## ATC communication from developers view

**Getting ATC statusitems/communication for Flightchecker from database:**

ATCParser processes records in *telex\_out\_rec*(outgoing FPLs) and *sender\_recipient\_rec*(incoming ATC telex) and writes the result into *telex\_to\_flight* and *fpl*.

(*telexout\_rec.id\_sender = telextext\_rec.id\_sender)*

*Telex\_to\_flight* and *fpl* are used by Flightchecker:

sp\_atc\_getstatusitems '03may12','sas',944, 'KORD','EKCH',213



Check if *dbo.telex\_to\_flight.responseid* is not NULL,

else add addresses to configure ATCParser “ATCParser.ResponseRequested”!

**Translate telexsource, type, suptype to a messagetype.**

*CFlightStatItem::GetMsgType()*

e.g. previous example messagetypes meant:

*1 // FPL to EC*

*10 // ACK IFPL MSGTEXT*

**All states we’re using:**

// + No valid FPL at ATC

// | + Valid FPL at ATC

// | | + FPL not acknowledged

// | | | + CHG not acknowledged

// | | | | + DLA not acknowledged

// | | | | | + CNL not acknowledged

// | | | | | | + Valid FPL at ATC; FPL revised

// | | | | | | | + Valid FPL at ATC; CHG revised

// | | | | | | | | + Valid FPL at ATC; DLA revised

// | | | | | | | | | + Valid FPL at ATC; duplicate FPL should be rejected

// | | | | | | | | | | + Valid FPL at ATC, but last message rejected

// | | | | | | | | | | | + Valid FPL at ATC; content unknown

// | | | | | | | | | | | | + Valid FPL at ATC; but last FPL should be negotiated

// | | | | | | | | | | | | | + Valid FPL at ATC; but last CHG should be negotiated

// | | | | | | | | | | | | | | + Valid FPL at ATC; but last DLA should be negotiated

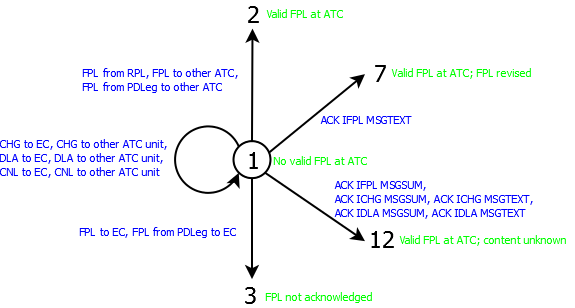
// | | | | | | | | | | | | | | | + Valid FPL at ATC; but last CNL should be negotiated

// | | | | | | | | | | | | | | | |

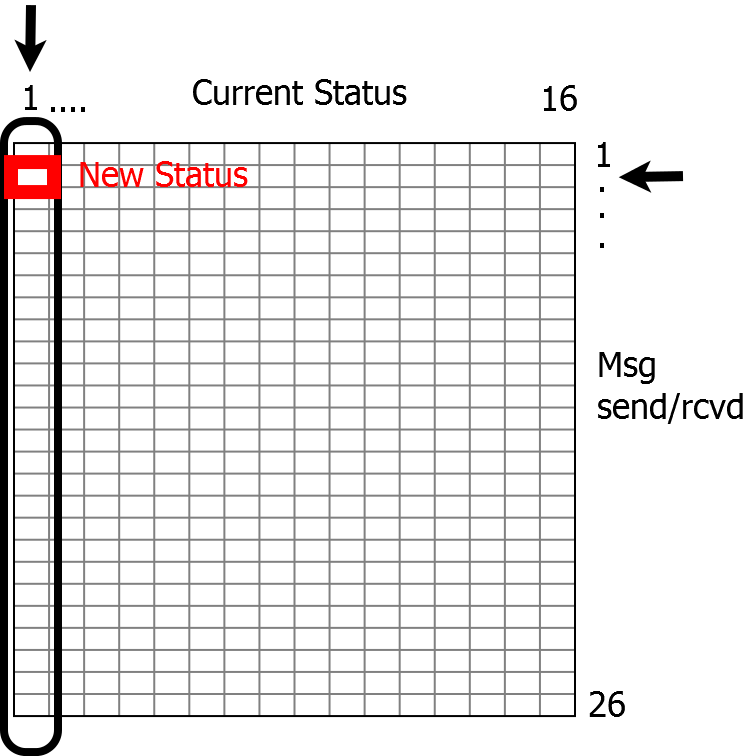
// 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

**Representation of transitions from one state to another:**

Graph of all transitions of starting state 1:



There are 16 states available and 26 messagetypes = 16 x 26 edges. A representation of the graph as a matrix is simpler:



e.g.

Current Status 1, sending a FPL to EC(msgtype=2) -> New Status:3

**Flightstat.cpp:**

// 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

{ 2, 10, 10, 10, 10, 2, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10 }, // FPL from RPL

{ 3, 10, 10, 10, 10, 3, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10 }, // FPL to EC

{ 2, 10, 10, 10, 10, 2, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10 }, // FPL to other ATC unit

{ 1, 4, 4, 4, 4, 1, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4 }, // CHG to EC

{ 1, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2 }, // CHG to other ATC unit

{ 1, 5, 5, 5, 5, 1, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5 }, // DLA to EC

{ 1, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2 }, // DLA to other ATC unit

…

* **Column** holds all edges from **“status”**

e.g. Column “1” has all edges of state 1

* **Row** holds all edges of a **“message type”**

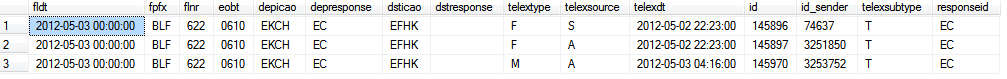
e.g. FPL to EC:

edge(1,3), edge(2,10),edge(3,10), edge(4,10), edge(5,10), edge(6,3) …

edge(1,3) means: edge between status 1 and 3

# Full Example - debugging:

sp\_atc\_getstatusitems '03may12','blf',622, 'EKCH','EFHK',385

****

***Get Messagetypes:***

Telexsource, telextype, telexsubtype, responseid -> *CFlightStatItem::GetMsgType()*

S F T EC 1 // FPL to EC

A F T EC 10 // ACK IFPL MSGTEXT

A M T EC 12 // ACK ICHG MSGTEXT

**Get final status:**

*CFlightStat::GetTransition(long prevStat, long msgType)*

**Matrix[Msg Type][curr.Status -1]**

*Matrix[1][1/\*start\*/ -1] - >state:3 // FPL not ACK*

*Matrix[10][3 -1] -> state: 7 //* Valid FPL at ATC; FPL revised

*Matrix[12][7-1] -> state: 12 // Valid FPL at ATC; content unknown*