CONSTRUCT/VizM: A Framework for Rendering Tangible constructions

Pamela L. Jennings
National Science Foundation, USA
✉ pljenn@gmail.com

David Antonio Castro Martínez
University de Los Andes, Colombia
✉ da-castr@uniandes.edu.co

ABSTRACT

The CONSTRUCTS Toolkit is a wireless sensor network system (WSN) for mixed-reality applications. Wireless sensor networks have become an accessible development platform with advances in the convergence of micro electro-mechanical systems technology, wireless communication protocols, integrated circuit technologies, and pervasive and embedded systems. As applied applications for wireless sensor networks in the manufacturing and health industries continue to grow there remains an opportunity to integrate these technologies into gaming and learning applications. This paper will present an overview of the CONSTRUCT/VisM application designed for transforming construction state messages from the WSN CONSTRUCTS Toolkit into a real-time 3D virtual environment.

KEYWORDS: mixed reality, tangibles, wireless sensor networks, graph systems.

The CONSTRUCTS Toolkit prototype consists of fourteen tangible blocks - two each of seven unique 2D and 3D pentomino shapes (Fig. 1). A pentomino is made from five cubes of the equal size. The pentomino shape was selected for the design because it supports three-dimensional cantilevered constructions, typically not a design quality of electronic construction kits such as those reviewed below (Fig. 2). The interchangeable magnetic sensor connector was designed to support gender-neutral block connections allowing the blocks to be connected in any configuration that does not conflict with physical constraints.

Figure 1. CONTSTRUCTS demonstration at ACM CHI'08

Each block is a node on an 802.15.4 mesh wireless network. When blocks are connected over-the-air messages about the block state, such as the connected faces and other block functions are transmitted to the CONSTRUCT/VizM application on the host computer. CONSTRUCT/VizM translates the wireless network messages to create an accurate 3D render of the tangible construction using the original block CAD models. Each 3D rendered block is an independent agent that can contain its own visual features including size, color, and embedded animated sprites. The user experience objectives include:

- Design a mixed-reality social networking game for public spaces.
- Design an application that encourages learning, discovery and exploration.

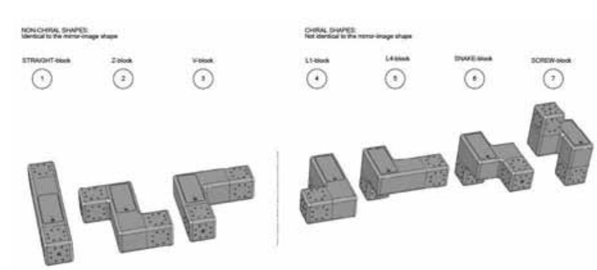

Figure 2. The CONSTRUCTS for the tangible block shape system

- Design an engaging application that attracts a broad demographic of users.

The technical objectives include:

- Develop a WSN toolkit that tracks nodes without device tethering or computer vision.
- Develop a tangible user interface that is updated in real-time in a 3D render.
- Develop a system architecture that is the basis for future WSN development platforms.

The CONSTRUCTS system can manage application domains based on other block shape systems as long as they can be represented geometrically in a grid. CONSTRUCTS toolkit applications align with human centered computing domains of computer supported collaborative play, learning and work (CSCP, CSCL, CSCW). This could include applications for exploring complex systems in science, art and humanities, as well as applications for disciplines that use prototypes for iterative design, such as architecture, design, and engineering.

Related Work

The CONSTRUCTS is part of a lineage of projects based on the early work of architect Jonathon Frazer's *Universal Constructor* generative system. (Frazer, 1995) The projects influenced from Frazer's work tend to fall into one or more of the following categories:

- Information navigation interfaces.
- Audio/visual feedback interfaces.
- Physical model replication interfaces.
- Post-console game interfaces.

Information navigation interfaces include experimental prototypes such as Navigation Blocks, a tethered singleton block used to navigate information (Camarata, 2001), and Media Blocks, tangibles used as phicons for navigating information (Ullmer, Ishii & Glas, 1998). The Triangles project is a set of identical flat plastic triangles that can be used to arrange media elements to reveal a screen-based narrative (Gorbet & Orth, 1997). The Strata ICC project uses transparent physical tile layers to illustrate building energy consumption. (Ullmer et al., 2001).

Audio/video feedback interfaces respond with light, sound, or color mapped onto a visual representation of the tangible units. BlockJam is a set of twenty-five half-cubed blocks that can be arranged to compose musical phrases (Newton-Dunn et al., 2000). Peano Cubes are a set of cuboid blocks whose physical construction is replicated on screen. The screen interface provides tools that can be used to change the LED color patterns inside the tangible cubes (Heaton, 2000). Instant City is a collaborative generative sound game. The rectangular blocks are made of semi-clear plexiglas. The sounds are triggered in the host computer system based upon the quantity of light passing through to photo sensors (Hauert & Reichmuth, 2003).

Physical model replication interfaces employ either complex algorithms that search for predetermined shape patterns. Or employ sensor schemes to deduct the tangible configuration based on the sensor data and understood tangible design constraints. ActiveCubes can recognize a few basic symbols such as a shape that references a toy airplane that is overlay as an image in the screen interface (Watanabe et al., 2004). Recent projects in this domain explore the development of children's toys. (Weller, Do & Gross, 2009). Recognizing a shape in a tangible construction can be an intractable problem. The MERL project uses a huge amount of computing power to analyze a complex shape like a block castle. It required fifty-three minutes to crunch data for a set of five-hundred and sixty blocks (Anderson et al., 2000).

Post-console computer game interfaces eliminate dependancy on a screen interface and embed all system feedback into the tangible device. Siftables™, a derivative of Block Jam, are individual half-cubed devices that act in concert as a single interface (Merrill, Kalanithi & Paes, 2007). Cublettes™, a derivative of the ActiveCube project, is equipped with both input sensors and output actuators to form a self-describing system for a children's robotics construction kit (Schweikardt & Gross, 2006). The Lumino project explores methods for vertical stacking of tangible objects on a horizontal touch-surface. Lumino blocks are passive devices packed with glass fibers and two bit opacity pattern filters that allow light to pass through the object. Lumino blocks are limited to two-level stacking (Baudisch, Becker & Rudeck, 2010).

CONSTRUCT/VizM Framework

CONSTRUCT/VizM is a Java3D application responsible for transforming the wireless device messages into data structures that can be used to render a real-time replication of the tangible construction as blocks are added and removed. The CONSTRUCT/VizM framework has five layers (user interface, application engine, network management, device management and communication, device firmware) (Fig. 3). The device firmware layer represents the device 802.15.4 firmware. The protocol used is a proprietary full mesh network. The device firmware relays the device state changes as they occur using a lightweight protocol. The *device management and communication* layer represents the serial interface between network devices and the host computer. The *network management* layer translates the raw byte data for *application engine* and *user interface* layer interpretation. The Network Management layer is developed on the agent kernel for Java™ Open Source Web Server Framework. Agent Kernel™ is an external java console that provides access to functional components of the application. For example, the lighting options in the 3D environment can be adjusted by turning the lighting agents on/off in the console.

The *application engine* layer converts the raw device network data into data structures used to build and traverse the application virtual render. It includes an end-user *application API* that decouples and hides the “under-the-hood” mechanics of the translation logic from application development. The application engine layer consists of four components - *network management, user interface*, and *translation logic* described elsewhere in this paper and *data management*, a series of high level block shape descriptions files. *Visualization clients* are end-user applications.

The Cubical Grid

The CONSTRUCTS tangible blocks can support an intractable number of possible connections. The *cubical grid* is a method used to analyze the configuration data and transform it into a manageable data structure. By doing this, it provides a robust scaffold for debugging the block transforms and identifying paths between block units.

The *cubical grid* partitions the space into a three-dimensional matrix of cube atoms. The cubical grid rotation rules is a generalizable algorithm that can be applied to all block connection patterns. Therefore, there is no block pattern hard-coding in the application. There are several rotation steps involved in the process of positioning blocks. The initial rotations are determined by the tri-axial accelerometer data from the tangible block and a data file. When a connection is detected the new block is positioned accordingly. If it is the first block read into a new construction, it is positioned on its center. Other attributes that are considered in the cubical grid system are face direction, block type, connected face and connection rotation. A color overlay of the cubical grid can be turned on/off in the user interface GUI.

Fig. 4 presents the process for translating physical events into the cubical grid. Image (a) represents a tangible block. Image; (b) shows the pentomino division of the block. Image (c) shows the end cube units for connecting. Image (d) shows the cubical grid debug overlay turned on show the block face mapping (Fig. 4).

The Java Universal Network/Graph (JUNG) framework non-directed sparse multi-graph data structure describes the connections between the tangible blocks. A non-directed graph is used to correlate to the gender-neutral connectors featured on the tangible devices and outputs block construction pairs to be rendered in CONSTRUCT/VizM. JUNG also provides a set of libraries for visualizing graph representations. We have incorporated those libraries in the GUI debug window and find it particularly useful to debug complex connection types. For example, blocks can be connected together on both ends - a “serpentine connection”. A singular block can be connected to a larger construction on both ends creating a structural feedback loop. And a block can be connected to several blocks simultaneously on both ends. It is our goal to exploit the advanced features of the JUNG framework to identify and render multiple simultaneous constructions, as well as to integrate the graph search algorithms for pattern recognition in future applications.

The *user interface layer* back-end was built as an ‘observer’ and ‘controller’ for managing over-the-air messages. This debug window replaces most debugging functions of the Agent Kernel™ described above. There are four tabbed areas in the debug window. The *application tab* controls features such as the cubical grid overlay and wireless network restart. Other low level network information is also presented such as the network channel. The *device tab* contains controls for communicating directly with the network devices for information about the accelerometer, signal and battery strength, and control of the device actuators. The device tab also controls visual and special effects in the three-dimensional render. The *connections tab* is dedicated to showing the state of the JUNG graph and complexity of the construction connections. Any device in play can be selected to view its state in the overall construction. The *demo module* was designed for debugging and demonstration. Real-time construction sessions can be recorded, saved and replayed.

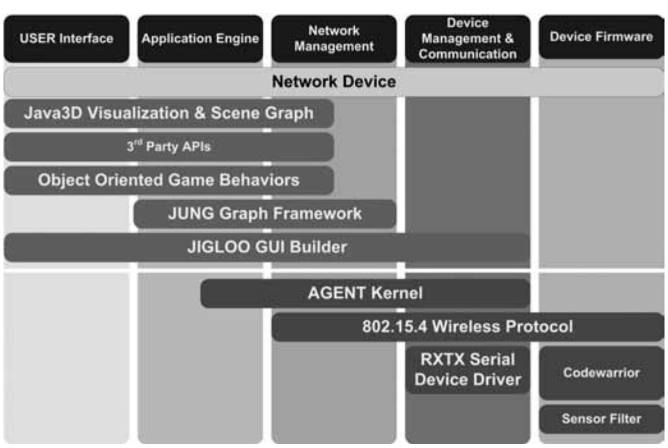

Figure 3. CONSTRUCTS Toolkit system architecture

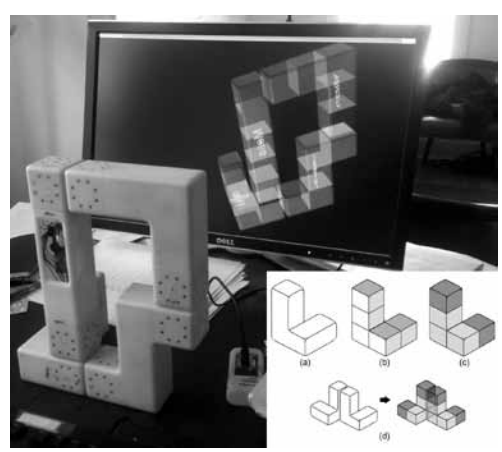

Figure 4. CONSTRUCTS construction of a simple loop and diagram showing the cubical grid shape encoding process

Conclusion

Further development of this architecture is planned to support alternative application development environments. JUNG allows us to build a very robust representation of the system. We plan to build an application service into the application API that will allow access to the JUNG graph system and its powerful graph search algorithms. The cubical grid will also be enhanced for mapping JUNG graph search results onto the three-dimensional render to be able to highlight specific paths through the construction as well as move animated sprites between individual block units.

Acknowledgements

We want to acknowledge those who have worked on the CONSTRUCTS project including Ted Hung, Drew Gislason, Tim Gillman, Paul Cunningham, and Daedalus-Excel, Inc. CONSTRUCTS was funded by the U.S. National Science Foundation Creative IT; and the Alberta Informatics Circle of Research Excellence (iCORE) in Canada.

Any opinion, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

Anderson, D. et al. (2000). *Tangible interactions and graphical interpretation: a new approach to 3D modeling*. Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH), pp- 393-402.

Baudisch, P., Becker T. & Rudeck, F. (2010). *Lumino: Tangible Blocks for Tabletop Computers Based on Glass Fiber Bundles*. Proceedings of the 28th International Conference on Human Factors in Computing CHI '10, pp. 1165-1174.

Camarata, K. (2001). *Navigation blocks: tangible navigation of digital information*. Unpublished Masters Thesis, University of Washington, USA.

Frazer, J. (1995). *An Evolutionary Architecture*. London: Architectural Association Publications.

Gorbet, M. & Orth, M. (1997). *Triangles: design of a physical/ digital construction kit*. Proceedings of ACM SIGCHI Computer Human Interaction Conference CHI 01, pp. 125-128.

Heaton, K. (2000). *Peano cube*. Unpublished Thesis Personal Information Architecture Group and Toys of Tomorrow, MIT Media Lab, USA.

Hauert, S. & Reichmuth, D. (2003). *Instant city*. Basil, Switzerland: Rosen & Spaderman Enterprise.

Merrill, D., Kalanithi, J. & Paes, P. (2007). *Siftables: towards sensor network user interfaces*. Proceedings of the 1st International Conference on Tangible and Embedded Interaction, pp 75-78.

Newton-Dunn, H., Nakano, H. & Gibson, J. (2000). *BlockJam*. Sony CSL Interaction Lab and Sony Design Center, Japan. Available from: http://www.csl.sony.co.jp/IL/projects/blockjam/contents.html.

Schweikardt, E. & Gross, M. (2006). *roBlocks: a robotic construction kit for mathematics and science education*. Proceedings of the 8th International Conference on Multimodal Interfaces, pp. 72 - 75, Banff, Alberta, Canada.

Ullmer, B., Ishii, H. & Glas, D. (1998). *mediaBlocks: Physical Containers, Transports, and Controls for Online Media*. Proceedings of the Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH), pp. 379-386.

Ullmer B. et al. (2001). *Strata/ICC: Physical Models as Computational Interfaces*. Proceedings of ACM SIGCHI Computer Human Interaction Conference CHI 01, pp. 373-374.

Watanabe, R. et al. (2004). *The soul of activeCube - implementing a flexible, multimodal, three-dimensional spatial tangible interface*. Proceedings of ACM SIGCHI International Conference on Advances in Computer Entertainment Technology ACE, pp. 173-180.

Weller, M., Do, E. Y-L. & Gross, M. (2009). *State Machines are Child's Play: Observing children ages 9 to 11 playing Escape Machine*. Proceedings of ACM Interaction Design for Children (IDC).