Transmitting Avatar Emotions over the Web

By Jing Yuan

Advisor: Dr. Chris Pollett

Committee Members: Dr. Ho Kuen Ng

Dr. Sin-Min Lee

December 9, 2003

Outline

- Introduction
- Design
- Requirement
- Implementation
- Conclusion

Introduction

- General description of the project:
  - A network poker game with 3D-characters being able to express emotional information based on their cards.
  - Interface
  - Poker
    - Five-card draw poker allows four persons to play.
    - Game rules and information etc.

File Edit View Favorites Tools Help

Back Forward Stop Refresh Home Search Favorites Media

Address http://localhost:8080/examples/servlet/startPage/

***** NETWORK POKER GAME *****

Enter a new game: ☐ Or select: ▾
Select player: rightplayer ▾

CONNECT Clear

whose turn:

Emotion:

Game: game1
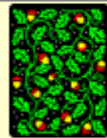
Player: rightplayer

Help

Return to Main Page

☐ **Hold**   ☐ **Hold**   ☐ **Hold**   ☐ **Hold**   ☐ **Hol**
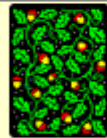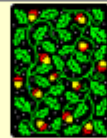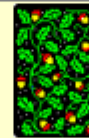
Click New Game to shuffer the cards. Click dice to get the turn of players.

BET ☐

Credits: 100

Refresh: 20 ▼

START

END

DEAL   DICE   Raising   Calling   Foldin
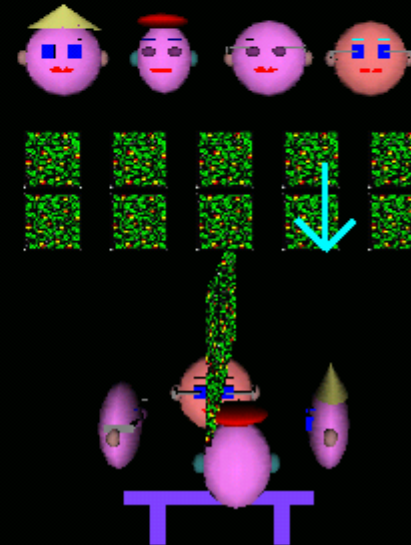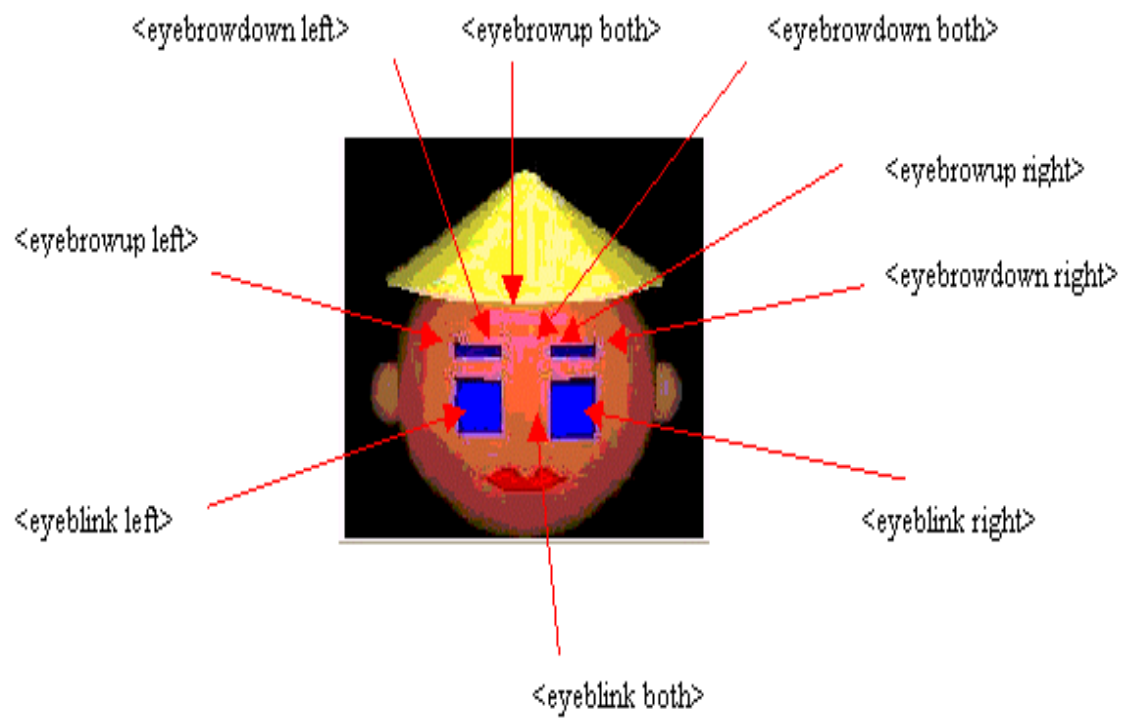
Introduction (Cont.)

- XML
  - An XML file is a tag-based document used to describe specified emotion that each player has.
  - A well-formed XML document.
    - All tags must have a corresponding ending tag
    - No overlapping tags
  - Init file is an XML document describing what emotions a player supports.
  - Also transmit emotions using an XML format of our creation.

Introduction (Cont.)

- VRML
  - A system for describing 3D scenes on the Web.
  - It needs a Plug-in (for example: Cortona) to view 3D graphic.

- JavaScript
  - A program that is included on an HTML page.
  - Allows us create an active user interface.
  - Used to display VRML content.

Introduction (Cont.)

- DOM (Document Object Model)
  - Used to traverse a tree-structured XML input source document.

- Servlet
  - A server-side software component, written in java, that dynamically extends the functionality of a server.

Start Page
(Java Servlet allows user to enter or choose a game to connect.

After game connects, it stores player.xml on the server.

The page allows user to play network poker game.

Top frame allows user to choose emotion message and play poker such as deal, calling, raising, and folding etc.

1. Passing info of emotion message.

2. Passing info of playing poker.

3. Using DOM API to parse player.xml file.

Content frame displays output of VRML 3D players' faces with emotion message and corresponding 3D poker cards.

gameServlet

- init()
- doGet()
- doPost()
- destroy()
- sendErrorToClient()
- printHTML()
- printOptionList()
- showDice()
- sendPlayerOrder()
- sendPlayerCards()
- sendDeal()
- sendCards()
- sendPlayerScore()
- readPlayerCards()
- readOrderFile()
- updateOrder()
- dealCard()
- getCardForPlayer()
- newGame()
- sendBet()
- updateDeck()
- loadOurCard()
- evaluateHand()
- readPlayerScore()
- readPlayerBet()
- readPlayerCards()
- calling()
- readPlayerCredit()
- raising()
- readOldPlayerBet()
- folding()

startPage

- init()
- doGet()
- doPost()
- createCSS()
- connectToServer()
- sendPlayer()
- sendPlayerToFile()
- connectPlayer()
- startGame()
- printOptionList()
- sendErrorToClient()

gameServletTwo

- init()
- doGet()
- doPost()
- sendErrorToClient()
- destroy()
- printHTML()
- printJavaScript()
- printFunctionCloseVRML()
- printFunctionGetVRML()
- printFunctionMove()
- printVRMLFile()
- writeVRMLRoute()
- writeRightPlayer()
- defineSensor()
- writeLeftPlayer()
- writeFrontPlayer()
- writeBackPlayer()
- writeMouth()
- writeMouthWithSmile()
- writeMouthWithSad()
- writeLeftEar()
- writeRightEar()
- writeTable()
- writeRightCardWithDeal()
- writeRightCardWithoutDeal()
- writeLeftCardWithDeal()
- writeLeftCardWithoutDeal()
- writeFrontCardWithDeal()
- writeFrontCardWithoutDeal()
- writeBackCardWithDeal()
- writeBackCardWithoutDeal()
- getPlayerFile()
- parseXMLPlayer()
- stepThroughAll()
- getSomeoneDealt()
- showcard()
- getCardFromFile()
- isCard()
- getValueAsInt()
- getSuitAsInt()
- getCard()
- getCardSuit()
- isFullHouse()
- isFlush()
- isStraight()
- isHighStraight()

Card.java

- Card()
- GetSuit()
- GetValue()
- GetSuitAsString()
- GetValueAsString()
- ToString()

Score.java

- Score()
- GetScore()
- GetPlayer()

Club.java

- Club()
- GetClub()
- GetPlayer()

SortScore.java

- sortScore()
- GetScore()
- addScore()
- getScoreCount()
- sortByValue()

Bet.java

- Bet()
- GetBet()
- GetPlayer()

Hand.java

- Hand()
- Clear()
- AddCard()
- RemoveCard()
- GetCardCount()
- GetCard()
- SortBySuit()
- SortByValue()

Deck.java

- Deck()
- Shuffle()
- CardsLeft()
- DealCard()

Requirements

- Browser
  - Cortona plug-in
  - Other software: Cosmo player, OpenWorlds etc.

- Software
  - Java Servlet API
  - Xerces2 Java Parser 2.2.1

Requirements (Cont.)

- Running Servlets
  - E.g.: http://localhost:8080/examples/servlet/startPage
- XML file
  - Meets a well-formed XML constraints and validated by DTD.
  - If there is any syntax errors, no output… etc.
- DTD file
  - Use xmlspy 5.0 checking DTD.

Requirements (Cont.)

- XML tags supported in this project

Tag	Attributes	Values
players	-	-
player	name	-
eyeblink	which	left | right | both
eyebrowup	which	left | right | both
eyebrowdown	which	left | right | both

Table 3.2: XML tags and attributes supported in the project

Implementation

- Emotions and Gesture Avatar DTD
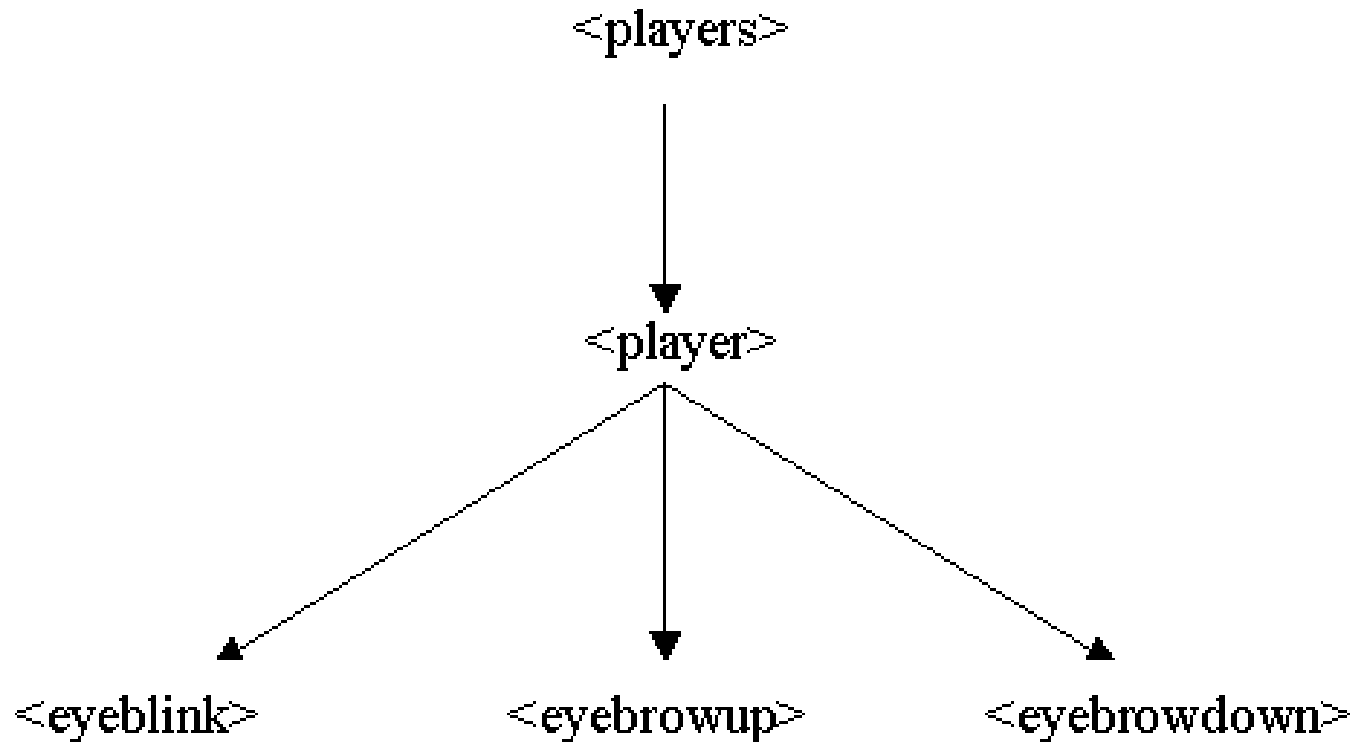 - Players
  - Player
  - Facial animation elements
  - Emotion elements
  - Gesture elements

Implementation (Cont.)
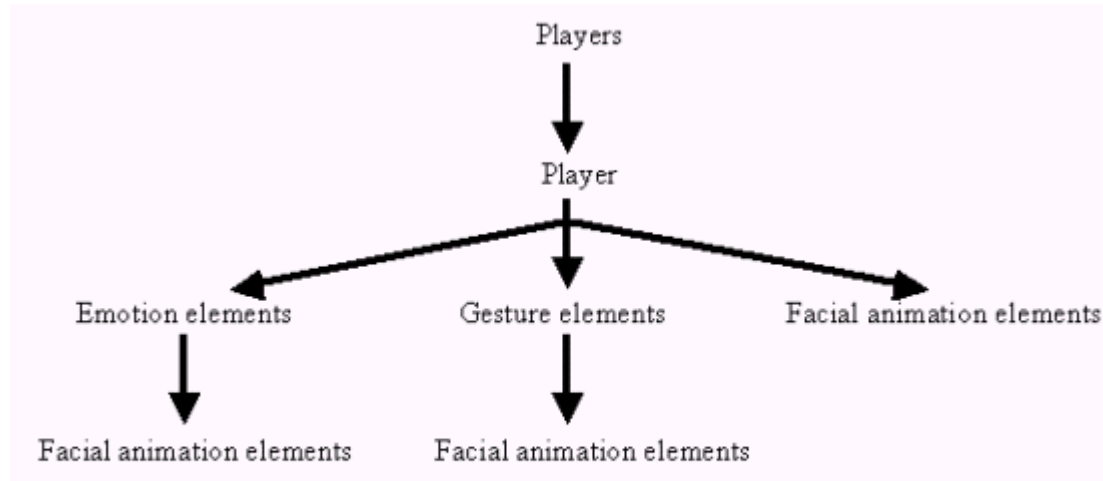
Figure 4.1: The element structure of DTD

Example of XML Document Structures

Implementation (Cont.)

- JavaScript
  - To force visualization in a frame by opening the frame document with a VRML mime type, and then having the JavaScript function write the appropriate VRML code.
  - Example of code.

```
out.println("<script language=\"javascript\" type=\"text/javascript\"> ");

out.println("<!-- Hide script from old browsers ");
out.println("function main () { ");
out.println("var doc = parent.content.document; ");
out.println("var doc = getVrmlDocument(parent.content.document); ");
out.println(" doc.open(\"model/vrml\"); ");
out.println(" doc.writeln(\"'#VRML V2.0 utf8    \"); ");
//More code here is deleted
out.println(" doc.writeln(\"'}                  \"); ");
out.println(" doc.close(); ");
out.println("}");

out.println(" function closeVrmlDocument() { ");
out.println(" var d = this.doc; ");
out.println(" d.open(\"text/html\"); ");
out.println(" d.write('<body leftmargin=\"0\" topmargin=\"0\" scroll=\"no\">'); ");
out.println(" d.write('<object id=\"Cortona\" width=\"100%\" height=\"100%\"
classid=\"clsid:86A88967-7A20-11D2-8EDA-00600818EDB1\">'); ");
out.println(" d.write('</object></body>'); ");
out.println(" d.close(); ");
out.println(" var e = d.all[\"Cortona\"].Engine; ");
out.println(" e.RootNodes.Add(e.CreateVrmlFromString(this.syntax)); ");
out.println(" } ");

out.println(" function getVrmlDocument(doc) { ");
out.println(" var isIE = navigator.appVersion.indexOf(\"MSIE\") != -1; ");
out.println(" if (!isIE)  ");
out.println(" return doc;  ");
out.println(" var adapter = new Object(); ");
out.println(" adapter.open = new Function(\"mimetype\", \"return;\"); ");
out.println(" adapter.write = new Function(\"s\", \"this.syntax += s;\"); ");
out.println(" adapter.writeln = new Function(\"s\", \"this.syntax += s; this.syntax
+='\\\\n';\");  ");
out.println(" adapter.close = closeVrmlDocument; ");
out.println(" adapter.doc = doc; ");
out.println(" adapter.syntax = \"\"; ");
out.println(" return adapter; ");
out.println(" } ");

out.println("// End hiding script from old browsers --> ");
out.println("</script> ");
```

Implementation (Cont.)

- VRML
  - Displays 3D face and cards.
  - Same code:
    - ➢ Smile (default emotion)
    - ➢ Eye blink

```
String  rs = "DEF eyeblinkTouchSensor
    TouchSensor { } " +

"ROUTE COLOR_PATH.value_changed TO
    eyeColorRightPlayer1.set_diffuseColor ";

out.println(" doc.writeln(\" " + rs + " \"); ");
```

```
String s = " geometry IndexedFaceSet { " +
   "  coordIndex [ 0 1 2 3 4 5 ]    " +
   "  convex FALSE  " +
   "  coord DEF COORD Coordinate {  " +
      "  point [ 1 0.5 0, 0.5 0.2 0, -0.5 0.2 0,
         -1 0.5 0, -0.5 -0.2 0, 0.5 -0.2 0 ]  } } } ] } ";
out.println(" doc.writeln(\" " + s + "  \"); ");
```

Implementation (Cont.)

- Poker Game
  - Number of players
  - Cards
  - Basic rules

  Score of Hand = (Value of Card – 1) * 4 + ranking of the hand

  - Ranking of hands
    - High card, one pair, two pair, three of a kind, straight, flush, full house, four of a kind, straight flush
  - Betting and Playing
    - Calling, raising, and folding.
  - Lock( ) and free( )

Conclusion

- Developed an encoding scheme for recording human facial expressions by using DTD, which is XML-based language.

- Using XML application to produce realistic facial emotional character agents in 3D graphics.

Conclusion (Cont.)

- Allows users to play real poker game through Internet and display facial emotion for each player based on the player's cards and emotion availabilities.

- Successfully used JavaScript to open VRML mime type and display 3D VRML of players' faces and poker cards.

Limitation

- Since all information is saved in files on the server side. Application is a little slow.

Future work

- Support more facial emotions in 3D character which are in player.dtd.

- Remove operation, viewpoint for each player.

Question & Thanks

?