NexusDSEditor — Integrated Tool Support for the Data Stream Processing Middleware NexusDS

Nazario Cipriani, Carlos Lübbe and Oliver Dörler
IPVS - Universität Stuttgart
{nazario.cipriani | carlos.luebbe}@ipvs.uni-stuttgart.de, oliver@doerler.name

Abstract: In this paper we present NexusDSEditor — an integrated tool for the stream processing middleware NexusDS. NexusDSEditor is an extension module for the NexusEditor and supports developers with designing new streaming applications by providing an integrated tool for orchestrating stream query graphs, define the deployment of query graph fragments to execution nodes, and analyzing data streams. In this paper we demonstrate these single steps and show how NexusDSEditor supports developing streaming data applications for the NexusDS platform by hiding complexity and providing an intuitive user interface.

1 Motivation

Context-based applications access data regarding the user's current situation to adapt accordingly [Dey01]. As an example consider the visualization pipeline as described in [CLM10]. A continuous stream of data flows through the visualization pipeline and results in a rendered image of a map showing nearby surroundings received by a client application. The application helps the user by adapting its behavior to the user context. To process such highly specific schemes, NexusDS [CEB+09] has been developed. NexusDS is a highly flexible and extensible stream processing middleware targeting the processing of context data streams in a highly distributed and heterogeneous environment. Nexus [NGS+01] as well as NexusDS build up a context data management platform where Nexus offers access to static context data and NexusDS provides access to streamed context data. The context data management platform is an open, federated platform for mobile, context-aware applications where arbitrary data providers can make their data available by the platform. A central component is the Augmented World Model (or short AWM) [NM04]. The AWM is a shared, global context model which can be extended by extension schemas.

In the past decade many data stream processing systems have been proposed. However, tool support for those systems is not considered by research comunity. An integrated tool that supports developers of streaming applications, when designing new or modifying existing ones, is beneficial. It helps reducing development time and prevent errors that occur at design time. As presented in [CWGN10], an integrated tool which hides the real task complexity and supports the developer during design time is highly beneficial. As a consequence, we have integrated support for the development of streaming applications within the NexusEditor [NN08]. Thereby, several requirements specific to the domain of

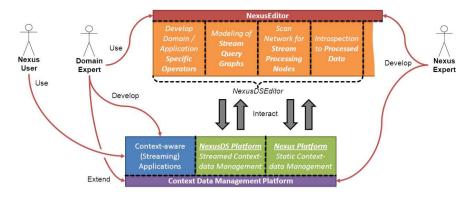


Figure 1: Embedding of the NexusDSEditor

streaming applications must be met: Support development of new domain and application specific operators, provide an intuitive graphical user interface that supports definition of stream queries, offer the possibility to scan networks for available stream processing nodes operators can be deployed to, and present a way for getting an introspection of data processed by the stream query graph at runtime.

2 Architecture

Figure 1 shows the embedding of the NexusDSEditor within the Nexus system. The NexusEditor is the central component in supporting the development process and bridging the world of *Nexus Experts* and *Domain Experts*.

On the right, Nexus Experts develop and maintain the context data management platform as well as the NexusEditor. The context management platform consists of the Nexus [NGS+01] and the NexusDS [CEB+09] platforms. Nexus Experts also develop extensions for the NexusEditor, such as the NexusDSEditor extension as displayed in Figure 1. NexusDSEditor supports Domain Experts during the development process of context-aware streaming applications (as we will show in our demonstration).

On the left, Domain Experts exploit the NexusEditor functionality, and here especially the NexusDSEditor functionality, to develop context-aware applications and context data management platform extensions respectively. By using the NexusDSEditor, productivity is increased since most potential conflicts are recognized and can be eliminated at design time. E.g., beside others the NexusDSEditor supports compatibility checking of interconnections between operators to guarantee a working query graph at design time.

Finally Users use context-aware applications and access functionality and data which Domain Experts have developed beforehand. These context-aware applications run on clients, such as desktop computers or mobile devices.

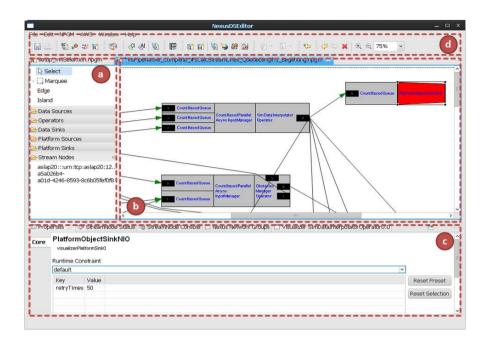


Figure 2: NexusDSEditor screenshot

3 Demonstration

Figute 2 shows a screenshot of the NexusDSEditor. On area (a) resides the toolbox offering different operators and (b) is the drawing area where the orchestration of the actual stream query graph is performed. (c) represents a properties view dependent on the currently selected item from (b). Finally, (d) offers some shortcuts to mostly used functionality grouped within a toolbar.

We demonstrate four scenarios, each focusing on a different requirement from Section 1:

Development of new domain and application specific operators Developers have the possibility to integrate specific operators in NexusDS. To do so, the developer has to provide the actual implementation of the physical operator and also a set of meta data that describes the operator's properties. To satisfy this requirement, NexusDS-Editor provides a modeling component [CWGN10] that allows to easily model the required operator meta data and package the operator for later deployment.

Definition of stream queries NexusDSEditor provides a graphical interface to orchestrate the stream query graph out of single units called operators. An operator is the basic concept of stream query graphs and represents a certain operation on the streamed data. After the stream query graph is modeled, NexusDSEditor supports

the fragmentation of the stream query graphs to so-called *isles*. An isle is a fragment of a stream query graph that is deployed on a stream processing node.

Scan network for available stream processing nodes In NexusDS, stream query graphs are processed in a distributed fashion. Each node, that runs the NexusDS system, can join and consequently be used as stream processing node running a stream query fragment. NexusDSEditor supports scanning the network for available stream processing nodes and present them to the developer. In a second stage the developer can pick the stream processing nodes he prefers and assign them to the corresponding isles. After that process, the query graph is fully defined and ready for deployment. NexusDSEditor therefore offers a deployment functionality that deploys the isles to their associated stream processing nodes initiating the query execution.

Introspection of data processed by the stream query graph After the query graph has been deployed, it is executed and data is processed. Nevertheless, errors may occur during operator development. Therefore it is beneficial, if developers can analyse processed data that is transferred between operators. For this purpose, NexusDS-Editor supports a special kind of operator class namely *Visualizers*. Visualizers consist of two components: One component for connecting to the query graph to retrieve the corresponding data and a second component to display the data within the NexusDSEditor.

References

- [CEB+09] Nazario Cipriani, Mike Eissele, Andreas Brodt, Matthias Grossmann, and Bernhard Mitschang. NexusDS: A Flexible and Extensible Middleware for Distributed Stream Processing. In *IDEAS '09: Proceedings of the 2008 international symposium on Database engineering & applications*, pages 1–8, New York, NY, USA, 2009. ACM.
- [CLM10] Nazario Cipriani, Carlos Lübbe, and Alexander Moosbrugger. Exploiting Constraints to Build a Flexible and Extensible Data Stream Processing Middleware. In *The Third International Workshop on Scalable Stream Processing Systems*, Atlanta, USA, 4 2010.
- [CWGN10] N. Cipriani, M. Wieland, M. Großmann, and D. Nicklas. Tool support for the design and management of context models. *Information Systems*, In Press, Corrected Proof:-, 2010.
- [Dey01] Anind K. Dey. Understanding and Using Context. *Personal and Ubiquitous Computing*, 5(1):4–7, 2001.
- [NGS+01] Daniela Nicklas, Matthias Großmann, Thomas Schwarz, Steffen Volz, and Bernhard Mitschang. A Model-Based, Open Architecture for Mobile, Spatially Aware Applications. In SSTD, pages 117–135, 2001.
- [NM04] D. Nicklas and B. Mitschang. On building location aware applications using an open platform based on the NEXUS Augmented World Model. *Software and System Mod*eling, 3:303–313, 2004.
- [NN08] Daniela Nicklas and Carsten Neumann. NexusEditor: A Schema-Aware Graphical User Interface for Managing Spatial Context Models. In MDM, pages 213–214, 2008.