Prototyping and User Interface Design for Augmented Coaching Systems with MATLAB and Delphi

Implementation of Personal Tennis Coaching System

Author: Boris Bačić, PhD
boris.bacic@aut.ac.nz

MATLAB®, MathWorks®, www.mathworks.com
Delphi®/Object Pascal, Embarcadero®, www.embarcadero.com
Technology-supported/Augmented coaching

“Coaching is essential to improving as an athlete.

Whatever your goal is
– a coach will help you see the things you can’t
– and help you improve so that you don’t end up to do the same things over again”

Andy Murray, Professional Tennis Player.

(https://youtu.be/9qw7kKSobJg?t=5s, accessed 22 May 2015)
Introduction – Development of Augmented Coaching Systems (ACS)

\[\text{Study efforts} = \text{ICT THEORY and INFRASTRUCTURE} + \text{MODEL FEEDBACK} + \text{REAL WORLD APPLICATION HUMAN EXPERIENCE} \]

(Bacic, 2004)
Introduction

Problem description – Augmented Coaching System (ACS) design and development:

- Scientific rigour and validation of:
  - Automated elements of qualitative analysis of human motion that are inherent to coaching and rehabilitation
  - Subjective expert/coach's assessment of observed motion
  - Machine learning-based model to work on previously unseen motion patterns (i.e., tennis swing)

- User Interface (UI) design challenge
  - How to convey machine learning, capturing expert insights and other AI concepts, …
  - Applied to motion data representing human sporting activity … to mixed/diverse audience?

(Bačić, 2013)

(Bacic, 2008)
Introduction

Problem description – Augmented Coaching System (ACS) design and development:

• Scientific rigour and validation of:
  – automated elements of qualitative analysis of human motion that are inherent to coaching and rehabilitation
  – subjective expert/coach’s assessment of observed motion
  – machine learning-based model to work on previously unseen motion patterns (i.e. tennis swing)

• User Interface (UI) design challenge
  – How to convey machine learning, capturing expert insights and other AI concepts, ...
  – applied to motion data representing human sporting activity ...
  – to mixed/diverse audience?
Personal Tennis Coaching System (PTCS)

Application **design objectives**

- Providing end-users with *interactive coaching experience*, aiming to improve motor learning and stylistic execution of tennis swings
- Assessment and feedback of captured motion data to be automated, personalised and flexible
- Integrating various research artifacts – combining *qualitative analysis of human motion* and AI approaches designed in MATLAB
Prototyping of ACS/PTCS

BACKGROUND – MATLAB prototyping

• Building MATLAB codebase:
  – proof of concepts, non-continuous, incremental, cyclic and investigative development;
  – modular and reusable design;
  – rapid algorithms prototyping;
  – visualisation; code refactoring; code readability; and
  – integration of heterogeneous code and 3rd party resources.

• User interface and MATLAB codebase prototyping using: GUIDE

• Platform-independent 3D stick figure player initially developed in MATLAB
  for each frame,
    redraw stick figure topology using
    line(X, Y, Z, 'Marker', '.')
...

• Other developed applications can be invoked from MATLAB
User Interface (UI) design using Delphi

- **Motivation** for combining Delphi and MATLAB
  - *3D Player* – need for fast, responsive, accurate and interactive animated 3D stick figure replay for expert validation, visual feedback and multi-platform distribution with 3D motion data set. Standalone ver. or part of UI application.
  - Advanced features and rapid UI design
    - Extending MATLAB development to using Delphi’s visual and run-time components
    - Principles: Uniform 'look and feel', interactive options ('drag and drop', shortcuts, etc.).
  - Control of system resources (single CPU instruction commands, messaging etc.) and existing Delphi codebase

- **Revised Design Objectives**
  - Portable *3D Player* – Improving interactive and visual performance for 3D animated stick figure player without relying on third party graphic libraries
  - Centralised data management
  - Client/Server n-tier processing integrating MATLAB and Delphi
    - MATLAB as a *computational server*
    - Delphi as a *client*. Multi-layered application.
  - Efficient implementation of the concepts including:
    - Augmented Coaching Systems and underlying AI-based architecture
    - External synchronisation for visualisation and replay
    - Orchestration of constituent modules
Design and implementation options considered

1. "Code Generation in MATLAB"
   - Convenient for rapid prototyping and implementation of critical algorithms using C/C++

2. Manually translate and optimise MATLAB code ✔ ... for 3D Player
   - Time consuming but able to optimise the target code for various design and implementation objectives

3. Design and produce a stand-alone application in MATLAB (using GUIDE and callbacks)
   - Convenient for platform-independent UI demonstrations and compiled applications distribution

4. Design a UI in another programming language according to MATLAB help documentation ✔ ... for UI design
   - Combining preferences, programming languages strengths, existing codebase and RAD programming tools
   - Delphi integration not documented but "MATLAB COM Integration" is well documented as a starting point
Solution/Results (1/2)

• User profiles: Coach, athlete/learner, AI and Kinesiology scientists
  – Coaching view (or Student view) and
  – Modelling view (or HMMA view)

• UI data import (via start-up CLI, Open File dialogue or drag-and-drop)

• The UI shows: flexible assessment related to common mistakes, coaching rules, coaching drills/scenarios and diverse skill levels

• Assessment of individual swings and summary of the swings collection from observed activity
ACS/PTCS: Implementation Results

User profiles - views

Coaching view

Modelling view
ACS/PTCS: Implementation Results

Assessment summary of the swings collection

UI and MATLAB COM Server visualisations
- Relative and
- Absolute assessment (MATLAB `plot()` function)
Solution/Results (2/2)

• UI and MATLAB session control:
  – Implemented as "Shopping Cart" concept in eCommerce
  – Can initialise, re-start, and shut down MATLAB server
  – MATLAB server running as a service or an application
  – Access to MATLAB's: Workspace, Command Window, Help, ...
  – Two-way data synchronisation between: UI – MATLAB
  – n-tier data synchronisation: 3D Player – UI – MATLAB
  – multi-layered UI and (animated and interactive) 3D Player architectures similar to ISO/OSI model
  – Data visualisation in MATLAB e.g. `plot()` graphs invocation
  – Multiple MATLAB commands and scripts invocation
ACS/PTCS: Implementation Results

MATLAB COM Server – can run as a (hidden) service or as another application (next slide ...
ACS/PTCS: Implementation Results

UI and MATLAB COM Server
– session control
– orchestration of constituent models
– external synchronisation for visualisation and replay

MATLAB COM Server
– Command Window (next slide...)
ACS/PTCS: Implementation Results

**MATLAB COM Server – Command Window**

- UI and MATLAB COM Server
  - session control
  - orchestration of constituent models
  - external synchronisation for visualisation and replay
Discussion

Limitations

• UI prototype constraints:
  – Single MATLAB session control
  – Singleton 3D Player invocation
    NB: Multiple instances of 3D Player could be invoked (with start-up file preloading and closing view) from MATLAB server, CLI or File Explorer.

• PTC prototype limitations
  – Platform dependent development: COM technology is MS Windows specific
  – Installation and distribution issues
  – Need for local installation of MATLAB
  – Computational power limitation – need for parallel processing scalability

Advantages

• Fast and responsive UI; run-time MATLAB modelling and data analysis. E.g. there is no need to recompile the UI when:
  – working directly on MATLAB server (e.g. using Command Window or modifying data) or
  – replacing MATLAB codebase modules.

• Relative ease of single machine development and testing vs. distributed client/server hardware architecture
Conclusion and findings

• Combining Delphi and MATLAB COM server is:
  – Suitable for research, development and concepts communication to diverse audience
  – Not recommended for applications distribution
  – Not suitable for commercial development
• Both Delphi and MATLAB development env. and code:
  – are relatively easy to use, maintain, test, debug and modify;
  – (re)gain familiarity for further advancements and reusability
  – are complementing each others as ‘best of both worlds’
  – have available open source codebase established over years
• Delphi/Object Pascal:
  – is universal software development tool, recommended for multi-platform RAD with similarities to C/C++
  – compared to MATLAB – it lacks rapid R&D capabilities, equivalent documentation and eco system support ...
Future work and opportunities

• Further advancements in ACS for coaching, injury prevention and rehabilitation.
• Acquiring and processing multimodal motion data from diverse input devices using MATLAB’s toolboxes (video, image and signal processing).
• Developing non-visual MATLAB components for Delphi (in collaboration with Embarcadero)
• Extending client/server development and computational power to distributed processing
• Platform independent developments
• Implement visual development environment for Human Motion Modelling and Analysis (HMMA) to:
  – bridge kinesiology and AI research
  – assist with proof of concept testing and
  – rapid prototyping
Thank you!

Questions?

Related work and references


Extra reading

MATLAB help topics:

• Run External Commands, Scripts, and Programs
• Packaging UIs as Apps
• MATLAB API for other languages
• MATLAB COM Integration