Composable Data Processing in Environmental Science - A Process View

Andreas Wombacher*
University of Twente,
7500 AE Enschede, The Netherlands
a.wombacher@utwente.nl

Abstract

Data processing in environmental science is essential for doing science. The heterogeneity of data sources, data processing operations and infrastructures results in a lot of manual data and process integration work done by each scientist individually. This is very inefficient and time consuming. The aim is to provide a view based approach on accessing and processing data supporting a more generic infrastructure to integrate processing steps from different organizations, systems and libraries. We propose an approach modelled in Colored Place/Transition Nets which has been implemented in a Web Service infrastructure.

1 Introduction

In the last years there has been quite some investment in e-science initiatives to support researchers in sharing resources, data and data processing results with each other. All these initiatives have a reasonable infrastructure budget. The resulting infrastructure is pretty closed and does not intend an Internet scale collaboration, also due to the related maintenance costs. Such a closed infrastructure allows to establish control mechanisms to achieve e.g. high data quality.

These maintained infrastructures are valuable for specific domains, while other domains require a less controlled and more open collaboration. An example of such a domain is environmental science, which aims to model and forecast environmental processes, like floods, avalanches, or debris flow. Many of their models are based on meteorological data measurable by nearly everybody. The higher the spatial resolution of meteorological data the better. However, buying, deploying, and maintaining weather stations is expensive and therefore re-use of data has a great benefit. Thus, an Internet style e-science infrastructure is beneficial for this community.

First parts of such an infrastructure are currently emerging with comparably low budget and open to all Internet users. An example are map based applications like for example SensorMap [3] or web sites providing access to sensor data and providing rather simple data processing and visualization functionality like for example in the SensorScope project [1].

The next step in the direction of an Internet style e-science infrastructure is to enable the integration of different data sources and to perform data processing services, like interpolations or visualizations. Right now the environmental engineers download the data from different sources, do a manual cleaning of the data, perform some pre-processing like aggregation of the data, and finally do the intended processing. The data are usually taken from databases or text files and the pre-processing as well as the intended processing is done in the users preferred tools. The different processing steps are hardly combined in a workflow. Most of the time the different processing steps are stated manually and the required data transformation from one processing step to the next one is either done manually or using some shell scripts. As a consequence, the data processing is a high manual effort. The selection of the relevant data set and the configuration parameters of the data processing is dependent on the experience of the user. In particular, the user needs to trace back the required data sets to the different sources and he needs to have a good understanding of the different databases to be able to retrieve the relevant data.

In an Internet scale network of sensors this is not feasible. We expect the infrastructure to support the composition of different sources and to provide means to perform data processing. An example of such a data processing service is the PIPES application [4] for composing, filtering, and processing RSS feeds. In addition, the same data processing should also be applicable to archived data stored in data bases. In particular, we expect the user to express the query based on the **intended processing result** rather than the **input of the data processing** to let the system handle the dependency between the different data sources and pro-

*This research has been conducted during my stay at LSIR department, EPFL, Switzerland.

cessing steps instead of letting it be handled by the user. Such an infrastructure has to address the general problems of data driven applications like for example syntactic and semantic data integration, data provenance, and all kinds of security issues. In this paper we investigate the requirements of such an infrastructure motivated by an example (Sect. 2). Further, we are discussing the data vs. the process architecture alternative in Sect. 3 focusing on querying relevant data. Further, a process based approach is proposed in Sect. 4.

2 Scenario

The scenario we are using to illustrate the work is taken from the SensorScope project [1]. In the course of the project several weather stations have been deployed at the EPFL campus and the aim was to understand better how different meteorological parameters are influenced by urban installations in a model with high spatial resolution. The example described here is on interpolating temperature values of several weather stations over a certain area of the campus under consideration of the buildings and ground types (grass, street, etc) in the area. The outcome of this data processing is a 2D or 3D picture containing the buildings of the investigated area as well as the visualization of the interpolated temperature values.

The general process consists of the following steps:

- define the study area and acquire the relevant information like the positions and height of the buildings, as well as the different ground types,
- digitize the buildings and add them to the GIS tool, which is later on also used to do the temperature interpolation calculation,
- determine the available environmental data by finding the weather stations being available in the area,
- specify the time frame of the study and retrieve the relevant data for this time frame from the corresponding database(s),
- use MatLab or a similar tool to do some filtering of data to clean the data and to further on aggregate the data (e.g. average per hour) based on a resolution suitable to the timeframe investigated in the study,
- feed the measured values into the GIS system and do the interpolation calculation within the GIS system using model specific parameters,
- the quality of the interpolation result can be evaluated using correlation coefficients; dependent on the quality of the interpolation the previous step might be repeated with different parameters to accomplish a better quality,
- the interpolation is visualized in a picture containing the buildings as well as the coloring according to the interpolated temperature values.

Right now this process is completely manual and the used raw data, all the intermediate results, knowledge about the coefficients, and the use of interpolation functions is hardly documented. In particular, this study has been performed by a master student and she documented the collected results and the corresponding knowledge in her thesis. However, the content of the thesis is quite an incomplete description of the knowledge, since for example not all acquired raw and processed data is contained in it.

In our research we interviewed several people from different disciplines and organizations to understand the complexity of the queries which are usually applied on the data collection. It turned out that users perform queries to find specific behavior or thresholds in the data collection mainly to determine time and location of specific events. Another class of queries are the ones described above on gathering data as an input for a model. In the latter case the data of different sources are always correlated (join or union) based on time. The **time correlation requirement** is a specific property of the e-science domain and has a strong influence on the design options for handling the data management.

Currently, environmental engineers do not have an infrastructure to support their work, but use different systems providing certain functionalities. The **integration of different systems** for a specific study is done by user specific tools. In particular, the user is setting up a specific infrastructure for a specific experiment, where he re-uses as much as possible from previous experiments. However, it is always a manual adaptation and an experiment specific solution. Further, the user has to collect the data from the different sources and integrate them himself.

This approach works fine for scenarios where the environmental engineer is in control of the complete set up (weather stations, data acquisition, database, and processing). In case the control is limited for example because the weather station is maintained by a different organization and therefore he does not know how to access the database or what the actual structure of the database is, this approach is not working anymore. Therefore, an **infrastructure supporting access to distributed data** according to the data needs of the study is required especially in an Internet scale network of sensor measurements. Furthermore, the **infrastructure has to support data processing** of the retrieved data in an easy way. In the following we will discuss two different approaches on addressing this requirement from a data and a process oriented perspective.

3 Design Options

The core of the problem stated above is data handling in a distributed environment. This can be understood as a data management problem addressed by a distributed database system, or as a data processing problem addressed by data

flow or workflow system.

3.1 Data Management

access to distributed data In a distributed data management system, the distribution of information is transparent to the user. The data management system takes care that the distributed information is gathered in an optimized way to answer the user request. Thus, a user of the system has to understand the schema of the database to retrieve the relevant data using a proper query. Furthermore, the user has to specify the corresponding data integration in every data request again.

The repeated specification of data integration can be avoided by introducing views for a set of integrated data sources. This is a common concept in data management systems and can be differentiated in materialized and transient views. In either case, the user can query directly the view and does not have to care about sources and query delegation. This is exactly what we are looking for.

However, a distributed data management system addresses the huge data amount and performance issues rather than the aspect of crossing organizational boundaries. In a cross-organizational usage scenario the data needed are stored in different (distributed) data management systems. Using a distributed data management system involving several organizations is quite unlikely due to communication overhead effecting the performance and due to management and administrative issues (like for example security) of such a system. In particular, access control in a cross-organizational scenario is a challenge, which has to be addressed on communication, application, and political level.

A distributed data management system is middleware and thus application level requirements should not be implemented in a generic middleware. Therefore, we are looking for a data management solution having the functionality of a distributed data management, but where the individual data management systems are loosely coupled with each other such that the lacking requirements can be realized on the application level.

support of data processing Besides the distributed data access, the support of data processing is essential. As described in Sect. 2 most of the processing is done using specific applications, which can not be directly applied by a database query. Data management systems support *stored procedures*, which are data management system specific implementations of additional functions. The usage of an extension in a query constitutes a kind of a data processing. The first option to address the data processing is to integrate external applications into the database. The idea is to define a generic interface and implement this as a stored procedure. This approach is too strong depended on the exter-

nal processing and not applicable in a cross-organizational case. The second option is to re-implement the processing function as a stored procedure. Again, re-implementation has high costs and can only be applied on local databases. The third option is to explicate data views and implement the processing chain using other technology. The idea is to define the integrated data as a materialized view and to implement the data processing outside the data management system based on these views. This approach still requires a data processing infrastructure and replicates the data for each processing step. However, the data integration part can be addressed using queries in the data management system.

None of the discussed options covers the requirement of flexible integration of external applications supporting the data processing. Further, going with one of these solutions seems to be quite costly but most likely will have the advantage of faster processing. However, in the scenario described in Sect. 2 we are discussing an offline processing, thus, it is less time critical. Therefore, we investigate next a slower, but more flexible and less costly approach.

3.2 Workflow

support of data processing In a workflow a complex functionality is assembled by calling several simpler processing steps (activities). Besides the control flow addressing dependencies between different activities, the data flow specifies which data is consumed and provided by which activity. Workflows can be distributed themselves, thus, the support of distributed data processing is inherently provided. Obviously, a chained or a loosely coupled workflow scenario with dynamic composition of workflows provides some challenges, like e.g. discovery of relevant workflows and the consistency of a composed workflow.

The security issues as discussed in Sect. 3.1 depend strongly on technology choices made for the workflow system and the communication with the activities. In this paper we propose to use Web Service technology, since WS-Security is independent of the used processing steps (activities). Furthermore, the provision of processing steps by an organization is very generic and can be used (if allowed by the security policies in place) by all people in the field. Thus, the integration of processing steps is much easier.

access to distributed data Workflows are hardly applied for distributed data processing. However, workflow models addressing these issues are e.g. Dataflow Process Networks [2]. This kind of models is based on the concept of input and output streams where the process is able to selectively read data from the input stream, to write data to the output stream, providing a back channel to the source, and to terminate. These models can be translated into classical Place/Transition Nets again. Since the scenario under inves-

tigation is not a streaming scenario, we are directly using a Colored Place/Transition Net (CPN) notation. The data available at a workflow execution state represent a view of the data at that particular state. I.e., each state of the workflow based data processing represents a data view. Thus, the concept of views is applicable although the workflow system does not support querying a view of the data.

As a consequence, the weaknesses of the data management approach are the strength of the workflow approach. In this paper, we propose a workflow view on the investigated scenario. I.e., we propose to use workflows and special activities as an infrastructure for data processing.

4 Approach

As stated in Sect. 3.2 the workflow based approach is lacking query delegation possibilities. As a consequence we have to model this functionality based on a notion of views.

Before query delegation is applicable, the data processing chain has to be established. This means that a new view is composed based on existing views on demand of the user. In particular, a user must be able to **compose different views** at run-time and to query the just composed views. After the view is initiated, the view is publicly available and supporting data access.

Querying these views means that a component called "view generator" receives the query, decomposes it based on the definition of the view generator and delegates the query to the source views. This **query delegation** is processed recursively until no further decomposition is possible or the view is materialized. An automatic query delegation requires a deep understanding of the dependency of output and input data. In a service based data processing each processing step is a black box to the workflow system, thus each data processing step has to specify how to delegate the query. Therefore, each query posed to the final view has to be **registered** with the processing step generating this view. This registration is then posed to the views used.

The basic idea behind retrieving and **accessing the data** is to use standard concepts like for example in JDBC. Here the database is queried and the results of the query are delivered using a cursor: In JDBC the query results are accessible via a cursor: The **next()** operation moves the cursor one row forward in the result set and returns a boolean value whether it is positioned at a valid row. The row content is accessible using getter operations for retrieving column values. For simplicity reasons we use a **get result** operation, which returns the complete result instead of a single column. In case the cursor is behind the last row, the **get result** operation will return nothing, i.e. representing the null object.

This approach is a pull based approach, where the consumer of the data is requesting a new data set. An alternative would be a push based approach, where the data provider delivers data without additional requests based on a before agreed data rate. We decided to go for the pull based approach since the data consumer may have to coordinate and synchronize the data rates of several data sources.

The implementation is based on Web Service technology, since they are applicable in cross-organizational scenarios. Further workflow tools exist to combine basic processing steps and required data integration functionality. The existing implementation covers different view generators and an optimization of the communication costs for the retrieving the data.

5 Conclusion and Future Work

Data processing of environmental measurement data has to deal with big data collections and has to support data processing in different applications. While data management solutions are very well suitable for handling the distribution at least within a single organization, the support of integrating external applications and the distribution over several organizations is challenging. However, a very valuable concept is the concept of views. In particular, we are proposing a distributed view based approach where we make use of the observation that the correlation of environmental measurement data is always done based on time. We present a process view of a query delegation mechanism and keep the issue of controlling the own resources in mind.

While in data management querying a view is a standard mechanism, this is not the case in comparable process based approaches. Data flow models support the processing of selected data, but do not interpret the resulting data as a view and thus, do not support to query the view. The proposed approach provides this capability. It has been implemented as a Web Service and has been applied in the context of a environmental data acquisition and processing environment.

In future work we will investigate performance in detail and see how we can optimize the implementation to accomplish scalability of the approach.

References

- [1] Sensorscope wireless distributed sensing system for environmental monitoring, 2007. <http://sensorscope.epfl.ch/>.
- [2] E. Lee and T. Parks. Dataflow process networks. In *Proceedings of the IEEE*, volume 83, pages 773 – 801, 1995.
- [3] Microsoft. Sensormap, 2007. <http://atom.research.microsoft.com/sensormap/>.
- [4] Yahoo.com. Pipes: rewire the web, 2007. <http://pipes.yahoo.com/pipes/>.