N94- 36489

Software Forecasting As It Is Really Done: A Study of JPL Software Engineers

55-61
12687
p. 28

19940319812

Martha Ann Griesel
Jairus M. Hihn
Kristin J. Bruno
Thomas J. Fouser
Robert C. Tausworthe

Jet Propulsion Laboratory/California Institute of Technology
4800 Oak Grove Avenue
Pasadena, Ca. 91109

Abstract

This paper presents a summary of the results to date of a Jet Propulsion Laboratory internally funded research task to study the costing process and parameters used by internally recognized software cost estimating experts. Protocol Analysis and Markov process modeling were used to capture software engineer's forecasting mental models. While there is significant variation between the mental models that were studied, it was nevertheless possible to identify a core set of cost forecasting activities, and it was also found that the mental models cluster around three forecasting techniques. Further partitioning of the mental models revealed clustering of activities, that is very suggestive of a forecasting lifecyle. The different forecasting methods identified were based on the use of multiple-decomposition steps or multiple forecasting steps. The multiple forecasting steps involved either forecasting software size or an additional effort forecast. Virtually no subject used risk reduction steps in combination. The results of the analysis include: the identification of a core set of well defined costing activities, a proposed software forecasting life cycle, and the identification of several basic software forecasting mental models. The paper concludes with a discussion of the implications of the results for current individual and institutional practices.

1.0 Introduction

In today's cost constrained environment, cost estimation is becoming an integral part of the engineer's job. Therefore, tools and databases are needed that are consistent with engineering based costing methods. Previous surveys have shown that engineers in general do not use tools and databases, finding them inconsistent with their intuitive engineering-based costing methods, in particular analogy-related techniques. (Hihn and Habib-agahi, 1991) This lack of correspondence between software forecasting practices and available computer-based tools prompted the current research.

To be able to design and develop tools and databases that are more consistent with engineering-based costing methods requires that there exist a relatively small number of costing activities and that these activities are primarily used in a few well defined sequences. A sequence of activities is what makes up a costing method or, in cognitive psychology terminology, the cost forecaster's mental model. The existence of a small number of basic forecasting mental models requires that the mental models depend on high-level domain and environment conditions, rather than personal style and low-level domain details.

To the best of the authors' knowledge, there have been only three attempts to develop such mental models of the forecasting process that are documented in the literature: Vicinanza et. al. (1991), Howard (1992), and Hihn et. al. (1993).

Vicinanza et. al. completed an exploratory study of the methods used by experts. In Vicinanza et. al. five respondents who ranked a series of cost drivers and then estimated the development effort that would be required for 10 projects. The forecasters' methods were categorized into four groups: algorithmic initial condition, algorithmic effort estimate, analogical initial condition, and analogical effort estimate. For a method to be algorithmic the forecaster had to mention and use productivity figures. For a method to be analogical the forecaster had to mention a reference project. Four of the estimators used an algorithmic approach and only one used analogy. Vicinanza et. al. propose a logic flow (mental model) for algorithmic and analogical forecasting (see Figure 1 for the analogy model). Given their simple categorization scheme it is unclear how they derived their mental model. Also, the experimental design required that the engineers use COCOMO cost drivers (Boehm, 1981) and function point descriptors (Albrecht and Gaffney, 1983), neither of which may have been natural to them; and the terms used in the proposed mental models are neither goals nor the vocabulary that are commonly used by software engineers.

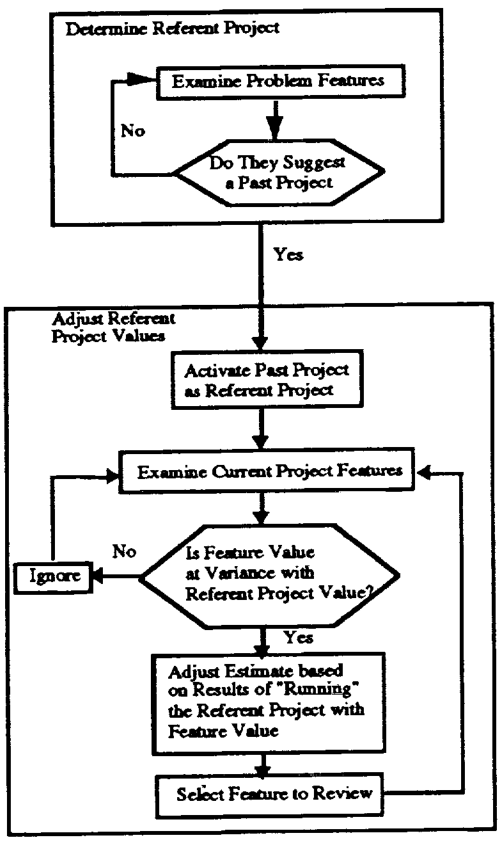


Figure 1 : Abstraction of Analogical-Estimation Strategy from Vicinanza et. al. (1991)

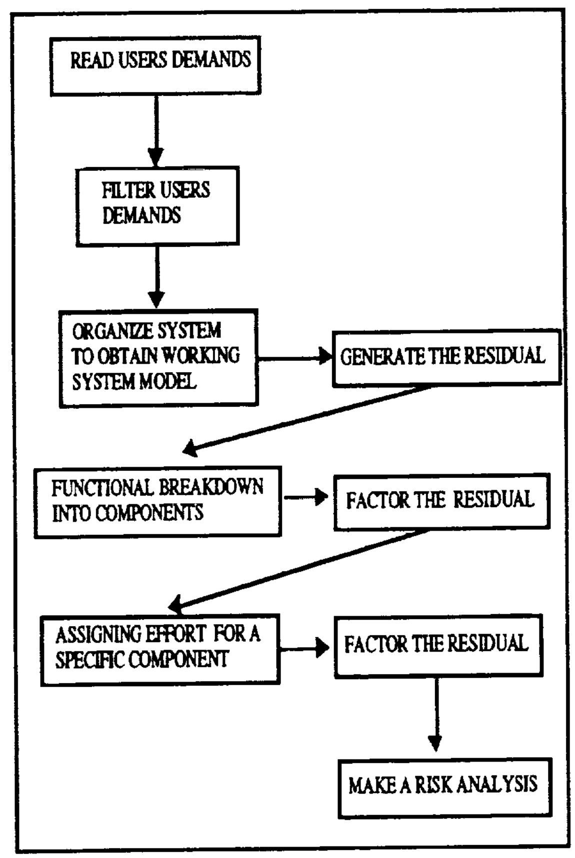


Figure 2 : A Bottom Up Approach to Estimation from Howard (1992)

Howard (1992) reports the results of two surveys on software cost estimation practices for standard information systems such as a banking transaction system. Approximately 50 observations were collected using a survey form. Twelve observations were collected using semi-structured face-to-face, interviews based on a case description, given to the subjects before the interview. The main objective of the research is to study how cost estimates are developed in group settings. The objective of the reported portion of the research task was to develop a mental model of the processing steps that estimators follow that could be used to support the study of group cost forecasting. A very high level model with about 20 possible steps based upon cognitive processing theories was proposed. Figure 2 illustrates the mental model of individuals applying the "bottom up" process. Interestingly, aggregation was never mentioned, even though "functional breakdown into components" is explicitly shown. The model proposed is intuitively appealing. However, the respondents provided quite generic responses in describing how they normally do cost estimating. Howard reports this is because the case example was found to be too poorly defined. Verbal reports of this type are well known to lead to biased, and very likely, inconsistent results [see Ericson and Simon, (1984)].

In both of the papers described above, the bases by which the proposed software forecasting mental models were derived is not explained. Howard followed some basic cognitive psychology techniques, but it was not clear that they were derived by a repeatable analysis. A significant problem, from the perspective of identifying a more detailed picture of the underlying mental model, was that most of what distinguishes an expert from a novice is in how they generate and "factor residuals" or, in other words, incorporate their cost drivers and adjustment factors.

Hihn et. al.(1993) attempt to address these problems by using a more precise data capture and analysis technique. In Hihn et. al.(1993) a combination of Protocol Analysis Ericson and Simon (1984) and Markov process modeling Papoulis (1991) is shown to be a viable technique for capturing the engineers' cost forecasting mental models in a repeatable manner. With this technique, Protocol Analysis was used to extract a common forecasting vocabulary across engineers and application domains by translating the engineers' self reports into verbal protocols, and Markov analysis was used to identify the common transitions, or steps, in the engineers' mental models. Seven primary cost forecasting activities were identified that clustered into 6 different, but not mutually exclusive, sequences (mental models) using this analysis technique. The 7 activities that were identified are requirements identification, attribute identification, attribute application, decomposition, estimation, aggregation, and adjustments. The definition of these terms are reviewed in Section 3.0. The original clusters of sequences were derived based upon purely data descriptive criteria. For example, a sequence that contains a single decomposition and single estimation activity is in a different sequence cluster then a sequence with multiple decomposition and multiple estimation activities. A very simplified example of the type of mental model this approach produces is displayed in Figure 3.

In this paper we are reporting an extension of these results that incorporates an increased number of cost forecasting activities and the identification of activity sequences (mental models) that correspond to software domain and development environment criteria. In addition, as part of identifying a number of basic mental models, it was possible to derive the components of a software cost forecasting life cycle based upon actual costing behavior.

2.0 Sample Definition and Institutional Background Information

Jet Propulsion Laboratory (JPL) is a Federally Funded Research and Development Center run by the California Institute of Technology under a government contract with National Aeronautics and Space Administration. As a national laboratory, it performs research and development activities in the national interest, primarily the development of robotic spacecraft for interplanetary studies. In addition, a portion of JPL's budget is supplied by non-NASA organizations such as the Department of Defense.

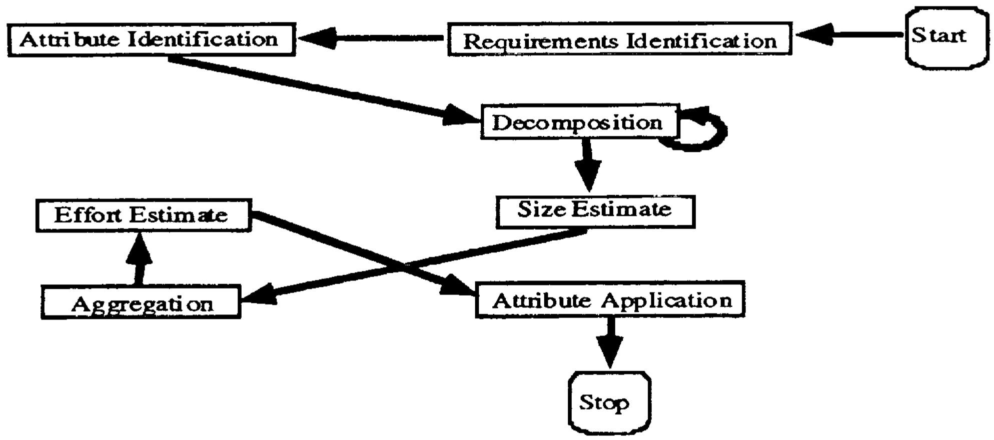


Figure 3 : Example of a CER-Based Forecasting Mental Model

A survey was conducted of the technical staff that had experience forecasting software development costs during the summer and fall of 1989. Over 185 software engineers were contacted for participation in the original survey. Of the 185 contacted, over 100 were identified who estimate effort, size and/or cost for software tasks. Of these, 83 were willing to complete a questionnaire on current software cost estimation practices. Of these, 28 responses provided sufficient information for use with the current analysis. For a detailed discussion of how the original data was collected see Hihn and Habib-agahi (1991).

The original purpose of the survey was to study the ability of software engineers to estimate effort and size given an architectural design document. In addition, the survey included a brief description of the typical approach each estimate used. The verbal protocols describing the cost forecasts used in the study were made during the system functional design and software requirements analysis phases (see Figure 4). Since data collected in this manner is not strictly appropriate for Protocol Analysis, conclusions drawn from this secondary analysis of the data may be questionable.

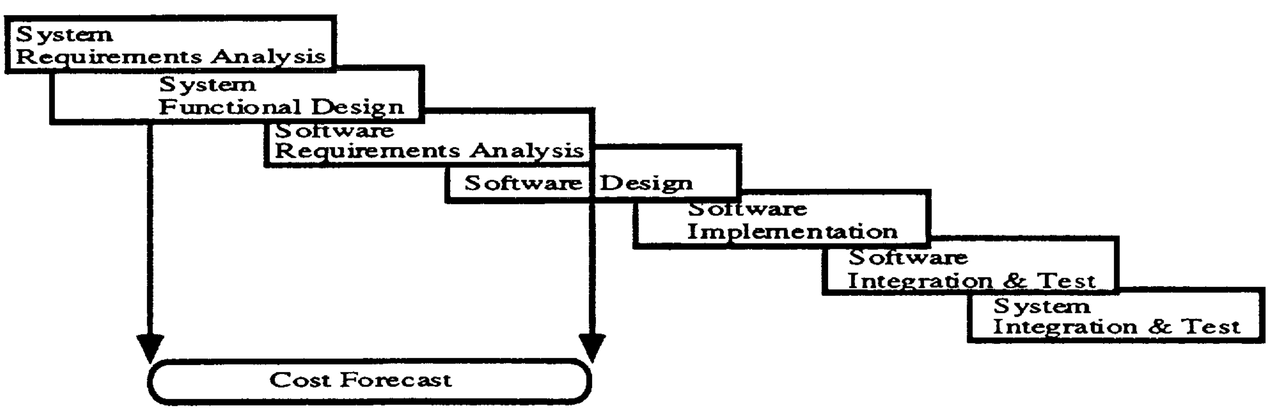


Figure 4: Timing of Cost Forecasting Verbal Protocol Collection Relative to the Software System Development Lifecyle

Table 1: Hypothetical Software Cost Forecasting Activities

Activity	Definition
Requirements Identification	The obtaining or retrieval of information. Key vocabulary words are: read requirements, talk to experts, review requirements, and obtain requirements.
Attribute Identification	Attributes are key aspects of a task that are used in forming the system mental model and are also used as analogy discriminators and cost drivers. This is one of the main products of the analysis of the requirements. Attribute identification is generally described by the basic activity that was undertaken with the result that precise attributes are rarely specified at this point. These consist of both product and process attributes. Key vocabulary words are: identify, understand, analyze, and include.
Decomposition	The breaking down of a software entity (system, subsystem, etc.) into smaller and simpler pieces. The types of decomposition that have so far been identified are: functional, work breakdown structure (WBS), new vs old system components requirements. Key vocabulary words are: breakdown (functions), identify sub-tasks, develop WBS.
Estimation	The prediction of future cost and other key project management dimensions. Three types of forecasts were reported: size, effort, and cost. Estimation was further divided by type of technique used: analogical expert judgement explicit analogy algorithmic rules of thumb cost estimating relationships Key vocabulary words are: use (analogy, rule of thumb), estimate (SLOC, effort), and cost.
Attribute Application	The explicit use of the system attributes to discriminate between systems for purposes of analogical comparison or as cost drivers when using an algorithmic approach. Identification primarily depends upon specific mention of attribute. While there is less homogeneity in the vocabulary some common phrases are: adjust, use (fog factor), add (change, fog factor, etc.), multiply.
Aggregation	The combination of forecasted values associated with the system pieces produced by decomposition. Key vocabulary words are: add-up, and run SRM (JPL resource management tool)
Adjustments	Multipliers used independently of the system being estimated. Usually applied at a higher level then attributes. Consist of adjustments for purposes of risk, scaling, and bias(error). Key vocabulary word is: add percent.
Evaluation	Any activity performed as part of checking that a forecast meets certain criteria. Most often this is the comparison of effort or cost estimate and is the last activity completed. Can also be a design-to-cost activity. Key vocabulary word is: compare to (cost of last task, budget).

3.0 Cost Forecasting Activity Definitions

Table 1 contains a list of the software forecasting activities and sub-activities that were identified in the process of converting the verbal protocols into data. These activities constitute an abstract vocabulary that was used to describe the forecasting process. The activities and their definitions were derived from the literature, JPL experiences documented in Lessons Learned, and the personal costing experiences of the authors, then modified by the data available in the verbal protocols to maximize the scoring of the linguistic units into one and only one scoring category. The level of granularity of the activities determined the information obtainable from analysis of the forecasters' activity sequences. An activity set defined at too coarse a granularity can not distinguish between sequences and all protocols will appear identical. An activity set defined with to much detail, at too fine a granularity, makes every protocol appear unique. Hence identifying the right granularity, or level of abstraction, is crucial. For a detailed description of the mapping of the vocabulary used in the verbal protocols to these activities see Appendix A in Hihn et. al. (1993).

The activities that have been added or changed since the analysis documented in Hihn et. al. (1993) are Evaluation and a re-grouping of the estimation sub-activities. The Estimation activity has been disaggregated into Size Estimation, Effort Estimation and Cost (dollar) Estimation. A distinction has also been made between Formal and Informal Effort Estimation. Formal Effort Estimation corresponds to the use of a CER or an analogical reference to a specific task or cost or size database, and Informal Effort Estimation corresponds to the use of a rule-of-thumb or any form of expert (engineering) judgment. When Effort Estimation is referred to as part of a specific mental model, it always should be understood to mean Informal Effort Estimation. The addition of the Evaluation activity to the activity list is the most fundamental change because it is a completely new activity. The specific activities that are used for describing forecasters mental models in the current analysis are Requirements Identification, Attribute Identification, Attribute Application, Decomposition, WBS Decomposition, New/Old Decomposition, Size Estimation, Cost Estimation, Informal Effort Estimation, Formal Effort Estimation, Aggregation, Adjustments, and Evaluation.

4.0 Software Forecasting Activity Analysis

The cost forecasting activities were analyzed several different ways in order to discern if there were any well defined patterns in the data. The purpose in this part of the analysis was to see if the frequency of use of an activity could be explained by some aspect of the system, environment, or an overall method that was being used. The most significant relationship we found is displayed in Table 2. For additional analysis of the activities see Hihn et. al. (1992). Some activities such as requirements identification and attribute identification were used by all the engineers interviewed. Some activities were used infrequently, e.g. adjustments and evaluation. There were three activities that were found to define relatively distinct sub-populations and correlated with the type of system being developed. These were the use of New/Old decomposition, size estimation, and the execution of a second effort estimate, which we shall call an assessment [1]. The other category consisted of cases where no pattern of activity use could be discerned. If a protocol used both a size estimate and an assessment it was counted twice. As will be seen in Section 6, the occurrence of these activities drives the whole sequence of activities.

The different types of software systems identified were rapid prototyping, formal military, research and development (R&D), evolving ground systems, and flight software. At JPL, rapid protoyping is used primarily to support military systems that automate support activities and also have vague requirements. There is a delivery at least once per year, with extensive user evaluation.

1. As will be seen in section 6 the use of multiple effort estimation activities was used to identify a Cost Assessment life cycle phase.

Documentation is kept to a minimum. The requirements are revisited with every delivery and a new rank ordering of the requirements is produced. Formal military systems follow DOD-STD-2167A. The R&D tasks cover a wide range of types of software from artificial intelligence to human-computer interface to network protocols. The evolving ground systems consist of software that supports the Deep Space Network and Space Flight Operations Center. Flight software consists of on-board or flight support software, such as software that helps to develop the navigation commands. Both ground and flight systems follow the JPL Software Management Standard. Our analysis indicates that forecasters working with Rapid Prototyping systems use assessment more extensively, Flight and Formal Military systems use size estimates more extensively, Evolving Ground Systems use New/Old Decomposition more extensively, and the R&D systems are uniform across the different key activities. The implications of these results are that, while there is diversity in engineering-based costing approaches, there is also a clustering around a few basic techniques.

Table 2: Sample Breakdown by Type of System and Forecasting Technique

System	New/Old Decomposition	Assessment	Size Estimate	Other	System Type Percentage
Rapid Prototype	20 %	60 %	20 %		13 %
Formal Military		20 %	80 %		13 %
Research	18 %	18 %	27 %	37 %	28 %
Evolving Ground System	43 %	21 %	14 %	21 %	36 %
Flight		25 %	75 %		10 %
Technique Percentage	23 %	26 %	33 %	18 %	100 %

5.0 Software Forecasting Life Cycle

As the focus of the analysis shifted from a static, or snapshot, view of what activities were verbalized to a dynamic view of the data, or time sequencing of the activities, the variation in the mental models due to personal style became even more apparent. The result is that most summaries of the mental models basically produced a blur. This is shown very well by the graph in Figure 5 , which maps the sequence of activities to the order that they were verbalized.

Thus, we needed objective criteria by which to partition the set of verbal protocols to determine if there was any clustering. The criteria could either partition the cases or partition time. As discussed above (see Section 1), a number of approaches were tried . These were refined as described in Section 4 to actually correlate the types of software systems with use of specific decomposition and estimation activities. However, this was not enough, as analysis of the probability transition networks revealed the existence of cyclic behavior. Breaking up these cycles required that the mental models be partitioned over time as well. One systematic way to define a partitioning over time is to specify a forecasting life cycle. Four phases were initially identified; Problem Definition, Problem Analysis, Cost Determination and Cost Assessment. Due to the nature of the verbal reports, it was not possible to distinguish between the first two phases, so for purposes of analysis they were combined into a single Problem Definition and Analysis phase.

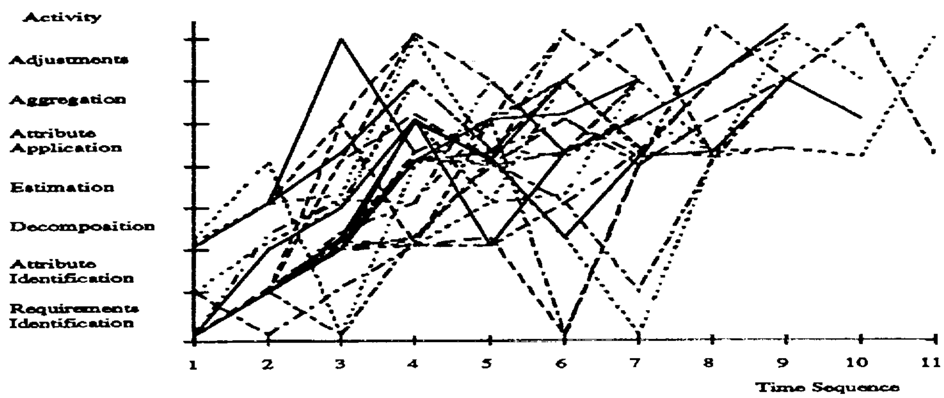


Figure 5 : Graphical Summary of Time Sequence of Activities (Hihn et. al., 1993)

Table 2: List of Activities by Cost Forecasting Phase

Problem Definition and Analysis	Cost Determination	Cost Assessment
Attribute Identification Attribute Application Requirements Identification Decomposition WBS New/Old	Attribute Identification Attribute Application Estimation Size Effort Cost Aggregation Adjustment	Attribute Application Estimation Informal Effort Formal Effort Evaluation

The assignment of activities, in the sample, to the phases is displayed in Table 2. The assignment is based on the protocols that were available. It is expected that the number of activities, with further studies, could increase in each phase due to access to more detailed protocols. Some activities, such as attribute Identification and Application, are ubiquitous, appearing in every phase. Other activities appeared only once, for example, Requirements Identification and Decomposition appeared only as part of the Problem Definition and Analysis phase. Some care had to be taken in determining when a verbal report transitioned between phases. The transition between Cost Determination and Cost Assessment was signalled by phrases such as "and then we did a backup estimate" or "compared our estimated cost to what it cost last time." The transition between the Problem Definition and Analysis phase and the Cost Determination phase was signalled when any type of estimate was mentioned. The one problem that arose in the verbal reports related to Attribute Identification that supported both Decomposition and Estimation activities. When Attribute Identification supported Decomposition, it was recorded in the Problem Definition and Analysis phase; when it supported estimation, it was recorded in the Cost Determination phase. When Attribute Identification occurred on the boundary between the phases, it was recorded as part of the Problem Definition and Analysis phase. In only one case was there compelling evidence to do otherwise.

Figure 6 displays how this costing life cycle relates to the software development life cycle for the verbal protocols used for this analysis is displayed in Figure 6. Cost estimates were made throughout the life of a software development task. Clearly, the amount of effort put into the different cost forecasting phases changes over the development life cycle. It is believed that, in the early stages of the development life cycle, more time tends to be spent in Assessment due to a lack of information required to do a comprehensive detailed cost estimate. The main changes in our model with respect to the Problem Definition and Analysis phase should be in the level of detail in the decomposition. The overall result should show a decrease in time spent in the first phase because each re-estimate builds on the previous one. The current data does not provide sufficient information to test these hypothesis.

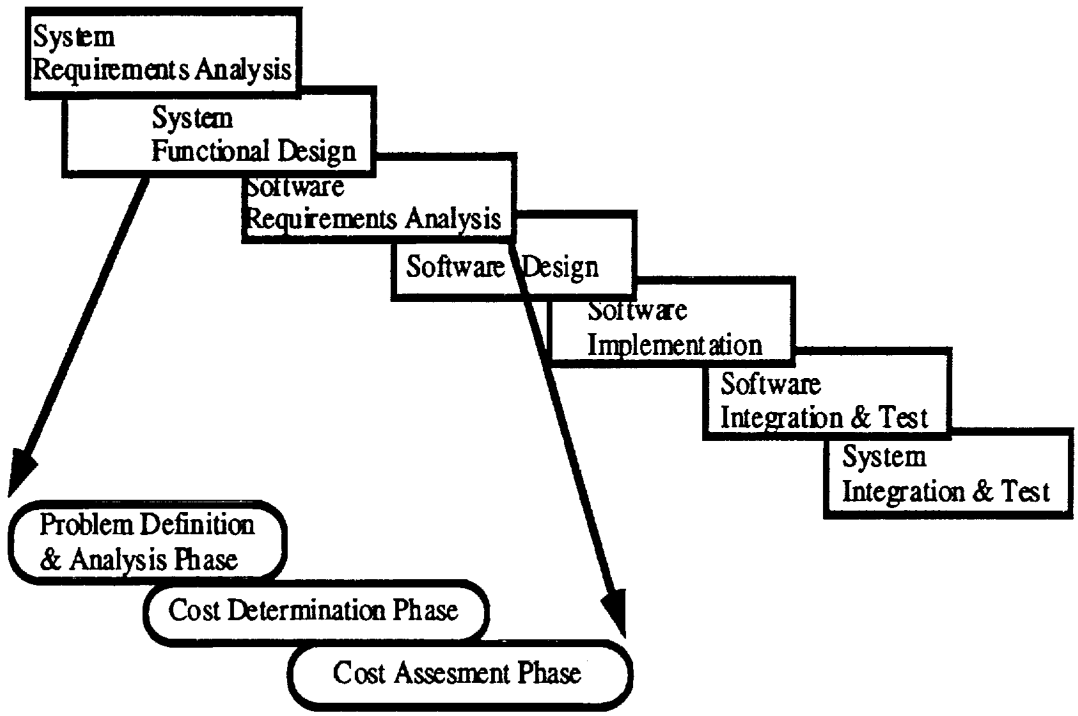


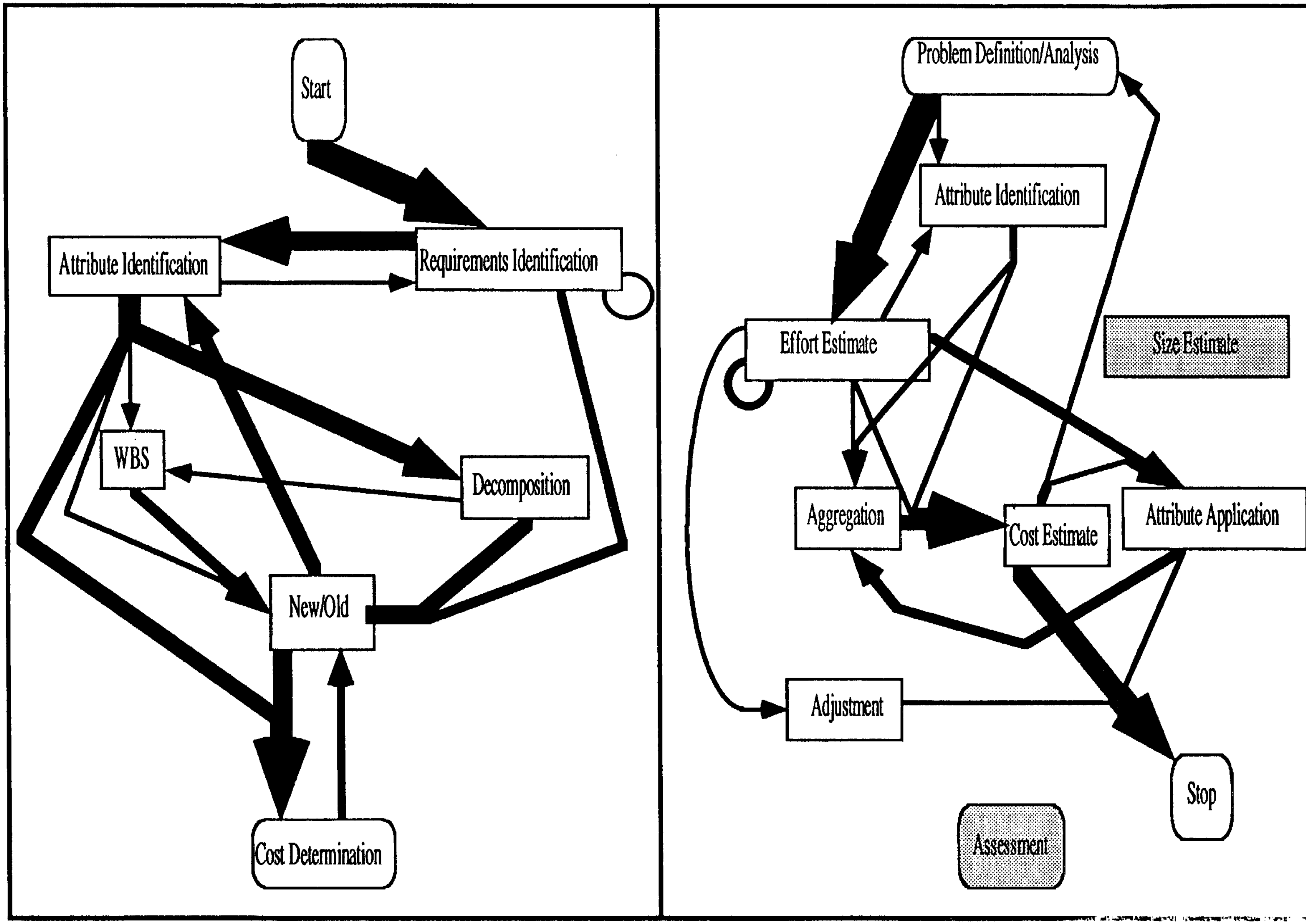
Figure 6 : Forecasting Life cycle Compared to the Software Development Life cycle

6.0 Software Forecasting Mental Models

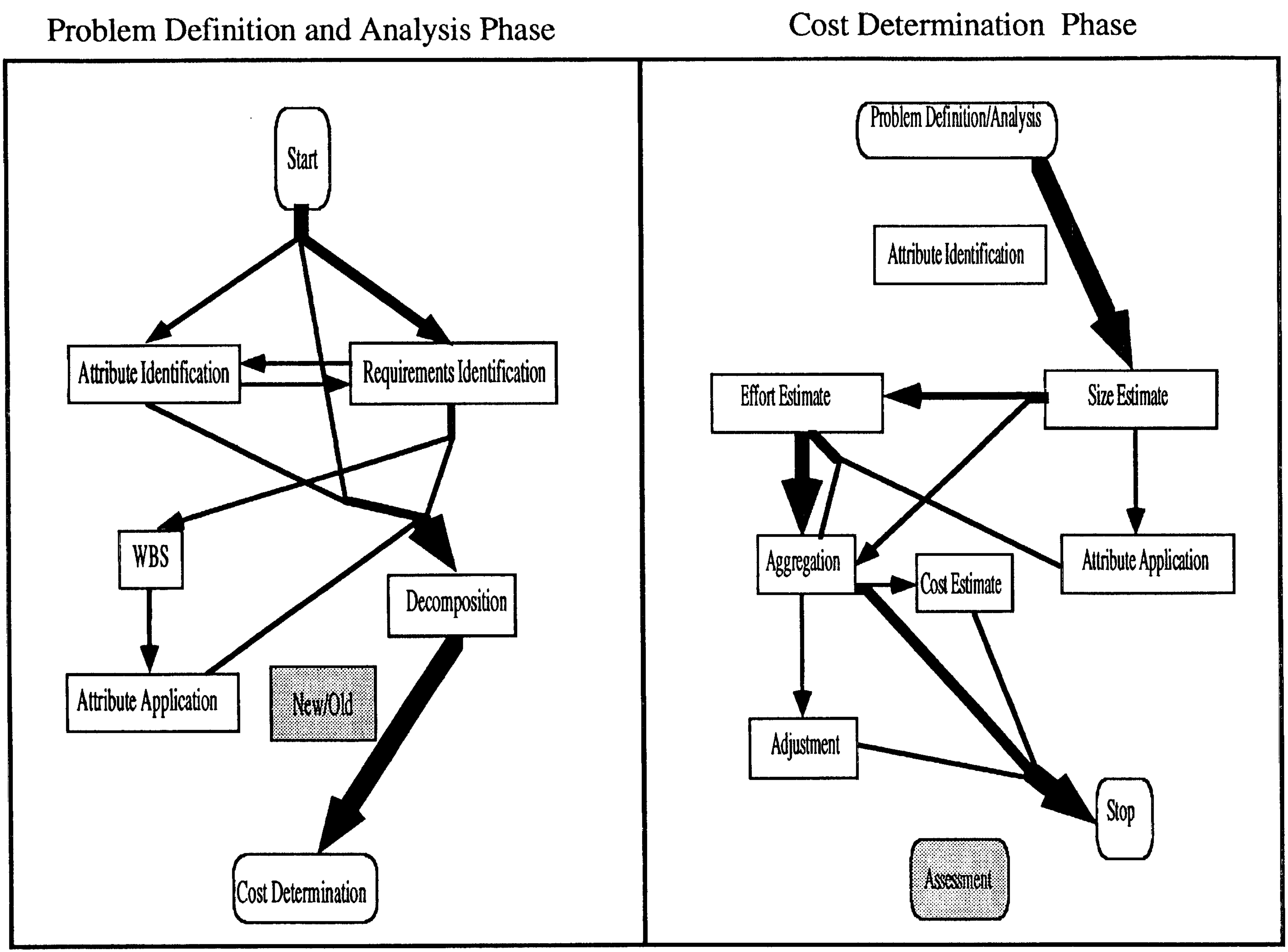
The forecasters' mental models can be represented, using Markov process modeling, by activity flow diagrams. It was possible to identify four mental models that partitioned the data. The activities and their transitions for each mental model are shown in Figures 7 through 11. Figure 7 shows the mental model of those who always used a New/Old Decomposition to support their cost estimate. Figure 8 shows the mental model of those who always used a size forecast to support their cost estimate. Figure 9 shows the mental model of those who always used an assessment effort estimate to support their cost estimate. Figure 11 shows the mental model of those who used both size and assessment. Figure 10 shows the activities and sequences for everyone in the sample who had a cost assessment phase. The thickness of the line indicates the number of transitions between activities, making it easier to visually discern where the major activity transitions occur. The thickness of the line is 2 pixels for each observation.

New/Old Only

Problem Definition and Analysis Phase | Cost Determination Phase



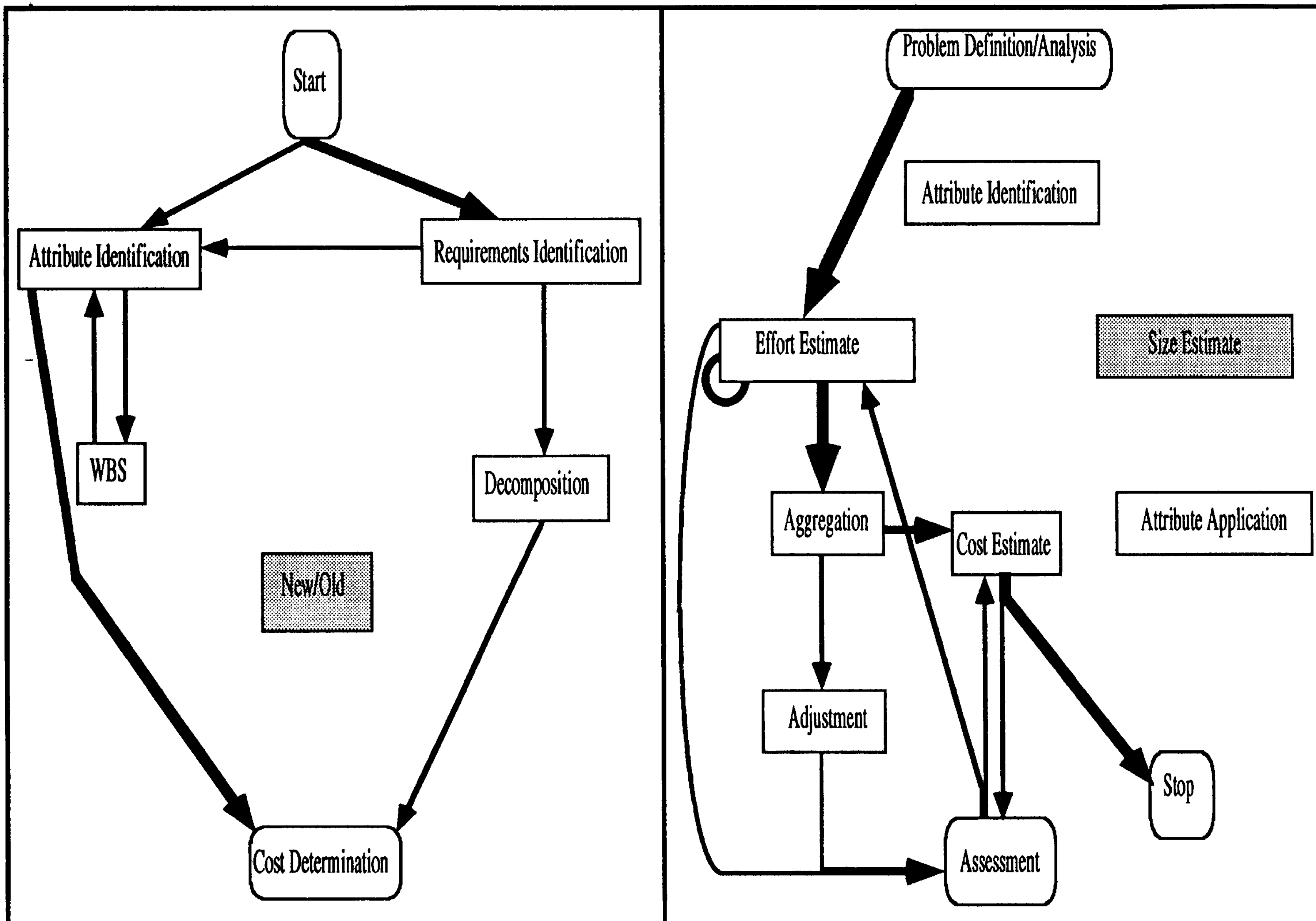
Size Forecasts Only



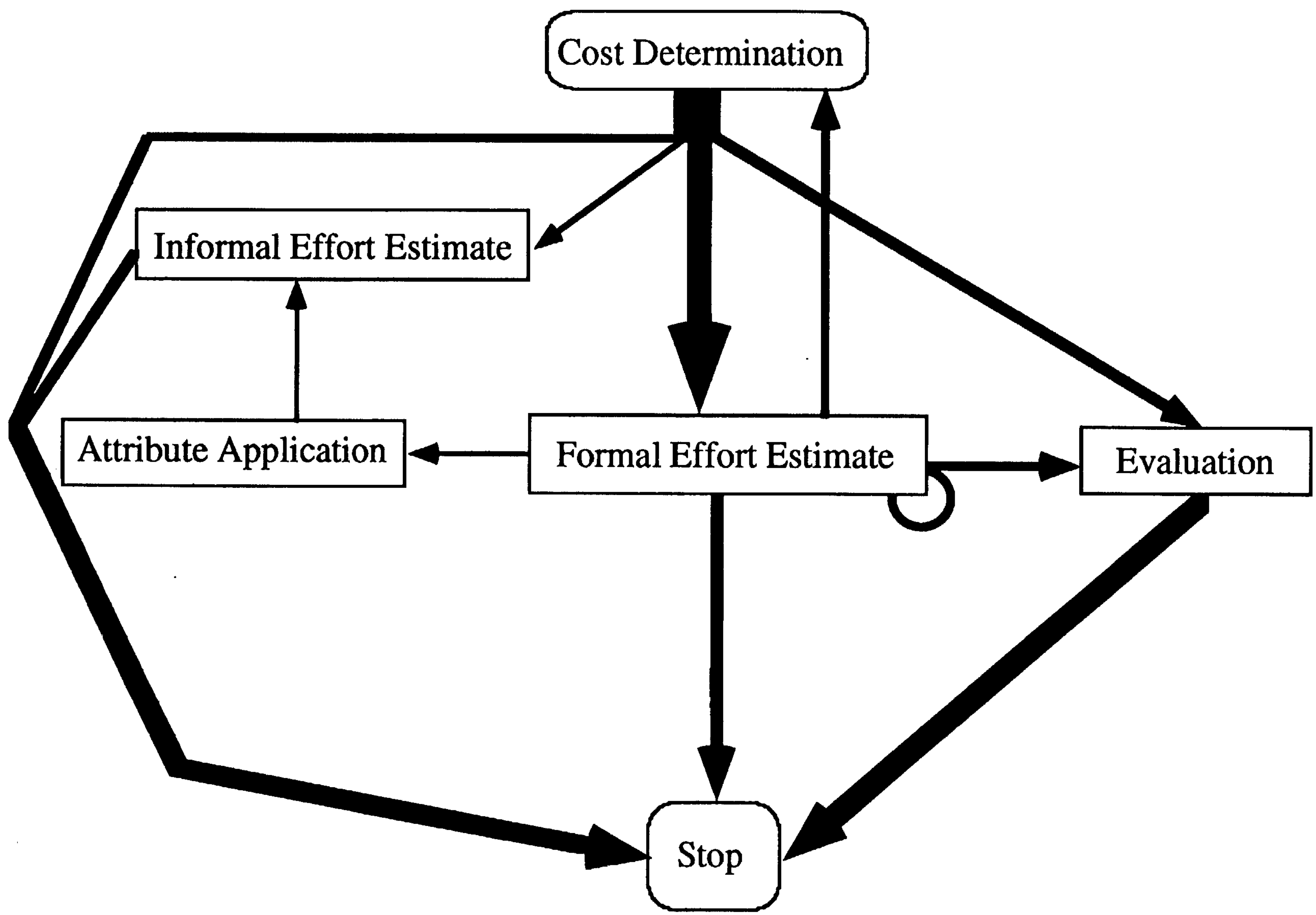
Assessment Forecasts Only

Problem Definition and Analysis Phase

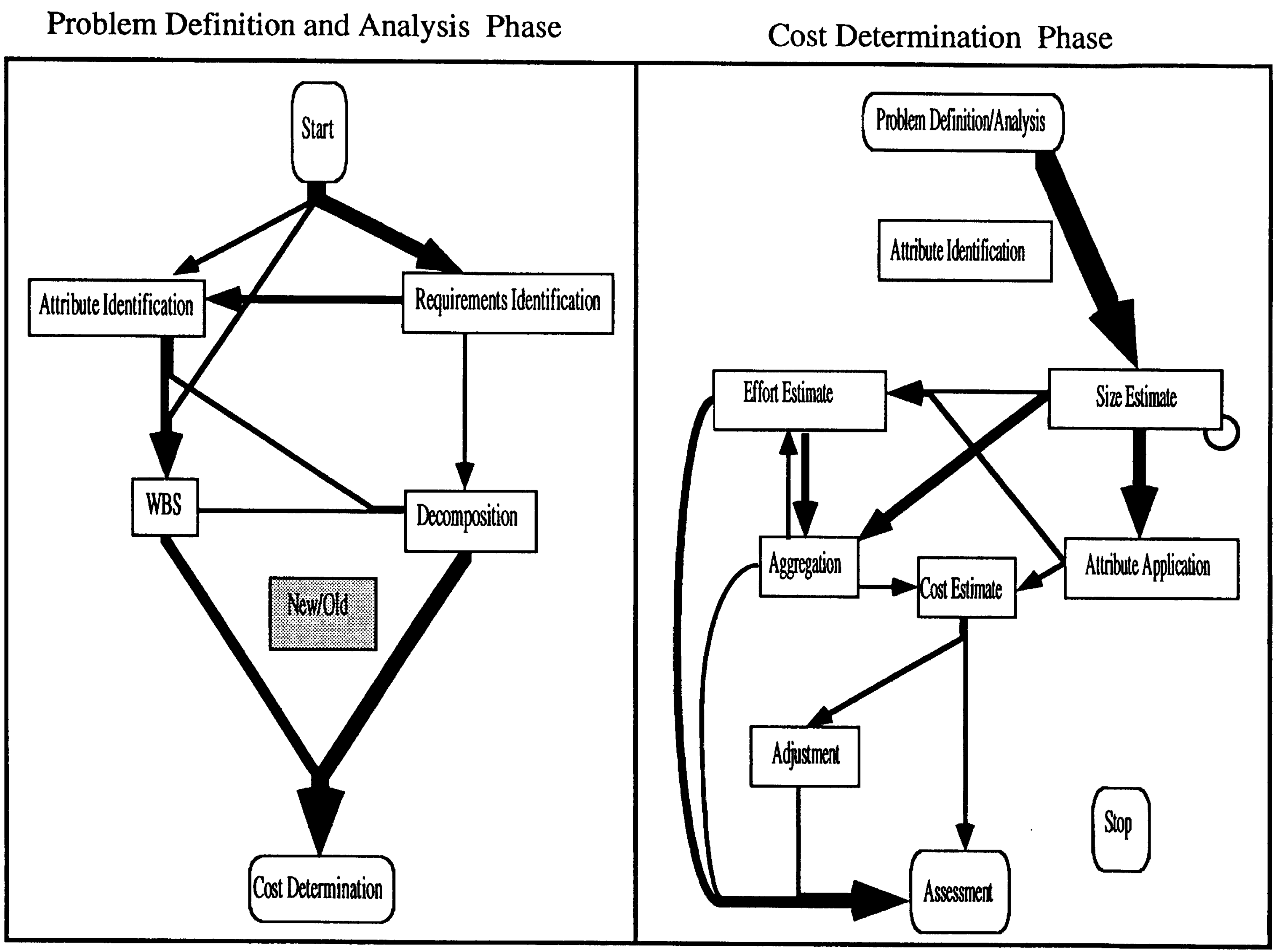
Cost Determination Phase



Cost Assessment Phase



Size and Assessment Forecasts



Note that in Figures 7-9 and 11 the Effort Estimate, Aggregation, and Cost Estimate activities are shaded in grey because there was some difficulty in discerning the actual sequence of these activities. This was primarily due to the way in which the System Resource Management (SRM) Tool, a cost accounting tool, was used. In many cases the respondent simply said and then "run an SRM". This tool can be used in a variety of ways, however, because it aggregates effort levels, adds planned procurement expenditures, and calculates overhead rates. It was frequently not clear how detailed the work was in determining the effort levels and procurements. Therefore, one level of interpretation of these activities in the mental models was simply into and out of the box that represents the combination of Effort Estimate, Aggregation and Cost Estimate.

It can be seen that, while there is a variety of activity sequences for each cost life cycle phase, there is also a clear dominant route. In Figure 7, the New/Old Decomposition Mental Model, the route was Requirements Identification, Attribute Identification, Decomposition (usually functional), New/Old Decomposition, a branch between exiting to the Cost Determination Phase or repeating Attribute Identification, finally exiting to the next phase. The Cost Determination Phase is less clear but the most likely route appears to have been: Effort Estimate, Attribute Application, Aggregation, Cost Estimate, Stop.

The dominant routes for the other mental models,while having similarities, do differ. Table 4 presents a summary of the sequence of activities for the main paths of the four mental models. Two interesting behavior patterns appear: the increased use of attributes among those using New/Old Decomposition and the lack of a Decomposition activity on the dominant path for those using only Assessment. The latter most likely occurs because those who reported only using Assessment did not have sufficient acçess to information: either because these were done as early, high level estimates or cost estimates for R&D tasks. In the New/Old mental model, the increased use of Attribute Identification reflects the impact of grouping functions by degree of inheritance. This is important because how the effort estimate was made depends upon the degree of experience of those developing the functions.

Table 4: Activity Sequence Summary of Major Activity Transitions for Forecasting Mental Models

Activity	New/Old	Size	Assessment	Size and Assessment A	Size and Assessment B
Requirements Identification	1	1	1	1	1
Attribute Identification	2,5		2	2	2
Decomposition	3	2		3	3
New/Old Decomposition	4				
Size Estimation		3		4	4
Effort and Cost Estimate	6,8	4	3	5	6
Attribute Application	7				5
Assessment			4	6	7
Stop	9	5	5	7	8

A cursory review of the different mental models revealed to us that there exist substantial personal style variations because there seems to be no one way to get a job done. However, there were dominant pathways, and the mental models are clearly different. We interpret the primary differences in the mental models as representing the different ways that forecasters attempted to reduce risk in their cost forecasts. The risk reduction techniques were based upon the use of either multiple-decomposition steps, in this case additional New/Old Decompositions or multiple forecasting steps. The multiple forecasting steps involve either forecasting software size or an additional effort forecast (Assessment). Very few used these risk reduction steps in combination.

7.0 Summary and Conclusions

A viable process for capturing and analyzing the mental models software engineers use for cost and size forecasting has been demonstrated. Our analysis demonstrates the existence of three interdependent cost forecasting life cycle phases. The data analysis of the last few sections provides a basis for us to begin to identify where software engineers can best use supporting methods, tools, and data. Unfortunately, the currently available costing methods and tools only support the Cost Determination phase. Methods, tools and data are needed that will:

support sequential estimation steps

support different techniques, save and assist in comparing results

store design information and supporting estimates

provide assistance in identifying task analogies

In addition the idiosyncratic nature of the individual protocols indicates that supporting methods and tools need to capture and record the steps followed and information used by the forecaster. This will provide a record of the assumptions and context within which the estimate was made, and should improve the quality of updated estimates.

Finally, previously published analysis of this data showed that for experienced forecasters, those who forecast frequently (at least every 6 months) on the average forecast effort 12% high, whereas those who forecast less frequently (at greater than 6 month intervals) on the average forecast effort 44% low. This suggests examining the mental models of those activities and transitions most dependent on memory and determining corrective support methods, tools and data.

forecasts. The risk reduction techniques were based upon the use of either multiple-decomposition steps, in this case additional New/Old Decompositions or multiple forecasting steps. The multiple forecasting steps involve either forecasting software size or an additional effort forecast (Assessment) Very few used these risk reduction steps in combination.

7.0 Summary and Conclusions

A viable process for capturing and analyzing the mental models software engineers use for cost and size forecasting has been demonstrated. Our analysis demonstrates the existence of three interdependent cost forecasting life cycle phases. The data analysis of the last few sections provides a basis for us to begin to identify where software engineers can best use supporting methods, tools, and data. Unfortunately, the currently available costing methods and tools only support the Cost Determination phase. Methods, tools and data are needed that will:

support sequential estimation steps

support different techniques, save and assist in comparing results

store design information and supporting estimates

provide assistance in identifying task analogies

In addition the idiosyncratic nature of the individual protocols indicates that supporting methods and tools need to capture and record the steps followed and information used by the forecaster.This will provide a record of the assumptions and context within which the estimate was made, and should improve the quality of updated estimates.

Finally, previously published analysis of this data showed that for experienced forecasters, those who forecast frequently (at least every 6 months) on the average forecast effort 12% high, whereas those who forecast less frequently (at greater than 6 month intervals) on the average forecast effort 44% low. This suggests examining the mental models of those activities and transitions most dependent on memory and determining corrective support methods, tools and data.

References

Albrecht, A. and J. Gaffney, ``Software Function, Source LOC and Development Effort Prediction, A Software Science Validation," **Transactions of Software Engineering**, Vol. SE-9, No. 6, November, 1983, p. 639-648.

Boehm, B., **Software Engineering Economics**, Prentice Hall, 1981.

Ericson, K. and Simon, H., **Protocol Analysis**, MIT press, 1984

Hihn, J, Griesel, A., Bruno, K, Fowser, T., and Tausworthe, R., Mental Models of Software Forecasting, **Proceedings of the 15th Annual Conference of the International Society of Parametric Analysts**, San Francisco, Ca, June 1-4, 1993, pp. K2-K28.

Hihn, J. and Habib-agahi, H., Cost Estimation of Software Intensive Projects: A Survey of Current Practices, **Proceedings of the 13th International Conference on Software Engineering**, May 17-19, 1991, pp. 276-287.

Howard, M., "The Creation of a Research Model for Estimation", **Proceedings of the European Software Cost Modelling meeting 1992** (ESCOM), Munich, germany, May 27-29, 1992.

Papoulis, A., **Probability, Random Variables and Stochastic Processes**, Mcgraw-Hill Inc, 1991.

Vicinanza, S., Mukhopadhyay and Prietula, M., Software-Effort Estimation: An Exploratory Study of Expert Performance, **Information Systems Research**, vol 2, December 1991, pp. 243-262.

Software Cost Forecasting As It Is Really Done: A Study of JPL Software Engineers

NASA Goddard Space Flight Center
SEL Software Engineering Workshop
December 1-2, 1993

Martha Ann Griesel
Jairus M. Hihn
Kristin J. Bruno
Thomas J. Fouser
Robert C. Tausworthe

JPL
Jet Propulsion Laboratory
California Institute of Technology
Pasadena, California 91109

Why Should We Care How Experts Forecast Software Costs?

In today's cost-constrained environment, cost estimation is an integral part of the engineer's job

Therefore, tools and databases are needed to support integrating cost analyses with traditional engineering practices

Previous surveys have shown that engineers in general do not use tools and databases they perceive to be inconsistent with their software cost forecasting mental models

The purpose of this study was to determine the requirements for methods, tools and databases that are consistent with engineers' software cost forecasting mental models

Questions That Needed to be Answered

Is there a set of well-defined software cost forecasting activities?

Do these activities combine into a small number of well-defined mental models?

To what degree are differences in the mental models dependent upon personal style, problem domain, and environment?

Do the different mental models fit within a single software forecasting lifecycle?

How can a better understanding of existing software cost forecasting practices improve the implementation of those practices?

Background

Literature on mental models of forecasting is sparse:

- results are not repeatable
- previous studies support the assumption that there are a small number of basic activities

Previous work by the authors identified more rigorous methods of data capture and analysis

- Cognitive Psychology provides a method of data capture (Protocol Analysis)
- Stochastic processes provide a method of analysis (Transition Probability Matrices)

Background (Cont.)

The current analysis uses data that existed from a previous study

We have been able to identify 28 observations that provide sufficient detail for analysis

Respondents types and number of years of software experience varied

Protocols reflect forecasts made during either System Architectural Design or Software Requirements Analysis

MAG/JMH/JPL/November 29, 1993 4

Forecasting Activities

Requirements Identification

Attribute Identification - People, Product, Process

Attribute Application - People, Product, Process

Decomposition - WBS, New/Old, Functional, Requirements

Aggregation

Size Estimation - Expert Judgement, Analogy, Rules of Thumb, CER

Effort Estimation - Expert Judgement, Analogy, Rules of Thumb, CER

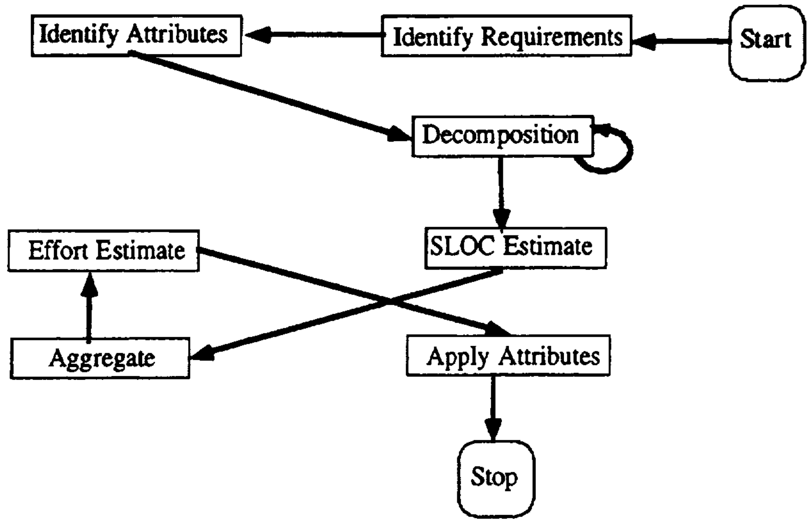
Cost Estimation - Generic, SRM

Adjustment - Risk, Scaling, Bias

Evaluation

MAG/JMH/JPL/November 29, 1993 5

An Example of a COCOMO Software Cost Forecasting Mental Model



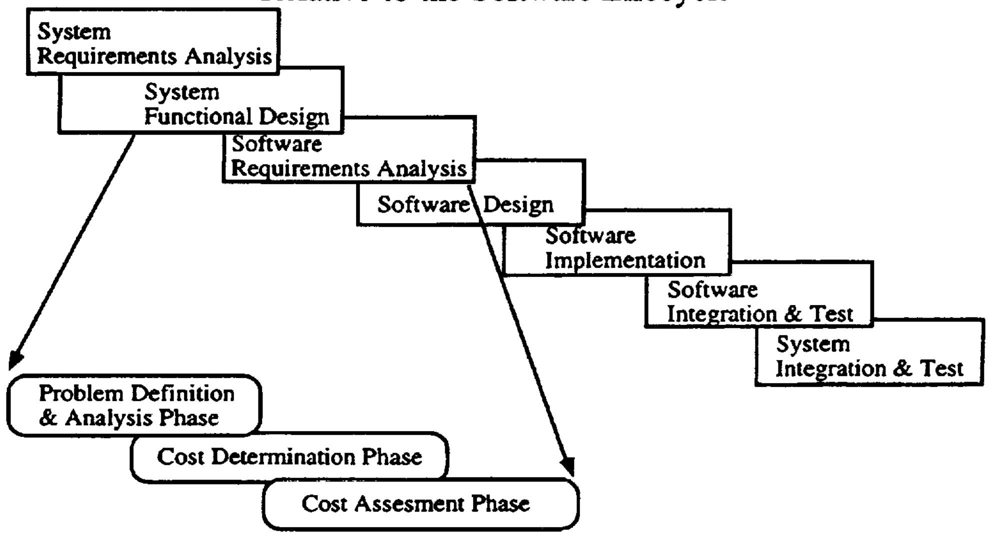
Activity Clustering and Differences

Sample Breakdown by Type of System and Forecasting Technique

System	New/Old Decomposition	Assessment	Size Estimate	Other	System Type Percentage
Rapid Prototype	20 %	60 %	20 %		13 %
Formal Military		20 %	80 %		13 %
Research	18 %	18 %	27 %	37 %	28 %
Evolving Ground System	43 %	21 %	14 %	21 %	36 %
Flight		25 %	75 %		10 %
Technique Percentage	23 %	26 %	33 %	18 %	100 %

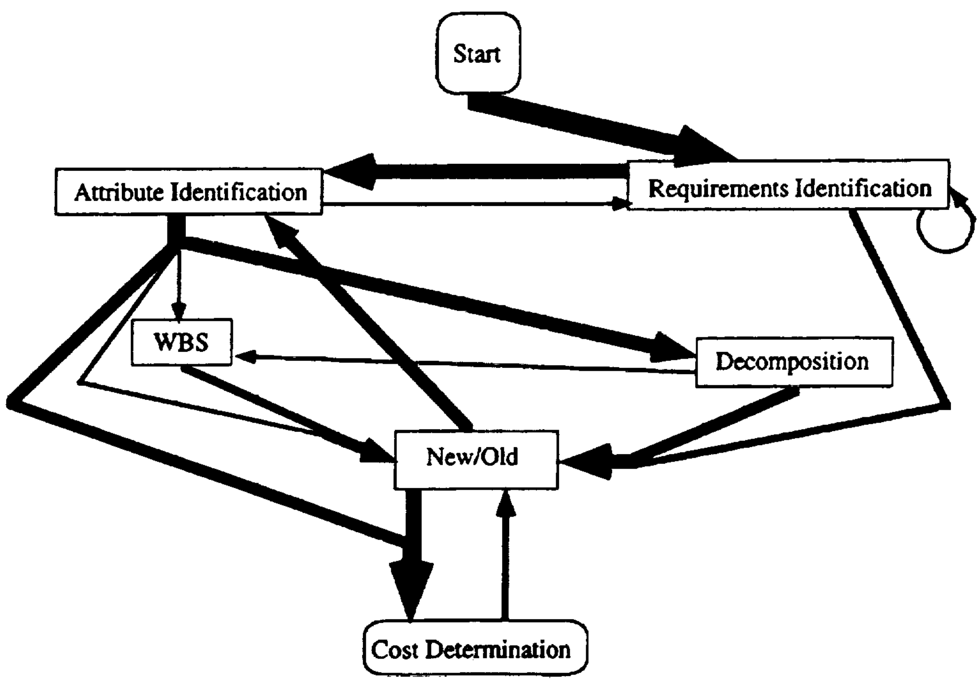
Sample Size equals 28, Due to use of multiple techniques total count is 39.

Scope of the Current Software Cost Forecasting Lifecycle Relative to the Software Lifecycle

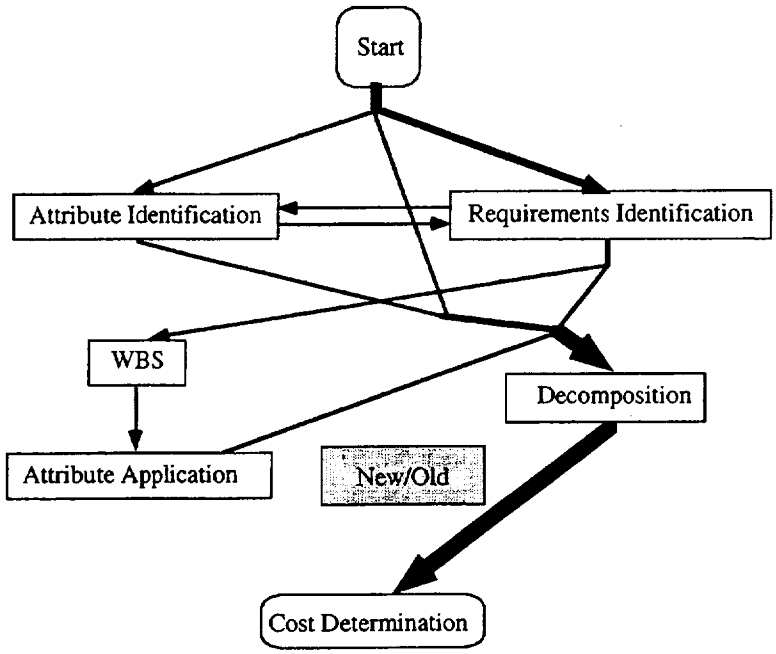


MAG/JMH/JPL/November 29, 1993 8

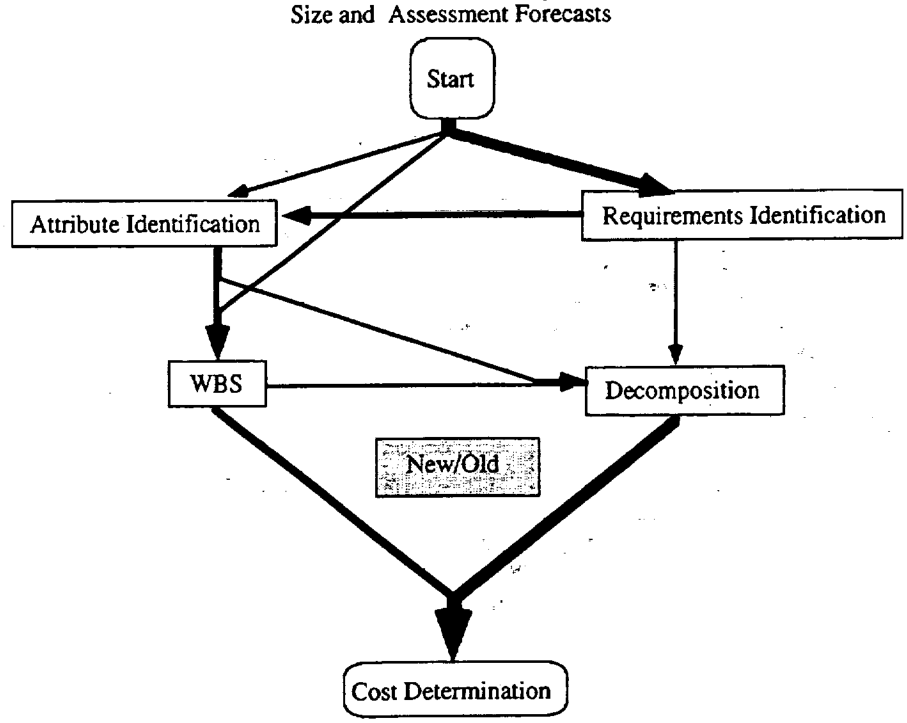
Problem Definition and Analysis Phase New/Old Only



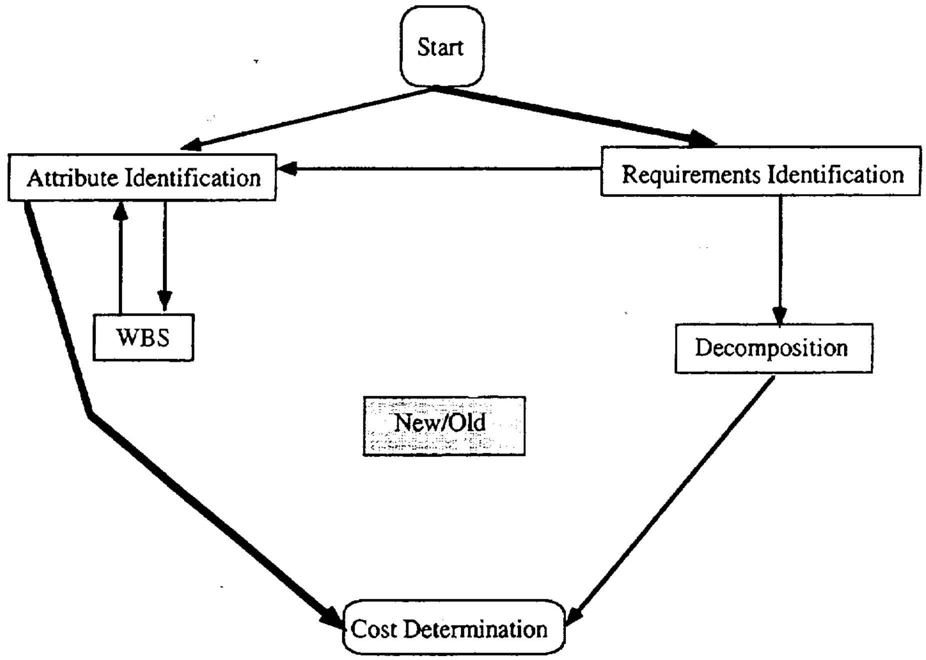
Problem Definition and Analysis Phase
Size Forecasts Only



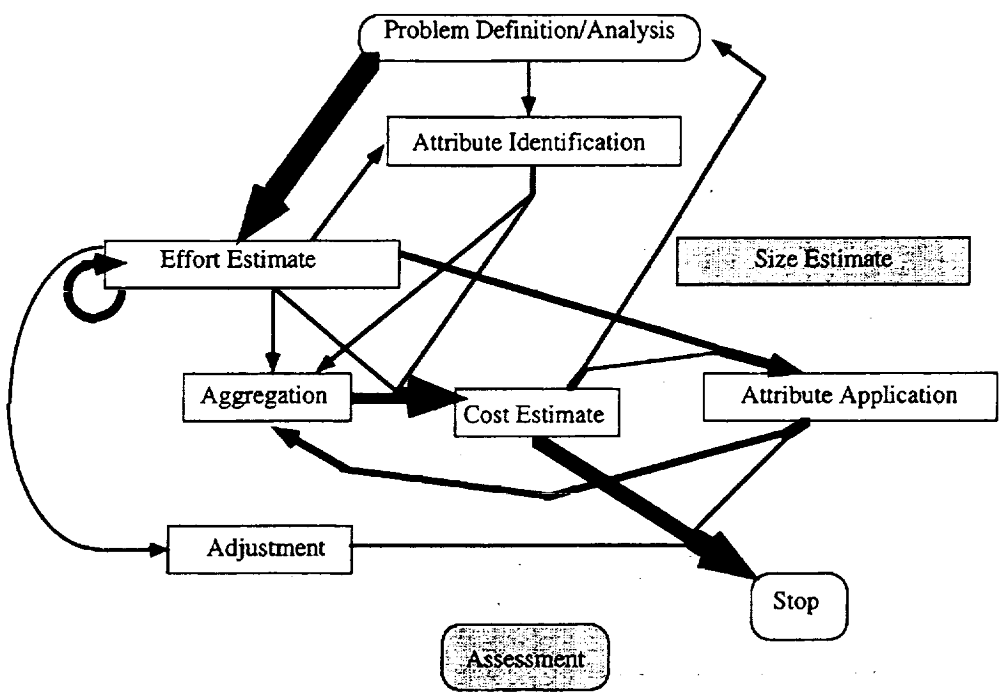
Problem Definition and Analysis Phase
Size and Assessment Forecasts



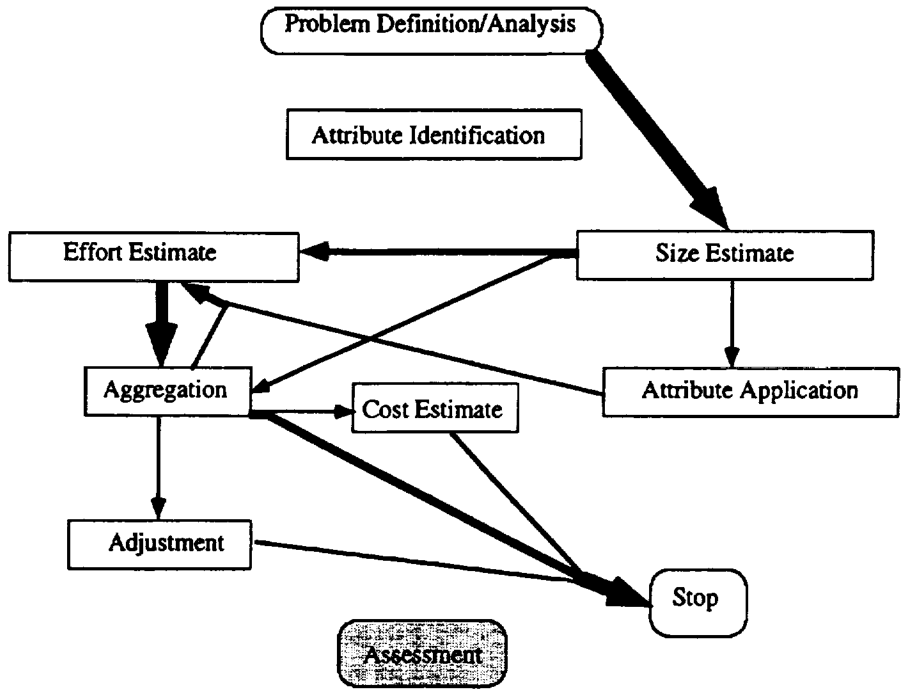
Problem Definition and Analysis Phase
Assessment Forecasts Only



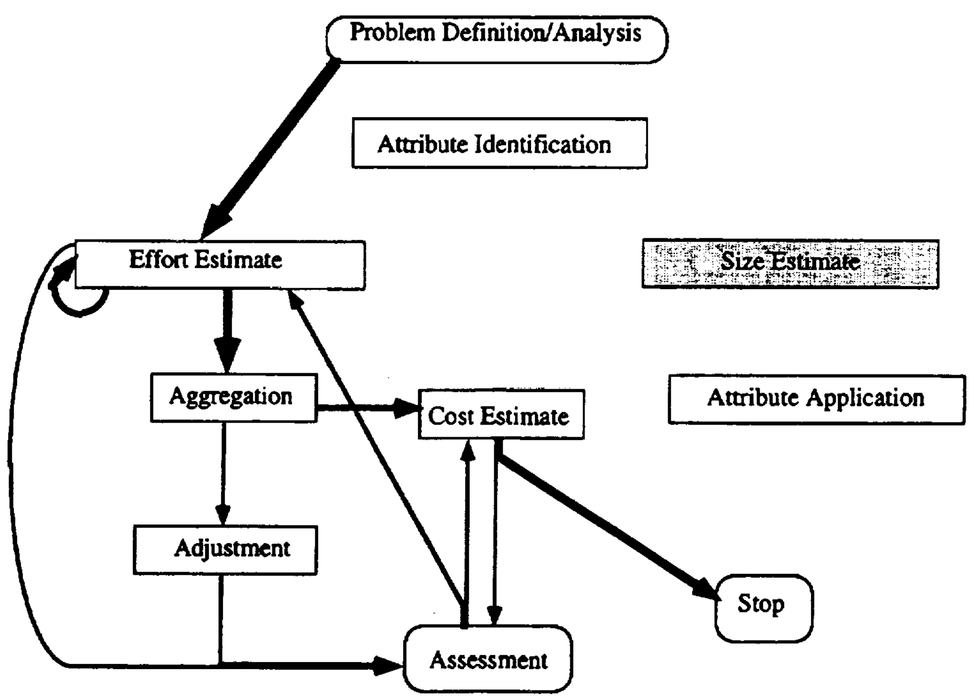
Cost Determination Phase
New/Old Only



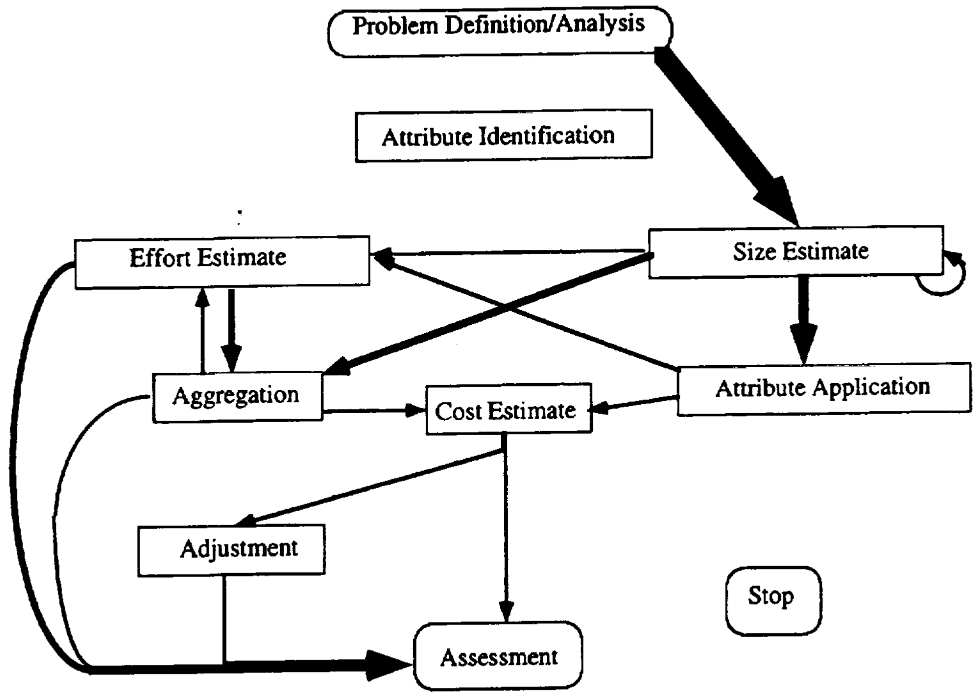
Cost Determination Phase
Size Forecasts Only



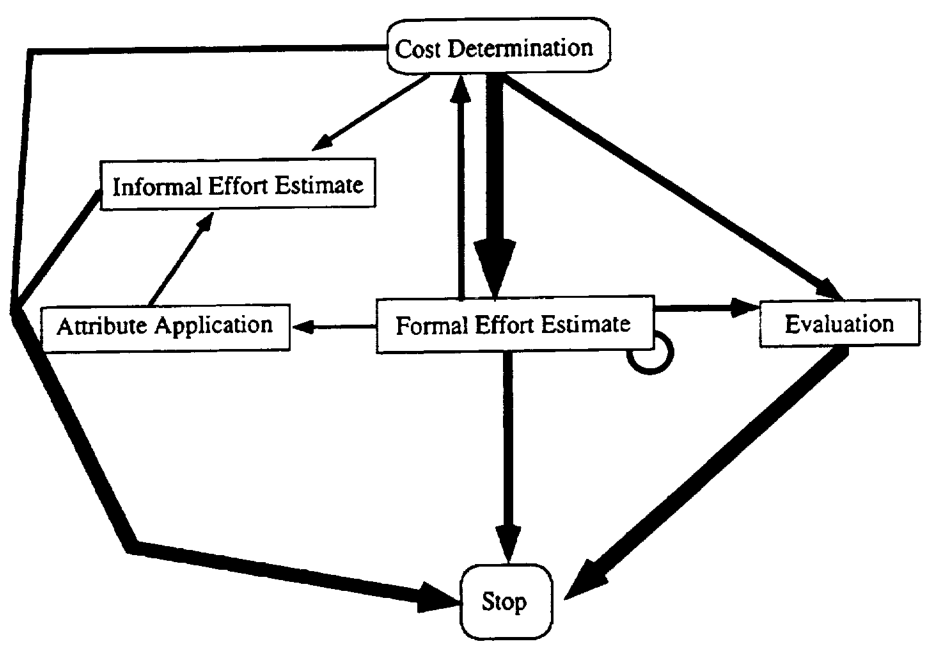
Cost Determination Phase
Assessment Forecasts Only



Cost Determination Phase
Size and Assessment Forecasts



Cost Assessment Phase



Conclusions

Expert forecasters use simplification in the face of complexity

- 86 % only use one technique to reduce cost forecast risk
  - more detailed decompositions
  - more detailed forecasts
- they keep cost techniques simple and use only a few cost drivers
  - Personnel Quality, Complexity, Language

Consistent with Cognitive Psychology findings in other fields

MAG/JMH/JPL/November 29, 1993 9

Conclusions

Experts tend to use techniques based on domain knowledge and rules of thumb

- single domain experts generally get into detailed forecasting quickly
- multiple domain experts do more abstract or generic forecasting

Design-To-Cost differs in that

- Attribute Identification is more likely to be used as a first step
- forecasts are iterated based upon cost-budget comparison

MAG/JMH/JPL/November 29, 1993 10

Summary
Spanning the Mental Model Problem Space