VIPL: Visual IoT/Robotics Programming Language Environment for Computer Science Education

Yinong Chen and Gennaro De Luca

School of Computing, Informatics, and Decision Systems Engineering
Arizona State University
Tempe, AZ 85287-8809, U.S.A.
Overview

• Introduction to IoT and RaaS
• IoT Standards and Protocols
• Definition of VIPLE
• Educational examples in VIPLE
• VIPLE and IoT devices
History of IoT

• Internet of Things
• RFID tags and Electronic Product Code (Auto-ID Lab)
• IoIT = IoT + computing power
• 15 billion IoT devices
• 5 billion IoIT devices
• 1 billion intelligent systems
Internet-based Computing

• Cloud computing
  – Desktop-based => Internet-based
  – Physical products => “Things” or services

• Cyber Physical Systems (CPS)
  – Extended/decentralized version of embedded systems

• Autonomous Decentralized Systems (ADS)
  – Loosely coupled/content-oriented systems
  – Industrial production lines, railway signaling, robotics
Robot as a Service

• Composed of services, a service directory, and service clients
• IoIT (and IoT) device
• IoT-enabled hardware:
  – Intel Galileo/Edison
• CPS
• ADS
IoT Standards

- Supports many protocols/standards
- Connection to physical world
  - Also supports various standards
  - ADS uses a content-oriented protocol
- DPWS
- RaaS

<table>
<thead>
<tr>
<th>Technologies</th>
<th>Applications</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cloud Computing and Big Data Processing</td>
<td>IoT</td>
</tr>
<tr>
<td>Service and Web-Based Computing</td>
<td></td>
</tr>
<tr>
<td>Web Data Representations: HTML, JSON, OWL, RDF, XML, etc.</td>
<td></td>
</tr>
<tr>
<td>Internet Protocols, HTTP, TCP, IP</td>
<td></td>
</tr>
<tr>
<td>IoT</td>
<td></td>
</tr>
<tr>
<td>Device Connection Protocols: ADS, DPWS, RaaS, Industry Control Systems, Industry Internet, etc.</td>
<td></td>
</tr>
</tbody>
</table>
IoT/RaaS Programming

- Visual programming environments for education
  - MIT App Inventor
  - Alice

- Robotics programming
  - MSRDS VPL
    - Used at ASU, FSE 100
    - Discontinued 2014
    - Lacks support for new robots
Definition of VIPLE

• Based on Microsoft VPL
• VIPLE supports similar applications
• Additional VIPLE services:
  – General services (e.g. User I/O, Timers)
  – Vendor robot services
  – Generic robot services
• Educational tool
Fundamental Programming in VIPLE
VIQUEL Multithreading

- Teaching parallel computing
  - Handles low-level synchronization, thread safety, data passing.
  - Allows building of multithreaded applications.
  - Race conditions may still occur.
- Hardware is fully utilized
  - Students can experiment with speedups/optimizations.
Parallel Computing in VIPLE

[Diagram showing parallel computing logic]
Event-driven Programming in VIPLE
VIPLE’s IoT Interface Definition

- **Internet Protocol**
  - Multiple supported standards

- **Data Format**
  - Standardized JSON

An object pair, with the second element an array of objects.
Connecting VIPLE to Devices
Drive-by-Wire
Maze Navigation with an IoT Device

Diagram:

- Start
- DistanceMeasured < 400
  - Turn Right
- DistanceMeasured >= rightDistance
  - Turning Right
  - Turning Left
- DistanceMeasured < rightDistance
  - Resume 180
- rightFinished
  - Turn Right
- leftFinished

Variables:
- Status
- String
- RightDistance
- Double

Running Program:

0.702118
0.8267809
1.018474
1.268449
1.820216
2.8881
Maze Navigation Implementation

Intel Edison-based robot with built-in Wi-Fi and Bluetooth components. A distance sensor is installed in front.
Multithreading and Maze Navigation

• **Multithreading is a key part of maze navigation.**
  – Many algorithms require concurrent sensor data readings.
  – Some actions do not need to be waited on.

• **Many applications are facilitated by multithreading.**
  – New users have more freedom and power in their application development.
Conclusion

• New Visual Programming Language: VIPLE
  – Extends Microsoft VPL
  – Supports Lego EV3
  – Supports all IoT devices based on an open architecture

• VIPLE has been pilot tested at ASU and several other universities globally.