Customizable Coordination of Independent Visual Analytics Tools

Lars Nonnemann, Marius Hogräfer, Heidrun Schumann, Bodo Urban, Hans-Jörg Schulz

Motivation

Background:

- Multitude of Visual Analytics (VA) tools with different functionality
- No tool can be top of the class at all possible tasks
- Some scenarios rely on multiple existing functionalities

In order to combine VA tools, we have to

Either implement a new system

→ Inconceivable Development Overload

Or run VA tools individually

→ Switching breaks the analytic Flow

Our Approach

Idea

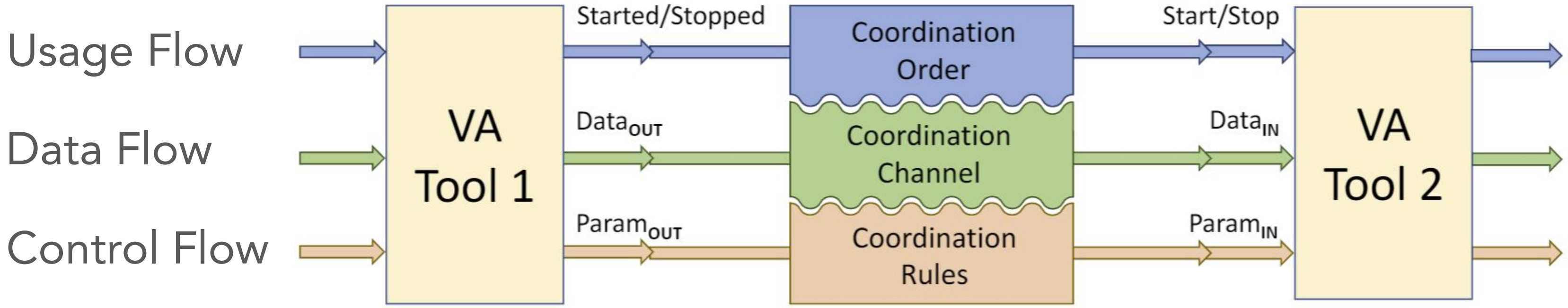
- Using independent VA tools with a lightweight coordination model

Constraints

- Opportunistic: Use any available data channel between two VA tools to exchange information
- Minimalistic: Exchange data only between subsequently or concurrently used VA tools
- Atomic: Utilize VA tools at different timesteps and switch in between them

Conceptual Base

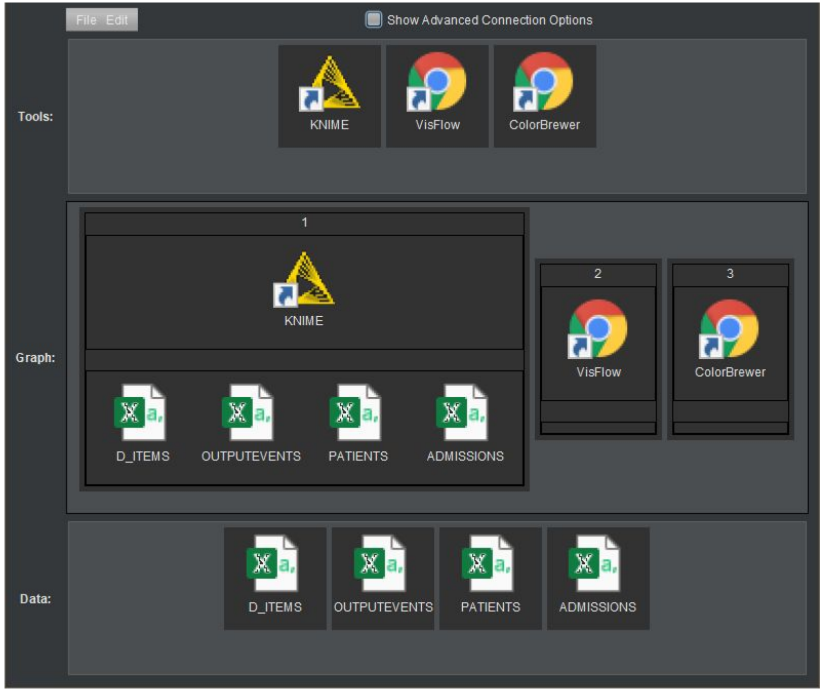
Based on previous concepts (Schulz.2020), we break the toolchain into layers:



The Analytical Process Constructor

Engaging with the three layers of tool coordination:

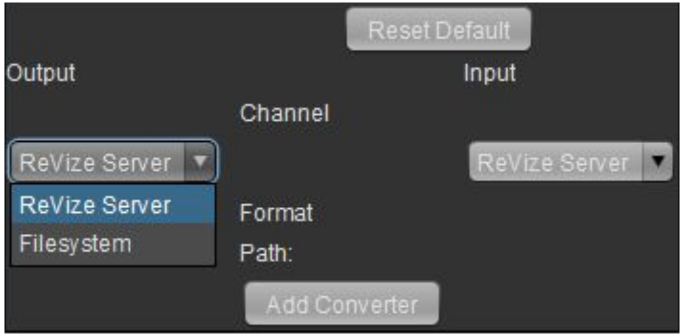
- Usage Flow: Providing a visual component for the assembly or toolchains called editor.
- Data Flow: Allowing the configuration of pairwise data exchange in terms of which channel to use and how the data exchange is to be performed.
- Control Flow: Providing a graphical control interface for the progression through toolchain called executor.



The Analytical Process Constructor

Engaging with the three layers of tool coordination:

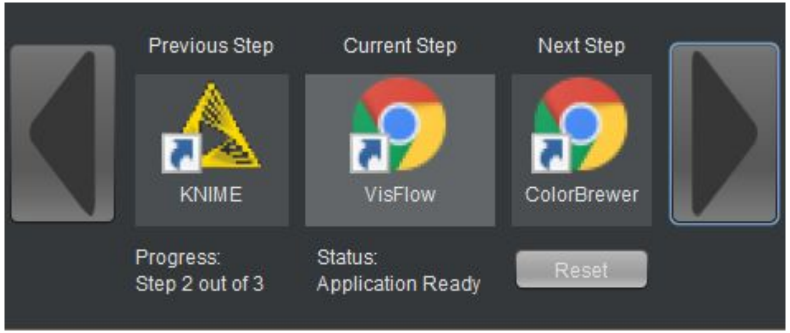
- Usage Flow: Providing a visual component for the assembly or toolchains called editor.
- Data Flow: Allowing the configuration of pairwise data exchange in terms of which channel to use and how the data exchange is to be performed.
- Control Flow: Providing a graphical control interface for the progression through toolchain called executor.



The Analytical Process Constructor

Engaging with the three layers of tool coordination:

- Usage Flow: Providing a visual component for the assembly or toolchains called editor.
- Data Flow: Allowing the configuration of pairwise data exchange in terms of which channel to use and how the data exchange is to be performed.
- Control Flow: Providing a graphical control interface for the progression through toolchain called executor.



Progress:
Step 2 out of 3

Status:
Application Ready

AnyProc Demo

EUROVIS
ZURICH 21

Summary & Future Work

What we showed today:

- Coordination of functionalities from independent VA tools
- Framework for the configuration and execution of VA toolchains
- Customization of data exchange for automatic information transfer

What we aim for in the future:

- Options for handling different formats regarding data, analysis and visualization in potentially ambiguous ways
- Visual overview and annotation on the progression during execution

Thank you for your attention!

Customizable Coordination of Independent Visual Analytics Tools

Lars Nonnemann, Marius Hográfer, Heidrun Schumann, Bodo Urban, Hans-Jörg Schulz

AnyProc code: https://github.com/nonnemann/AnyProc

Further materials: https://vis-au.github.io/anyproc

UnIVA research project: https://nonnemann.github.io/UnIVA/

Contact us: Lars.Nonnemann2@uni-rostock.de mhograefer@cs.au.dk

EUROVIS ZURICH 21

Platzhalterfolie: Inhalt 30 Sec. Teaser