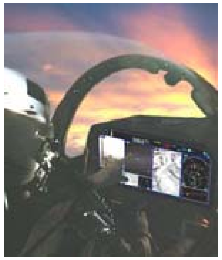
Full Operator Choice Model for Air Traffic Control Systems

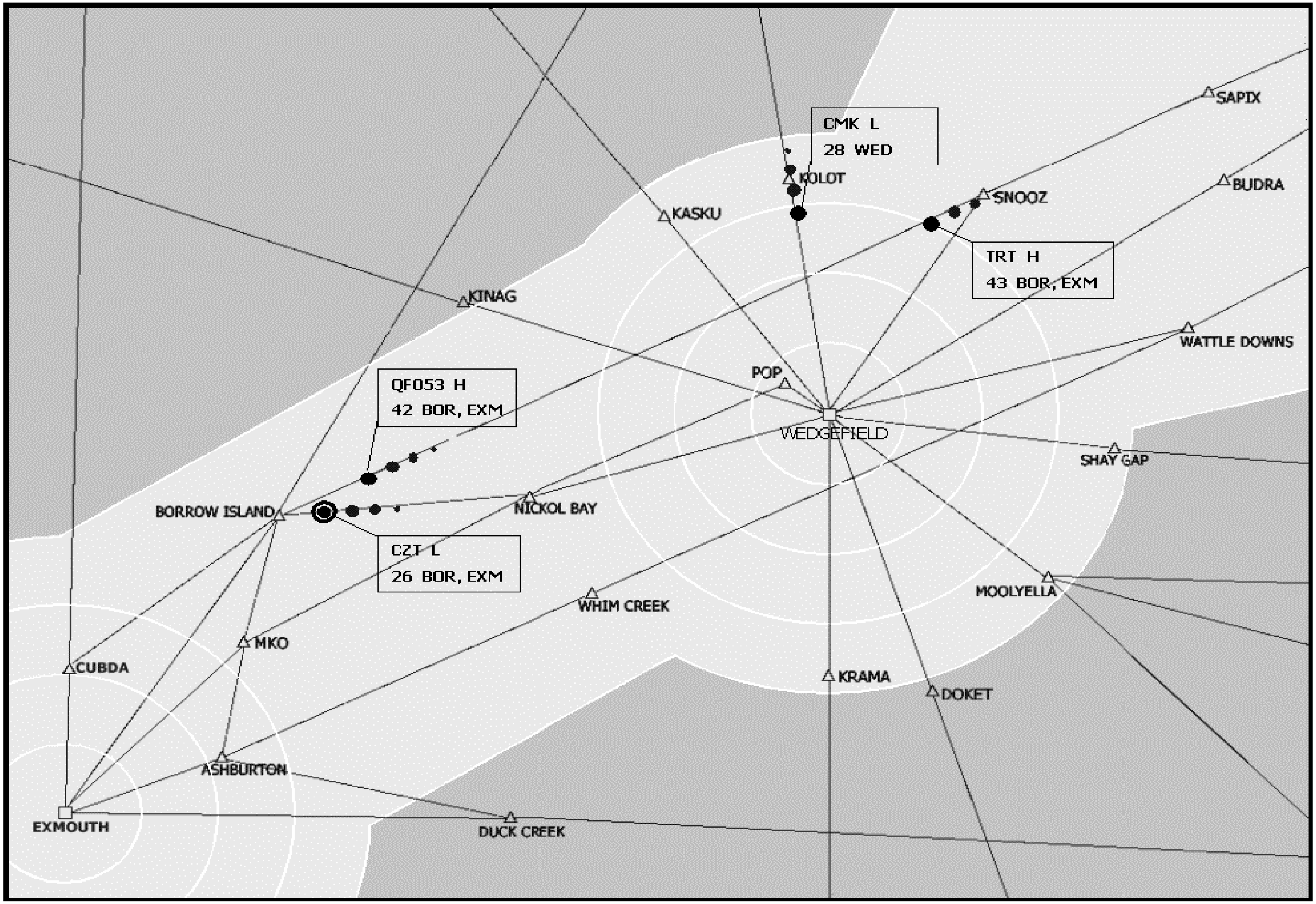


Thaizel Fuentes Martinez
Universitá di Pisa, Italia
fuentes@di.unipi.it
http://www.di.uni.it/~fuentes

Summary

- ATC and HCI
- Basic knowledge
  - **O**perator **C**hoice **M**odel
  - Object-**Z** language
- ATC (modeling and hazard identification)
- Conclusions

Air-Traffic Control System

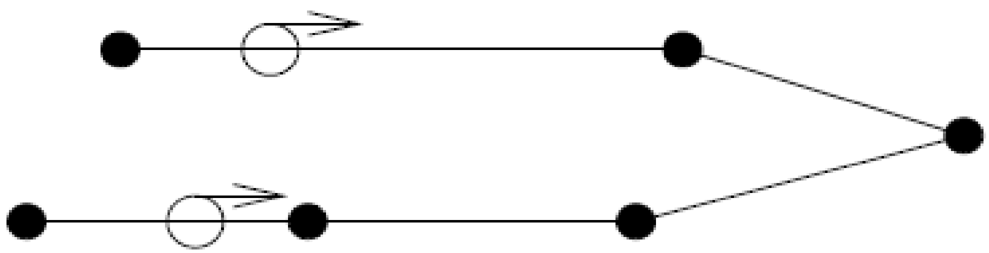


Modeling Hazardous situations

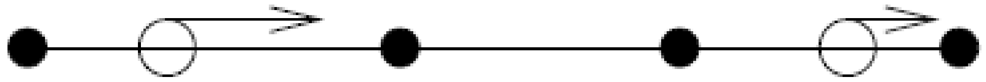
- To combine improved understanding of the psychological causes of human errors, with formal methods for modeling human-computer-interaction.
- To incorporate mathematical models of operators.
  - Object-Z language
  - Operator Choice Model
- Simulation and detection of conflicts

Air-Traffic Control (Hazards)

Convergence

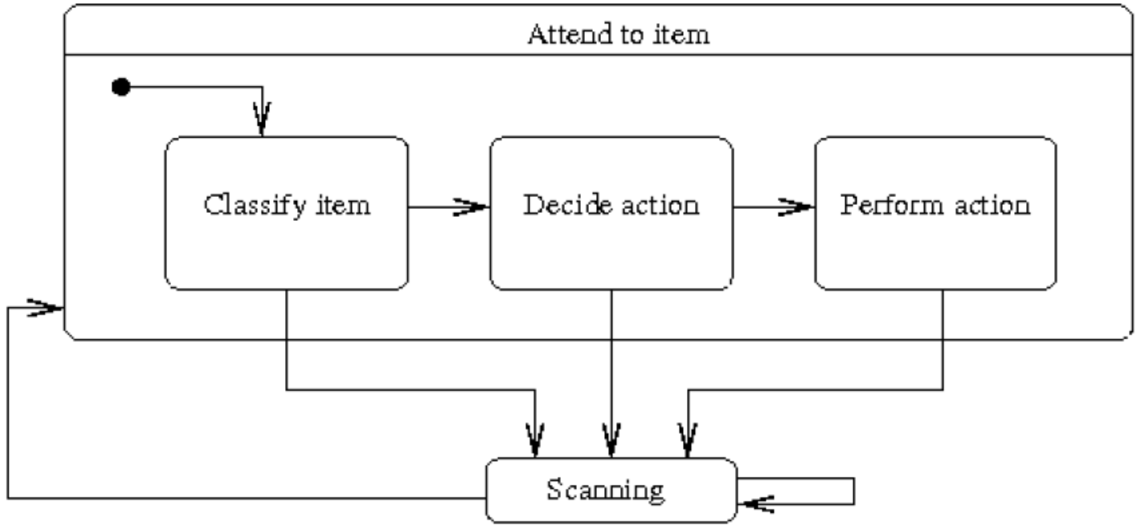


Overtaking



Operator Choice Model

- Designed to model human behavior in a real-world system. Simulation of realistic operator behavior in complex task evolving unpredictably through time.
- Cycle of: Scan→Classify→Decide Action→Perform Action



Object-Z notation

- What is Object-Z ?
  - Formal specification language. Formal method for modeling human-computer-interaction.
  - Z is based in a notion of "set" (∈,∪,∩,#,…),
    - "|" and "●" means *such as*
    - "seq" and "‹›" means *sequence*
    - "==" means *equivalence*
    - Functions "head" and "tail" applied to a sequence
    - "∩" means *sequence concatenation*

More about Z…

- Schemas: *Variable + predicates constraining the variable*
  - *Traditional logic operator (¬,^,v,→,↔)*

Coord

$x : \mathbb{Z}$

$y : \mathbb{Z}$

$x \geqslant 0 \land y \geqslant 0$

$$Origin \mathrel{\widehat=} [\, Coord \mid x = 0 \land y = 0 \,]$$

  - *Operation Schema:* Model states transition between variables. (?,!,’,Δ)

AddCoord

$\Delta Coord$

$x?, y? : \mathbb{Z}$

$x' = x + x?$

$y' = y + y?$

  - *Schema Calculus: (¬,^,v,→,↔)*

Object-Z

- Object-Oriented Z
- Execution Operators
  - Conjunction (“^”)
  - Parallel (“||”)
  - Sequential (“;”)
  - Choice (“[]”)
- Predicates can optionally contains temporally logic operators : *always* (“□”), *eventually* (“◊”), *next* (“○”)

StackPair

$s_1, s_2 : NatStack$

$s_1 \neq s_2$

$\#s_1.nats \leqslant \#s_2.nats$

INIT

$s_1.\mathrm{INIT} \wedge s_2.\mathrm{INIT}$

$Push_1 \mathrel{\widehat=} s_1.Push$

$Push_2 \mathrel{\widehat=} s_2.Push$

$Pop_1 \mathrel{\widehat=} s_1.Pop$

$Pop_2 \mathrel{\widehat=} s_2.Pop$

$PushBoth \mathrel{\widehat=} Push_1 \wedge Push_2$

$Transfer \mathrel{\widehat=} (Pop_1 \parallel Push_2) \setminus (nat!)$

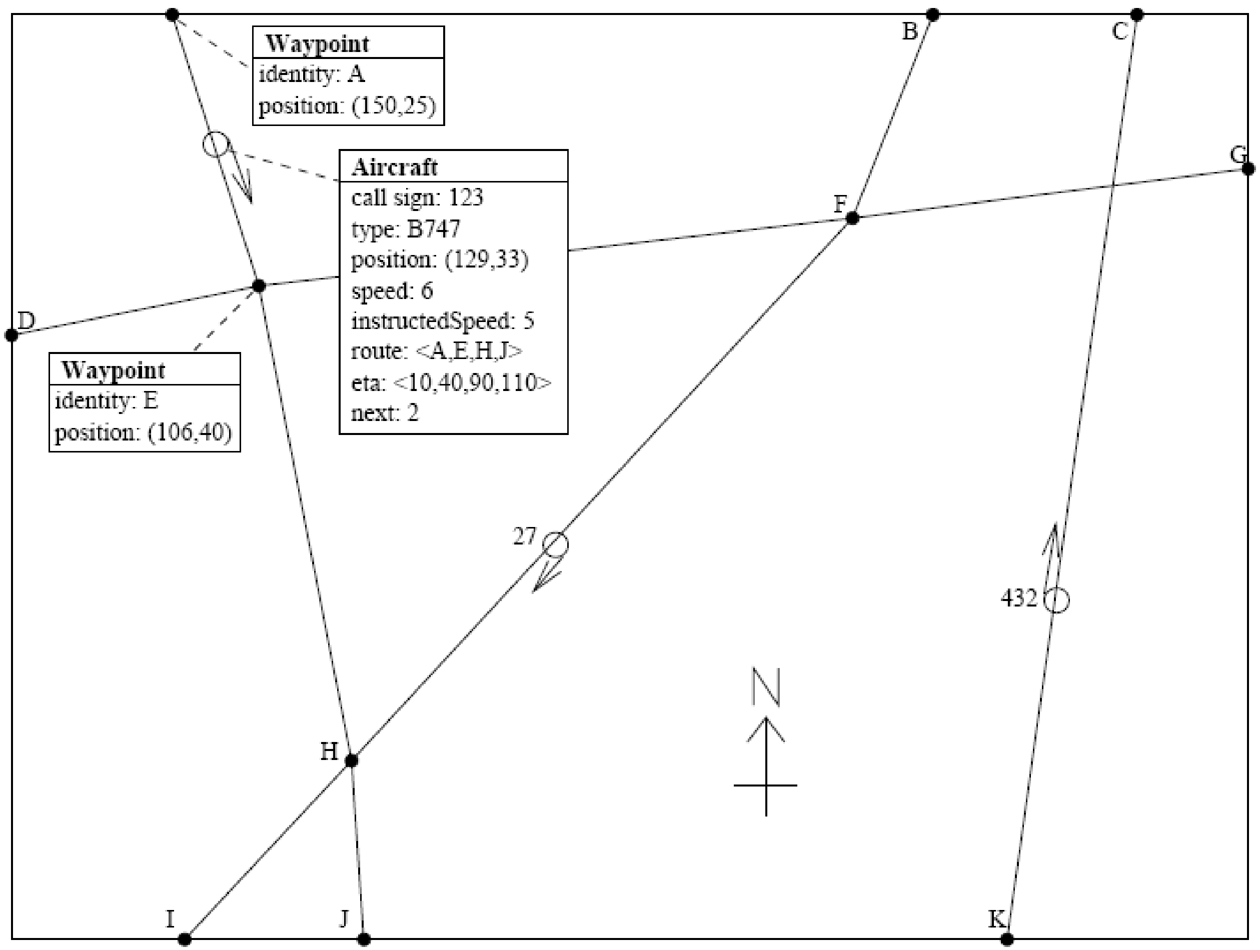
$PushOne \mathrel{\widehat=} Push_1 \,[]\, Push_2$

$TransferAll \mathrel{\widehat=} Transfer \fatsemi TransferAll$

$[]$

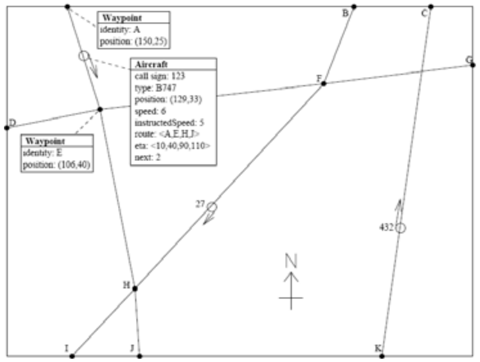
$[\, s_1.items = \langle\rangle \,]$

The problem



Waypoint
identity: A
position: (150,25)
Aircraft
call sign: 123
type: B747
position: (129,33)
speed: 6
instructedSpeed: 5
route: <A,E,H,J>
eta: <10,40,90,110>
next: 2
Waypoint
identity: E
position: (106,40)
B
C
D
F
G
H
I
J
K
27
432
N

The formal model



- ✓ Position
- ✓ Waypoint
- ✓ Manhattan Distance
- ✓ Route
- ✓ SubRoute
- ✓ RouteLength

$$Latitude == \mathbb{N}$$
$$Longitude == \mathbb{N}$$
$$Position == Latitude \times Longitude$$

$$\begin{array}{l} \textbf{Waypoint} \\ \hline identity : WaypointID \\ position : Position \end{array}$$

$$\begin{array}{l} distanceBetween : Position \times Position \to \mathbb{N} \\ \hline \forall pos1, pos2 : Position \bullet distanceBetween(pos1, pos2) = \\ \quad abs(first(pos1) - first(pos2)) + abs(second(pos1) - second(pos2)) \end{array}$$

$$Route == \mathrm{seq}_1\ Waypoint$$

$$\begin{array}{l} subRoute : Route \times \mathbb{N} \times \mathbb{N} \nrightarrow Route \\ \hline \forall route : Route;\ start, end : \mathbb{N} \mid \{start, end\} \subseteq \mathrm{dom}\ route \bullet \\ \quad subRoute(route, start, end) = (start \mathrel{..} end) \upharpoonright route \end{array}$$

$$\begin{array}{l} routeLength : Route \to \mathbb{N} \\ \hline routeLength(\langle\,\rangle) = 0 \\ \forall wp1, wp2 : Waypoint;\ route : Route \bullet \\ \quad routeLength(\langle wp1 \rangle) = 0 \land \\ \quad routeLength(\langle wp1, wp2 \rangle \frown route) = \\ \qquad distanceBetween(wp1.position, wp2.position) + \\ \qquad routeLength(\langle wp2 \rangle \frown route) \end{array}$$

The formal model

✓ Aircraft schema

Aircraft

$$\begin{aligned}
&callsign : Callsign \\
&type : AircraftType \\
&position : Position \\
&speed : Speed \\
&instructedSpeed : \mathbb{F}\, Speed \\
&route : Route \\
&eta : \mathrm{seq}_1\ Time \\
&next : \mathbb{N}
\end{aligned}$$

$$\begin{aligned}
&\#instructedSpeed \leq 1 \\
&\#route \geq 2 \\
&\#route = \#eta \\
&\forall\, num : 1 \,..\, \#eta - 1 \bullet \\
&\qquad eta(num) < eta(num + 1) \\
&next \in 2 \,..\, \#route
\end{aligned}$$

The formal model

- ✓ DestinationReached
- ✓ DistanceToWayPoint
- ✓ Aircraft movement

$$destinationReached : Aircraft \rightarrow \mathbb{B}$$

$$\begin{aligned}
&\forall\, ac : Aircraft \bullet destinationReached(ac) \Leftrightarrow \\
&\quad ac.next = \#(ac.route) \land \\
&\quad ac.position = (last(ac.route)).position
\end{aligned}$$

$$distanceToWaypoint : Aircraft \times \mathbb{N} \nrightarrow \mathbb{N}$$

$$\begin{aligned}
&\forall\, ac : Aircraft;\ wpnum : \mathbb{N} \mid ac.next \leq wpnum \leq \#ac.route \bullet \\
&\quad distanceToWaypoint(ac, wpnum) = \\
&\qquad distanceBetween(ac.position, ((ac.route)(ac.next)).position) + \\
&\qquad routeLength(subRoute(ac.route, ac.next, wpnum))
\end{aligned}$$

move

$$\Delta Aircraft$$

$$\begin{aligned}
&callsign' = callsign \\
&type' = type \\
&route' = route \\
&(1 \mathrel{..} next - 1) \upharpoonleft eta' = (1 \mathrel{..} next - 1) \upharpoonleft eta \\
&next' = if(position' = (route(next)).position \land next \neq \#route) \\
&\quad then\ next + 1 \\
&\quad else\ next \\
&distanceBetween(position, position') \in \{0, 1\} \\
&speed - speed' \in \{-1, 0, 1\}
\end{aligned}$$

The formal model

The ATC System HCI Screen Schema

Screen

$$sector : Sector$$
$$aircraft : \mathbb{F}\, Aircraft$$

$$\forall\, ac1, ac2 : aircraft \bullet ac1.callsign = ac2.callsign \Rightarrow ac1 = ac2$$
$$\begin{aligned}
&\forall\, ac : aircraft \bullet\\
&\quad sector.south \leq first(ac.position) \leq sector.north \wedge\\
&\quad sector.west \leq second(ac.position) \leq sector.east \wedge\\
&\quad (\forall\, wpnum : 1 \,..\, \#ac.route - 1 \bullet\\
&\qquad ((ac.route)(wpnum), (ac.route)(wpnum + 1)) \in\\
&\qquad\quad sector.flightPaths) \wedge\\
&\quad head(ac.route) \in sector.handOverPoints \wedge\\
&\quad last(ac.route) \in sector.handOverPoints
\end{aligned}$$

ScreenInit

$$Screen$$

$$aircraft = \varnothing$$

aircraftEnterScreen

$$\Delta Screen$$
$$craft? : \mathbb{F}\, Aircraft$$

$$\{ac : craft? \bullet ac.callsign\} \cap \{ac : aircraft \bullet ac.callsign\} = \varnothing$$
$$\begin{aligned}
&\forall\, ac : craft? \bullet\\
&\quad \mathrm{ran}(ac.route) \subseteq sector.waypoints \wedge\\
&\quad ac.position = (head(ac.route)).position \wedge\\
&\quad ac.next = 2\\
&aircraft' = aircraft \cup craft?\\
&sector' = sector
\end{aligned}$$

aircraftExitScreen

$$\Delta Screen$$
$$craft! : \mathbb{F}\, Aircraft$$

$$\begin{aligned}
&craft! \subseteq aircraft\\
&\forall\, ac : craft! \bullet destinationReached(ac)\\
&aircraft' = aircraft \setminus craft!\\
&sector' = sector
\end{aligned}$$

The formal model

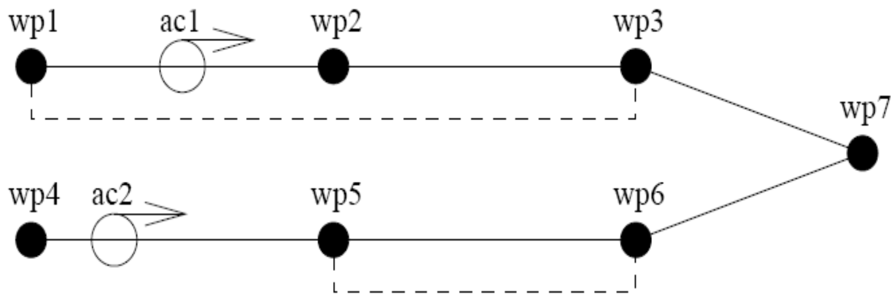
The HCI functionality

$$ChangeAircraftRoute \mathrel{\widehat{=}} selectAircraft \land reroute$$
$$ChangeAircraftSpeed \mathrel{\widehat{=}} selectAircraft \land changeSpeed$$

$$updateScreen \mathrel{\widehat{=}} aircraftExitScreen \fatsemi aircraftMove \fatsemi aircraftEnterScreen$$

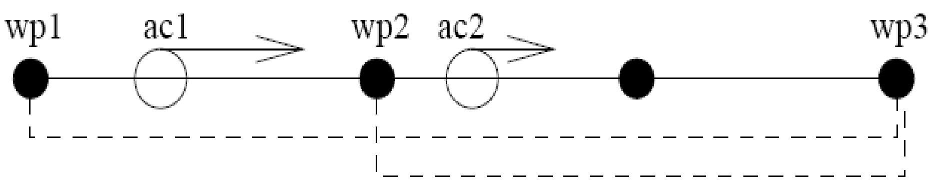
Convergence Hazard

$$
\begin{array}{l}
timeDifference : Time \times Time \rightarrow Time \\
\hline
\forall t1, t2 : Time \bullet \\
\quad timeDifference(t1, t2) = abs(t1 - t2)
\end{array}
$$



$$
\begin{array}{l}
convergenceHazard : ScenarioIdentifier \\
\hline
\forall ac1, ac2 : Aircraft \mid ac1 \neq ac2 \bullet convergenceHazard(\{ac1, ac2\}) \Leftrightarrow \\
\quad (\exists point1 : ac1.next \mathinner{\ldotp\ldotp} \#ac1.route;\ point2 : ac2.next \mathinner{\ldotp\ldotp} \#ac2.route \bullet \\
\qquad (ac1.route)(point1) = (ac2.route)(point2) \land \\
\qquad (ac1.route)(point1 - 1) \neq (ac2.route)(point2 - 1) \land \\
\qquad \mathrm{ran}(subRoute(ac1.route, ac1.next - 1, point1 - 1)) \cap \\
\qquad\quad \mathrm{ran}(subRoute(ac2.route, ac2.next, point2 - 1)) = \varnothing \land \\
\qquad timeDifference((ac1.eta)(point1), (ac2.eta)(point2)) \leq \\
\qquad\quad separationTime)
\end{array}
$$

Overtaking Hazard



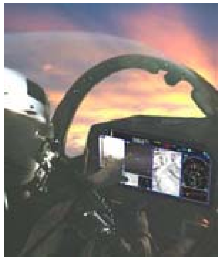
$$overtakingHazard : ScenarioIdentifier$$

$$\forall ac1, ac2 : Aircraft \mid ac1 \neq ac2 \bullet overtakingHazard(\{ac1, ac2\}) \Leftrightarrow$$
$$(\exists point1 : ac1.next \mathrel{..} \#ac1.route;\ point2 : ac2.next \mathrel{..} \#ac2.route \bullet$$
$$subRoute(ac2.route, ac2.next - 1, point2) \text{ suffix}$$
$$subRoute(ac1.route, ac1.next - 1, point1) \land$$
$$distanceToWaypoint(ac1, point1) > distanceToWaypoint(ac2, point2) \land$$
$$((ac1.eta)(point1) < (ac2.eta)(point2) \lor$$
$$timeDifference((ac1.eta)(point1), (ac2.eta)(point2)) \leq separationTime))$$

Conclusions

- A model of the operator’s cognitive states.
- Represents the human error into the model.
- Stores results in long term memory.
- Pattern-based techniques (predictions).

Full Operator Choice Model for Air Traffic Control Systems



Thaizel Fuentes Martinez
Universitá di Pisa, Italia
fuentes@di.unipi.it
http://www.di.uni.it/~fuentes