FSharpGephiStreamer: An idiomatic bridge between F# and network visualization

Kevin Schneider[1] and Timo Mühlhaus[1]

1 Computational Systems Biology, TU Kaiserslautern, 67663 Kaiserslautern, Germany

Summary

Network science empowers researcher across different disciplines to study complex relationships between a huge number of elements and reveal the underling wiring diagram of a system. Therefore, network visualization tools have become a very important component in various data analysis workflows (Barabási, 2011). `Gephi` is an open source software for explorative analysis and visualization of large-scale networks. It facilitates real-time visualizations and analysis to improve the exploratory process by high quality layout algorithms and intuitive user interface (Bastian, Heymann, & Jacomy, 2009). Due to universality of network science, their representation emerges of different data sources from nature or society. Therefore, the need to efficiently connect to rich information spaces and create complex repeatable workflows exceeds the capabilities of automation features in any point-and-click-based network visualization and analysis software available. The functional-first programming languages F# is specialized for type save information-rich programming and domain driven explorative programming (Don Syme, 2006) (Donald Syme, Battocchi, Takeda, Malayeri, & Petricek, 2013). This enables data collection, processing and analysis tasks leveraging the flexibility of the programming environments.

We created `FSharpGephiStreamer` to enable automated access to the `Gephi` network and visualization models directly from within the F# interactive programming environment. The functional approach used for the implementation of `FSharpGephiStreamer` provides a short grammar that models the direct graph data streaming process to transfer the data over the `Gephi` Rest API. This allows an immediate insert, update or delete of individual graph elements directly in the `Gephi` network and visualization model without the necessity to create the network on the client side. The client-host communication works exception-free with error handling using error type monads. `FSharpGephiStreamer` closes the gap between F# and the functionality of the `Gephi` software project. Hence, effectively improving the integration of `Gephi` into large scale data analysis pipelines and increasing workflow reproducibility.

Figure 1: Schematic overview of an interactive network analysis and visualization workflow using F# and Gephi. FSharpGephiStreamer interfaces with Gephi and allows the direct streaming of node and edge data into the Gephi network model.

References

Barabási, A.-L. (2011). The network takeover. *Nature Physics*, *8*, 14. Retrieved from https://doi.org/10.1038/nphys2188

Bastian, M., Heymann, S., & Jacomy, M. (2009). Gephi: An open source software for exploring and manipulating networks. Retrieved from http://www.aaai.org/ocs/index.php/ICWSM/09/paper/view/154
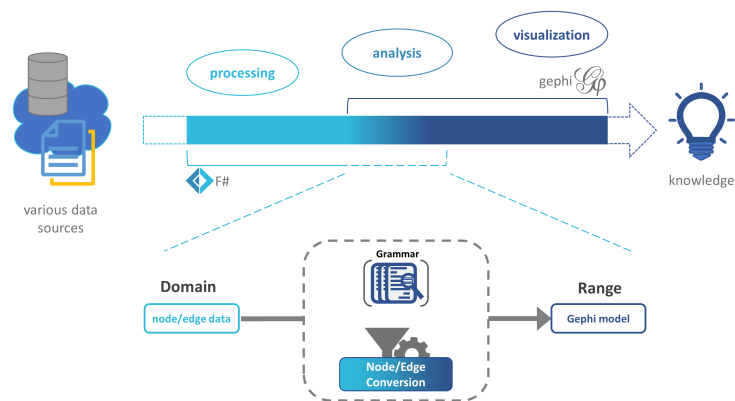
Figure 1: WorkflowOverview

Syme, D. (2006). Leveraging .NET meta-programming components from f#: Integrated queries and interoperable heterogeneous execution. In *Proceedings of the 2006 workshop on ml*, ML '06 (pp. 43–54). New York, NY, USA: ACM. doi:10.1145/1159876.1159884

Syme, D., Battocchi, K., Takeda, K., Malayeri, D., & Petricek, T. (2013). Themes in information-rich functional programming for internet-scale data sources. In *Proceedings of the 2013 workshop on data driven functional programming*, DDFP '13 (pp. 1–4). New York, NY, USA: ACM. doi:10.1145/2429376.2429378