Softwarefehler in der Logistik am Beispiel des Denver Gepäcktransportsystems

Prof. Thomas Huckle
Institut für Informatik
Technische Universität München

Tacoma Bridge

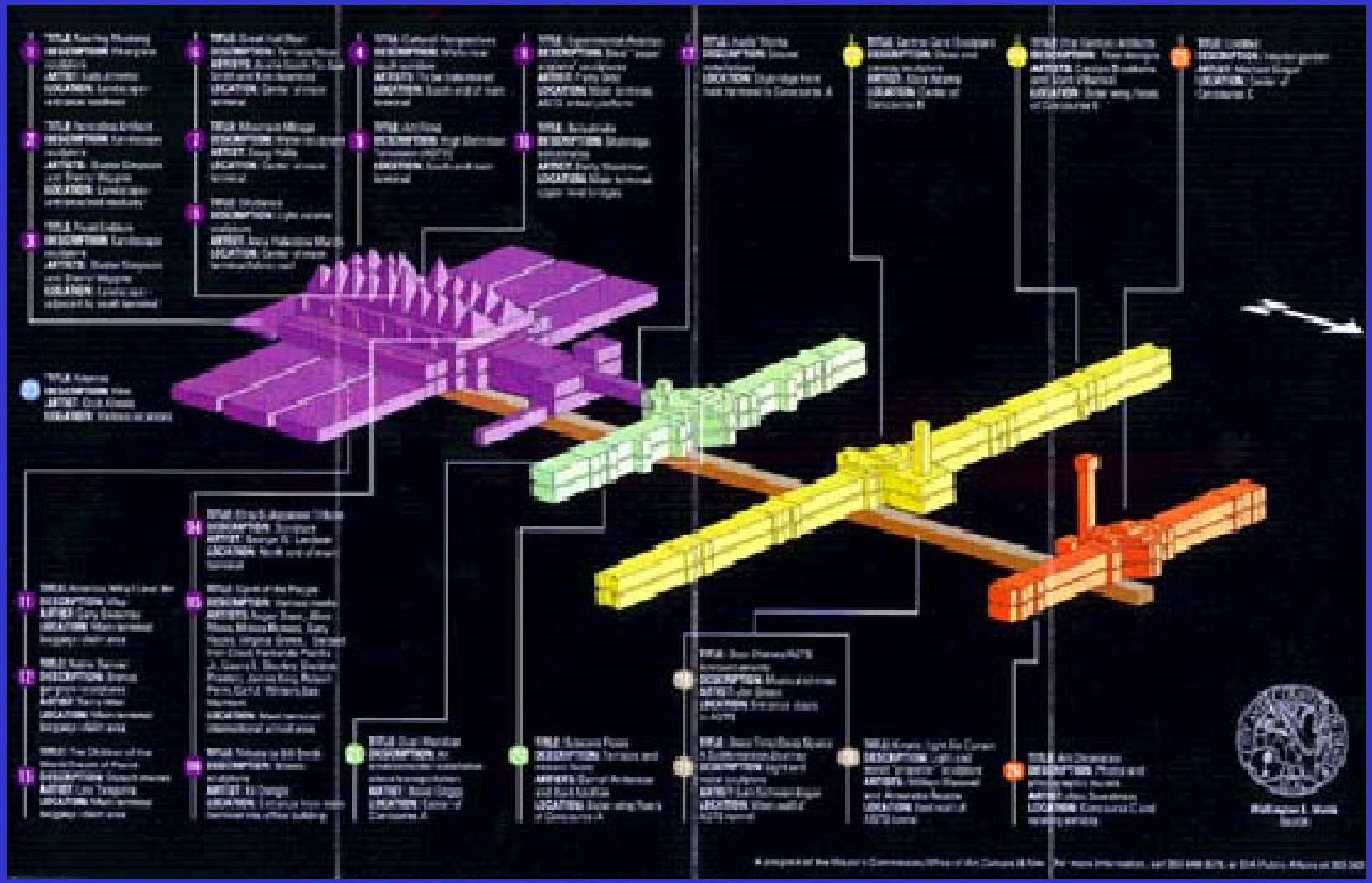
London Millenium Bridge

Denver International Airport

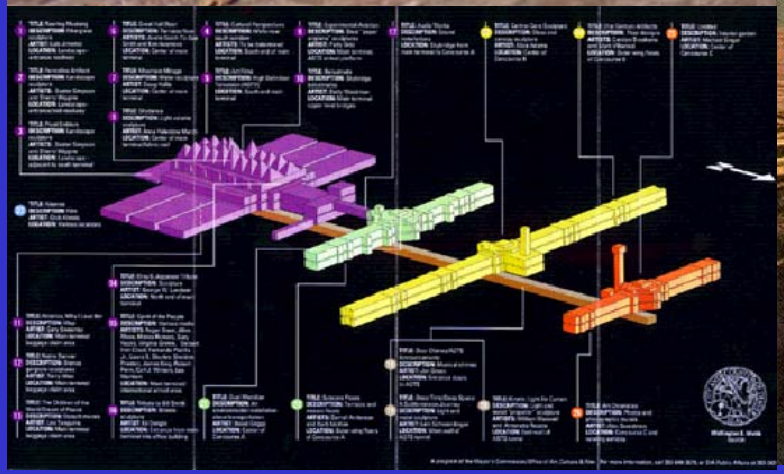
Bau eines Flughafens in Denver, Beginn 1989

- Denver International Airport (DIA)
- Mayor Webb: "This project is of the same magnitude as the Panama Canal or the English Channel Tunnel"
- Flughafen der Superlative

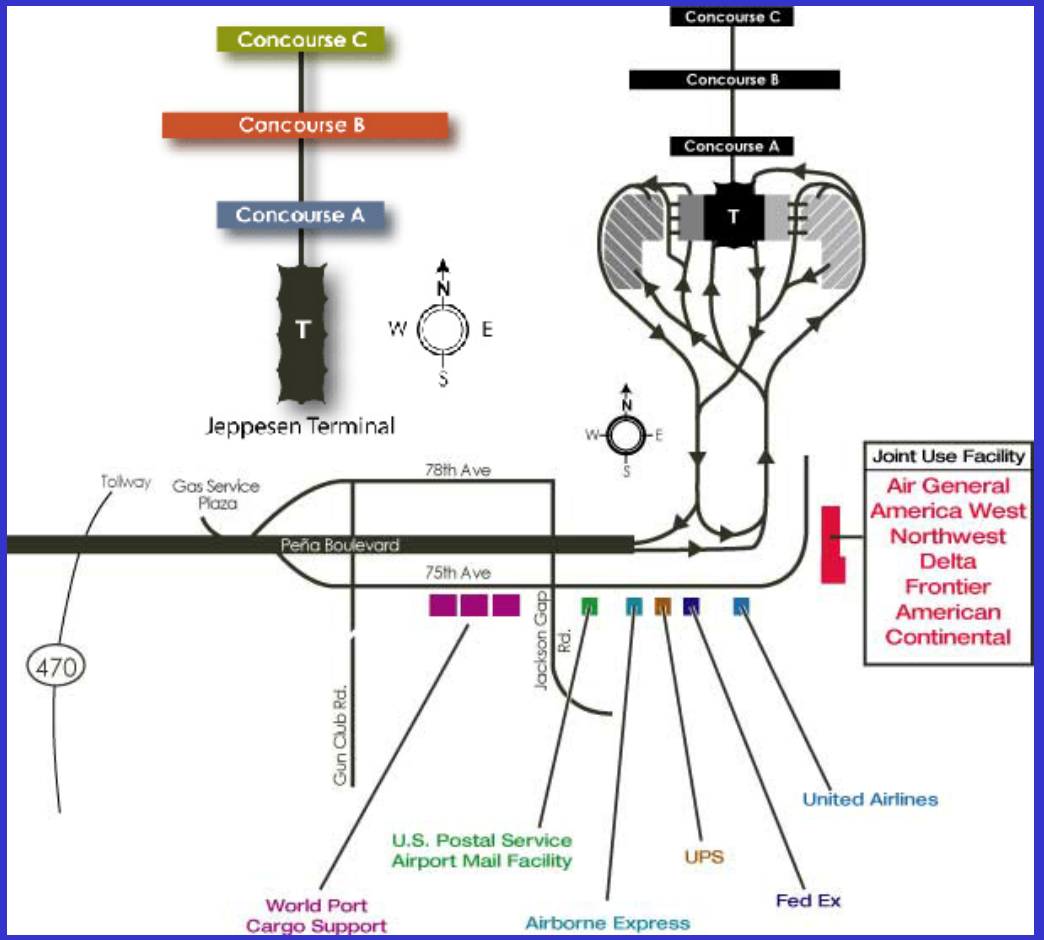
Modell des Flughafens

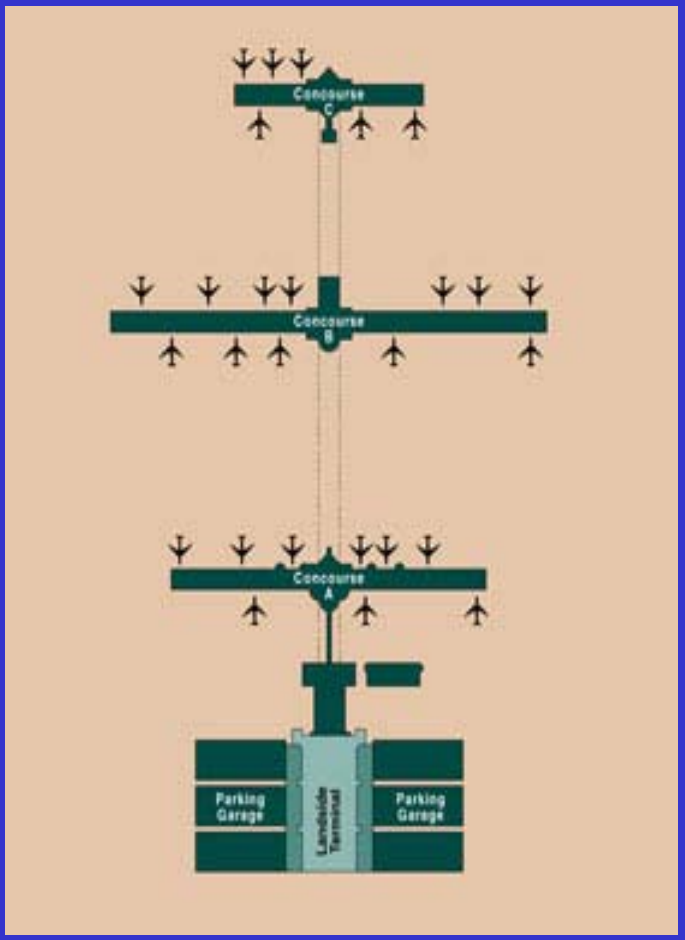


Denver International Airport



Denver International Airport





Denver International Airport

Fläche: 53 Quadratmeilen
Grundsteinlegung: September 1989

Geplante Eröffnung: 31. Oktober 1993
Eröffnung: 28. Februar 1995
(16 Monate Verspätung)

Geplante Kosten für das Transportsystem: $ 193 Millionen
Tatsächliche Kosten: $ 311 Millionen

United Airlines

- größte Fluggesellschaft des Flughafens
- UA entscheidet sich für ein automatisches Gepäcktransportsystem
- Dezember 1991: United Airlines beauftragt BAE Automatic Systems, Inc. zum Bau des Systems

Aber: Andere Fluggesellschaften unternehmen kaum Anstrengungen in dieser Richtung.

Jede Fluggesellschaft sollte zunächst selbst für ihr Gepäcktransportsystem verantwortlich sein.

Zwei Jahre später...

Wunsch nach einem gemeinsamen automatischen Transportsystem für den ganzen Flughafen

- Drastische Verkürzung der Bodenzeiten
- Schnellere Gepäckausgabe
- Manuelle Sortierung ist zu arbeitsintensiv
- Überwindung der großen Distanzen am DIA

BAE bekommt Auftrag für flughafenweites Transportsystem

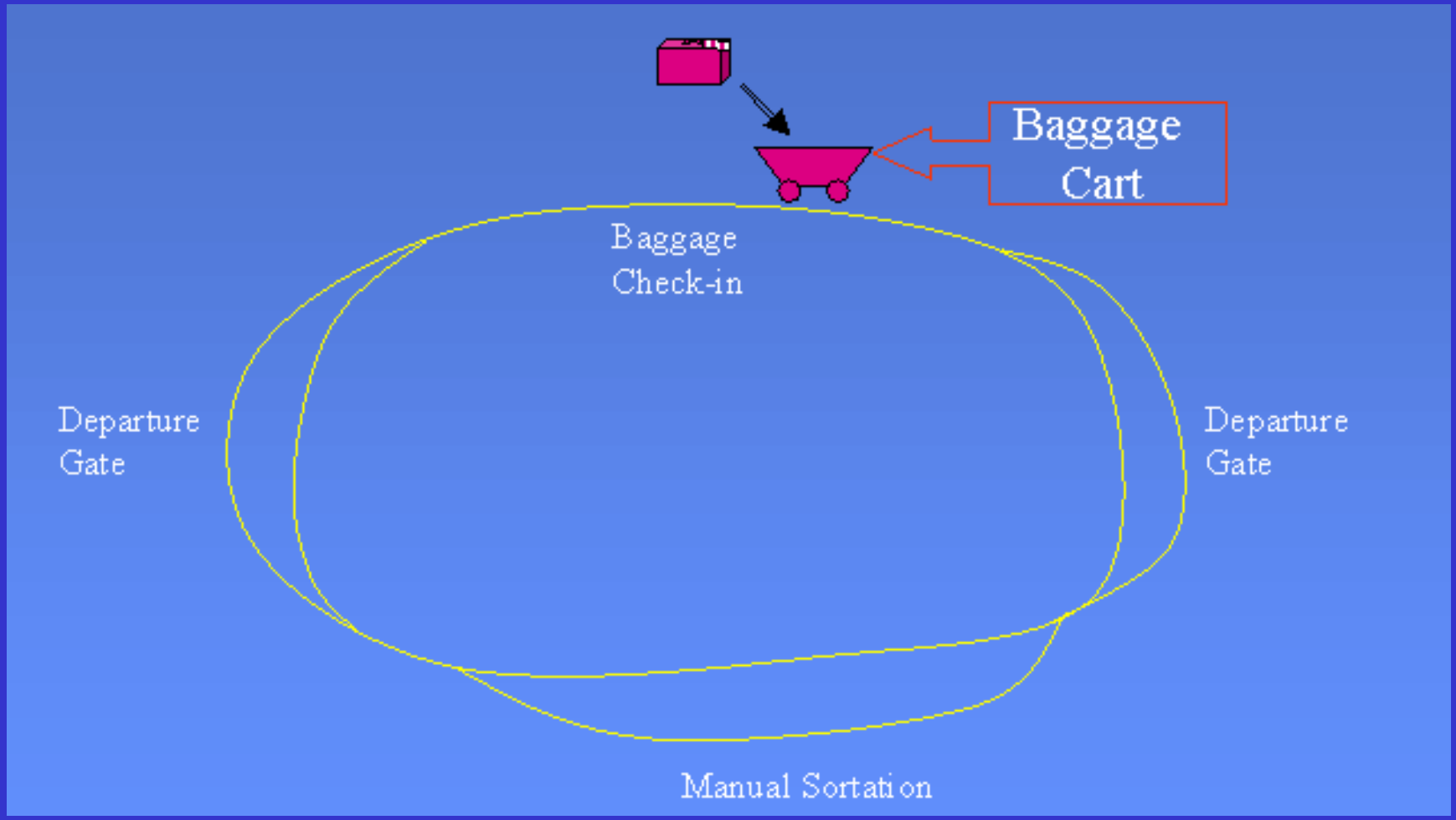
Ziele des Systems

- Vollautomatischer Lade- und Entlade-vorgang für alle Fluggesellschaften
- Schnelles Zurücklegen der langen Transportwege (27 km Gleis)
- Gepäck benötigt gleiche Zeit wie Passagier selbst (⇨ keine Wartezeiten)

Design Planung:

- 300 Rechner (486er) in 8 Kontrollräumen
- Raima Corp. Datenbank auf Netframe NF250 Server
- High-Speed Glasfaser Netz
- 14 Millionen Fuß Kabel
- 56 Lasereinheiten (Bar Code Leser)
- 400 Frequenzleser
- 22 Meilen Schienen, 6 Meilen Fließband
- 3100 Standardwagen und 450 Wagen für Übergrößen, funk gesteuert,
- 60 „Destination Coded Vehicles“ (DCV) pro Minute/Track
- 10000 Motoren und 92 PLC‘s (Programmable Logic Controller) für die Kontrolle der Motoren und der Weichen

Transport mit Wagen



Funktionsweise

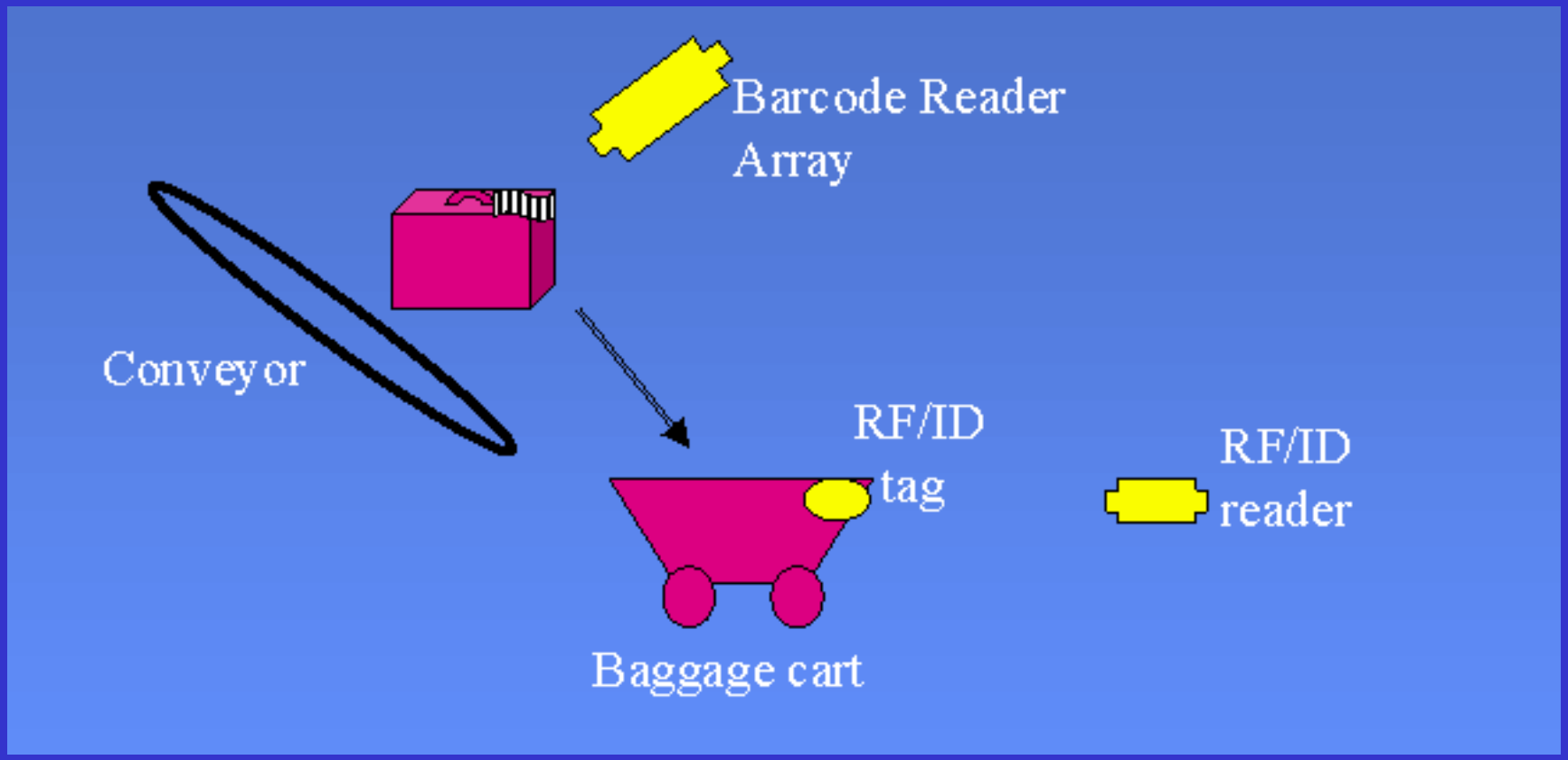
Klebe Bar Code Label, bzw. Photozelle, auf Gepäckstück mit Flugnr., Besitzer, Ziel, Zwischenhalt.

Fließband zum eigentlichen Track.

Anfordern eines leeren Wagen (Funk gesteuerter Telecar).

Dynamic Loading:
Ankommender Wagen verlangsamt auf 4,5 Meilen pro Stunde,
Fließband schießt mit „Kanone“ an T-Kreuzung Gepäck auf Wagen;
Wagen beschleunigt auf 19 Meilen pro Stunde;
Wagen bremst vor Weichen auf 8.5 Meilen pro Stunde ab;
jeder Wagen hat Koffereinsatz mit drei Stellungen für Beladen, Transport und Entladen

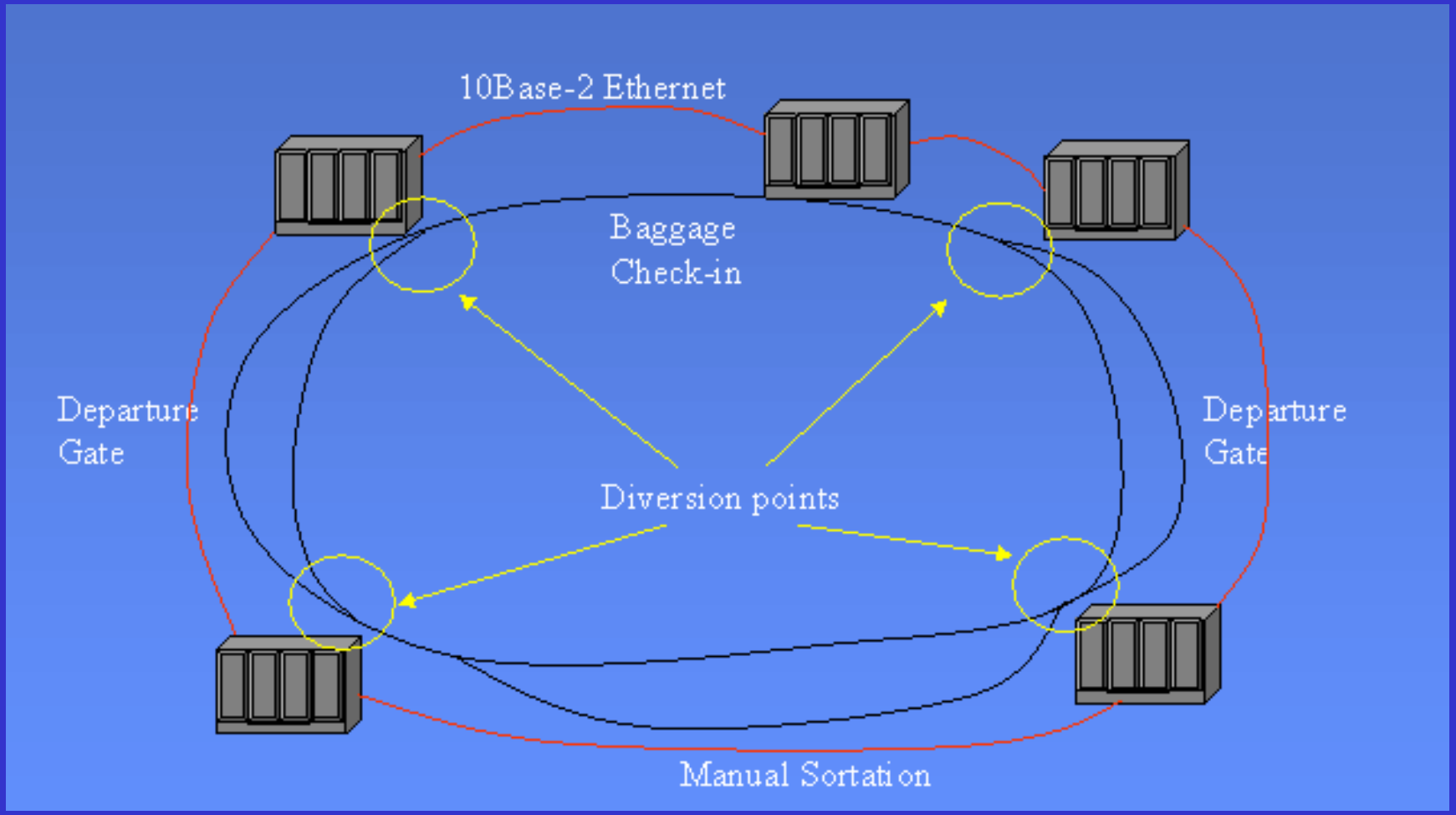
Erfassung


Barcode Reader
Array
Conveyor
RF/ID
tag
RF/ID
reader
Baggage cart

Funktionsweise II

- Beim Beladen liest ein Barcode-Scanner das Label und sendet die Daten an den Sortier-Computer
- Sortier-Computer schlägt Flugnr. aus Tabelle nach und schickt gespeicherte Routing-Information an Führungs-Computer
- Positionsbestimmung und Steuerung des Wagens und der Weichen per Funk

Netzwerk


10Base-2 Ethernet
Baggage
Check-in
Departure
Gate
Departure
Gate
Diversion points
Manual Sortation

Der Bumerang

War einmal ein Bumerang;
War ein Weniges zu lang.

Bumerang flog ein Stück,
Aber kam nicht mehr zurück.

Publikum – noch stundenlang –
Wartete auf Bumerang.

J. Ringelnatz

Auftretende Probleme

- Kurven in Tunnels zu eng – Entgleisen und Beschädigungen
- Fahrtwind
- Fehlerhafte Verschlüsse an Wagen
- Schlecht ausgedruckte Labels
- Justierung der Barcode-Scanner
- Verschmutzung der Photozellen
- Stoßstangen interferieren mit Photozellen
- Stromausfall

Auftretende Software Probleme

- Kombination mehrerer Programmiersprachen → fehlerhafte Datentyp-Konvertierung,…
- Fehlerhaftes Timing von Wagen und Lade-Kanonen
- Zu schnelle Ausgabe der gedruckten Label
- Zu frühes oder zu spätes Losschicken von Wagen
- Schwieriges Debugging:
  - Nur Trial & Error
  - Kein Funkverkehr in den Tunnels möglich
  - Viele verschiedene mögliche Fehler-Ursachen

Essentielle Probleme

- 10-Megabit Netzwerk zu langsam für Echtzeit-Steuerung
- Komponenten mussten schon an der Leistungsgrenze arbeiten
- Line Balancing:
  Steuerung der Wagen so, dass
  - keine vollen Laufbänder entstehen und
  - keine große Zahl leerer Wagen unterwegs sind
- Gesamtkomplexität

Auftretende Fehler

- Wegweiser zum Baggage Claim, die auf Betonwand zeigen
- Koffer fliegen aus Wagen
- Zerbrochene Koffer, Wäsche blockiert Schienen
- Entgleiste Wagen + demolierte Schienen
- Zusammenstöße von Wagen, teilweise orientierungslos
- Abladen der Koffer an falscher Station
- …….

Ursachen

- Erst Gebäudeplanung, insbes. Tunnel, später Baggage Handling
- Erweiterung von UA auf Gesamtsystem
- Installieren von BAE, Betrieb aus politischen Gründen an Aircraft Service International, Miami (Kosten, Arbeitsbedingungen)
- Nachträgliches Verlegen von Gebäuden
- Wenig Zeit für Tests und Anlernen
- Wenig Personal
- Schnittstelle zu Airline Reservierungs-Computer

Hauptursachen

- Netzwerk zu langsam
- Komplexität der Aufgaben wächst exponentiell mit der Größe!
  z.B. Line-Balancing: vgl. öffentliche Verkehrsmittel, Stoßzeiten mit überfüllten Bussen, Verpassen von Anschlüssen, z.B. Anfordern eines leeren Wagens
- Exponentielles Wachstum:
  10, 100, 10 000, 10^8, 10^{16},...

Entwicklung

- Viermaliges Verlegen der Flughafeneröffnung aufgrund des nicht funktionierenden Transportsystems
- Reduzierung des Systems: Transportsystem nicht flughafenweit, sondern separat für jedes der 3 Terminals
- Nur die Hälfte der 84 Gates werden versorgt, der Rest mit herkömmlichen System

Consulting

- Netzwerk durch 100 Megabit ersetzt
- Zusätzlich redundante Bauteile
- Beschränkung auf Teilbetrieb, 30 DCV pro Min., nur Concourse B, nur Start/Ende in Denver
- Deutsche Firma Logplan wurde hinzu gezogen, um Fehler zu finden und das System in modifizierter Form zum Laufen zu bringen
- Auf dieser Basis wurde dann die abgespeckte, funktionierende Version installiert.

Eröffnung

Entwicklungsstand zur Eröffnung:

- United Airlines ist einzige Fluggesellschaft die das System nützen kann (Terminal B)
- Aber auch hier nur begrenzter Einsatz
  - Nur Ankunft und Abflug von Denver
  - Weiterflüge und Zwischenlandungen werden manuell verarbeitet
- Traditionelles Transportsystem mit manueller Sortierung am restlichen Flughafen

⇨ entspricht 12% der geplanten Kapazität

Zusammenfassung der Fehler

"The main reasons for these problems were that the complexity of the system exceeded the understanding of the people designing and building it."

Kosten

16 Monate Verspätung: $ 500 000 pro Tag

$ 71 Millionen für Kauf des traditionelles Ersatz-Transportsystem

⇨ jeder Passagier muss $ 20 Flughafen-gebühr zahlen

Literatur

- Schloh: Analysis of the DIA baggage system
- de Neufville: The Baggage System at Denver: Prospects and Lessons
- Montealegre et.al.: BAE Automated Systems
- Donaldson: A case narrative of the project problems with the Denver Airport

Lehren

- Vorsichtige Planung
- Berücksichtigung der Komplexität
- Kleine Schritte
- Genügend Testzeit
- Backup-System vorsehen

Neue Probleme

19. Oktober 2000:

- automatische Passagier-U-Bahn fährt am Hauptterminal vorbei
- Sicherheitsmechanismus stoppt Bahn

Ursache: Datenverlust eines Speichers führt zum fehlerhaften Auslesen eines Barcodes, der für das Haltesignal notwendig gewesen wäre

Neue Probleme II

Januar 2002:

Wireless LAN Netzwerke des DIA ohne Verschlüsselung

- Möglichkeit der Datenmanipulation
- auch andere amerik. Flughäfen betroffen, z.B. San Jose

Daten zu Großprojekten

- 55% übersteigen die geplanten Kosten
- 68% können den gegebenen Zeitplan nicht einhalten
- 88% benötigen tiefgreifende Änderungen

⇨ Hohe Wahrscheinlichkeit des Scheiterns bereits beim Start

- 45% - 80% der Systemkosten entstehen durch Wartungsarbeiten

⇨ System muss wartbar sein

Meilensteine der Softwarefehler

- Ariane 5 (Zahlumwandlung)
- Patriot (Rundungsfehler)
- INTEL Pentium Divisionsfehler
- Mars Climate Orbiter (Einheiten)
- Therac-25 (z.B. 8 Bit Zähler)
- Stellwerk Altona (zu wenig Speicher)
- Wintersturm Lothar oder Ozonloch (Ignorieren von Messdaten)
- Toll Collect
- SDI, bzw. MD (Missile Defense)

Software-Bugs

Laut INTEL: 80-90 Bugs in Pentium, genauso viele wie bei den Vorgängern.

Normale Software: 25 Fehler pro 1000 Programmzeilen.

Gute Software: 2-3 Fehler pro 1000 Zeilen.

Space Shuttle Software: weniger als 1 Fehler pro 1000 Zeilen (laut NASA)

- Beispiel Handy:
  200 000 Zeilen Programm → ca. 500 Fehler
- Beispiel Space Shuttle Software:
  3 Mill. Zeilen → ca. 300 Fehler
- Beispiel Windows 2000:
  27 Mill. Zeilen → ca. 50 000 Fehler
- Beispiel SDI (Raketenabwehr USA):
  25-100 Mill. Zeilen → 10 000 Fehler

Stichwort Bananensoftware:
Lass die Software beim Kunden reifen!

It´s not a bug, it´s a feature!

Quelle: Ingolf Giese, GSI Darmstadt

Ursachen von Software-Bugs

- Offensichtliche Fehler (Tipp~, Konzeptions~)
- Fehlende Sicherheitsabfragen (unvorhergesehene Fälle treten auf; Division durch Null, Typumwandlung,…)
- Schnittstellen- ~ (Programmteile passen nicht zusammen)
- Fehlinterpretation von Ein/Ausgabe-Daten
- Ungeprüfte Wiederverwendung alten Codes (Ariane 5)
- Software und Hardware passen nicht (mehr) zusammen
- Computerentwicklung wesentlich schneller als die Entwicklung der restlichen Technologie (NASA)
- Numerische Rundungsfehler
- Nicht ausreichende Tests
- Gigantismus/Komplexität
- Unterschätzen der Aufgabenstellung