The GLYPH/GRG Game Development Environment: Requirements Progress Report

Walt Scacchi
Institute for Software Research
Donald Bren School of Information and Computer Sciences
&
Game Culture and Technology Lab
University of California Irvine
Irvine, CA 92697-3425 USA
http://www.ics.uci.edu/~wscacchi

Overview

- GLYPH/GRG Problem
- (Original) Proposed Solution
- Requirements Analysis
  - Non-functional Requirements
  - Functional Requirements
- Conclusions

The GLYPH/Game Research Grid problem

- What is the *best* way to rapidly create heterogeneous networked game worlds and play experience across devices?
  - "best" =>
    - faster, better, cheaper
    - open source (e.g., BSD/MIT style license)
    - (global) community-based development, contribution and support
    - Fun, enjoyable, intrinsically motivating, disruptive, etc.
    - Games can span cell phones, PDAs, PCs, LANs, Grids
- Modification, Construction, or Generation?

(Original) Proposed solutions

- Modification
  - Hack existing game content, levels, engine
  - Repurpose content/data from other sources
- Construction
  - Scripting (*UnrealScript vs. C-shell/Perl/Python/…*)
  - Custom programming using SDK and other tools
- Generation
  - Parameter value instantiation
  - Macro expansion
  - Language-directed (game) application generation
  - Meta-environments tailored for (game) domain

Requirements Analysis

- Non-functional Requirements:
  - Goals to achieve
  - Industry stack model
  - Updates to “Notes” memo of 4 March 2004
- Functional Requirements:
  - Operational
  - “Solutions” under study
  - Architectural implications
  - (Run-time platform) Implementation issues

Non-functional Requirements

Industry stack model

- Game user experience – *GLYPH focus*
- Game content/world – *GLYPH focus*
- Game engine/client: game world media players – *SDSC focus*
- Game grid middleware services – *SDSC focus via Butterfly.net*
- Game grid servers (software, hardware) – *Sun focus* (plus Nokia, Motorola, Sony, etc.)
- Telecomm carriers/broadcasters – TBD

Updates to “Notes”

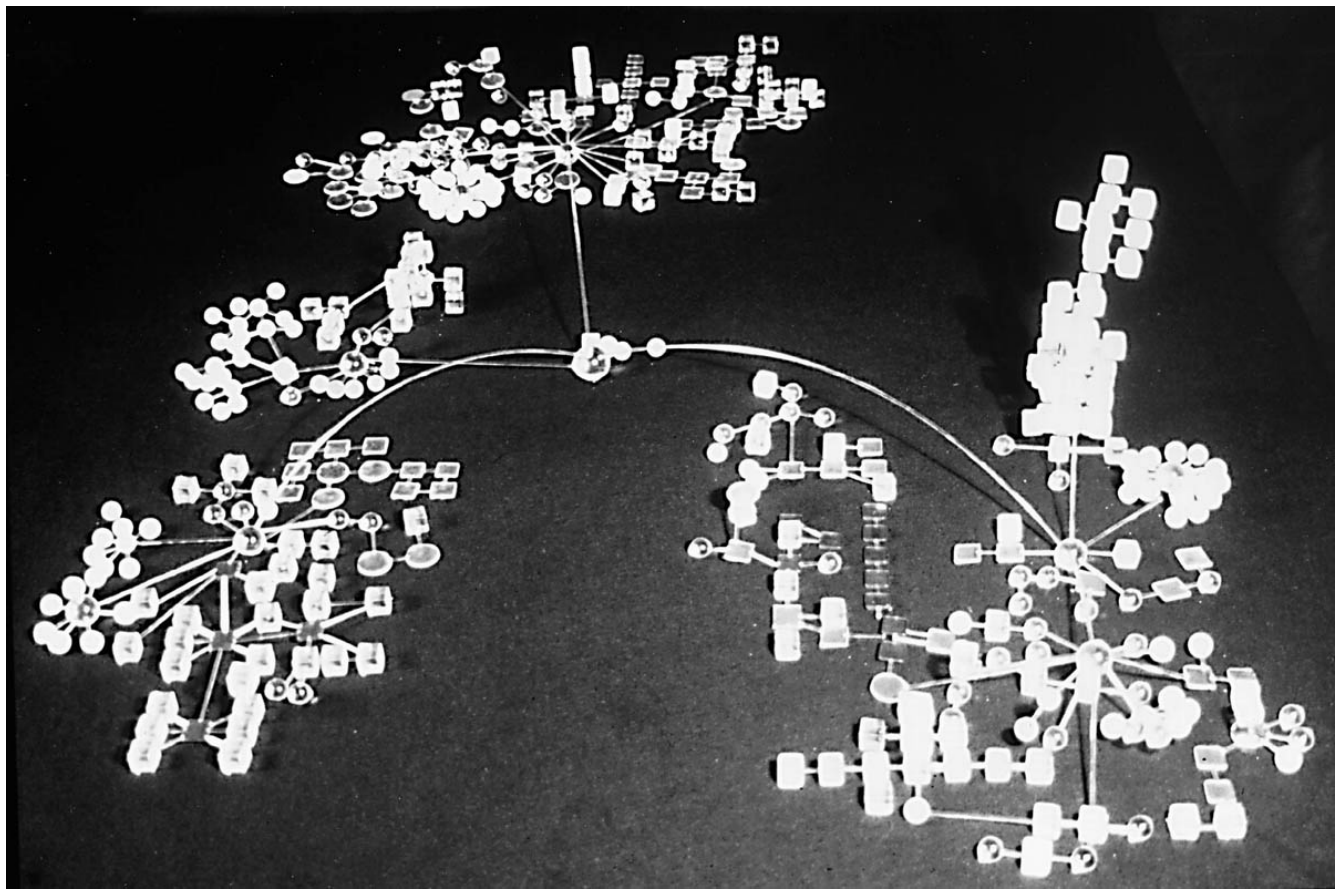
- Open source software development methods for networked games
  - Requirements, version coordination and release, continuous development, project mgmt and career (role) dev., tech transfer and licensing
- Game clusters and game development portal
- TBD: Game-based enterprise applications may follow!
  - Patent implications

Functional Requirements

Solutions under study

- Modification
  - Hack existing game content, levels, engine*
  - Repurpose content/data from other sources
- Construction
  - MUD/MOO-style object/room composition (GLYPH proposal)
  - Scripting (*UnrealScript vs. C-shell/Perl/Python/…*)
  - Extensible architecture (plug-ins, helper apps)
  - Custom programming using SDK and other tools*
- Generation
  - Parametrics* (texture/world generation)
    - Pre-compiled vs. on-the-fly* terrain generation
  - Macro expansion (Lisp-style, not Excel-style)
  - Language-directed (game) application generation
    - Game-based “storyline” languages and environments (GLYPH proposal)
  - Meta-environments tailored for (game) domain
- *Techniques currently being employed in Unexceptional.net prototype

Architectural implications: Game Clusters and Portal



Architectural implications:

- MUD/MOOs
  - Information space (*intra*-game) model
  - Information infrastructure (*inter*-game) model
    - Master game (MOG/MMOG—central nodes)
    - Intermediate games (branch nodes)
    - Simple or introductory games (peripheral/leaf nodes)
- Game space development and usage span:
  - Game engine (client-side), server browser (server-side), level and object (“brush”) editors, content management (game web site), bug tracking, computer-mediated communication (inside/outside of game play), and more

Implementation issues

- Cell phones are a moving target
  - No surprise
  - Need input from sponsor on where to aim
- Small screen vs. big screen phones
- Linux (Java Mobile System?) phones
- PDA, PSP, Nokia NGage, Gameboy phones?

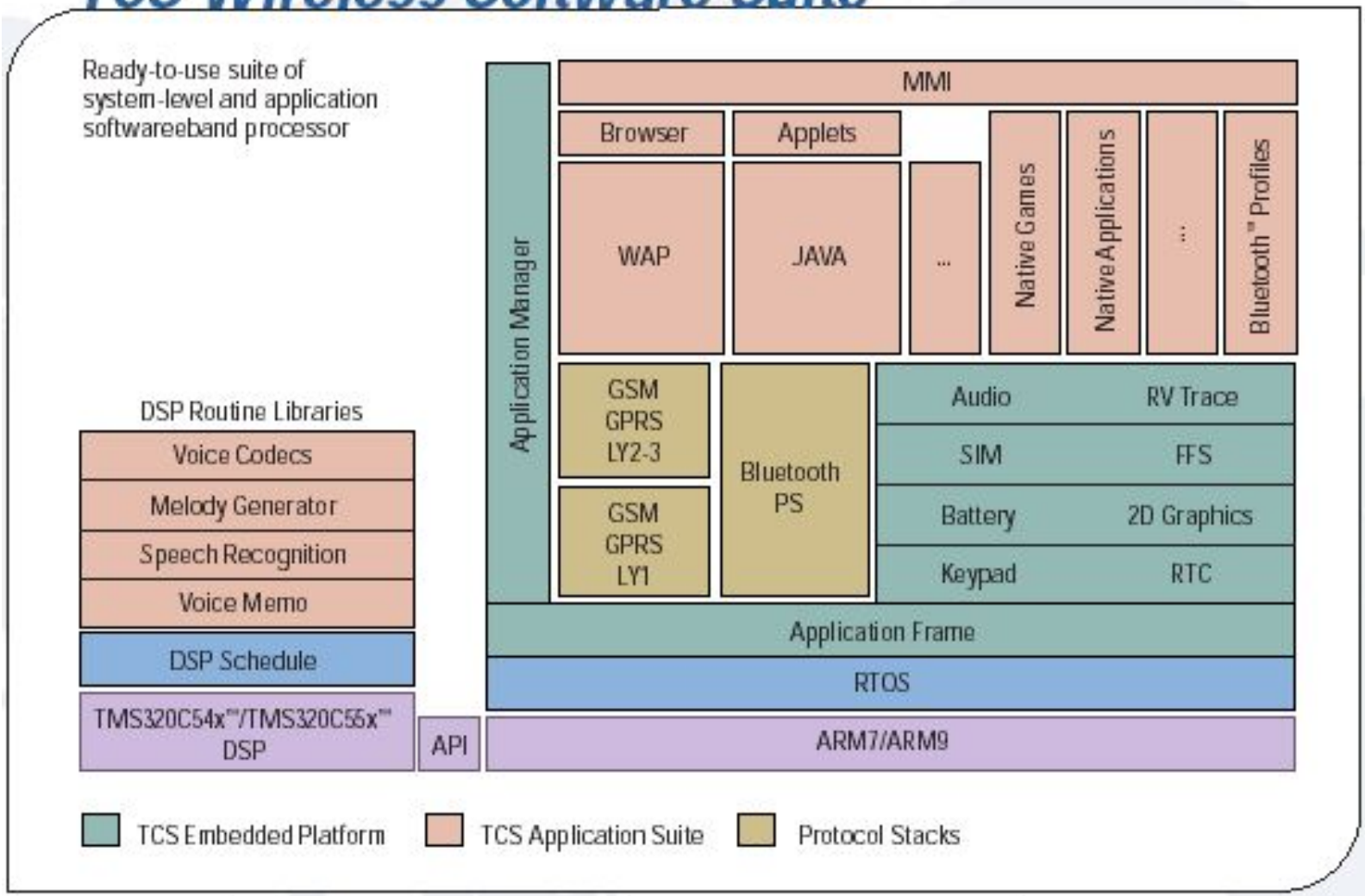
Implementation issues

- Telecomm. infrastructure
  - 2G, 2.5G, or 3G (or 4G)—which to target?
- "Everything integrated" vs. common phones
  - GPS (spatial resolution, map services)
  - MP3/WMA/AAC audio download,playback
  - Still/video download, playback, upload (w/Camera)
  - Haptic interfaces: hard buttons, soft buttons (touch sensitive), keyboards

Implementation issues: Cell Phones

- Graphics (lack of standards)
  - 2D: native SDK libraries (vs. *Flash*)
  - 3D: Java3D vs. OpenGL-ES vs. “DirectX10-CE”
  - Industry platform capture opportunity?
- Phone emulators—don’t really emulate phones
- Processor technology stacks (phone vs. telecomm) have limited communication
  - Games cannot easily access phone functions

TCS Wireless Software Suite



Conclusions

- Making significant progress on *Requirements* for GLYPH/GRG game development environments
- *Architectural models* create new options
- Need input from sponsor on *implementation platform targets* like cell phones regarding focus options