USE OF GENERALIZED ACTIVITY NETWORK MODELS FOR ANALYSIS OF EUROPEAN ATM DEVELOPMENT PROJECTS

Peter F. Kostiuk, Scott Houser, LMI, McLean, Virginia

Patrick Ky, Eurocontrol Experimental Centre, Bretigny, France

Introduction

Many of the challenges facing the ATM research community are well-known and include substantial technical risk and difficulty demonstrating benefits for stakeholders. To these concerns, we propose to add another: inadequate understanding of the cost and schedule risks of development programs. Of these risks, political influences are particularly difficult to analyze. Political decisions can effect the choice of development processes, which can drastically effect the final cost, completion time, and probability of success for the project. In this paper, we propose a method to analyze the impacts of technical risk and political effects on the time to complete ATM development projects. We recommend the use of this approach for all major development projects as a way to improve the management of ATM programs, identify sources of risk, and better predict project completion times and their costs. We illustrate the approach by analyzing a simple but relevant example from a recent ATM development project in Europe.

Generalized Activity Networks

This paper demonstrates how Generalized Activity Networks (GANs) can model the development process used in ATM programs and assess the degree of uncertainty in their completion time and cost completion. GANs are generalizations of the popular Program Evaluation and Review Technique (PERT) method. They enable an analyst to specify in detail the different activity in a development process and incorporate the distribution of completion times for each activity. GANs are also capable of modeling test outcomes, and of specifying the rework activities that may be necessary should a test show that the project has failed to meet a milestone criterion. The method has been applied for a couple of decades in the construction industry,[1] but only recently has it been applied to development programs with significant research content. LMI analysts pioneered the application of the approach to programs for the United States Department of Defense with good success.[2]

A GAN analysis provides program managers with an improved understanding of the risks associated with delivering their projects on time and within budget. When used during project planning, they can identify the impacts of alternative development approaches that introduce different types of risk. The approach is particularly useful when evaluating programs that:

- Require testing, with the possibility of test failures and the need for rework;
- Integrate two or more activities;
- Allow for the possibility of branching into different development paths.

Each of these characteristics is common to ATM research projects. These projects often include tests at various milestones. The outcomes of these tests may determine the course of future work, or reveal the need to revise previously completed work to correct deficiencies. ATM projects also tend to include work delivered by multiple organizations, either during the research phase or the development phase. Development may occur in different countries, or must be implemented by many other parties (airlines, manufacturers, or air traffic services providers, for example).

Like a PERT chart, a GAN diagram describes a process as a set of nodes, connected by arcs. The nodes represent the states of the process as it evolves toward completion, and the arcs represent the activities required to make the transition from one state to another. Figure 1 shows a very simple example.

[1] The definitive text is [1], written in 1977.

[2] See [2] and [3] for recent publications on this topic.

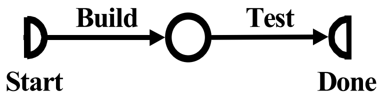

Figure 1. A Simple GAN

The simple example of Figure 1 could be a PERT chart as well as a GAN; GANs comprise a superset of PERT charts. But GANs can do much more than PERT charts: every arc of a PERT chart must execute, while the arcs of a GAN may or may not execute. The probabilities of execution for such arcs may be specified, or computed based upon previous events in the GAN.

Also, arcs in PERT charts execute exactly once: looping is not allowed (formally, PERT charts are directed, acyclic graphs). GANs can model feedback loops. This is important for characterizing cycles of test-rework-retest, and the cycles in spiral development.

Each arc of a GAN may have a probability distribution for time or cost to execute the arc or have a joint probability distribution of cost and time. More generally, GAN arcs can be represented as vectors of random variables for crucial values. Time and cost are the most obvious of these variables. It is also possible to include metrics that express technical progress as well: for example, software defects fixed or research objectives accomplished. These random variables can be dependent.

Figure 2 shows a more interesting case of a GAN diagram of an iterative, abrogable process. We define a work task (W) with an iteration task (I), a finalize-on-success task (F), and a complete-on-cancellation task (A) in this network. We define the probability s that the project team succeeds in completing the activity as a function of the iteration cycles c ($s = f(c)$). The dependence upon cycles assumes that the team is able to learn from each iteration, and can therefore improve their chances for success in a predictable fashion. We also allow for the probability a that the work will be canceled to depend upon the number of cycles. It is possible, and common, to condition a on s and c. For example, after a certain number of failed attempts, the stakeholders may cancel the process if success appears unlikely in the next attempt. As an alternative to cancellation, the activity A could define an alternative activity that would be followed after a certain number of attempts to achieve F had failed.

In developing a GAN model of a project, the analyst defines the project's activities and their relationships with other activities in the entire process. The activities can be part of a straightforward sequence, such as the one shown in Figure 1, or a

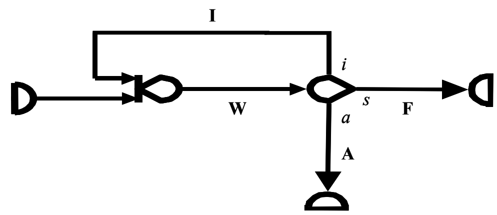

Figure 2. A GAN of an Iterative Process

more complicated, iterative sequence, as shown in Figure 2. GANs are robust and accommodate the modeling of multiple types of failures.

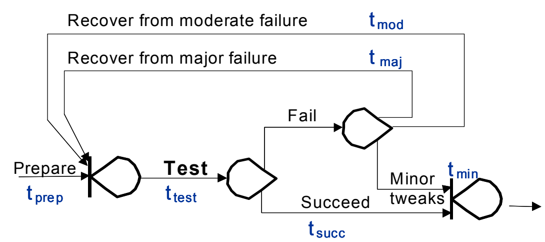

Figure 3. GAN For a Major Test With Multiple Outcomes

Figure 3 shows a GAN that models a development test with three possible failure severities. We modeled an entire test program for a U.S. Department of Defense development project as a sequence of these test GANs. Working with the project managers, we assigned probabilities each of the possible outcomes of the test activity: success, a failure requiring minor rework, a failure requiring moderate rework, and a failure requiring major rework of the defense asset. We modeled the completion time of each activity using Weibull distributions. These times included time for initial preparation (tprep), test time (ttest), transition time after a success (tsucc), and times for recovery from the three failure modes (tmin, tmod, and tmaj).

Perhaps the most important feature of this simple model is the requirement that the test team redo the test after a moderate or major failure. Test planners understand the need for such retests; however, they often fail to realize the potential impacts of these retests on the total completion time. By decomposing the process into a set of outcome probabilities and activities with relatively easy-to-estimate completion times, we were able to help the test planners make a better estimate for the completion time of the full test program.

Application to European ATM Projects

In this section, we apply GANs to estimate the likely time to complete for different development approaches. Our analysis attempts to demonstrate the added value of a common, *ab initio* approach to the development of highly interoperable systems.

This project requires the implementation of interoperable ATM systems in all European countries. European States could therefore undergo the effort at the beginning of the project to develop common specifications and standards which would then be implemented, possibly in parallel but also possibly in a centralised manner.

Alternatively, each of the states could develop its own specifications and standards, then implement them. To obtain interoperability, a more difficult integration activity would be required to bring the three implementations together.

For this particular program, we chose to identify another scenario, which consists in an attempt to develop common standards that is abandoned after some years because of a failure to come to political agreement, at which point groups of States then pursued their own standards and development activities independent of the other projects. For this study, we evaluate the three development scenarios and generate predictions of total project completion time.

To summarize, the three scenarios are:

1. A European centralised approach: Develop a common set of specifications and standards, followed by three local implementations of those standards, and a final integration phase.

2. A localised approach: Three independent efforts to develop specifications and standards, followed by three local implementations, completed by integrating the three distinct implementations.

3. A "typical European scenario": An initial effort to develop common standards, which is stopped after three years, at which point we begin to implement the local standard approach in Scenario 2.

Each of these development approaches presents different levels of risk at different phases of development. In Scenario 1, the greatest risk is during the initial phase of developing common specs and standards. In addition to the usual technical risks involved with such efforts, there is the additional challenge of overcoming the political risk involved when different countries, with potentially conflicting agendas, must reach agreement before moving on to the next development phase. Once agreement is reached, however, the following phases can focus on meeting technical milestones. In particular, the risk of integration is substantially reduced due to the use of common standards throughout implementation.

Scenario 2 bypasses the political risk of reaching agreement in the first phase of the project by allowing each group of States to pursue their own standards and development. This approach transfers the political risk into a substantial technical risk at the integration phase, when the developers must build in interoperability among three independent development efforts. Our third scenario reflects both the difficult task of reaching agreement in the presence of different political agendas, as well as the greater technical challenges that result when agreement fails.

Figure 4 describes the development process for a European ATM project. The process begins by defining system operating specifications, followed by a standards-development phase. Once the standards are developed, they are implemented in three different political jurisdictions. After the local developments are completed, the systems are integrated to provide interoperability across States.

In Figure 4, the top horizontal line describes the sequence of development activities during a centralized development process. All three organizations work jointly during the specifications and standards phases, with the goal of defining products that are politically acceptable and technical feasible. If agreement is reached on those activities, they move on to the "A-B-C" box, which depicts the local implementations process. Once each of the three organizations completes implementation, they move to the final integration phase.

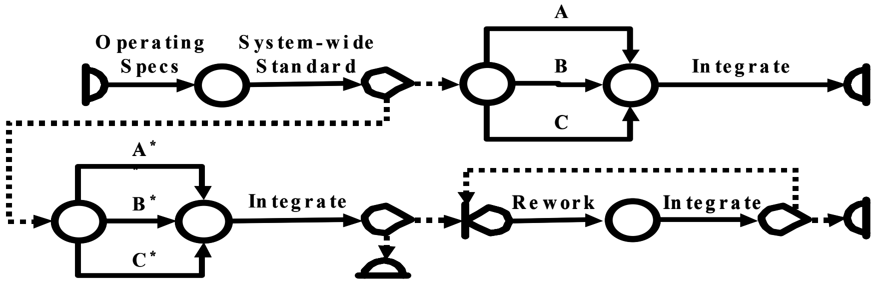

Figure 4. GAN Diagram of a European ATM Development Project

GAN modeling of that process requires us to specify the random process that describes the length of time that it takes to complete each activity in the process. The GAN method is flexible and can accommodate a wide variety of distributions. In our work, we have found that Weibull distributions provide good descriptions of many development activity durations, as they can exhibit the right-tailed asymmetry typical of development programs and their tendency for schedule creep. The Weibull was used to model duration because it has a long history of use in modeling time random variables, is flexible (its shape can resemble an exponential curve, and it can also look almost Gaussian), easy to simulate, and can be characterized by only three parameters. When modeling time, a convenient method exists for computing the Weibull parameters from three intermediate parameters: the most likely time, the minimum time, and a risk level (low, medium, or high) for the activity. Technical experts can provide these values without extensive knowledge of statistics.[4]

Figure 5 shows typical examples of Weibull distributions used in duration analysis. The top diagram depicts a Weibull with a nearly symmetric shape, with only a small right tail, that would represent a project activity with low or medium risk. The lower diagram shows a much greater risk of long duration times, a characteristic not unknown in ATM projects.

In projects with multiple activities, each with significant right-tails in their duration distribution, it is important to properly account for the asymmetry when estimating total project duration time. Project managers commonly estimate total duration by adding the "most-likely" estimate for each activity, sometimes accounting for the risk when choosing the most likely value. However, when the durations are right-skewed, adding the most likely duration for each activity will significantly under-estimate the expected value for the total duration because of the high probability of longer durations at each activity. This is even more likely when some activities have higher risk than others.

To account for differences in risk among the project phases, we use differently-shaped Weibull distributions. Table 1 lists the descriptive parameters that describe the shapes of those distributions. These parameters describe the overall shape, which is then scaled by providing a time value for the mode.

Table 1. Weibull Descriptive Parameters

Risk	Ratio of Mode to Most Likely Low Value	$Prob(t>t_{mode})$
Low	1.15	0.65
Medium	1.25	0.70
High	1.50	0.80

We chose a different way to model the duration of the standards development phase. Since the challenges to this phase are both technical and political, a Weibull or other distribution with right-skewness seems inappropriate because external forces are likely to intervene if the process takes too long because of the political disagreements. As a technical issue, we believe that it would take about 3-4 years to complete the standards phase without undue political interference. To model the effects of different levels of interference (or perhaps political willpower, if you prefer), we modeled the duration of this phase with three uniform distributions of 3-4, 4-5, and 5-6 years duration. This enables us to assess quantitatively the impact of external pressures on the technical process of developing standards to guide development

[4] See Gladstone and Miller [4].

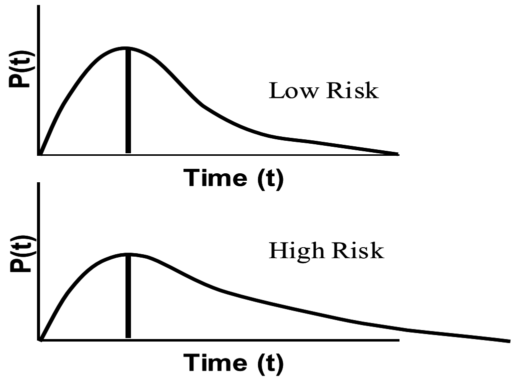

Figure 5. Weibull Distributions

For this analysis, we modeled the specification phase as a moderate risk Weibull, the standards phase as a uniform distribution, the implementation phase as a moderate risk Weibull, and the centralized integration phase as a moderate risk Weibull. We also selected a mode for each Weibull distribution to specify its measure of central tendency. Table 2 summarizes the parameters for each activity.

The dashed line connecting the system-wide standard decision point to the A-B-C box on the lower line of Figure 4 depicts the transfer to a local development process after failed attempts to develop centralized standards. As in the centralized standard scenario, we model the local implementation with a moderate risk Weibull, but with a slightly higher mode to account for the need to develop local standards before initiating implementation. We model integration differently, to account for the difference in risk due to the need to integrate across different standards, and for a probability of significant rework. The initial integration activity is modeled with the same moderate risk Weibull as in centralized development, but because we are integrating systems developed under different standards, we estimate a probability of 0.50 for rework. If rework is required, we model its duration as a moderate risk Weibull with a mode of 3.5. This is due to the expected need to rework some of the local standards and implementations. As a result, if the initial developments are not compatible, we expect the duration of rework to be about half as long (measured by mode of the distributions) as that of the initial standards phase.

We generate statistics on the distribution of possible duration times by conducting a Monte Carlo simulation for each of the development scenarios. For each scenario, we generated 5,000 runs to create the probability distribution of total duration time.

Table 2. Simulation Parameter Values

Phase	Distribution	Risk Level	Weibull Mode (years)
Define Specifications	Weibull	H	1.75
Define Centralized Standards	Uniform	L(3-4) M(4-5) H(5-6)	
Implement Standard Locally (3X)	Weibull	M	5.25
Integrate System Standard	Weibull	M	2.00
Define & Implement Local Standards	Weibull	M	7.00
Integrate Local Standards	Weibull	H	2.00
Rework Local Standards	Weibull	M	3.5
Integrate Reworked Standard	Weibull	M	1.00
Rework Probability	0.50		
2nd Rework Probability	0.25		
Additional Rework Probability	0.125		

Table 2 lists the principal parameter values for the simulation runs. A few of these values require additional discussion.

- Define specifications: we model this as a high risk activity when developing centralized standards, and moderate risk if it is solely a local activity
- Define Centralized Standards: as discussed above, this is analyzed using three

uniform distributions to reflect varying degrees of political interference

- Implement Standard Locally: when using centrally-defined standards, this phase is modeled as a Weibull distribution with a mode of 5.25 years; note that this process occurs simultaneously and independently at three locations

- Integrate System Standard: we assume that when working with previously-agreed to centralized standards, this can be represented as a moderate risk Weibull distribution with a mode of 2.0. The completion time includes rework time

- Define & Implement Local Standards: we model this process as a Weibull distribution with a mode of 7.00 to account for the need to develop and refine local standards after the failure of the centralized effort

- Integrate Local Standards: this phase is modeled as a high risk Weibull distribution with a mode of 2.0, to account for some probability that the lack of common standards can be overcome within a reasonable period of time

- Rework Local Standards: if the initial integration fails because of incompatible systems, significant rework is required, so we model this phase as a Weibull distribution with mode of 3.5

- Rework Probability: we expect a high probability of 0.5 of rework when trying to integrate systems that have not specifically identified the need, and allocated the resources, to ensure compatibility

- Integrate Reworked Standard: with sufficient time for rework of the local standards and implementations, integration should be relatively straightforward

- 2nd Rework Probability: this activity is included to allow for missteps in the development of revised standards and their local implementation, since revising work frequently generates additional problems

An Example of Centrally-Developed Standards

Figure 6 shows a sample of the model output for the case of centrally developed standards under the moderate risk assumption.

As Figure 6 shows, the centralized development scenario demonstrates significant dispersion in the duration distribution, with a mean of 15.9 years and a median of 15.8 years. The range of outcomes is substantial, however, with a 20% probability of a duration exceeding 17 years. In comparison, if we assume a stronger political consensus to achieve common standards and specifications, the mean duration falls to 14.9 years and the probability of exceeding 18 years falls to less than two percent. If we allow for greater difficulty reaching consensus, but assume that the process will eventually converge by the end of six years, the mean duration increases to 16.9 years, and a 20% chance of exceeding 18 years to complete.

Results for the Typical European Approach and Local Development

Let us now examine the estimated duration in the case in which central standards development is abandoned after a few years of failed attempts and the States proceed with their own independent programs (Scenario 3). In that case, shown in Figure 7, the mean duration increases to 21.0 years with an enormous variability: the simulation generates a 30% probability of exceeding 23 years in development.

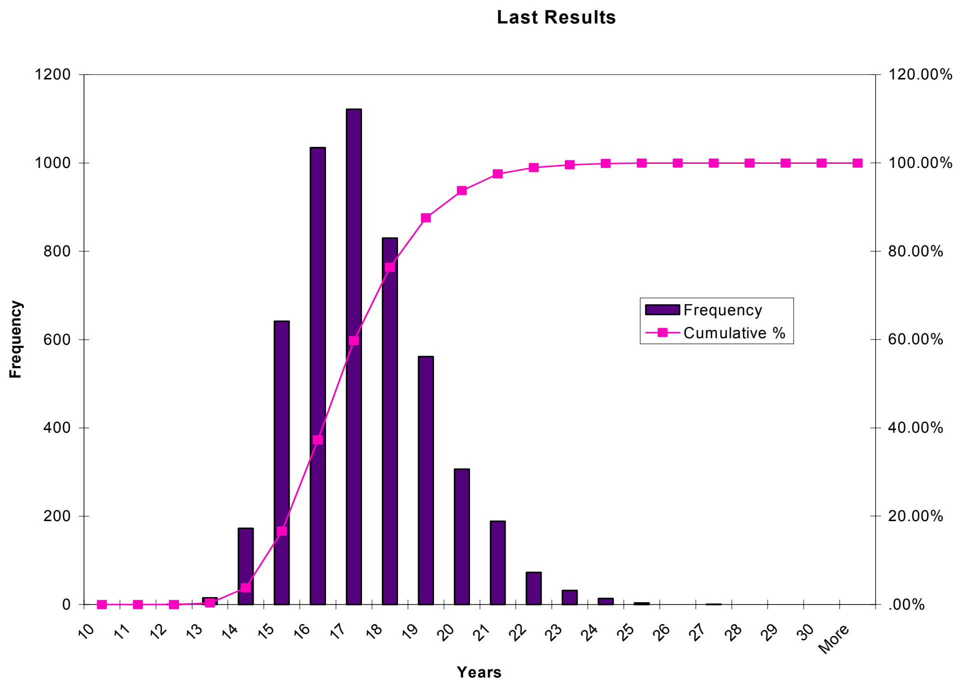

Figure 6. Simulation Results for Moderate Risk Under Centralised Development (Scenario 1)

To some extent, this case is the worst-case scenario since it includes the time wasted in the attempt to develop centralized standards. We analyzed a hypothetical case in which we eliminated the central development phase, but replaced it with a shorter phase (modeled as a uniform distribution over 0.5-1.0 years) in which the developers would share information and objectives, but not attempt to develop common standards. (Scenario 2) We also included the initial activity to develop a common set of system specifications.

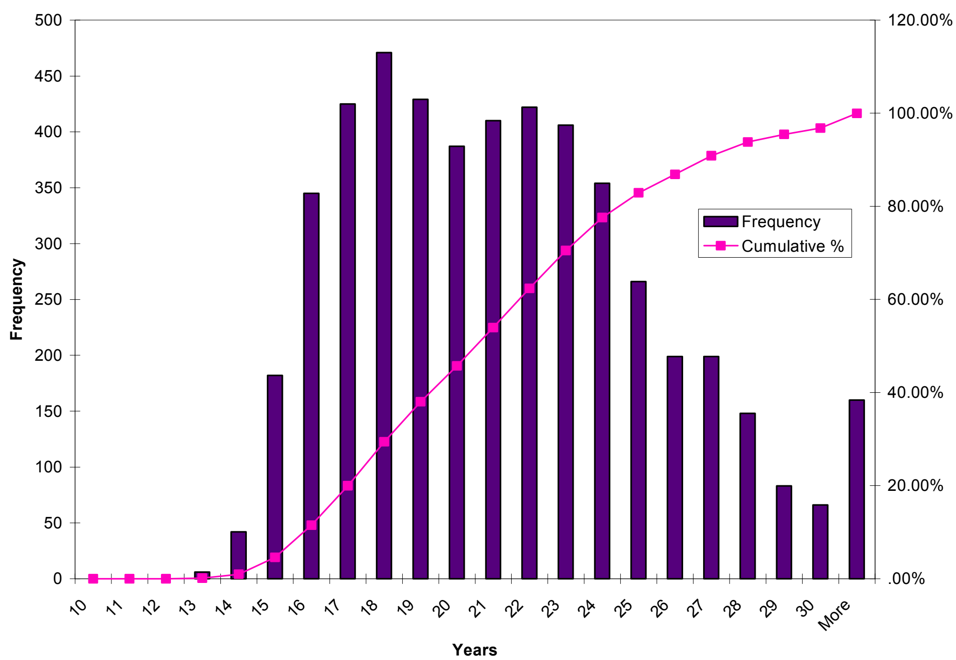

Figure 7. Simulation Results Under a "Typical European Approach" (Scenario 3)

The results for that hypothetical case are shown in Figure 8. The mean duration time now falls to 18.7 years, but with substantial variability on the upper end. The probability of the duration exceeding 22 years is 20 percent, and the chance of taking more than 26 years is over 5 percent.

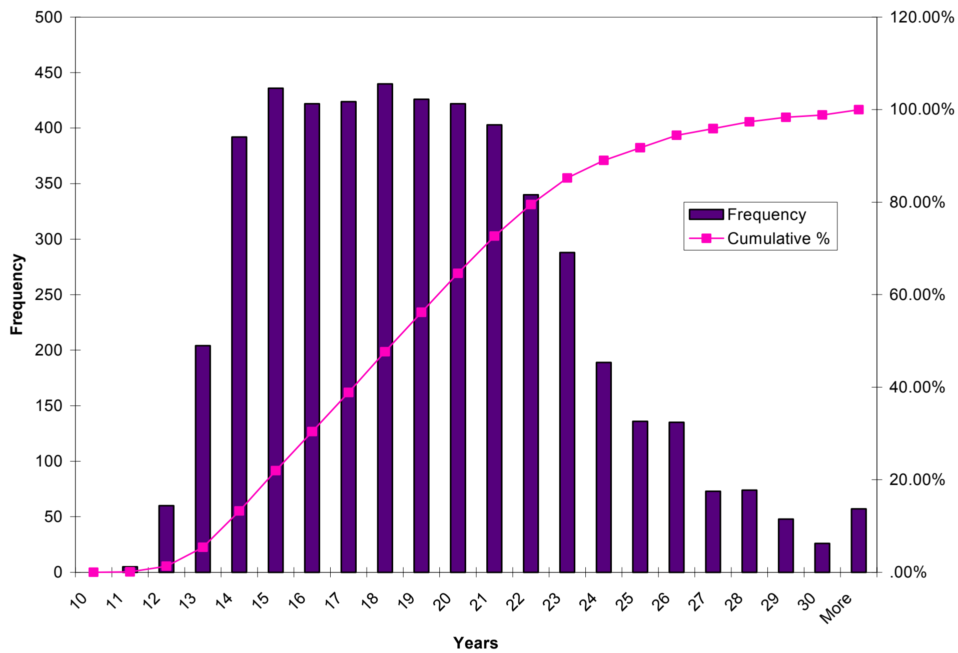

Figure 8. Duration for a Localised and Independent Approach (Scenario 2)

The results of our findings are summarized in Table 3. We characterize the different scenarios by the mean duration and the range for duration outcomes within the span of the middle quartiles of the estimated distribution (the 25-75% range). The findings clearly illustrate the implications for different development strategies on the predicted time-to-complete. Although the difficulties in reaching agreement on common standards may be severe, success in that activity serves to significantly reduce the risk and duration of a complex development project that requires integration across political entities. By reaching agreement on common standards during the preliminary stages of the development project, managers reduce their technical risk in later phases. Moreover, they reduce the technical risk in the system integration activities that are inherently more risky, especially when time and resources are not devoted at the design stage to clearly specify the standards.

Clearly, the results are strongly affected by the assumptions we made about the relative risk of different activities. In particular, we modeled the integration phase differently depending on whether the systems being integrated were developed under clearly specified common standards. Without those standards, we believe that the probability of significant rework is high, and that the likely duration of the rework phase will be a substantial fraction of the original development time. Regardless of the particular probabilities used in the analysis, this feature of the development process is well documented. The decision to avoid difficult design choices in the early stages of a project, whether motivated by technical, cost, or political considerations, is known to incur significantly higher costs later in the development process.

Table 3. Summary of Results

Scenario	Mean Duration (Years)	25-75% Range (Years)
Centralised Approach (Scenario 1)		
Low Risk	14.9	14 -16
Moderate Risk	15.9	15 - 17
High Risk	16.9	16 - 18
Local and Independent Approach (Scenario 2)	18.7	16 - 22
Typical European Approach (Scenario 3)	21.0	18 - 24

Our simple example, although its underlying assumptions are, of course, debatable, illustrates clearly the added-value, for a Europe-wide implementation of ATM systems, of a coordinated approach. It also shows that the median or indecisive scenario is the worst case.

Application of the GAN Method

The example we presented in this paper is a relatively simple application of the generalized activity network analysis approach. To keep the analysis simple, we modeled only one phase with rework iterations. In many ATM projects, there are testing and integration milestones that must be attained before development continues, which implies multiple points at which rework may be required. More generally, the GAN method allows for modeling the branching process common to many development programs, in which the outcome of a test or specific research results (or possibly a funding decision by political authorities) affects the choice of alternative to follow. The method is particularly well-suited for analyzing the implications of choosing alternative development strategies. For example, ATM system improvements often require changes in several capabilities. However, it is usually not known beforehand whether all of the capabilities should be developed in parallel and then joined at the end", or whether smaller incremental improvements pursued jointly provide for less risk.

In addition to modeling alternative development strategies, this paper suggests the value of expanding the assessment of ATM projects to modeling the development process itself. The history of delays, schedule slips, and other complications is no different in ATM than other areas.[5] Better understanding of the development process and the corresponding risks to schedules and budgets could lead to better mitigation strategies and more accurate estimates for funding requirements.

References

[1] Elmaghraby, Salah, 1977, *Activity Networks: Project Planning and Control by Network Methods.*

[2] Belcher, Gerald, John Dukovich, Bobby Jackson, and David Lee, May 2000, *Better Development Cost Estimating Through Modeling and Simulation*, LMI Report PA903T1, McLean, Virginia.

[3] Lee, David A., February 2001, *Generalized Activity Networks for Cost Estimation and Program Management*, 34th Annual DoD Cost Analysis Symposium, Williamsburg, Virginia.

[4] Gladstone, Brian, Steve Miller, 2002, *Chemical Demilitarization Program Schedule Risk Assessment*, 35th Annual Department of Defense Cost Analysis Symposium, January 2002, Williamsburg, Virginia.

[5] In the United States, the Government Accounting Office provides a steady supply of reports on government projects at the FAA and other agencies that fail to meet their planned budget and schedule.