High Performance Linked Data Processing for Virtual Reality Environments

Felix Leif Keppmann[1], Tobias Käfer[1], Steffen Stadtmüller[1], René Schubotz[2], and Andreas Harth[1]

[1] Karlsruhe Institute of Technology (KIT)
{felix.leif.keppmann, tobias.kaefer, steffen.stadtmueller, andreas.harth}@kit.edu
[2] Airbus Group
rene.schubotz@eads.net

1 Introduction

The success of Linked Data (LD) [1] has enabled an environment in which application data can easily be enriched by the abundance of available information on the Web. Many recent approaches of the Linked Data community go beyond the mere exposure of static data and propose the combination of Linked Data and Representational State Transfer (REST) [3, 5, 7] to enable dynamic systems. However, in highly dynamic environments, where near real-time data integration and processing with high update frequencies are required, the perceived overhead of Linked Data query processing and stateless communication pattern often prevents the adoption of resource state exchange-oriented systems.

Nevertheless, in our demonstration, we show a Virtual Reality (VR) information system that leverages the REST principles and the integration capabilities of LD. We specifically chose a VR setting, because it requires very low latency [2] in order to enable a natural interaction of the user with the system. Our system consists of loosely coupled components [4] as implicated by REST, and provides an interactive experience by seamlessly integrating existing LD sources from the Web as well as high dynamic body tracking data in a VR environment.

We show how sensor data exposed as LD, can be processed with high update frequencies and be rendered in a VR environment. Constantly evaluated queries are employed to realise both gesture recognition and collision detection of objects in the VR. Derived actions like data retrieval from the Web and the subsequent integration of the retrieved data with the sensor data are performed on-the-fly. With our system we contribute by demonstrating:

- the applicability of Linked Data in a VR environment
- the feasibility of a REST-based distributed system with high frequencies
- the capability to execute high frequency on-the-fly declarative integration of Linked Data in a REST environment

In the following we present the experience provided by our demonstration from both a user's and a technological point of view (Section 2). Afterwards, we elaborate on the underlying technologies (Section 3) and conclude shortly at the end (Section 4).

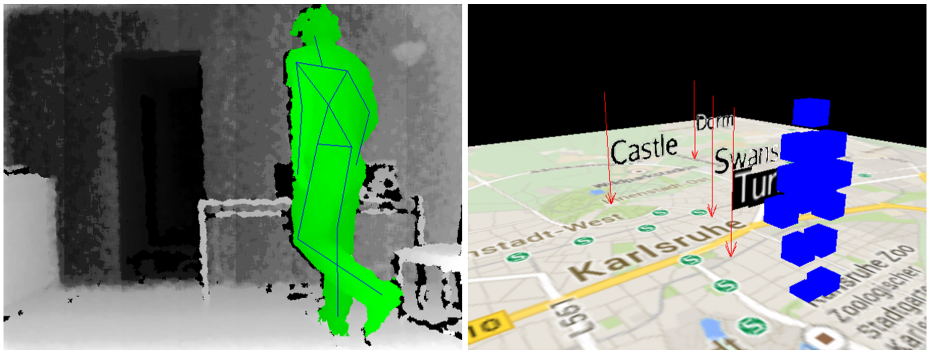

Fig. 1. Depth Video with Skeleton Tracking and Demo System Visualization

2 Demonstration

Our demonstration system let the user experience an interactive, integrated and responsive Virtual Reality. The system displays an avatar representing the user and information from various sources integrated on-the-fly in response to user commands. In our system, the user inputs commands to the system via Natural Interaction (NI), e.g. by interacting with the system via movements and gestures. The user in our set-up stands in front of a motion tracking sensor that tracks all parts of the body. As the user moves, for example, a hand to form a specific pose, the system executes a particular action. Figure 1 shows both, a visualization of the depth video input data (including skeleton tracking) and the VR which the user is remote controlling via NI.[3]

The user is represented in the VR by a human-like avatar (blue in Figure 1). Each recognized joint of the user's skeleton is mapped to the avatar, e.g. a knee of the user is mapped to knee of the avatar and moved accordingly. Instead of walking on a floor, the avatar is placed on a map (in Figure 1 the city of Karlsruhe in Germany) which will move in a specific direction if the avatar steps on the corresponding border of the map. Further, Points of Interest (PoIs) are visualized on the map, e.g. important buildings or upcoming concerts in the area (represented by red arrows in Figure 1). Via gestures of the user at a PoI, more detailed information will be requested, integrated on-the-fly and displayed in the VR. Thereby, a user is able to navigate through the map by walking and requesting additional information with gestures.

Beside the user experience of our system, we demonstrate from a technological point of view, loosely coupled components representing several internal and external data sources and sinks with a variety of exposed update frequencies, which are integrated on-the-fly in a declarative manner. All data sources and data sinks expose their data and functionality as LD via REST interfaces. The data sources include sources of a relatively static nature, e.g. map data or PoIs, and highly dynamic data sources, e.g. the sensor data of the body-tracking video sensor. Nevertheless, all sources and sinks are integrated via one component without the need for programmatic hard-coded integration of the interfaces.

[3] More screenshots and videos of the demonstration system are available at: `http://purl.org/NET/HighPerfLDVR`

In particular, a set of rules defined in a declarative rule language specifies the data flow between sources and sinks. A corresponding engine handles the query processing, evaluates the rule set in each cycle and executes low level tasks, e.g. retrieving data from the REST interface of the video sensor and a PoIs service, transformation and integration in the target schema and updating or creating resources in the REST interface of the visualization. Currently, our demonstration system processes data with a frequency of 20 Hz under high load up to 28 Hz under low load, which is close to the ideal 30 Hz update frequency at which the body tracking sensor updates.

3 Technology

We use Linked Data-Fu (LD-Fu) [6] as a central component for integration of data sources and sinks. LD-Fu is both a declarative rule language and an engine to handle the execution of programs consisting of these rules. The LD-Fu engine is able to interact with LD resources via REST interfaces. Rules defined in a program are evaluated and the interactions defined in the rule body are executed if the conditions are met. These interactions can include the retrieval of data from LD resources, the integration of data and the manipulation of LD resources. We developed the engine as multi-threaded forward-chaining data processor, thereby supporting fast parallel data processing with reasoning capabilities.

In our demonstration system LD-Fu handles 1) the data flow between body tracking of the depth sensor and visualization on the screen or wall, 2) collision detection in the interaction of the user with virtual objects, 3) execution of actions as result of the user interaction and 4) the integration of additional data sources on the web, e.g. information about PoIs or map data. All these interactions are defined as rule sets in LD-Fu programs.

With Natural Interaction via REST (NIREST) we expose body tracking data as LD via a REST interface. It utilizes data extracted from depth video sensors, e.g. Microsoft Kinect[4] devices. The position information of recognized people in the depth video, their skeleton data and device metadata are exposed as LD resources on the REST interface and are updated at a frequency of 30 Hz. We developed NIREST in Java as application container that encapsulates all required components and is deployable on application servers. The application is build on top of the OpenNI[5] framework and the NiTE middleware. OpenNI is as an Open Source framework providing low-level access to colour video, depth video and audio data of sensor devices. NiTE acts as middleware on top of OpenNI and provides body, skeleton and hand tracking.

We use NIREST in our demonstration system as 1) high dynamic data source for 2) body positions and for 3) the position of skeleton joint points of all people in front of the sensor. LD-Fu programs use this data via the REST interface for collision detection and visualizations in the virtual reality.

[4] http://www.microsoft.com/en-us/kinectforwindows/
[5] https://github.com/openni

Our user interface is based on jMonkey[6], an Open Source 3D engine for the development of games in Java. We combine jMonkey with a REST interface to expose data about objects in the 3D scene graph using LD resources. A scene graph is a data structure in VR engines which represents all objects in the VR as well as their interrelations. The LD interface on top of jMonkey allows for the retrieval and modification of data about the 3D scene graph nodes. In our demonstration we employ the visualization based on jMonkey as user interface which can be displayed on a monitor or on a wall using a beamer.

4 Conclusion

With our system we demonstrate data integration in a highly dynamic environment using Semantic Web technologies. We applied successfully the LD and REST paradigms that facilitate on-the-fly declarative data integration in our VR scenario. Moreover, the user will be part of the demonstration system and be able to control the integration and visualization of information via natural interaction all programmed using a declarative rule language.

Acknowledgements

This work was supported by the German Ministry of Education and Research (BMBF) within the projects ARVIDA (FKZ 01IM13001G) and Software-Campus (FKZ 01IS12051) and by the EU within the projects i-VISION (GA #605550) and PlanetData (GA #257641).

References

1. Bizer, C., Heath, T., Berners-Lee, T.: Linked Data – The Story So Far. International Journal on Semantic Web and Information Systems 5(3), 1–22 (2009)
2. (ITU), I.T.U.: End-user multimedia QoS categories. ITU-T Recommendation G.1010, International Telecommunication Union (ITU) (2001)
3. Krummenacher, R., Norton, B., Marte, A.: Towards Linked Open Services and Processes. In: Proceedings of the Future Internet Symposium. Springer Berlin Heidelberg (2010)
4. Pautasso, C., Wilde, E.: Why is the Web Loosely Coupled? A Multi-Faceted Metric for Service Design. In: Proceedings of the International World Wide Web Conference (2009)
5. Speiser, S., Harth, A.: Integrating Linked Data and Services with Linked Data Services. In: Proceedings of the Extended Semantic Web Conference (2011)
6. Stadtmüller, S., Speiser, S., Harth, A., Studer, R.: Data-Fu: A Language and an Interpreter for Interaction with Read/Write Linked Data. In: Proceedings of the International World Wide Web Conference (2013)
7. Verborgh, R., Steiner, T., van Deursen, D., van de Walle, R., Gabarró Vallès, J.: Efficient Runtime Service Discovery and Consumption with Hyperlinked RESTdesc. In: Proceedings of the International Conference on Next Generation Web Services Practices (2011)

[6] http://jmonkeyengine.org/