

The end-point is not enough

Duncan Martin, Mark Truran, Helen Ashman

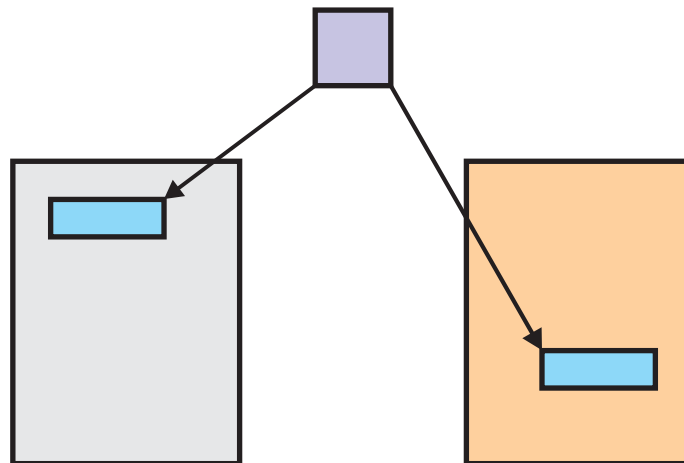
Web Technology Research Group

University of Nottingham

`{djm,mat,hla}@cs.nott.ac.uk`

Traditional link modelling

- A link is a semantic connection between parts of a document structure
- Link structures reference a number of end-points with each end-point describing a location within the document structure



Activating a link

- A link may be 'activated'
- Activation causes the reader's view to be affected in a way defined, implicitly or explicitly, by the link structure
- Links can be activated in a number of ways:
 - ★ Clicking on the area occupied by the end-point
 - ★ Clicking on a marker next to the end-point
 - ★ Selecting an area and requesting from the system links whose end-points fall within the area

The role of end-points

- Link structures may have **many** end-points
- But a particular activated link has **two** end-points
- One end-point is the area of the view affected — the ‘subject’
- The other is the ‘requested’ content — the ‘object’
- These are context-sensitive terms, for a bi-directional link each end-point can take the role of subject or object

Link actions

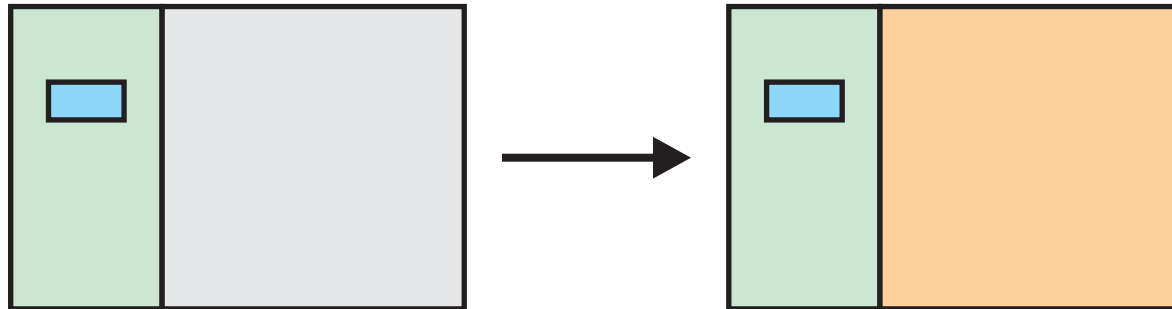
- Common linking actions are 'traversal' and 'appear':
 - ★ Traversal — replace the entire window with new content
 - ★ Appear — create a new window containing the new content

Link actions

- But also consider these actions:
 - ★ Include — bring new content into the current window
 - ★ Replace — replace content in the current window with new content
- Replace highlights that the activator is not always an end-point of the link

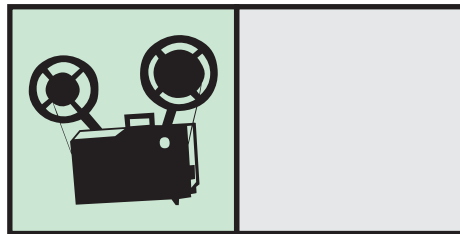
Replacement example

- Consider a page where there is a sensitive area which when clicked changes some of the content
- The subject end-point is the content to be replaced, the object end-point is the new content. The activator is neither end-point

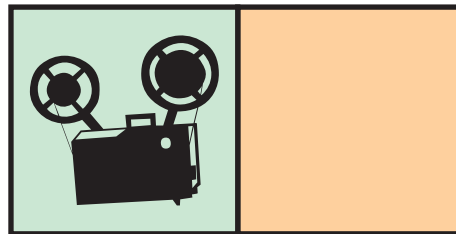


Video with annotations

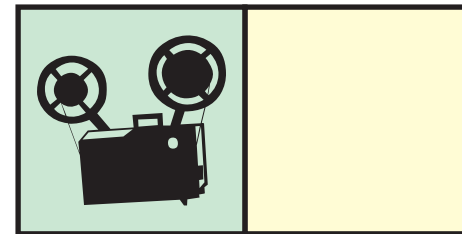
- A video may play in part of the view with content being updated at set time-points
- The link activators exist in/on the video, the subject and object refers to the content part of the view



time=0



time=1



time=2

So...

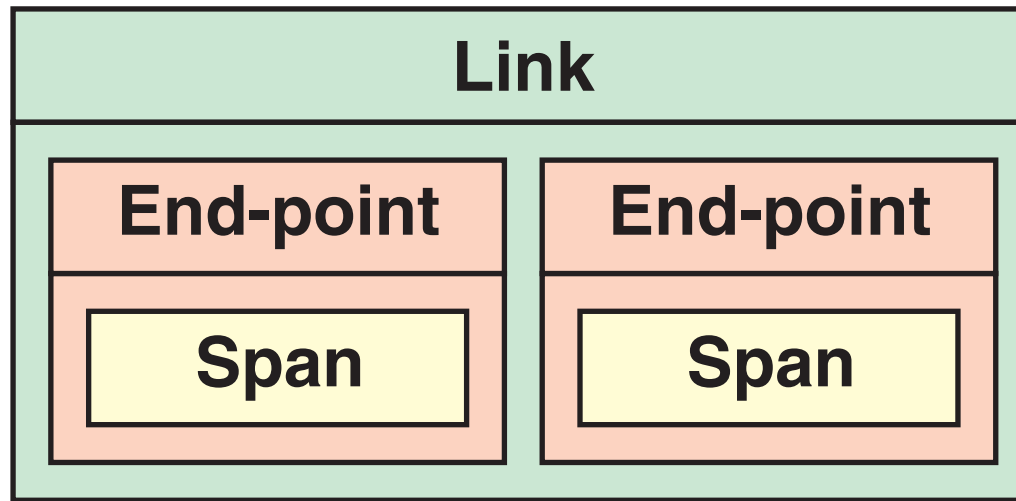
- The modelling of link activators should be separated from the modelling of end-points
- We describe the activator of a link as a **'trigger'**
- A trigger has three components; Span, Event and Presentation

Spans

- A 'span' describes an amount of media-space within the hypermedia
- The Span type should be capable of referring to any type of media used by the system, e.g.:
 - ★ A range of text
 - ★ An area of an image
 - ★ An area and time period of a video

Spans

- Spans are used as parts of end-points



- Spans are useful for describing where a trigger exists

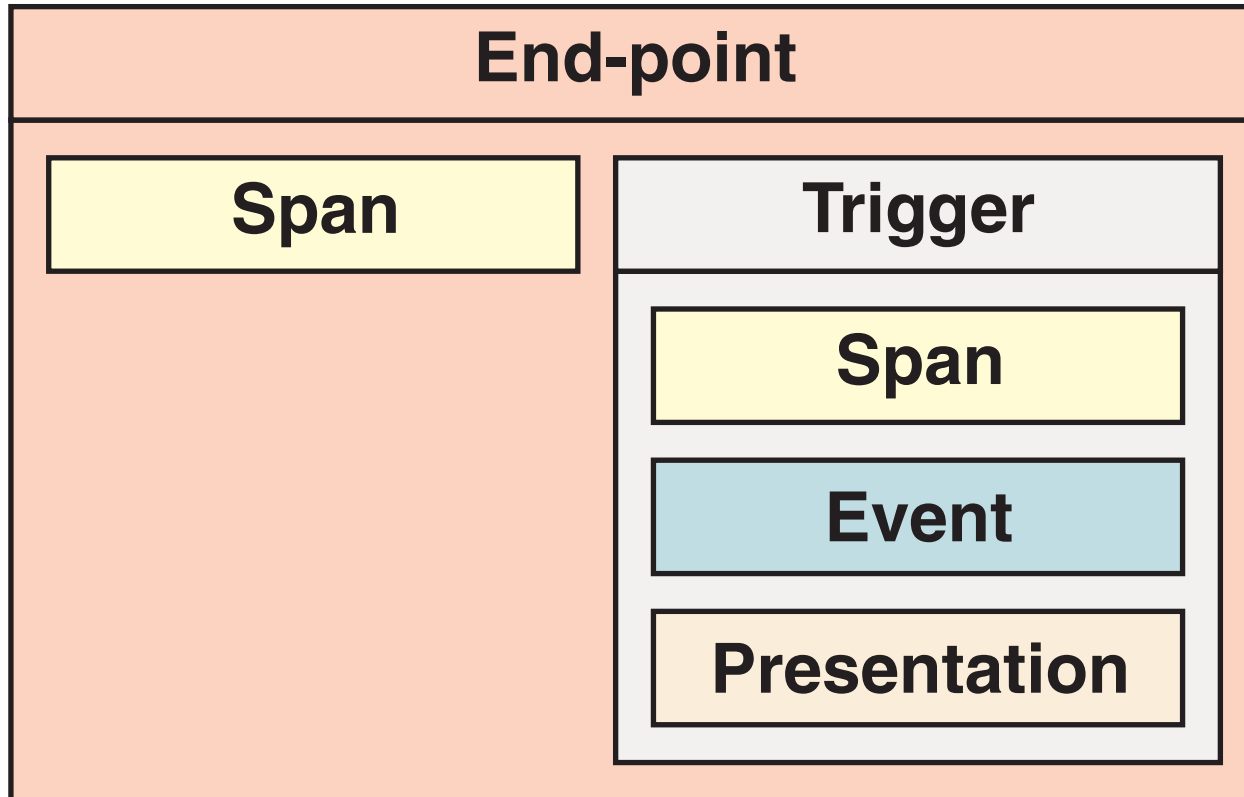
Events

- The 'Event' member describes the event which causes the trigger to activate. For example:
 - ★ `click` — Clicking on the span area
 - ★ `hover` — Hovering over the span area
 - ★ `instant` — Link is activated as soon as the trigger is presented, e.g. the annotated video example
- May include 'qualifications' such as the number of seconds to hover for

Presentation

- The 'Presentation' member describes how the trigger appears, e.g.:
 - ★ The span is not styled in any way
 - ★ The span area should be back-shaded
 - ★ An icon is placed by the side of the span area
 - ★ Multiple destinations are displayed in a pop-up
 - ★ Multiple destinations are displayed in a block
- A specified title may be necessary for some presentation styles

Modified end-point



Other considerations

- In a link model based around single-direction one-to-one links the trigger can exist directly under the link structure with no effect on interpretation
- Triggers may be raised to '1st class' objects in a system with end-points referencing the trigger objects rather than containing them

Summary

- End-points alone are insufficient to model hypermedia behaviours
- ... a separate trigger concept is necessary
- Triggers can be easily integrated into existing hypermedia models

Any questions?

