SWIMming in Tcl

Frank Morlang
Overview

- INTRODUCTION
- PROVOCATION
- MOTIVATION
- CHALLENGE
- SOLUTION
- OUTLOOK
Introduction

• **Future Commercial Space Traffic assumption:**
  • Will return as a hypersonic glider
  • What does a (Columbia comparable) fatal break up event (ca. 231000 ft. Alt., *speed > Mach 20*) mean?
    • **Debris** raining down on conventional air traffic will cover a **footprint** of about **300 by 35 nm**
    • No collision of Columbia debris with air traffic was just **luck** (*Casualty probability* for passengers was about **0.3**)
Provocation

• Commercial Space Traffic
  • Only a few movements per year = research + entertainment for private super millionaires = no air traffic integration considerations needed = If ever relevant, in the very far future!

Really?
Motivation

- 10 years between
Motivation

• Now

• Future (Who knows when ?)
Motivation

SWIM

= System Wide Information Management
Motivation

SWIM "Intranet for ATM" concept requests **all** the future air traffic participants acting as communicating sub-systems.
Motivation

SWIM ⇒ Why?

Motivation

**What?**

### SWIM Scope

- Different applications
  - SWIM-enabled applications

#### Information Exchange Services
- Service specifications for information exchange
- **AIXM, FIXM, WXXM**
- **Information Exchange Models**

#### SWIM Infrastructure
- Interface management, message comm. protocols
- **SWIM Infrastructure**

#### Network Connectivity
- Communication networks (Ground/Ground, Air/Ground)
  - **Network Connectivity**
Motivation

SWIM ➔ What?

Motivation

• Technical profiles:
  • Yellow → non critical information
  • Blue → critical information
  • Purple → Air / Ground info exchanges
Motivation

SWIM = What?

Source: Manual on System Wide Information Management (SWIM) Concept, ICAO Doc 10039 AN/511
Motivation

SWIM = What?

Enterprise Service Management
- SNMP
- JMX
- WS-Distributed Management
- WS-Manageability

QoS
- WS-Reliable Messaging
- WS-RM Policy
- DDS Reliability

Security
- WS-Security
- TLS
- DDS Security

Others
- DDS Standard QoS

Policy
- WS-Policy Association
- WS-Policy Attachment
- WS-Policy Framework

DDS QoS Discovery

Interface Management
- WSDL
- UDDI
- IDL
- WS-Metadata
- DDS Automatic Discovery

Inter Operability
- WS-DDS Bridge
- ...

Data Representation
- XML
- XSD
- XPath
- XQuery
- CDR
- ASN.1

Messaging
- SOAP
- MTOM
- DDS
- WS-Attachments
- WS-Addressing
- WS-Notification
- JMS

Transport
- HTTP
- DDSI
- HTTPS
- AMQP
Motivation

Benefits of acting SWIM compliant

• Access to real-time, relevant aeronautical, flight, and weather information $\Rightarrow$ faster dedicated response possibilities

• Reduced implementation, operating and extension costs because of SWIM’s standardized character

• SWIM = requested fundament of the future for info based collaboration in ATM (Air Traffic Management) $\Rightarrow$ being prepared for the future
Challenge

Safe global space traffic integration by taking into account data distribution of its changing debris (= hazard) area during reentry!
Solution

Input:
Hypothetical spacecraft's (returning) runtime data:
- id
- lat
- lon
- alt
- heading
- path_velocity

Output:
Lat_Lon of 4-point-HazardZonePolygon

TFR airspace in AIXM
Solution using:

- **TclHttpd** as the web server
- **Web Services for Tcl** for the server side web service creation on top of TclHttpd
- **TclITLS** for using HTTPS
- **Rpcvar** for complex data type definitions
- **CriTcl** for improved performance by the usage of C code runtime embedding
- **BaseXClient-Tcl** for using the BaseX server protocol to communicate with the hazard area model database server
Solution structure:
Solution

[Diagram showing the solution involving DLR, Data Server, EMS, NEAR GEMS, Services, Consumers EFB Display, SESAR SWIM, FAA SWIM, and a bar chart showing reaction times of the SpacecraftReentryHazardAreaService with varying numbers of connected clients.]
Solution CriTcl usage (excerpt):

critcl::cproc c_calcheading {double lat1in double lon1in double lat2in double lon2in} double {
    /* this is C code */
    double localoperator1;
    double localoperator2;
    double localheading;
    localoperator1 = cos(torad(lat2in)) * sin((torad(lon2in)) - (torad(lon1in)));
    localoperator2 = cos(torad(lat1in)) * sin(torad(lat2in)) - sin(torad(lat1in)) * 
    cos(torad(lat2in)) * cos((torad(lon2in)) - (torad(lon1in)));
    localheading = atan2(localoperator1, localoperator2) * (180 / pi);
    if (localheading < 0)
        localheading += 360.0;
    return localheading;
}
Solution (performance enhancement)

Reaction Times of the
SpacecraftReentryHazardAreaService
pure Tcl

Calculate Heading and Calculate Hazard Zone procedures in C using the Critcl
package

- 1 connected client
- 2 connected clients
- 3 connected clients
- 4 connected clients
- 5 connected clients
- 6 connected clients
- 7 connected clients
- 8 connected clients
- 9 connected clients
- 10 connected clients
Solution (scalability)

Sufficient for most small and medium sized cases!

Anyhow ....
Solution (scalability)

Anyhow ....
Outlook

SecSWIM

Big Data River

$M_1$$M_2$$M_3$$...$$M_n$
Outlook

Machine Learning

HDFS  Mongo DB  Data Lake  Kafka

Big Data River

Data Ingest Streaming  Views / Visualization
Outlook

SecSWIM .foreseen to be developed in ?  
Make an educated guess… ^

• Kafkatcl
• TensorFlow (C++ API used inside Critcl)