

Learning from deployed middleware

Whirlwind Retrospective
Adrian Friday,
Lancaster University

<http://ecampus.lancs.ac.uk>



Ethos

A dark, rectangular video frame, likely a still from a broadcast. It contains a blurred background image and white text centered in the lower half. The text reads "TETRA Mountain Rescue System" on the first line and "BBC Northwest (Breakfast)" on the second line.

TETRA Mountain Rescue System
BBC Northwest (Breakfast)

Applied systems research

“You can’t have Ubicomp without deployment”

Exemplars

- Create & deploy systems for use by the general public
 - 1996: MOST: a system for use by electricity field engineers - **adaptive middleware**
 - 1998: **context sensitive** tour-GUIDE for visitors to Lancaster
 - 2000: Emergency Multimedia: **L²imbo coordination middleware** supporting spontaneous/ ad-hoc collaboration
 - 2006: Equator (**Equip**, **ECT**, mixed reality experiences)

MOST: Mobile CSCW

- Traditional ‘**transparencies**’ model of distributed systems
 - Mask failures, RPC == LPC etc.
 - Mobile environments broke all that
 - Expose information from network, protocol stack, middleware
 - Let applications and users **adapt**

GUIDE



The Lancaster GUIDE Project : <http://www.guide.lancs.ac.uk/>

Application Bookmark

Back Forward Stop Refresh Info Collaborate Locator Help network connection

Locator



We seem to be lost! The last time I knew our location was when we were in The Tourist Information Centre area of Lancaster. Do you know if we're still in this area?

[Yes](#) [No](#)

Last know location cell
The Tourist Information Centre

Status
Last location update received 2 mins ago.

Tour Guide / Route Guidance

Follow A Tour Repeat Last Instruction Show Next Instruction

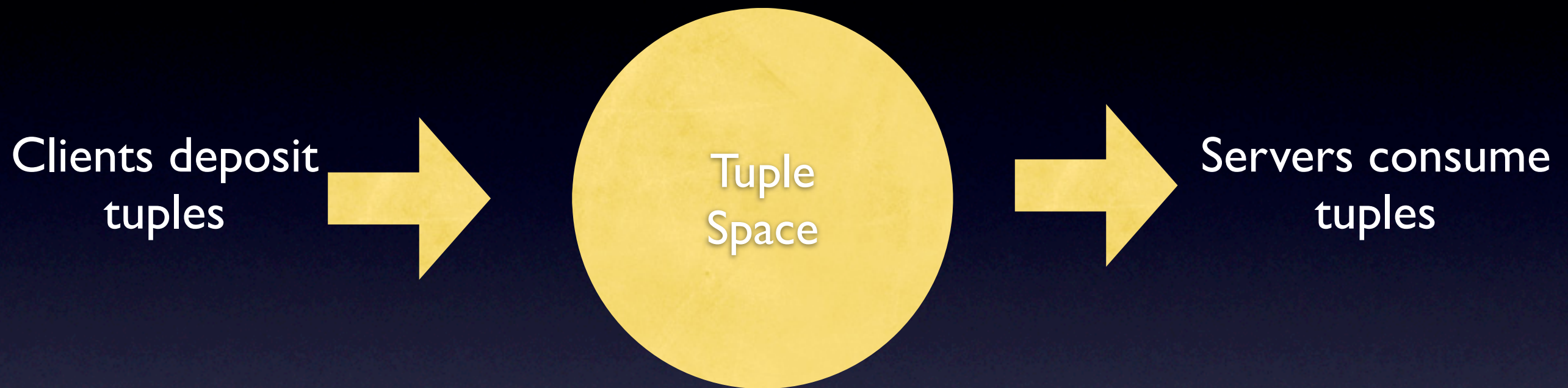
Interactive Services

Message Tickets

We didn't see that coming

- GUIDE was all about network protocol & cache design (no, really!); learnt *many* lessons including:
 - Public's **trust** of the system depended on content, **when** it was delivered and **honesty** about systems' coverage
 - Most painfully: **context-awareness** means **more content**; early tests blighted by us having to make excuses like:
 - “of course **it could** tell you where the nearest ATM is”
- Solution: employ students to work on content and nothing else - **no dissertations here** - provider model **not sustainable**
 - ~10 cells meant 15.3Mb of hand crafted content & 409 navigation points
- Retrospective: **Guide as a an experimental platform & technology probe for learning about use of such systems**

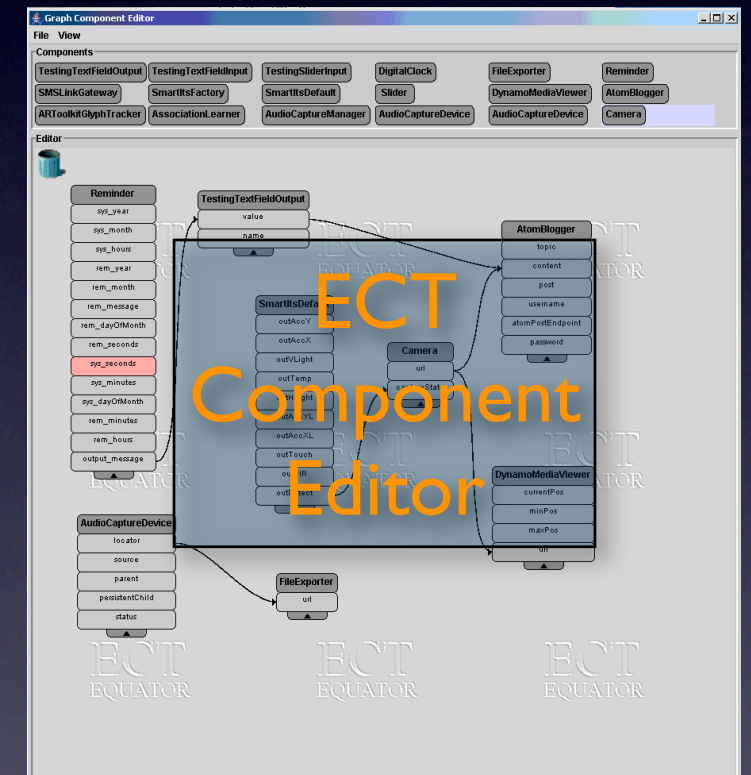
Tuples are Events?



- **L²imbo** was the answer?
 - No more RPCs, producers and consumers decoupled in time and space
 - No need for synchronous availability
 - Introspectable, externalised state, layer breaking TS
 - But, bootstrap problem, spotting changes, clumsy API(s)

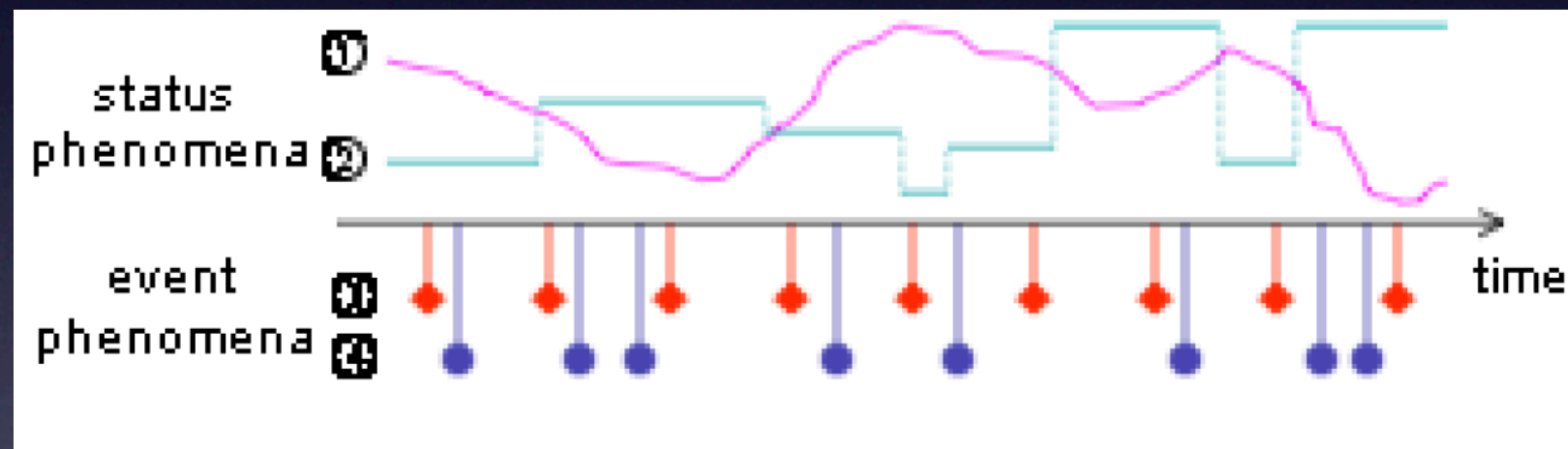
Equip and ECT

- A dataspace platform to underpin ‘mixed reality experiences’
 - Tuples become typed objects
 - ‘Update in place’ supported
 - Events when tuples are created, updated, deleted etc.
 - Subscriptions are also tuples
 - Peer and master-slave configurations
 - Problems with garbage collection and transient connections



Events and more events

- As toolkit components led to yet more events



- Optimisation problem - when are events necessary? Dependencies?
- how frequently should we sample?

A. Dix, J. Leite, and A. Friday, "Xsed – xml-based description of status–event components and systems," in *DSVIS - 14th Conference on Design Specification and Verification of Interactive Systems*, (Salamanca, Spain), LNCS, Mar 2007

Ongoing Work

- Simple Mobile Services (**SMS**)
 - Streamlining interaction with web based services for use on the move
- **RESTBroker** (with UBC)
 - A REST based middleware for client-less mobile-screen interaction
- **e-Campus**

e-Campus

- Large-scale deployment of networked displays across the campus
- Aims to provide one of the largest community of pervasive computing users in the world – engage students and staff on a daily basis
- Focus: software infrastructure for simplifying deployment & supporting context-aware and interactive applications (c.f. ‘the Grid’, ‘PlanetLAB’)
- Moving toward a wide range of **contributed** and **interactive** content - leverage/ engage community



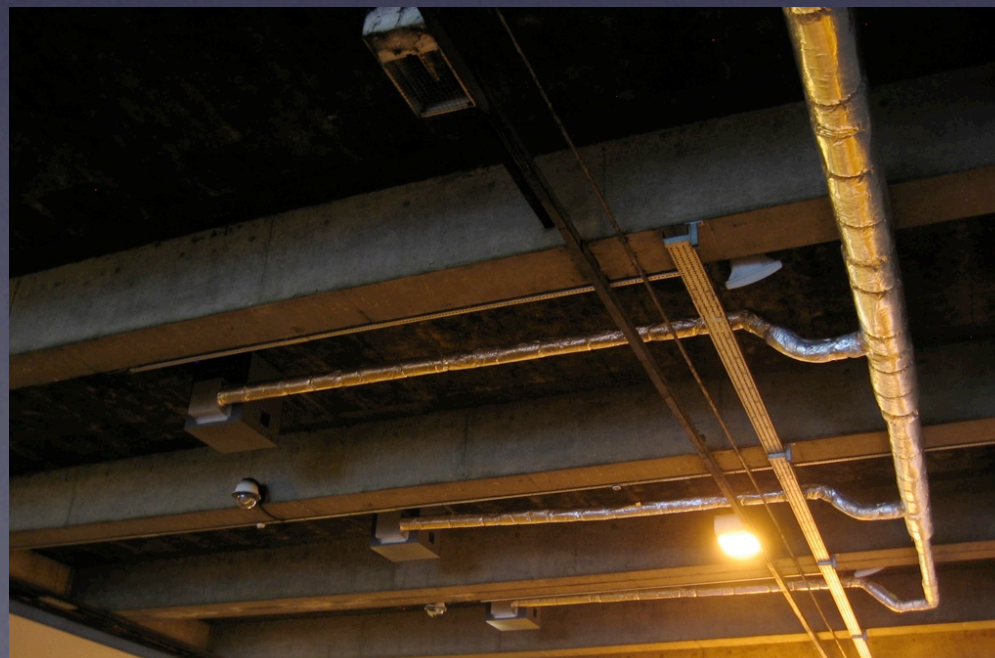
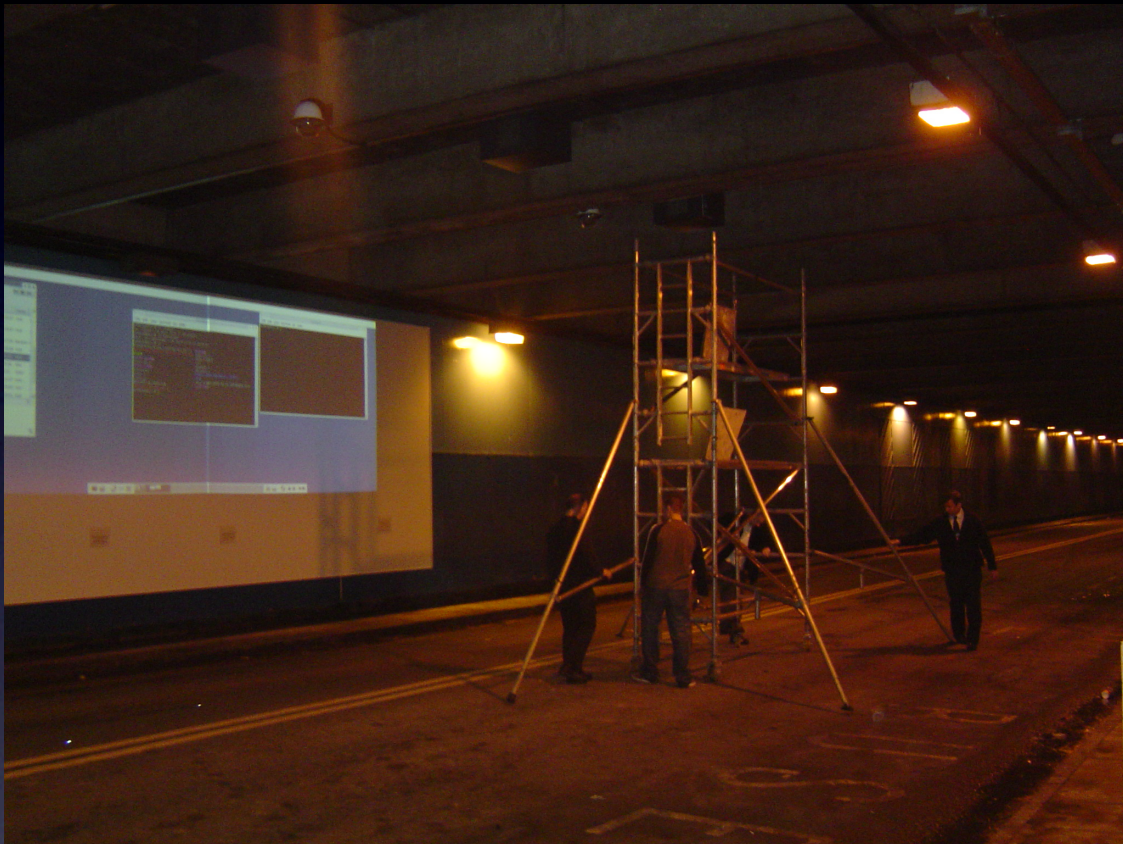
You get what you're given

- Totally unprepared for the quality of contributed content [[video](#)]
- Issues familiar to digital archivists
- Format, longevity etc.
- Still content that we were not allowed/ didn't want to show - **workflow**
- The ownership of mistakes - **determinism and control** of the content & presentation

Content management

- Legal issues: copyright, privacy, fair use etc.
- Inadequate tools for managing location of data & process
- Numerous versions of the same data
 - Raw, PC, Mac versions, formats, etc.
 - Quickly lost track of the meta-data
- Tools, formats & connectivity made decentralised workflow problematic
 - German naming scheme :-)

Prototype III: Underpass





Metamorphosis: Welfare State Intl./ ∴The Pooch∴

Tuesday, 29 April 2008

Artists care about their art

- Lots more requirements for the infrastructure
 - Carefully orchestrated “performances”
 - Ability to author and deploy in different domains
 - Support for control flow in the schedule
 - Support for special purpose hardware and software (‘tools of the trade’)

Current e-Campus Architecture

- Flexible distributed infrastructure
 - Low level API for building range of schedulers and presentation tools
 - Separate out scheduling from low level control (AV switching, playout etc.)
 - Provide support for **transactions on displays** - use them **in concert**
 - Provide abstraction for specialised hardware
- Written in **Python** with **Elvin** as the underlying event distribution system

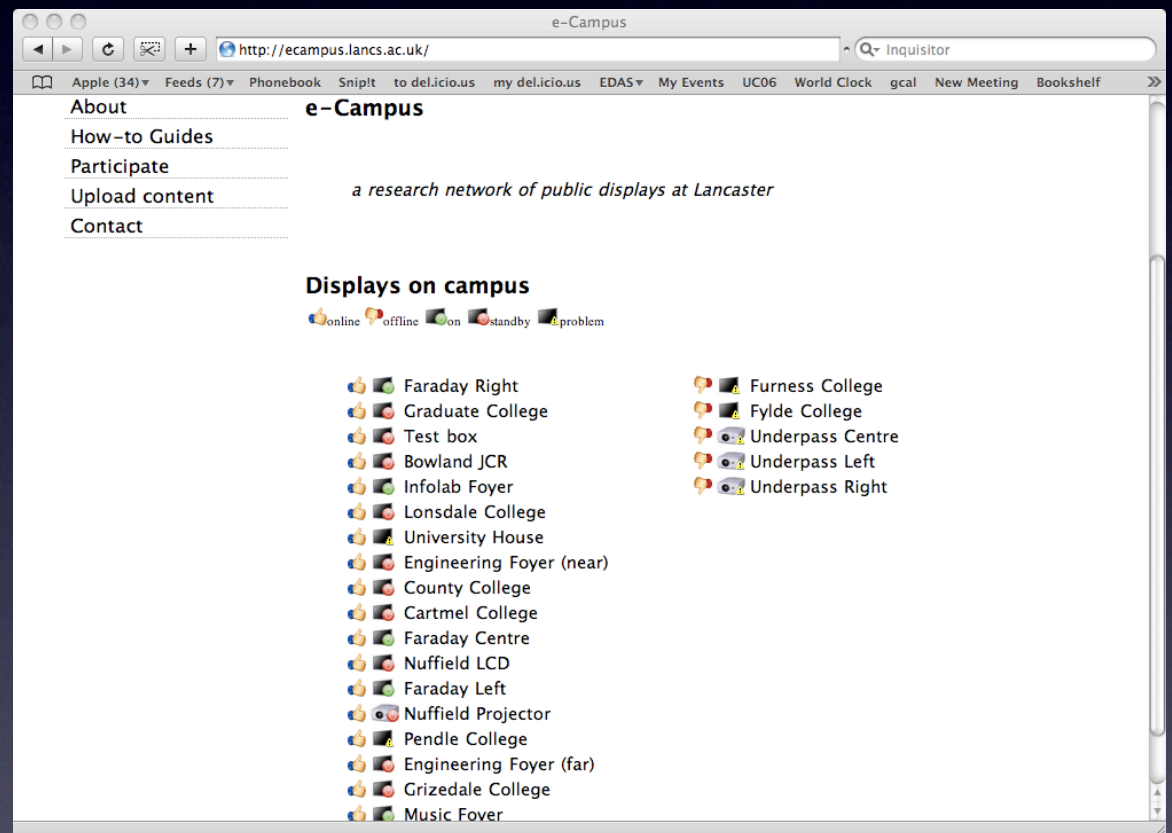
Actual code

```
a = api( connection )
gid = a.MakeGroupId()
t = Transaction( a, None )
t.CreateApplication( "ecampus-02", "file:/home/
ecampus/some-video.mpeg", gid )
t.ChangeState( gid, APPLICATION_STATE_PREPARED )
t.Commit()
```

- Group Id (**gid**)
- Create application adds an application to group **gid**
- **ChangeState** can take an application id or a **gid**
- N.B. Transactions are **visual** - algorithms have an aesthetic dimension

Issues Arising

- Centralised event broker
- Liveness of software components are global heartbeats
- Where should the state live...
- Content transport + workflow
- Web 2.0 high level API + 'simple' portal



Conclusion

- We definitely learn through deployment, **but...**
 - effort not to be underestimated!
- Roll out of e-Campus network is work in progress
 - 18 live, plans for 2 more external sites well advanced and 40 door displays ongoing
 - Steady stream of content being contributed
 - Platform gives us solid foundation for higher level abstractions and interaction work

e-Campus References

- O. Storz, A. Friday, and N. Davies.
Supporting content scheduling on situated public displays.
Computers & Graphics, 30(5):681-691, 2006.
- O. Storz, A. Friday, N. Davies, J. Finney, C. Sas, and J. Sheridan.
Public ubiquitous computing systems: Lessons from the e-campus display deployments.
IEEE Pervasive Computing, 05(3):40–47, 2006.
- Rafael Ballagas, Michael Rohs, Jennifer G. Sheridan, Jan Borchers
Sweep and Point & Shoot: Phonecam-Based Interactions for Large Public Displays
CHI '05: Extended abstracts of the 2005 conference on Human factors and computing systems, ACM Press, pp. 1200-1203, Portland, Oregon, USA, April 2005.