEEE/ACM SOFTWARE TOOL FAIR, SAN DIEGO, CA, MARCH, 1981, PP. 49-57.

[TAUSWORTHE 79]
R.C.TAUSWORTHE, STANDARDIZED DEVELOPMENT OF COMPUTER SOFTWARE: PART II STANDARDS, PRENTICE-HALL, ENGLEWOOD CLIFFS, NEW JERSEY, 1979.

[VALETT 87]
J.D.VALETT, "THE DYNAMIC MANAGEMENT INFORMATION TOOL (DYNAMITE): ANALYSIS OF THE PROTOTYPE, REQUIREMENTS AND OPERATIONAL SCENARIOS," M.SC. THESIS UNIVERSITY OF MARYLAND, 1987.