Airport 95: Automated Baggage System?

A. John Swartz
1915-B Alex Avenue
Austin, TX78728
alexander.swartz@polaris.dhs.state.tx.us

Abstract

The Denver International Airport automated baggage system was a major news story spanning the years 1994-95. Reconstruction of the events of the project management of this system serves as an example of project summary reporting, which is stipulated in every project management methodology, but which is seldom or never done. The author provides sufficient detail to enable simulation of the design approach alternatives. If other projects are reported in the same format, it will be possible to compare projects on a design phase and/or event-by-event basis. The author recommends establishment and maintenance of a knowledge base of specific causes for failed software development projects.

Background

On 28 February, 1995 the opening of the Denver International Airport represented the first new major airport since Dallas/Fort Worth in 1975. The DIA automated baggage system was a major news story of 1994/1995. In the Open Channel column of the February, 1995 issue of the IEEE Computer magazine, I advocated further investigation into the problems associated with this project.

The DIA project is of interest because of its use of contemporary design concepts and its reasonably bounded scope. (See the sidebar.) It is a large-scale, real-time distributed system, not a batch environment sited on a monolithic mainframe. It is an example of program-as-component (see figure 1). It is a component of a megaproject–the characteristics of a megaproject are: capital cost exceeds $1 billion, stretch available resources to the limit, having a high profile within the sponsoring agency, and success is crucial to their sponsors [1]. It is characteristic of the systems that will be developed in the next few years. Lessons learned from this project can be immediately folded into future developments–especially airport baggage systems presently in the industry backlog.

Every project management methodology directs us to create a written summary of the lessons learned from the project itself. Due to non-allocation of resources ("there is no project charge code for this") or to an organizational non-disclosure mentality, this is seldom or never done. If a project is documented, it is in the form of a generic case study presented as a glowing report of success. Any mistakes made in the course of the project are glossed over or simply not reported.

We only learn lessons when mistakes are made, documented, reported and analyzed. The details of failed project are extremely difficult/impossible to obtain. It is no participant's interest to part with this information.

Writers are seeking the parallel between the current state of software engineering and the history of the larger engineering profession. Today, software engineering is at the point where the turbojet airliner industry was in 1954. Our showcase projects are not succeeding! As a profession, our course of action is to emulate the engineers from de Havilland: analyze the results and report objectively.

The role of program-as-component arises in large heterogeneous systems. Such systems include programs in multiple languages for significantly complex hardware systems; they may have mechanical constraints, produce noisy data, or impose real-time constraints on operation. To capture the nature of this shift of attention, we can consider the same attributes as before:

- *Characteristic Problems:* The major focus of design is shifting from algorithms and interfaces to the integration of the system as a whole.
- *Dominant Data Issues:* We need integrated databases that include not only symbolic and numeric information, but also information about the physical status of the system that may in fact be a physically distributed – in which communication is a very significant issue.
- *Dominant Control Issues:* Software must now provide control over complex systems that may include data subject to physical or mechanical constraints as well as the usual purely symbolic data.
- *Specification Issues:* Software specifications must address interfaces with non-software elements of the system as well as with other software elements.
- *Character of State Space:* The state space of a large heterogeneous system may be very large. In addition, the structure may be dynamically reconfigured, and it may contain physical elements as well as symbolic elements.
- *Management Focus:* The heterogeneous character of these systems increases demands on management to coordinate design, development, construction, and integration schedules which have very different characteristics.
- *Tools and Methods:* These systems require real-time control and interfaces for lay users; they must be capable of running complex control problems with very little human intervention.

Figure 1. Attributes of Program-as-Component [2]

The purpose of this study is not to highlight flaws in the DIA automated baggage system or the Denver Airport Authority. Any and all failures represent failures of the larger software engineering community. If this were not so, then in the summer of 1994, when DIA appealed in the press for anybody who knew how to make the system operational, somebody would have stepped forward with the solution to the problem and reaped the rewards. This investigation concerns the practice of project management itself.

Think of this report as sort of an autopsy: it is not pretty, but if we do it right now, we will not have to repeat it in the future.

The software engineering body of practice will gain from the reporting of information concerning one or more failed projects. The present study is intended as a prototype for future investigations. If future studies use a similar format, the measures will be comparable.

History of the Automated Baggage System [5]

When DIA's baggage handling system was planned in 1990, the consultants recommended carts pulled by tugs for the long hauls and belt conveyors for the shorter distances. At that time, United Airlines was opposed to the new airport and had not agreed to occupy the big Concourse B designed for them. In early 1991, United agreed to sign but stated a requirement for a baggage handling system that would enable aircraft turnaround in 35 minutes despite the fact that the gates are more than a mile from the terminal.

Neither tug-and-cart nor belt conveyors could meet the time requirement.

United recommended BAE Automated Systems on the basis of a demonstration of a prototype system.

Since the Denver city charter mandates a bidding process, system requirements were defined and the specifications sent out for bids.

Formal Proposals [5]

A bid was prepared by a consortium of Harnishfeger Engineers, Inc., Syscon Corporation, UTDC, CCC-Pentek and ElectroCom Automation. The Harnischfeger-led consortium received a request for proposal in July, 1991. The company spent $150,000 to prepare a proposal and submitted it in mid-September. The bid had the following characteristics:

- Throughput that met the throughput requirements of the city's request for proposal.
- A destination coded vehicle (DCV) system with an installation record more current than BAE's San Francisco job of 16 years earlier.
- Experience in integration of large-scale material handling systems Conveyor expertise in baggage handling systems.
- (Most critically) enough depth in empty cart management software to run DIA's baggage handling system, and them some.

This bid came from a consortium of Harnischfeger Engineers, Inc., Syscon Corporation, UTDC, CCC-Pentek and ElectroCom Automation. The Harnischfeger proposal is a careful mix of manual handling, high-speed transport and reliable sortation.

As a systems integrator, Harnishfeger Engineers had completed more that 300 material handling projects, several of equal or greater complexity than the DIA baggage handling system.

Syscon, a player in big-league software projects in government and industry, was to provide computer hardware/software, configuration management and training.

UTDC was to provide the multi-bag DCVs for the system. UTDC completed such a system for Changi International Airport (Singapore) in the early '90's. (BAE's last car system was San Francisco in 1978.)

ElectroCom Automation, a major supplier of sortation systems for parcels and baggage for government, postal service and industry, would provide sorters that would read the bar code labels on the bags and distribute them to gates.

CCC-Pentek, veteran supplier of airport material handling equipment and was the conveyor arm of Docutel when the early car systems were installed, would provide the input and take-away conveyors needed in the system.

The system was to function as follows: at check-in, outbound bags receive bar code tags and are conveyed to the baggage loading area, where they are manually loaded onto DCVs, up to 10 bags per vehicle. The empty cart management software directs the vehicle to the concourse of the airline doing the check-in. The DCV takes bags dumped off at United to the United concourse.

On 4 November, 1991, the airport baggage handling system evaluation committee rejected the bid. "your proposal did not fully respond to the operational requirements of the request for proposal (RFP) in that the set forth criteria for a full integrated, automated baggage handling system were not clearly demonstrated to the satisfaction of the evaluation team." [5]

BAE Design Approach [3]

The design is an automated system based on single-bag destination coded vehicles (DCV's). BAE's name for its system is Telecar. The vehicles are referred to as cars or carts. In proposal language they are called destination coded vehicles (DCVs).

- Bags are conveyed from check-in
- Scanners read the bar code labels as the bags are being conveyed.
- Data from the bar code scanner are processed to a radio frequency identification transponder mounted on a car that is barrelling into the loading area.
- The car is loaded on the fly and is directed to its destination gate by the radio frequency identification transponder.

DIA's baggage handling system is centered on track-mounted cars propelled by linear induction motors. The cars slow down, but do not stop, as a conveyor ejects bags onto their carrying platform. During baggage transfer, a scanner reads the bar coded label on the bags and transmits the data through a programmable logic controller to a radio frequency identification tag on a passing car. Now the car knows the destination of the bag it is carrying, as does the computer software that routes the car to its destination.

Design Issues [3]

Cars propelled by linear induction motors.
In the Denver system a fin on the bottom of the car passes through a slot in the motor mounted under the track. The electromagnetic field of each motor drives the car forward to the next motor. Fast, straight-ahead travel is no problem. Problems develop when stops, slowdowns and diverts are built into a high-speed system that has plenty of curves, inclines and declines. The use of vertical friction wheels and permanent magnets to fine-tune a linear induction motor system is more an art than a science in a system the size of DIA.

Bar code scanning and radio frequency identification.
Handing off information from bar code scanners to a radio frequency identification system has been a proven technique in industrial material handling systems. It has never been tried in a high-speed baggage handling system that employs conveyor-to-car transfers on the fly.

Empty cart management software.
The DIA car-on-track system, is a high-risk automation approach in which the whole system must respond in real time to an incredible number of questions and commands generated by the empty cart management software.

Recommendations by Consultants. [5]

The airport baggage handling system evaluation committee was aided in its decisions by consultants from Breier Neidle Patrone Associates. In 1990, a BNP study stated that the advantages of a multi-bag DCV system include:

- Minimum processing times, providing essentially equal service to all concourses
- High throughput capacity
- High reliability
- Oversize baggage–skis, golf clubs–capability
- Minimal development risk
- Availability from two proven suppliers: BAE and UTDC.

In 1990, Breier Neidle Patrone Associates further stated: "With regards to the single-bag DCV, considering the prototype state, we strongly feel it is not capable of being implemented within the project schedule." [5]

Contract Award

In early 1991, before the decision to make baggage handling an airport-wide system, United signed a contract with BAE to design a Telecar system for its concourse. In December, 1991 the city gave BAE $20 million contract to install Telecar track. In June, 1992, BAE was awarded a $193 million sole-source contract, versus the $160 million bid by the Harnischfeger consortium [5].

Project Management Paradigm

Since the first event in the project was a prototype demonstration, the project management form is Rapid Prototyping or Rapid Application Development (RAD). Since the demonstration clinched the contract award, the tacit agreement was that "we essentially have the required system and we only need to tinker with the software."

System Hardware Components

The automated baggage handling system comprises two main components:

1. high-speed, bag-carrying telecarts mounted on tracks
2. connecting conveyor belts to load and off-load baggage.

The tracks are suspended from the basement ceilings of the terminal and concourses. Electric motors and synchronous drives move the telecarts along the tracks at varying speeds. Photocells and radio frequency reading devices direct each telecart to the right location. In total the original system included:

- over 17 miles of track
- 5.5 miles of conveyors
- 4,000 telecarts
- 5,000 electric motors
- 2,700 photocells
- 59 laser bar code reader arrays
- 311 radio frequency readers
- over 150 computers, workstations, and communications servers [8].

The computers comprise 30 Texas Microsystems Inc. 386 and 486 PCs running customized control software. BAE used Microsoft Windows on workstations and IBM OS/2 on servers. A fault-tolerant database server from Vista Corp. tracks baggage movement and confirms the delivery of carts at particular sites [9]. The physical layout of the DIA automated baggage system is shown at figure 2.

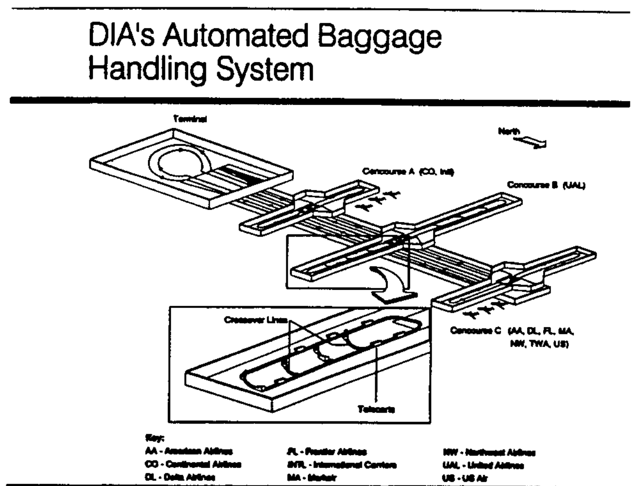

Figure 2. DIA's Automated Baggage Handling System [8]

Events subsequent to Contract Award

In May, 1992, the airlines and the city ordered a major revision of the automated baggage system while it is under construction [7].

21 October, 1992: a BAE superintendent complained another contractor was denying his crews access to the work site. Infighting continued through 1993 [7].

Automated baggage handling system integration tests ran in April, 1994 and July, 1994. The July test used a 1,000-foot loop and 200 cars; the system was supposed to run ten hours a day for two days. On both days the baggage jams were so frequent that the tests had to be called off [4]. The test results are presented at figure 3.

Problems with DIA's Automated Bagge Handling System
Significant mechanical and software problems have plagued the automated baggage handling system. In tests of the system, bags were misloaded, were misrouted, or fell out of telecarts, causing the system to jam. Video cameras were installed at several known trouble spots to document problems, such as the following:

- The baggage system continued to unload bags even thought they were jammed on the conveyor belt. This problem occurred because the photo eye at this location could not detect the pile of bags on the belt and hence could not signal the system to stop.
- The baggage system loaded bags into telecarts that were already full. Hence, some bags fell onto the tracks, again causing the telecarts to jam. This problem occurred because the system had lost track of which telecarts were loaded or unloaded during a previous jam. When the system came back on-line, it failed to show that the telecarts were loaded.
- The timing between the conveyor belts and the moving telecarts was not properly synchronized, causing bags to fall between the conveyor belt and the telecarts. The bags became wedged under the telecarts. This occurred because telecarts were bumping into each other near the load point.

Although the contractor, BAE Automated Systems Incorporated, believes that these problems have been resolved, other problems remain. To resolve these problems and make the system operational, a number of critical tasks must be completed. A recent BAE system status report listed 72 hardware, software and testing activities that must be completed such as the following:

- The telecarts' front bumpers have to be replaced so that they will not slip under the back bumpers when telecarts collide. These collisions have caused telecarts to lock together.
- Additional track, synchronous drives, and related software changes must be installed to improve "empty car management," that is, allocating the correct number of telecarts to specific locations at appropriate times.

BAE and City of Denver officials recognize that the system's testing might uncover additional problems.

Figure 3. DIA Automated Baggage System Integration Test Results [8]

In September 1994, the German consulting company Logplan submitted a 12-page report that concluded: the BAE automated car-on-track system can be made to work in maybe five months or possibly up to two or three years. Logplan admitted that it was unable to investigate control systems and operating strategies because its questions "have not been answered completely" by BAE [6].

At the time of airport opening, only a small-scale version of the automated baggage system was running at just one concourse. This reduced scale system ran at a 90% success rate. "That means 10% of the bags did not go to the right place" [9].

Findings

The success of the airport hinged on the automated baggage system at **two distinct times** in the construction process: in July 1991, United became a tenant airline only on the guarantee of the automated baggage system; the unavailability of the automated baggage system delayed the opening of the airport.

The May, 1992 massive system redesign adversely affected project completion schedules.

The October, 1992 and subsequent "turf wars" among the contractors adversely affected project completion schedules.

The scale and scope of the integration test (1000 feet of track/200 cars) was between 1% and 5% of the entire system (17 miles/4000 cars). The contract was awarded on the basis of a prototype demonstration. One wonders what was the scope of the prototype demonstration; I.e., how much trackage and how many cars were involved? And beyond hearing the sound of running machinery, what exactly happened?

Between contract award and integration testing, none of the hardware design issues were resolved. Between contract award and integration testing, the software design issue was not resolved.

Conclusions

The contracting agency was not insulated from the consequences of the May 1992 massive system redesign.

The contracting agency was not insulated from the consequences of the October 1992 and subsequent inter-contractor "turf wars."

Recommendations

When a system redesign occurs after start of contract, an adequate risk analysis should be performed.

The contracting agency and/or the prime contractor should maintain active control over adverse interactions between contractors.

Extension

This paper provides sufficient information to enable simulation of design alternatives. Recommend that a simulation be undertaken to assess design alternatives.

This study can serve as a template for future analyses of failed major software development projects.

When several such studies have been completed, a knowledge base of causes for project failures should be established and maintained by the software engineering community.

If the software engineering profession does not create a knowledge base of causes for project failures, projects will continue to fail for **the same reasons**–and nobody will know **what those reasons are**!

Freedom of Information Act Requests

In researching this report, I sent Freedom of Information Act requests to various participants in the baggage system development effort. I requested the following data items:

- A copy of the Logplan company report
- System proposal
- System specification
- All system design documents
- All planning documents including milestone dates
- Status reports by BAE
- Status reports by the United Airlines project team
- Status reports by the Denver Airport Authority

- Source code listings
- BAE Organization Charts
- BAE staff resumes
- Project Invoices
- Project man-hour reports by project phase
- A list of agencies investigating the DIA baggage system development
- All transcripts and reports of investigations of the DIA baggage system development.

The agencies to which I sent these requests and their responses follow:

- U. S. Department of Transportation: letter of acknowledgment but no information.
- U. S. General Accounting Office: letter of acknowledgment but no information.
- U. S. Securities and Exchange Commission: for a fee, they reproduced their news clippings file, but I was given no investigatory information.
- Denver Airport Authority: no response.
- United Airlines: no response.

References

1. E. Merrow, Understanding the Outcomes of Megaprojects, RAND Corporation, 1988, p. 3.
2. M. Shaw, Beyond Programming-in-the-Large: The Next Challenges for Software Engineering, Software Engineering Institute, 1986, p.14.
3. B. Knill, Flying blind at Denver International Airport, Material Handling Engineering, Vol. 49, No. 7, July 1994, p. 47.
4. B. Knill, Denver International Airport has happened before; failure of materials handling systems, Material Handling Engineering, Vol. 49, No. 8, August 1994, p. 7.
5. B. Knill, Reality checks into baggage handling system; Denver International Airport , Material Handling Engineering, Vol. 49, No. 8, August 1994, p. 51.
6. B. Knill, New baggage handling solution ... sort of; Denver International Airport baggage system, Material Handling Engineering, Vol. 49, No. 10, September 1994, p. 43.
7. K. Flynn, How the Baggage System was Botched, Rocky Mountain News, Vol. 100, No. 1, 29 January 1995, pp. 26A-29A.
8. GAO/RCED-95-35BR, New Denver Airport Impact of the Delayed Baggage System, U. S. Government Accounting Office, October 1994, pp. 14-17.
9. B. Violino and P. Kapustka, Is It Really Ready to Fly?, Information Week, Issue 518, 13 March 1995, p. 86.
10. C. Pancake, The Promise and the Cost of Object Technology: A Five-Year Forecast, Communications of the ACM, Vol. 38, No. 10, October 1995, p. 42
11. ———, Denver International Airport Scheduled to Open, CNN Video Archives, Cassette No. 87-3752, Story No. 4, 17 February 1994.

Sidebar → Why a case study?

In the Training Issues section of a recent object-oriented roundtable discussion, a panel member commented "One problem is the lack of good didactic examples based on real-world needs ... I'm getting tired of ATM examples and the vending machine. It is something I can read and say 'well, yes, I see how they did that.' But when I go back to do something in a manufacturing setting–or if I'm going to automate an oil refinery–it doesn't mean I'm going to automatically make the transition." [10]

In contrast to the glowing project reports that constitute the contemporary software engineering case study, the classic business school case study is a medium of discourse. It is a presentation of good methodology and, just as important, bad methodology. A valid project summary report possesses these same attributes: here is the project report, these are the parts that went well and these are the parts that went badly and why.

In the staff level training of senior military officers, historical and hypothetical tactical situations (scenarios) are used to assess and extend the participants' abilities. The participants solve these scenarios in a seminar context. After the group present and discuss the solutions, a "school solution" is offered. Rarely, a student arrives at a superior solution which then becomes the "school solution". After a time, familiarity with a small set of these studies is part of each military practitioner's professional repertoire. Some of this "institutionalized medium of discourse" is currently being developed by the patterns people. Patterns are more abstract than a case study.

In software engineering, we have a set of these established "medium of discourse" problems. I am thinking of the "bank of elevators" example and Dijkstra's dining philosophers.

The problem may purposely be unsolvable. NP complete and the Strategic Defense Initiative ("Star Wars") problems from the 1980's are examples in software engineering.

The quality of the problem/case study must be commensurate with the caliber of the group asked to work on it. I propose that the DIA Baggage System is such a problem. It has already established its pedigree in that nobody has solved it and everybody has something to say about it.

The DIA Automated Baggage System is a Swiss army knife of a case study in that it is a microcosm of every type of problem. It has the advantage of being a flawed historical development project. The historical attribute is important in that when the analysis gets specific, findings and conclusions cannot be discounted on the basis that the context is hypothetical. The flawed attribute is important in that one is reluctant to critique component phases of a successful system. In the case of a faulty system, one is willing to examine each component of the system as the potential weak link.

Finally, the DIA Automated Baggage System has a visual component: there are videotapes of the failed baggage handling tests [11].