TAMPERE UNIVERSITY OF TECHNOLOGY

European Union
European Regional Development Fund

Leverage from the EU 2014–2020

Toward Manageable Data Sources

Pekka Sillberg

Tampere University of Technology, Pori, Finland

EJC 2018: The 28th International Conference on Information Modelling and Knowledge Bases, Riga, Latvia, June 4-8, 2018

Motivation

- Combine tiny bits of data together easily
  - Visualization of the operation?
- Data management
  - Storing of the raw data, intermediate results or the final results?
- Experimenting/playing around with the data

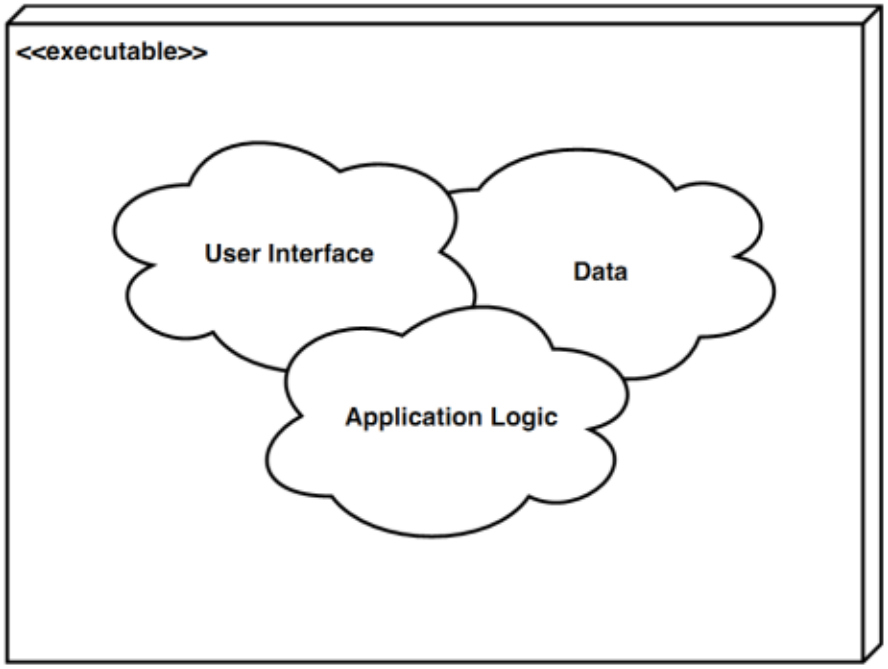
➔How to gain knowledge from the data?

Goals

- Systematic management of data sources
- How the produced data *can* and *should* be managed and utilized by the software
- Identify a generic data processing model for creating a framework to process data requests

Background

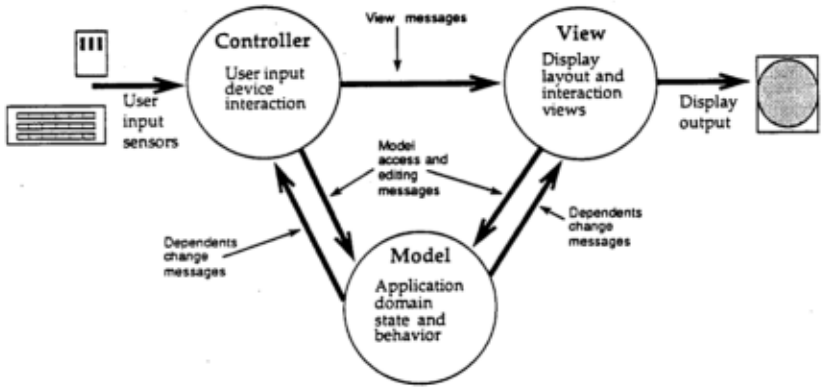
- Unstructured source code and applications
  - May lead to unmaintainable programs
  - Boundaries between different roles are not clear enough
  - Modifications to one place may have unpredictable effects somewhere else
  - Reusability of source code is low
  - …



Improving Software Quality

- Decoupling of responsibilities into components
  - Business logic
  - User Interface
  - Data Access
  - …
- Re-usability of modules and components

Figure 1. Model-View-Controller State and Message Sending

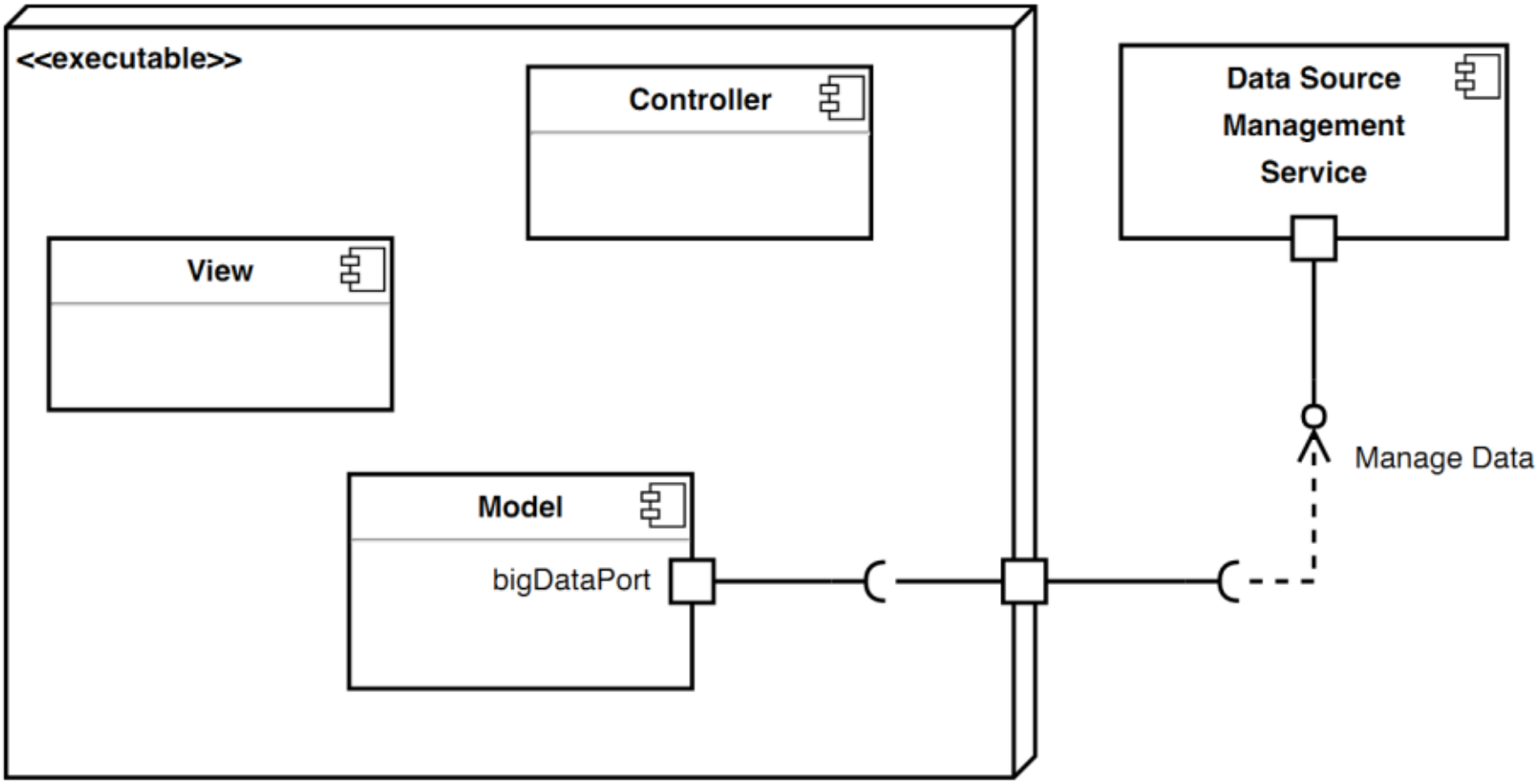


Glenn E. Krasner and Stephen T. Pope. 1988. A cookbook for using the model-view controller user interface paradigm in Smalltalk-80. *J. Object Oriented Program.* 1, 3 (August 1988), 26-49.

Improving Software Quality

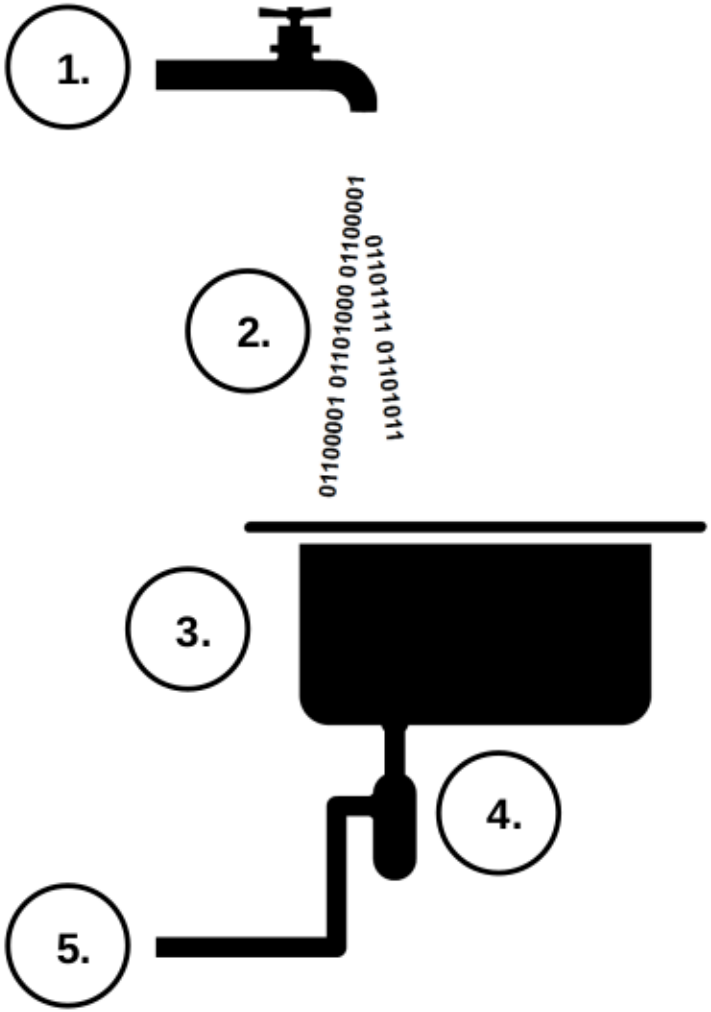
- Design patterns
  - e.g., facade, factory, observer, singleton
- Architectures, e.g.
  - Model-view-controller (MVC) and friends (Model-view-presenter, MVP; Model-view-viewmodel, MVVM)
  - Layered architecture
- Programming paradigms
  - e.g., declarative, imperative, object-oriented

Toward Manageable Data Sources



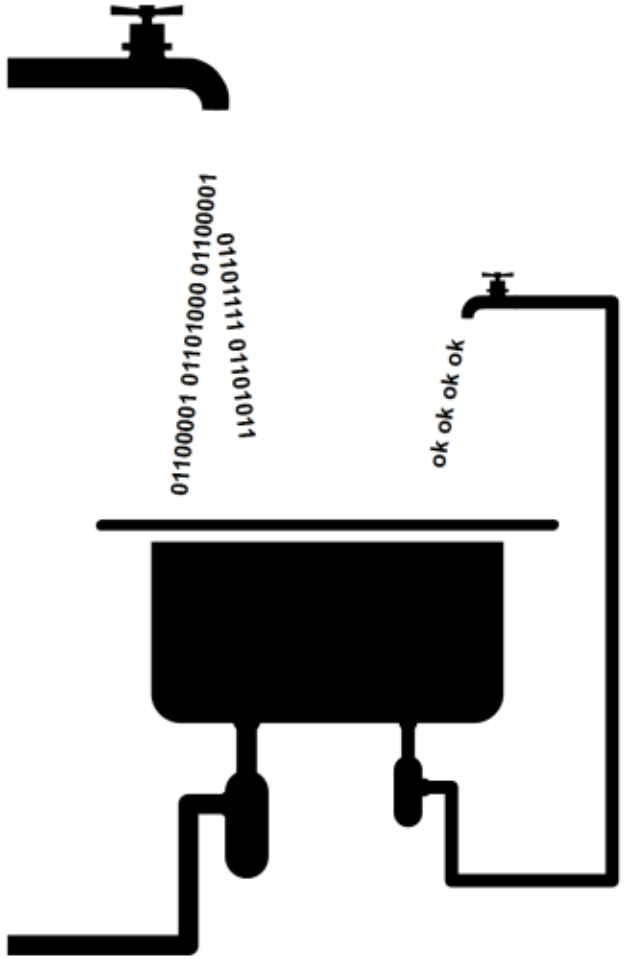
Conceptual Model of Data Processing

- Key components of Faucet-Sink-Drain Model
  1. Faucet
  2. Stream
  3. Sink
  4. Sieve
  5. Drain



Faucet-Sink-Drain Model

Extendability



- Addition of ”feedback loop” to refine the original data
- Any stream of data can be processed as many times as needed
- Drains and faucets may be ”attached” to multiple different sinks
- Utilized in Manageable Data Sources (MDS) framework

Data Faucet

- Represents one source of data
- Defines the characteristics of the data source (e.g., location, update intervals and so on)
- Can be used to embed “user preferences” to the data streams

- Simple examples:
  - Car live data
  - Stock market data
  - Statistical weather data

Data Stream

- Describes the type, format, and origin of the associated data
- Container for the actual data

- Examples:
  - Faucet: Car live data
    - Description: “My Car ABC-123”; Update interval: 1 second; Real-time data: true; Raw data: true
    - Data Stream #1
      - Name: Engine load
      - Description: Calculated engine load
      - Unit: %
    - Data Stream #2
      - Name: Vehicle speed
      - Description: Reported speed of the vehicle
      - Unit: km/h
    - Data Stream #3
      - Name: Engine RPM
      - Description: Engine speed revolutions per minute
      - Unit: rpm

Data Sink

- Central part of the model where each other component is related to (one way or another)
- Storage device where all Data Streams, and associated data will be available at
- May be used for visualization of the data

Data Sieve

- Device that can find and filter the requested Data Streams from the Data Sink
- Can process and combine the selected data into a new Data Stream(s)
  - May consume the selected data (to remove it from the Data Sink) or create a copy of it (to retain the original data in the Data Sink)

- Simple example:
  - Car fuel efficiency (liters/100 km; miles per gallon) is not available as a raw data, but it can be calculated by compounding vehicle speed & time (distance travelled) and mass air flow (fuel used) together

Data Drain

- Device to transfer data streams around
- Can be used to discard the data that is not needed to prevent information overflow
  - Someone (or something) could be interested of the discarded data...
- Faucets that are attached to drains will be able to gather the associated data

- Example:
  - Some entity, e.g. public administration, may be interested of the average fuel efficiency of cars. Users may agree to share a part of their data with this entity.

Future Research Paths

- Proof-of-concept implementation
  - Experimenting with the data
  - Integrate together with visualization methods
  - Evaluation of use cases
  - Carry out a user study to evaluate usability and applicability
- Data protection?
- Data source management on different layers
  - E.g., application, user, operating system…

Summary

- Proposed a new conceptual data processing model (Faucet-Sink-Drain model)
  - Simple and easy?
- Introduced a framework for systematic data source management based on the Faucet-Sink-Drain model (Manageable Data Sources, MDS)
  - User can control what raw data is selected for processing, and how it is refined into information and/or knowledge
  - How the produced data can and should be managed and utilized by the software?

Thank You