

Temporal reasoning models of a targeting decision chain

Robert J. Houghton
Chris Baber

University of
Birmingham, UK



www.hfidtc.com

Human Factors Integration Defence Technology Centre

Purpose

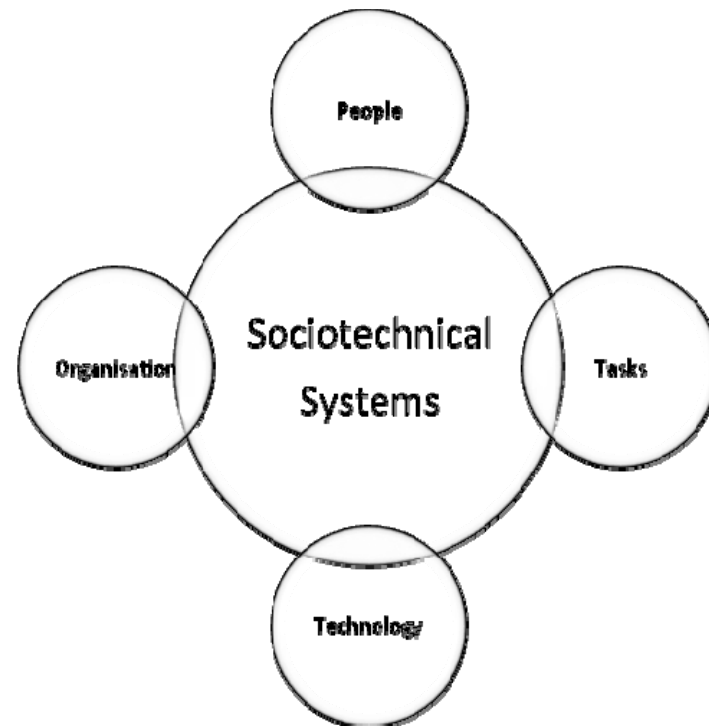


- To demonstrate and pilot a new (old) idea for analysing systems
 - A logical method: the event calculus
- Domain of interest:
 - “Strawman” model
 - Activities that occur between calling for a UAV having spotted a target to the point of tasking assets
 - Full F2T2EA model in progress

Sociotechnical systems



- Heterogeneous
- Hard to describe consistently using one method
- Elicitation
 - Verbal descriptions
 - Procedures
 - Narratives/scenarios
- Analogy with a contract



Event calculus



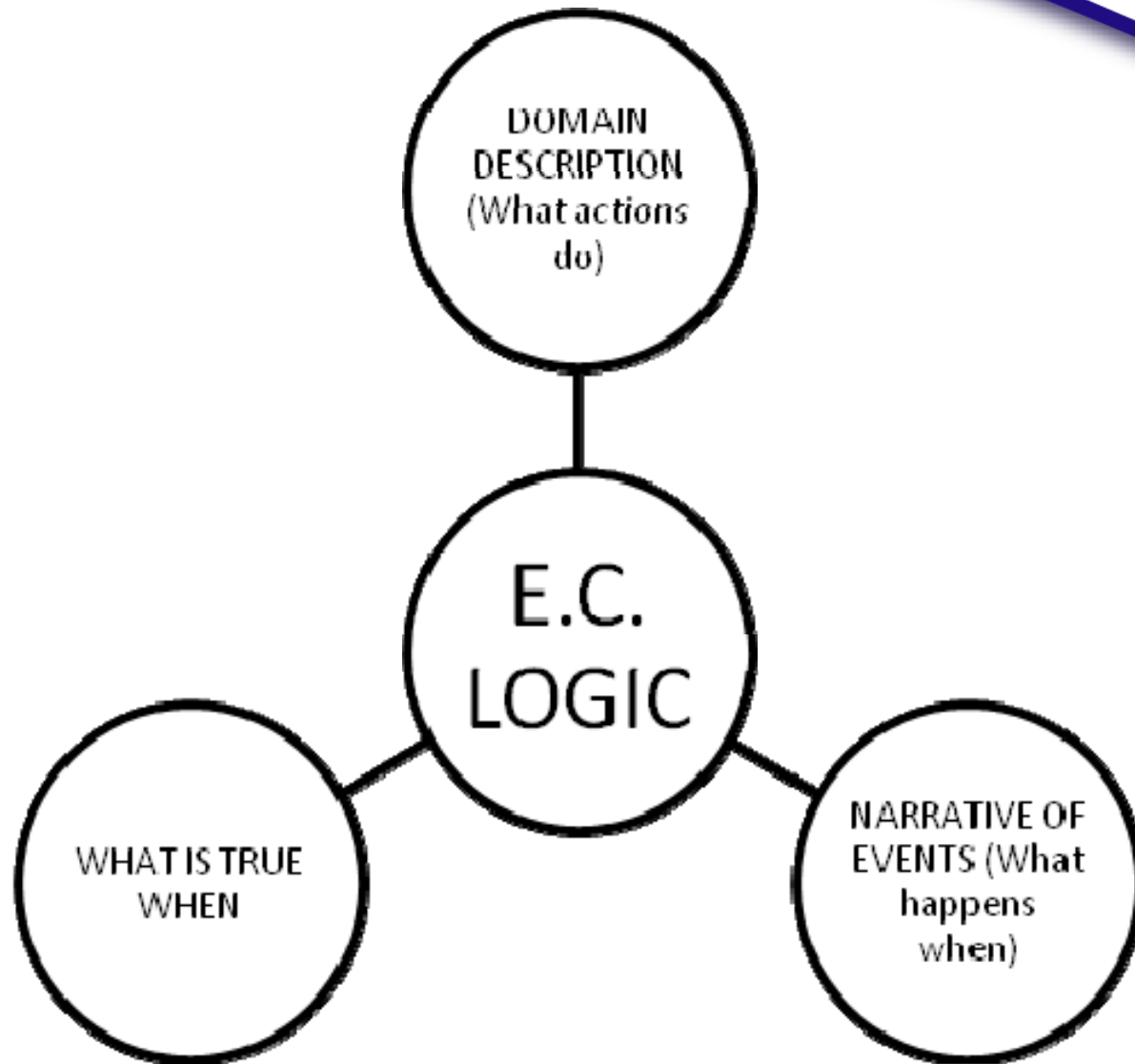
- Logic-based approach to describing events over time
- Consists of
 - Fluents (time-varying variables)
 - Events (things that happen that affect fluents)
 - Sorts (things in the world; objects, people, devices etc.)
 - Time
- Constructed from these are a domain description ('a formalisation') and a narrative of events
- This representation can be shown to be consistent on the basis of first principles

Event calculus 2



Predicate	Explanation
$\text{Happens}(e, t)$	Event e happens at time t
$\text{HoldsAt}(f, t)$	Fluent f is true at time t
$\text{Initiates}(e, f, t)$	If event e occurs at time t , fluent f will be true after time t
$\text{Terminates}(e, f, t)$	If event e occurs at time t , fluent f will be false after time t

Event calculus 3

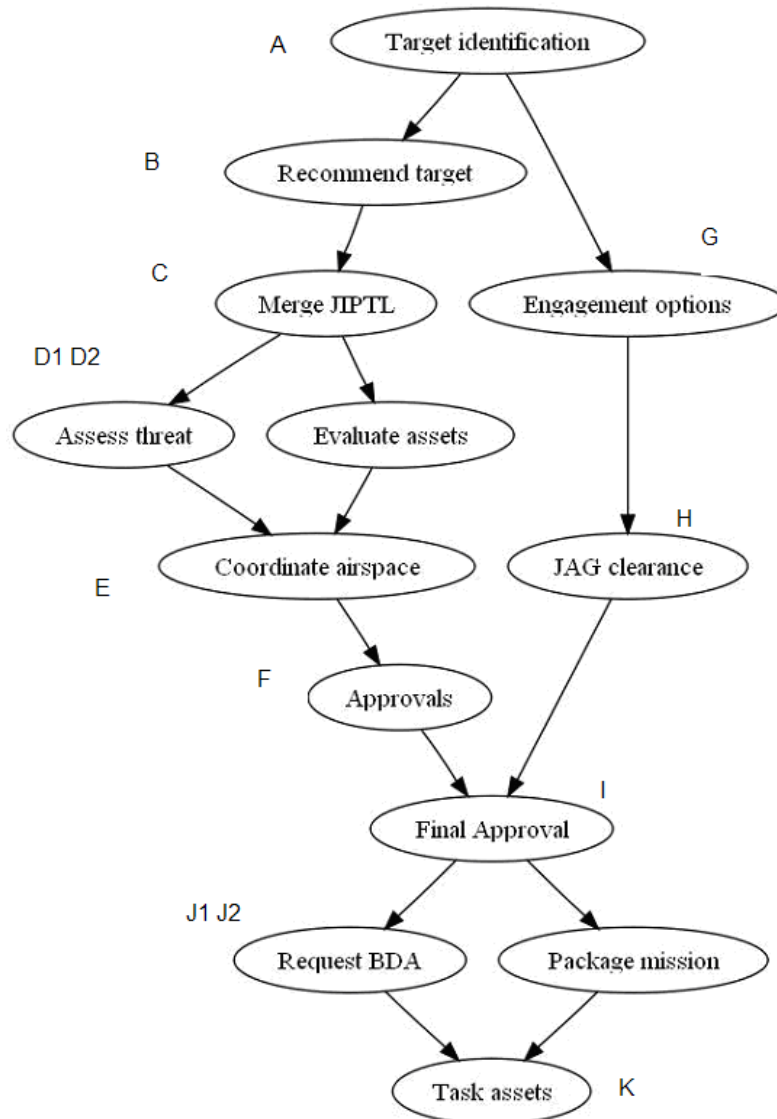


Task and people



- Task representation:
 - Linear flow
 - Event B follows event A etc.
 - Some AND-split/joins, Event I follows Events F & H
 - No attempt made in the model to understand the content of actions, just their sequential relationship to each other
- People representation:
 - Can only do one thing at a time
 - Scope for future work

Linear workflow



Approach to representation



- Based on Cicekli & Yildirim “Formalising workflows using the event calculus” (2000)
- Two states for each activity (active/completed) and two events that change them (Start/End)

```
Initiates(Start(activity),Active(activity),time).  
Terminates(Start(activity),Completed(activity),time).  
Initiates(End(activity),Completed(activity),time).  
Terminates(End(activity),Active(activity),time).
```

- Events can only occur once precursor events have occurred

```
Happens(Start(B),time -> !HoldsAt(Active(B),time) &  
!HoldsAt(Repeatlock(B),time) &  
HoldsAt(Completed(A),time).
```

How it works...



TIME	0	1	2	3	4	5	GOAL
EVENT	Start(A)	End(A)	Start(B)	End(B)	Start(C)	End(C)	no event
AGENT	Available	Busy	Available	Busy	Available	Busy	Available
TASK A		Active	Completed	Completed	Completed	Completed	Completed
TASK B				Active	Completed	Completed	Completed
TASK C						Active	Completed

Information requirements



- ...possibly touches on “Organisation”
 - Reflects NEC/NCW future systems
 - Information can be pulled/pushed from network
 - More flexibility in flow of activities
 - Difficulties:
 - How can we be assured dangerous paths of events will not occur
 - Are procedural safeguards sufficient?
 - Swiss cheese error model
 - How can this process be managed?
- Axiomisation:
 - Events can occur when their information needs are met

Representation



- Example of what events do:

```
Initiates(Targetidentification(target), KnowLocationTarget  
(target), time).
```

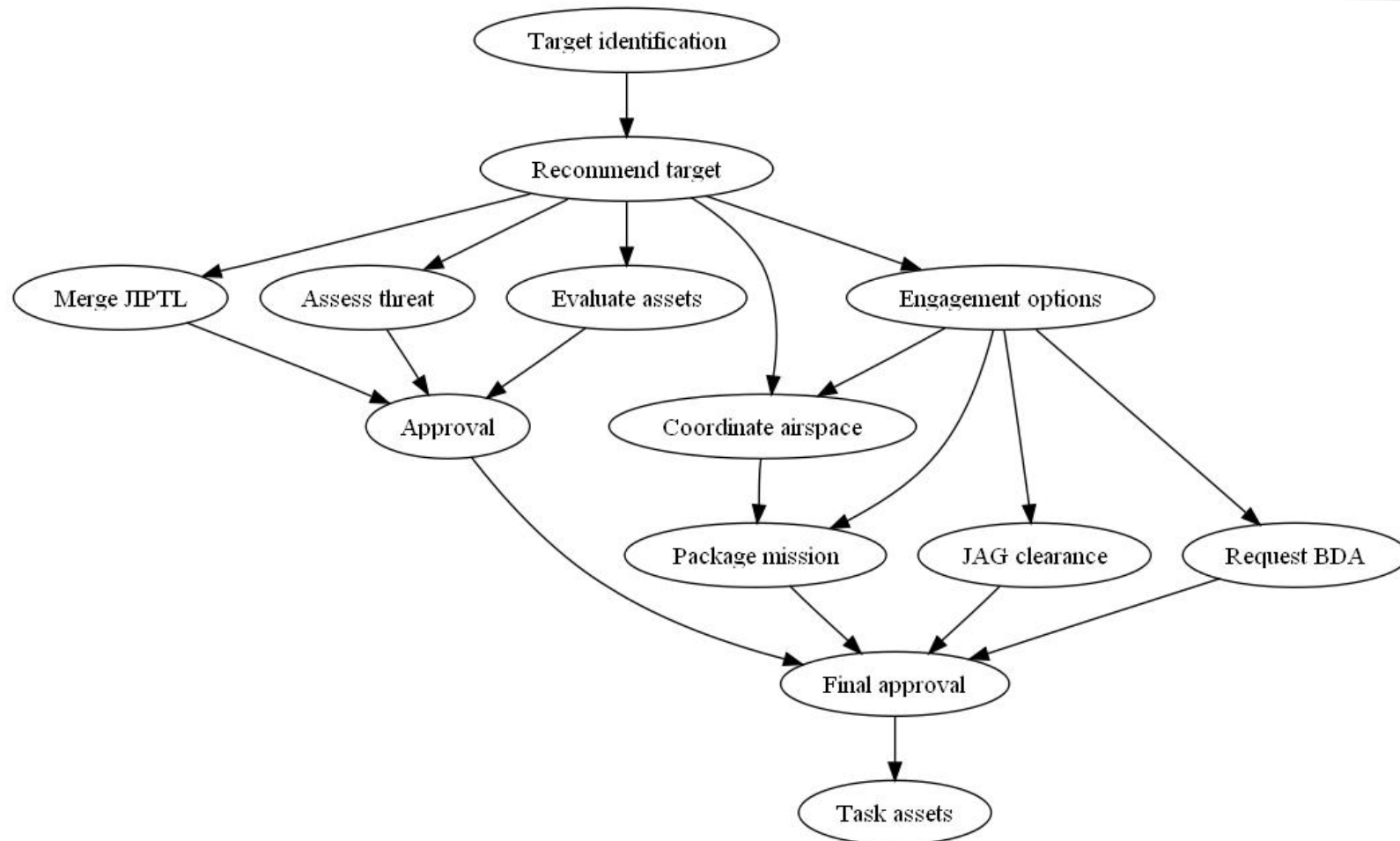
- When we have identified a target, it is true we know the location of that target

- Example of what events require:

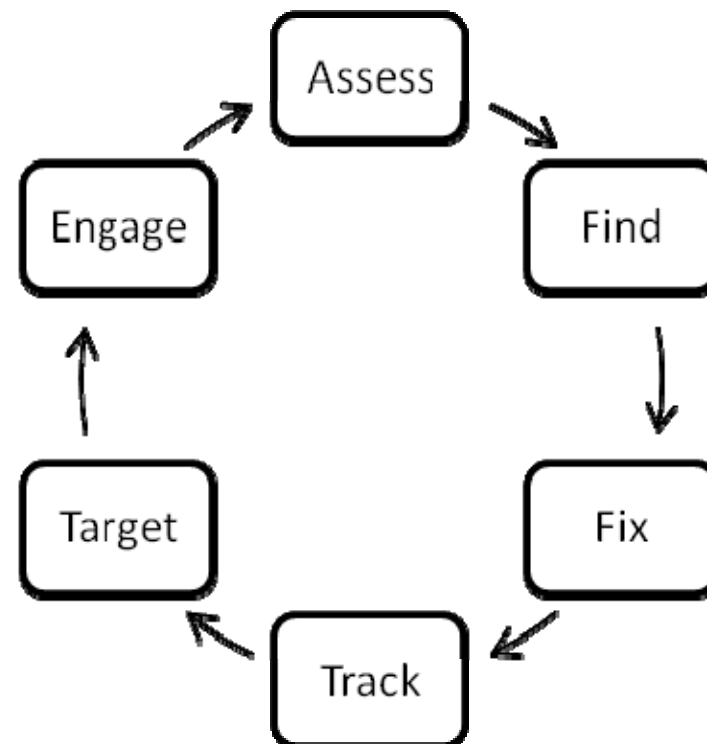
```
Happens(Finalapproval(target), time) -> HoldsAt(Approved(target), time) &  
HoldsAt(JAGcleared(target), time)  
& HoldsAt(KnowLocationTarget(target), time)
```

- Final approval can only be given once it is true that the mission has been approved, JAG has given clearance and we still know the location of the target.
- Differences from workflow:
 - Events considered in more fine grained 'meaningful' manner
 - States of fluents must remain true throughout the operation of the system, not just at specific points in time (earliest event directly influences the final event)
 - Analogy can be drawn here with a contract

One possible output...



F2T2EA Model



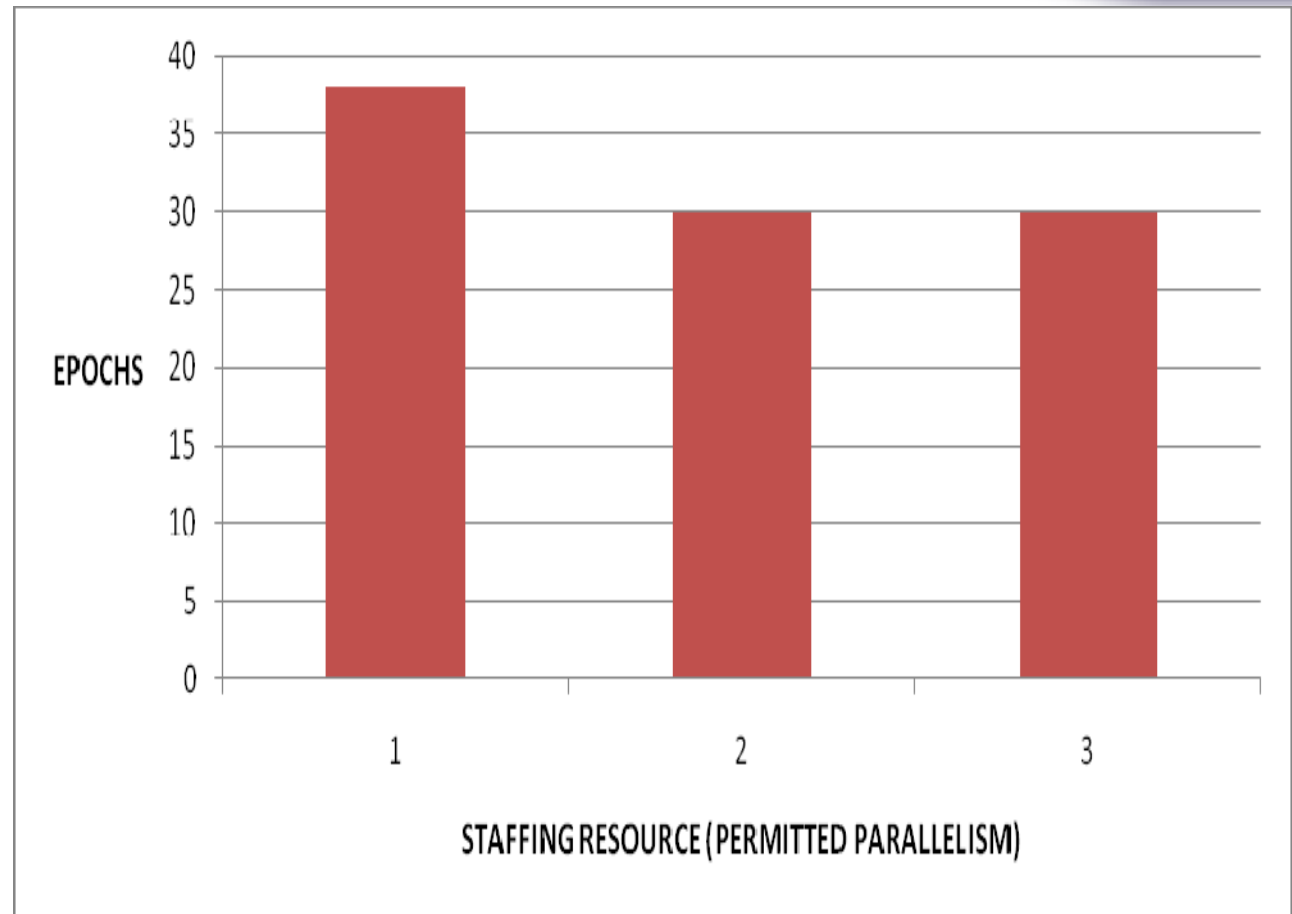
Engage and assess



Preliminary results



Time	Agent1	Agent2	Agent3
0	A start		
1	A active		
2	A comp		B start
3			B active
4	C start		B comp
5	C active		
6	C comp		D start
7			D active
8	F start		D comp
9	F active		
10	F comp/G start		
11	G active		
12	G comp		I start
13			I active
14		E start	I comp
15		E active	
16		E comp	
17			
18			
19			
20			H start
21			H active
22			H comp
23	J start		
24	J active		M start
25	J comp		M active
26			M comp
27			
28			
29			
30	R start		
31	R active		
32	R comp	P start	
33		P active	
34	S start	P comp	
35	S active		Q start
36	S comp		Q active
37	T start		Q comp
38	T active		
39	T comp		U start
40			U active
41			U comp
42		V start	
43		V active	
44		V comp	
45	X start		
46	X active		
47	X comp		
48	MISSION ENDS	MISSION ENDS	MISSION ENDS



EC conclusions



- Learning curve in use
 - But outputs convert naturally to English
- Allows different ways of thinking about things to be built into models to allow comparisons
 - Other techniques often have a fixed perspective on how a problem should be thought about
- Time consuming
 - Construction of models has potential to be automated
- Potential for an EC model when run in real time to function as the backbone for a decision support/management system

How this all fits in



- Analytical prototyping

