International Journal of Smart Home
Vol.7, No.6 (2013), pp.367-376
http://dx.doi.org/10.14257/ijsh.2013.7.6.36

An XML Based Communication System for a Ubiquitous Game Simulator

KeeHyun Park

Department of Computer Engineering
Keimyung University, South Korea
khp@kmu.ac.kr

Abstract

There are many game simulators users can enjoy indoors. An indoor golf game simulator is a typical example of such indoor game simulators. Usually, players experience some inconvenience when they get together to use the same game console. In this paper, a communication system to support a ubiquitous game is proposed. For ubiquitous games, players do not need to get together to use the same game console. Instead, they can enjoy the game together while being in different places so long as they are playing at the same time. Because players might use different kinds of game consoles in different places, there needs to be an interoperable communication system for the games. In this paper, therefore, an XML based communication scheme is proposed and XML documents are defined to represent all the information generated by the game simulator and to process the game progress.

Keywords: *Game simulator, Ubiquitous game, XML document, Interoperability*

1. Introduction

There are many game simulators users can enjoy indoors. An indoor golf game simulator [1-4] is a typical example of such indoor game simulators. Usually, players have to put up with some inconvenience if they want to get together to use the same game console. In this paper, a communication system to support a ubiquitous game simulator is proposed. For ubiquitous games, players do not need to get together to use the same game console. Instead, they can enjoy the game together while in different so long as they are playing the game at the same time. Because players might use different kinds of game consoles in different places, there needs to be an interoperable communication system for the games. In this paper, therefore, an XML based communication scheme is proposed. In other words, XML documents [5-8] are defined to represent all the information generated by the game simulator and to process the game progress. XML is proposed by W3C as a standard document on the Web [8]. This study proposes how to generate and process XML documents as the game proceeds. The flows of the XML documents inside the game simulator are proposed also.

The rest of the paper is organized as follows: Section 2 describes an indoor golf game simulator as background information. Section 3 introduces the structure of the communication system for a ubiquitous golf game simulator. It also explains the definition of XML documents and the flow of the documents over the communication system of the game simulator to represent the game status. Finally, Section 4 draws conclusions and discusses some future research directions.

2. Indoor Golf Game Simulator

An indoor golf simulator is an example of a game simulator and consists of a screen, a sensor system, a projector and a PC console [1-4]. The simulator allows users to enjoy their golf games indoors. When a player logs into the golf game simulator, the golf game simulator executes various actions to create a simulated golf environment on the screen. The projector displays the images of real golf courses on the screen. When a player hits a golf ball on the ground, the sensor system analyzes the strokes a player makes in order to measure such things as the movement of the ball resulting from the speed at which it was hit, the direction, angle and distance, etc. Based on the information of the ball transmitted from the sensor system, the PC console generates the images of ball movements. And then the projector displays the images of the ball movements on the screen.

Figure 1 shows the communication system for the conventional indoor golf game simulator. To play the same game, players have to be in the same place to use the same PC game console.

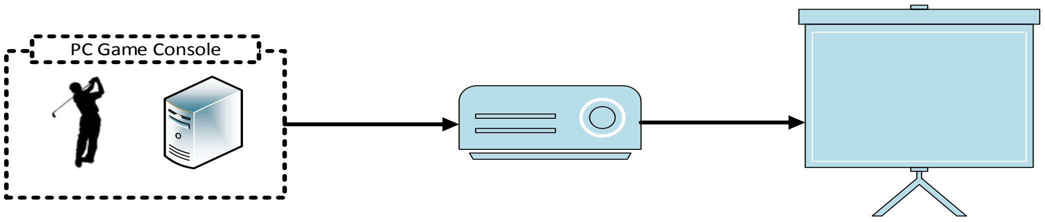

Figure 1. Communication System for the Conventional Indoor Golf Game Simulators

3. Communication System for a Ubiquitous Golf Game Simulator

3.1. System Structure

The Figure 2 shows the structure of the communication system for a ubiquitous golf game simulator proposed in this study. The communication system consists of PC console clients, PC console P2P hosts, proxy servers, a user data server and a game contents server.

- PC console client: It is installed in a PC game console, where the conventional game simulator executes actions. It communicates with the related PC console P2P host in a P2P fashion in order to send/receive information on the game status.
- PC console P2P host: It creates and manages a game lobby which is a logically arranged group consisting of a PC console P2P host and PC console clients. It communicates with the related proxy server as well as with the PC console clients which belong to the game lobby the PC console P2P host takes care of. P2P communication is used between PC console clients and a PC console P2P host. A PC console P2P host sends the proxy server game status of its related lobby, using Web communication schemes.
- Proxy server: It relays authentication information regarding players between PC console P2P hosts and the user data server. After successful initial authentication by the user data server, a proxy server caches the authentication information for later use in order to lessen traffic load centered on the user data server.
- User data server: It keeps user information and manages player authentication. It also stores users' game results and histories.

- Game contents server: It communicates with PC console P2P hosts in order to keep track of the progresses of all of games currently activated. It also multicasts the current game status to PC console clients which take participate in the same game, via the related PC console P2P host. After a game is finished, all information related to the game is transmitted to the user data server.

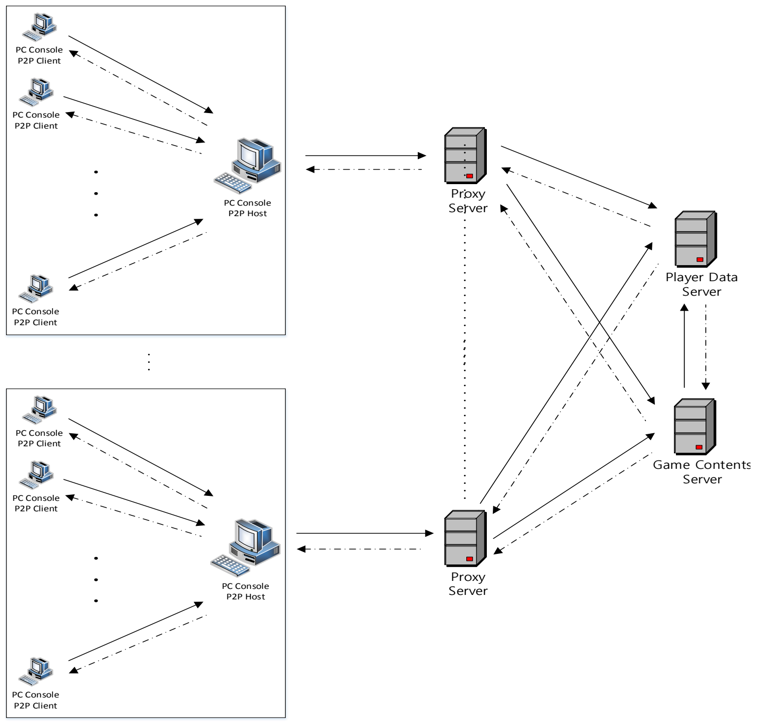

Figure 2. Structure of the Communication System

3.2. XML Elements

Table 1 shows XML elements used in this study. The subset of Table 1 was discussed in earlier study [9].

Table 1. Elements of XML

Elements	Comments	Elements	Comments
ProtoType	Type of a XML document	nHoleListCurrentNo	Current hole number
Auth	Authentication	nScore	Game scores
AuthHash	Hash value for authentication	fTShotDistance	Tee shot distance
PlayerInfo	Player's information	nTShotFairWay	Number of shots on the fairway
cBallPos	Position of a ball	nGreenShot	Number of shots on the green
nSelectFieldNum	Current field number	nPuttingCount	Number of putts in this hole
nSelectHoleNum	Current hole number	nBunkerShot	Number of bunker shots
nPlayerNum	Number of players	fTshotAverage	Average distance of tee shots
nPlayerSequence	Player number	nPuttingTotalCount	Number of putts in this game
nRemHoleCount	Staring hole number	dwHitCount	Number of hits in this hole
strPlayerName	Player name	dwTotalHitCount	Number of hits in this game
nDiff	Remaining distance	strClubName	Name of a club being used
nHoleListCurrentNo	Current hole number	strClubNo	Number of a club being used

3.3. XML Documents

Creation and the flow of the XML documents for lobby creation, lobby join, lobby game start/end processes will be discussed. Creation and the flow of the XML documents for Login, authentication and hole start/end processes were discussed in an earlier study [9].

Figure 3 illustrates the flow of a lobby creation process. When a player logs into the game simulator installed on a PC game console for a new game, a PC console client on the PC game console generates a 'LobbyCreateRequest' document in XML to request lobby creation to the game contents server via the related PC console P2P host and proxy server. On receiving the 'LobbyCreateRequest' document, the game contents server creates a lobby and sends a 'LobbyCreateResponse' document to the PC console P2P host via and the related proxy server. Then, the PC console P2P host creates a logical group for the lobby and relays the 'LobbyCreateResponse' document to the PC console client.

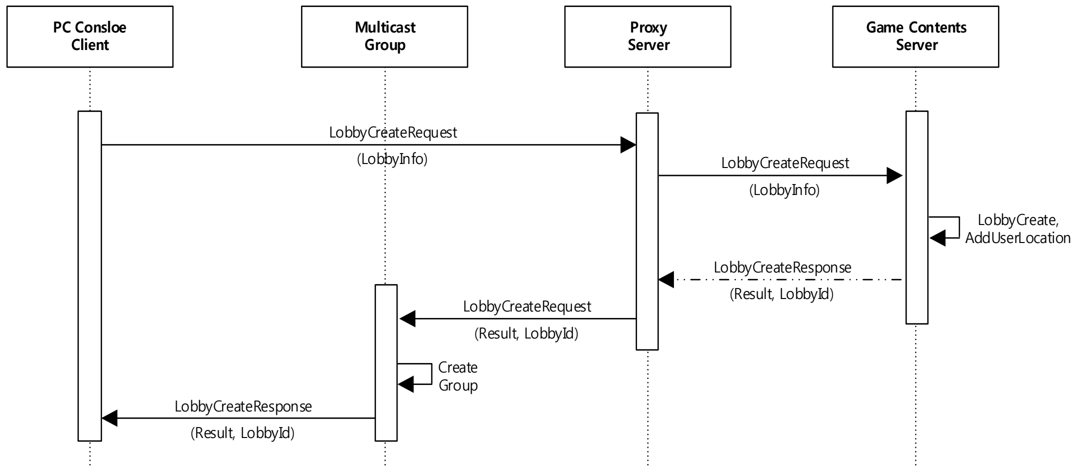

Figure 3. The Flow of XML documents for a Lobby Creation Process

Figure 4 shows an example of the 'LobbyCreateRequest' XML request document generated by the PC console client when a player logs into the game simulator. Figure 5 shows an example of the 'LobbyCreateResponse' XML response document generated by the game contents server after the server has successfully executed the lobby creation process.

```
<OneClickGolf Length = '343'>
  <Auth>
    <ClientVer>0.1</ClientVer>
    <AuthHash>id_user</AuthHash>
    <Prototype>LobbyCreateReq</Prototype>
  </Auth>
< Body>
  <LobbyInfo>
    <LoobyTitle>lobby_title</LoobyTitle>
    <LobbyPasswd>lobby_passwd</LobbyPasswd>
    <LobbyMaxUser>4</LobbyMaxUser>
    <LobbyHoleNumber>18</LobbyMaxHoleNumber>
    <LobbyFieldType>0</LobbyFieldType>
  </LobbyInfo>
  </Body>
</OneclickGolf>
```

Figure 4. XML Request Document for a Lobby Creation Process

```
<OneClickGolf Length='1375'>
   <Auth>
      <ClientVer>0.1</ClientVer>
       <ProtoType>LobbyCreateRes</ProtoType>
   </Auth>
<Body>
<Status>1</Status>
<nSelectFieldNum>1</nSelectFieldNum>
<nSelectHoleNum>18</nSelectHoleNum>
<nPlayerNumber>1</nPlayerNumber>
<nPlayerSequence>1</nPlayerSequence>
<nHoleListStartNo>1</nHoleListStartNo>
<PlayerInfo>
<strPlayerName>id_user</strPlayerName>
<strClubNo>1</strClubNo>
<strClubName>wood</strClubName>
<nScore>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</nScore>
<nPuttingTotalCount>0</nPuttingTotalCount>
<nPuttingCount>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</nPuttingCount>
<nPlayerCount>1</nPlayerCount>
<nMuliganCnt_remain>3</nMuliganCnt_remain>
<nGreenShot>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</nGreenShot>
<nDiff>0</nDiff>
<fTShotDistance>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</fTShotDistance>
<fTShotAverage>0</fTShotAverage>
<dwTotalHitCount>0</dwTotalHitCount>
<dwHitCount>0</dwHitCount>
<nTShotFairWay>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</nTShotFairWay>
<nBunkerShot>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</nBunkerShot>
</PlayerInfo>
</Body>
</OneClickGolf>
```

Figure 5. XML Response Document for a Lobby Creation Process

Figure 6 illustrates the flow of a lobby join process. When a player logs into the game simulator for a game where other players have logged in already, a PC console client representing the player generates a 'LobbyJoinRequest' document in XML along with the lobby ID the player wants to join with. The 'LobbyJoinRequest' document is sent to the game contents server via the related PC console P2P host and proxy server. On receiving the 'LobbyJoineRequest' document, the game contents server adds the player's information (player ID, location, *etc.*,) to the lobby and sends a 'LobbyJoinResponse' message to the PC console client. Then, the PC console client sends the related PC console P2P host the 'LobbyJoinRequest' document again to notify existing lobby game members about the addition of a new game member to the lobby.

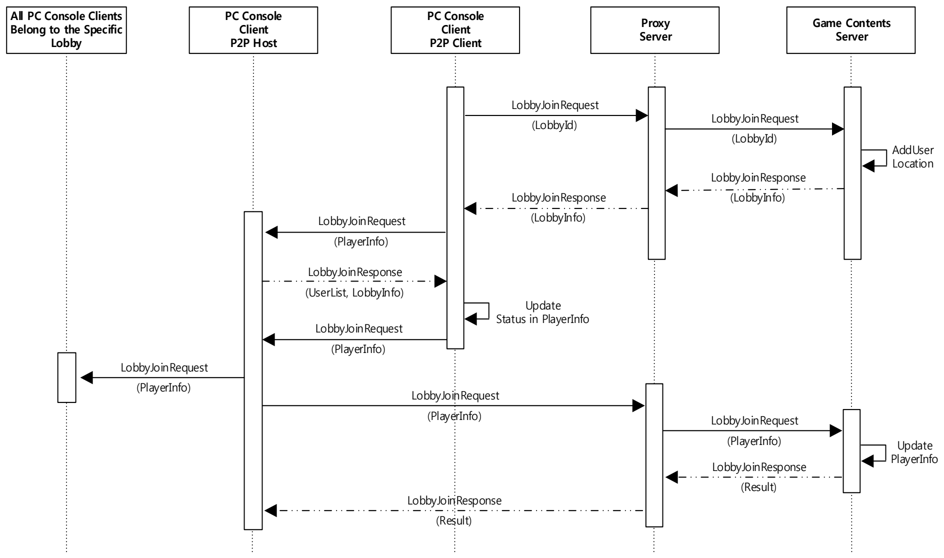
All PC Console Clients Belong to the Specific Lobby
PC Console Client P2P Host
PC Console Client P2P Client
Proxy Server
Game Contents Server
LobbyJoinRequest (LobbyId)
LobbyJoinRequest (LobbyId)
AddUser Location
LobbyJoinResponse (LobbyInfo)
LobbyJoinResponse (LobbyInfo)
LobbyJoinRequest (PlayerInfo)
LobbyJoinResponse (UserList, LobbyInfo)
Update Status in PlayerInfo
LobbyJoinRequest (PlayerInfo)
LobbyJoinRequest (PlayerInfo)
LobbyJoinRequest (PlayerInfo)
LobbyJoinRequest (PlayerInfo)
Update PlayerInfo
LobbyJoinResponse (Result)
LobbyJoinResponse (Result)

Figure 6. The Flow of XML Documents for a Lobby Join Process

Figure 7 shows an example of the 'LobbyJoinRequest' XML request document generated by the PC console client when a player logs into the game simulator. Figure 8 shows an example of the 'LobbyJoinResponse' XML response document generated by the game contents server after the server has successfully created the lobby creation process.

```
<OneClickGolf Length = '306'>
   <Auth>
      <ClientVer>0.1</ClientVer>
      <AuthHash>id_user</AuthHash>
      <Prototype>LobbyJoinReq</Prototype>
   </Auth>
   <Body>
      <LoobyTitle>lobby_title</LoobyTitle>
      <LobbyPasswd>lobby_passwd</LobbyPasswd>
   </Body>
</OneclickGolf>
```

Figure 7. XML Request Document for a Lobby Join Process

```
<OneClickGolf Length='2485'>
  <Auth>
    <ClientVer>0.1</ClientVer>
    <ProtoType>LobbyJoinRes</ProtoType>
  </Auth>
  <Body>
    <Status>1</Status>
    <LobbyInfo>
      <GameInfo>
          <nSelectFieldNum>1</nSelectFieldNum>
          <nSelectHoleNum>18</nSelectHoleNum>
          <nPlayerNumber>1</nPlayerNumber>
          <nHoleListStartNo>1</nHoleListStartNo>
      </GameInfo>
               <PlayerList PlayerCount="2">
               <PlayerInfo PlayerSeq="1">
                 <strPlayerName>id_user</strPlayerName>
                 <strClubNo>1</strClubNo>
                 <strClubName>wood</strClubName>
                 <nScore>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</nScore>
                 <nPuttingTotalCount>0</nPuttingTotalCount>
```

```
                <nPuttingCount>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</nPuttingCount>
                <nPlayerCount>1</nPlayerCount>
                <nMuliganCnt_remain>3</nMuliganCnt_remain>
                <nGreenShot>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</nGreenShot>
                <nDiff>0</nDiff>
                <fTShotDistance>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</fTShotDistance>
                <fTShotAverage>0</fTShotAverage>
                <dwTotalHitCount>0</dwTotalHitCount>
                <dwHitCount>0</dwHitCount>
                <nTShotFairWay>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</nTShotFairWay>
                <nBunkerShot>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</nBunkerShot>
              </PlayerInfo>
              <PlayerInfo PlayerSeq="2">
                <strPlayerName>id_user2</strPlayerName>
                <strClubNo>1</strClubNo>
                <strClubName>wood</strClubName>
                <nScore>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</nScore>
                <nPuttingTotalCount>0</nPuttingTotalCount>
                <nPuttingCount>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</nPuttingCount>
                <nPlayerCount>1</nPlayerCount>
                <nMuliganCnt_remain>3</nMuliganCnt_remain>
                <nGreenShot>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</nGreenShot>
                <nDiff>0</nDiff>
                <fTShotDistance>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</fTShotDistance>
                <fTShotAverage>0</fTShotAverage>
                <dwTotalHitCount>0</dwTotalHitCount>
                <dwHitCount>0</dwHitCount>
                <nTShotFairWay>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</nTShotFairWay>
                <nBunkerShot>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</nBunkerShot>
              </PlayerInfo>
          </PlayerList>
        </LobbyInfo>
    </Body>
</OneClickGolf>
```

Figure 8. XML Response Document for a Lobby Join Process

Figure 9 illustrates the flow of a playing manoeuvre process. Every time a player makes a playing manoeuvre (like swing a golf club), a PC console client representing the player generates a 'SwingDataRequest' document in XML to send to the game contents server via the related PC console P2P host and proxy server. On receiving the 'SwingDataRequest' document, the game contents server updates the game status and generates a 'SwingDataResponse' document. Then, the server multicasts the document to other PC console clients which belong to the same lobby in order to notify that the player has made a playing manoeuvre.

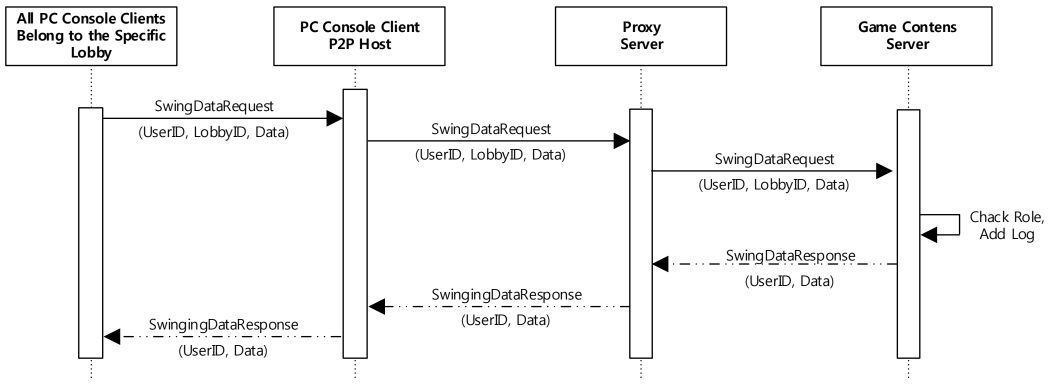

Figure 9. The Flow of XML Documents for a Playing Manoeuvre Process

Figure 10 and Figure 11 show examples of the ‘SwingDataRequest’ XML request document and ‘SwingDataResponse’ XML response document, respectively.

```
<OneClickGolf Length = '306'>
   <Auth>
      <ClientVer>0.1</ClientVer>
      <AuthHash>id_user</AuthHash>
      <Prototype>SwingDataReq</Prototype>
   </Auth>
   <Body>
      <LobbyInfo>
         <LobbyTitle>lobby_title</LoobyTitle>
         <LobbyPasswd>lobby_passwd</LobbyPasswd>
      </LobbyInfo>
   </Body>
</OneclickGolf>
```

Figure 10. XML Request Document for a Playing Manoeuvre Process

```
<OneClickGolf Length = '1307'>
   <Auth>
      <ClientVer>0.1</ClientVer>
      <AuthHash>id_user</AuthHash>
         <Prototype>SwingDataRes</Prototype>
   </Auth>
   <Body>
      <status>"1"</status>
      <SwingData>
         <PlayerSeq>"1"</PlayerSeq>
         <BallPos>
            <x>100</x>
            <y>112</y>
            <z>0</z>
         </BallPos>
   </SwingData>
   </Body>
</OneclickGolf>
```

Figure 11. XML Response Document for a Lobby Join Process

4. Conclusion

In this paper, a communication system to support a ubiquitous game simulator is proposed. For ubiquitous games, the concept of a lobby is proposed where a group of players enjoy the same game, possibly in different locations. Because players might use different kinds of game consoles in different places, there needs to be an interoperable communication system for the games. In this paper, therefore, an XML based communication scheme is proposed. In this paper, the structure of the communication system for ubiquitous game simulators was proposed. In addition, this study proposes how to generate and process XML documents as the game proceeds. Currently, the communication system proposed in this paper is being developed.

5. Acknowledgements

This work (Grants No. 0002985) was supported by Business for Academic-industrial Cooperative establishments funded Korea Small and Medium Business Administration in 2012.

References

[1] S.-I. Lee, “The effect of motivation of virtual reality sports and service factors on leisure satisfaction”, Sejong University, Master's Thesis, **(2011)**.
[2] S.-H. Ahn, “Development of a simulator for Interactive 3D Golf Game”, Hallym University, Master's Thesis, **(2007)**.
[3] B.-M. Moon, “The Study of Screen Golf’s fun factors on exercise immersion experience, participating satisfaction and exercise continuation behavior”, Silla University, Master's Thesis, **(2010)**.
[4] K. H. Park and S. Lim, “Indoor Golf Simulator for Continuous Golf Games”, International Journal of Smart Home, vol. 7, no. 3, **(2012)**, pp. 75-84.
[5] Y. Song, K. Choo and S. Lee, “Design of Index Schema based on Bit-Streams for XML Documents”, International Journal of Software Engineering and Its Applications, vol. 6, no. 4, **(2012)**, pp. 131-136.
[6] S. Youn Hong and D. Hoon Han, “XML Principles and Applications”, Hanbit Media Inc., **(2011)**.
[7] H. Deitel, XML, Prentice Hall Pub. Co. **(2001)**.
[8] W3C, http://www.x3.or.kr/, accessed, **(2013)** August.
[9] K. H. Park, J. Park, S. Lim and S Lim, “Application of an XML Processing Scheme to an Indoor Golf Simulator”, submitted for publication **(2013)**.

Author

KeeHyun Park, received his B.Sc. and M.Sc. degrees in Computer Science from Kyungbook National University, Korea, and from KAIST, Korea, in 1979 and 1981, respectively, and his Ph.D. degree in Computer Science from Vanderbilt University, USA, 1990. He has been a professor of Computer Science and Engineering Department at Keimyung University, Korea since March 1981. His research interests include Mobile/Network Communication System, Embedded System and Parallel Processing System.