Environmental Systems Research
a SpringerOpen Journal

RESEARCH Open Access

Conceptual model for environmental science applications on parallel and distributed infrastructures

Denisa Rodila[1,2*], Nicolas Ray[1] and Dorian Gorgan[2]

Abstract

Background: The global changes that are currently threatening the natural environment demand appropriate answers and solutions by the environmental science community. The increasing amount of heterogeneous data—Big Data—needed for that endeavor typically requires large computational and storage resources. This manuscript presents a general conceptual model for easily porting environmental applications on different parallel and distributed infrastructures.

Results: We developed the conceptual model for a general environmental application and illustrate it through a use case on hydrological modeling. We also positioned this concept in a general methodology that will be used for efficiently porting applications on different computing environments.

Conclusion: The proposed conceptual model of an environmental application facilitates and simplifies not only the understanding of the structure of the application but also the general execution flow and the data flow. It provides a platform-independent, flexible and convenient way to execute the described application in a heterogeneous computing environment.

Keywords: Environmental applications, Conceptual model, Big Data, Cloud, Grid, SWAT, OGC

Background

At the beginning of the 21st century, global changes linked to climate, biodiversity and habitat loss, environmental degradation and pollution, are threatening our natural environment and the human society at large, with already tangible negative outcomes [see Climate Change 2014 Synthesis Report—IPCC (2014)]. Intensified droughts, ocean acidification, global sea level rise, increases in frequency of extreme weather events and glaciers melting are examples of such outcomes that are thought to intensify if appropriate international policies are not endorsed and applied.

Responding effectively to all these complex changes has become an important challenge for policy makers, but also for the scientific community that demands access to continuously increasing quantities of heterogeneous data and resources [see e-IRG Report on Data management—ESFRI (2009)]. Scientists need to understand the interlinkage between natural phenomena and human-induced activities and an important aspect for achieving this is the accessibility and processing of environmental data from various disciplines and geographic scales (local, regional, national and global).

Turning this data into knowledge is not an easy task, especially when locating and accessing the right resources (e.g. data, information, tools and services which can be information about the state of the Earth, relevant services, project results, applications, etc.) is done in a very scattered way through different state organizations, operators, service companies, data catalogs, scientific institutes, etc.

In the domain of Earth and environmental science there is an unprecedented avalanche of data due to a large extent to the fast evolution and availability of sensor/

*Correspondence: Diana-Denisa.Rodila@unige.ch
[2] CGIS Lab, Computer Science Department, Technical University of Cluj-Napoca, G. Baritiu 28, 400027 Cluj-Napoca, Romania
Full list of author information is available at the end of the article

detector technologies. The advances in IT that enabled the capture, analysis and storage of massive amounts of data contributed also to this avalanche of data. The so-called "digital data deluge" is a phenomena caused not only by the ease with which these large quantities of new data can be created but also by the output of re-analysis of already existing archived data [e-IRG Report on Data management—ESFRI (2009)]. This phenomenon is considerably changing the way science and research is being conducted in many disciplines as they are dealing with unprecedented sizes of data that needs massive computing capacities to handle it. The concept of "Big Data" emerged in this context.

Big Data are usually defined not just as massive data sets but also as data having very complex and varied structures, making further actions (e.g., storage, analysis, visualization, processing) very difficult. New satellite, airborne and ground-based remote sensing systems characterized by high spatial, temporal and radiometric resolution are, or will be soon, available. With the launch of three families of Sentinels satellites, Copernicus will be producing for example, 8 TB of Earth Observation data per day (approximately 3000 TB per year) [Big Data Workshop—Copernicus (2014)], which will lead to an increase of data volume, diversity and also value. Based on this, the main characteristics of Big Data are gathered around the "5V" (Demchenko et al. 2012):

- Volume: available amount of data;
- Velocity: rate of data collection;
- Variety: the variety of sources producing Big Data but also the implementation of services dealing with these different types of data;
- Veracity: validity and accuracy of the data must be taken into account considering that data sources can be of different qualities, especially when it comes to coverage, accuracy and timeliness;
- Value: how meaningful the row data is and how valuable is the obtained information (the main purpose of Big Data is to produce meaningful Small Data).

Big Data is already embedded in environmental science studies and is mainly produced by three important sources (Yang and Huang 2013): (1) From the impressive array of sensors that are placed in space (via remote sensing satellites) and in situ, used to measure and monitor weather, precipitations, vegetation, land cover, water quality, as well as other geophysical parameters. These collections of data sets satisfy all the characteristics of Big Data (the 5 Vs). (2) From the various scientific model simulations used for predicting physical phenomena. Climate change for example can be considered one of the largest use cases of scientific modeling and simulations. Nowadays climate simulation can be run on a daily basis with increasingly higher horizontal (hundreds of meters rather than tens of kilometers) and vertical (more model layers in the atmosphere) spatial resolution, as well as higher temporal resolution (minutes or hours rather than days or weeks). The update of these models is done more frequently and with much higher quantities of new data. Therefore the amount of data coming out of these simulations is very large, reaching typically petabytes of data from just one simulation. Based on this we can conclude that this data can as well be considered Big Data. (3) From data assimilation, the process by which models are updated with the latest observational data to be able to correct and validate the assumptions made in the model due to different factors like missing parameters, incorrect data, etc. Analysis of this Big Data can give unprecedented possibilities for better decision making for understanding and mitigating the effects of climate changes.

Nativi et al. (2015) emphasize the Big Data challenges in Global Earth Observation System of Systems—GEOSS (GEO 2005)—and particularly its common digital infrastructure (GEOSS Common Infrastructure—GCI). The presented challenges can be identified along all the Big Data dimensionalities: volume, variety, velocity, veracity and visualization.

Environmental data are most of the time spatially referenced (i.e., referring to a geographic location) and as such belongs to geospatial data or geodata. Geospatial data describes geographical locations by giving attributes/information about their spatial and/or temporal extents (Giuliani et al. 2011). The amount of geospatial data has grown dramatically in the last 30 years mostly due to the rapid progress of communication means, as well as technologies to capture this type of data (e.g., GPS, sensors, satellites). Geospatial data is typically voluminous, complex, heterogeneous and geographically distributed. All these attributes make it generally difficult to access, share and distribute geospatial data, often with challenges to combine it with other types of data sets. Nowadays geospatial data is used and analyzed most of the times within a Geographical Information System (GIS) that has capabilities such as assembling, storing, manipulating, displaying, and merging data from different sources (Giuliani et al. 2011). In environmental sciences, GIS can be used in conjunction with Spatial Data Infrastructures (SDIs) that are widely used to share, discover, retrieve and visualize geospatial data through standardized services [e.g., Open Geospatial Consortium services—OGC (1994)]. SDIs are therefore more than just data repositories, although suffering from limited analytic capabilities. Making use of GIS and SDI, a wealth of geospatial applications, technologies and initiatives have emerged recently in order to handle the increasing amount of environmental data, and to extract useful information out of it.

IBM for example, offers free supercomputing hours in the World Community Grid (http://www.worldcommunitygrid.org/) for researchers who are analyzing and studying climate change. Google has also donated 1000+ terabytes of cloud storage for satellite observations and climate models. After the White House's Climate Data Initiative (https://www.whitehouse.gov/the-press-office/2014/03/19/fact-sheet-president-s-climate-data-initiative-empowering-america-s-comm), in March 2014 a large amount of climate data has been made public from different organizations and agencies (NOAA, NASA, US Geological Survey, US Department of Defense). The goal of this was to encourage data providers, scientists and the public in general to share data and make use of the obtained information. From March to June 2014, ESRI hosted the Climate Resilience App Challenge (http://www.esri.com/software/landing_pages/climate-app) for governments, private industries and non-profit organizations to submit climate resilience applications. The number of useful submissions, addressing different aspects of climate change, was outstanding. In May 2014 the United Nations started a new initiative on climate change—the Big Data Climate Challenge (http://unglobalpulse.org/big-data-climate/)—that aims to use Big Data for supporting climate change actions, i.e. "to bring forward data-driven evidence of the economic dimensions of climate change".

Applications that are used to solve different environmental issues, use specific data as input and produce outputs that are useful for the Earth and environmental community at large can be labeled as "environmental science applications" (or simply "environmental applications" hereafter). Since the 1990s, the number and diversity of environmental applications have increase dramatically. Many software systems were developed to integrate data coming from various thematic areas such as agriculture and soil science, ecology, terrain modeling, hydrology, land use/land cover, population distribution, education and health planning, energy resources, etc. The specificity of the majority of these environmental applications is the requirement of large computational and storage resources due to the massive amount of input and/or output data that is typically due to a combination of high spatial and temporal resolutions. Other reasons for high performance requirements include also the utilization of compute-intensive algorithms, the execution of large number of scenarios, the urgent need of responses, etc. Different parallel and distributed infrastructures, such as Grids, Clouds, and High Performance Computing (HPC) systems can satisfy the necessary requirements for running these applications (Nativi et al. 2013).

Despite the popularity of Big Data nowadays and the existence of solutions to handle the afferent challenges (with respect to storage, management, interoperability, governance and analysis), putting these solutions into practice is still a time consuming endeavor. Big Data storage management is indeed among the most important challenges for computing environments since many data intensive applications usually involve a high degree of data access locality. Data locality is thus a key aspect in providing performance for Big Data processing as transferring such large amounts of data would considerably slow down the process. Typical high performance computing systems did not take data locality into consideration as they used to focus on performing CPU-intensive computations over a moderate to medium volume of data (Assuncao et al. 2015), where the ratio of data transfer between the computing units to processing time is still small. Considering the context of Big Data, this solution is in most of the cases inefficient. The alternative is to move the computation as close as possible to where the data is. Existing parallel and distributed infrastructures already have built in mechanisms for efficient transfer of data among the computing units, although, considering the increasing volume of data we are dealing with, this option is no longer efficient. In (Assuncao et al. 2015), the authors argue different existing and on-going solutions for dealing with data locality in Cloud environments; similar initiatives are carried for other computing infrastructures such as Grid (Kumar and Bawa 2012).

Considering all the efforts of computing infrastructures to keep up with the increasing demands of Big Data, parallelism and distribution are still good solutions to efficiently execute data intensive applications. Some examples of environmental applications taking advantage of the capabilities offered by parallel and distributed infrastructures are those using parameter estimation, model calibration (Vrugt et al. 2006; gSWAT 2011), Web Processing Service on the Grid (Giuliani et al. 2012), numerical weather prediction (Maity et al. (2013)) and satellite images workflows over the Grid (GreenLand 2011).

The choice of the appropriate parallel or distributed infrastructure depends on the application features, data model, and processing requirements of the environmental application. To run on one or several of these distributed or parallel infrastructures (i.e., an heterogeneous computational environment), the application has to be modified to have a particular structure or to use particular programming interfaces for accessing the resources of the infrastructures. This is typically done without knowing too much details about the final infrastructure(s) on which the application will run.

However, to our knowledge there is no convenient tool/framework to allow a user to easily express and control the execution of an environmental application in a heterogeneous computing environment, without having expertise in sophisticated workflow systems or control of the backend functionality. The main goal of this manuscript is to fill that gap and to propose a conceptual model of environmental applications, which will be a key component in a general methodology for porting these applications on different parallel and distributed infrastructures. The conceptual model facilitates and simplifies not only the understanding of the application structure but also the general execution on different computational platforms. It provides a platform-independent, robust, convenient and easy way to use a mechanism that allows a user to execute an application on a heterogeneous computing environment, and as such provides a first step towards the automation of this process.

Figure 1 shows the overall conceptual model architectural context and the goal of our final methodology.

The methodology consists in proposing solutions to easily and efficiently port and execute environmental applications on different parallel and distributed infrastructures, using the conceptual model proposed in this manuscript. The main steps in this general methodology are: (1) conceptualize the environmental application (i.e. create the conceptual model), (2) instantiate the conceptual model with specific application data, (3) collect user specifications (data formats, application type, execution preferences, etc.), (4) check for similar executions performed in the past (history), (5) execute the application, and (6) collect the results. The execution of the applications and the selection of the computing environment(s) can be done automatically by a Mediator component, based on a complete conceptual model as well as on application related information provided by the user and other useful information such as availability of computing environments, previous execution history of the application, etc.

The development of this general methodology is still a work in progress. The purpose of the current manuscript is to detail the Conceptual Model which is a key component in this methodology.

The final methodology, based on the proposed conceptual model, will bring important contributions to the environmental science community but also to the parallel and distributed computation field. Despite the fact that the conceptual model is based on environmental applications experiments, it is flexible enough to be reused in other scientific areas. The simultaneous usage of different computational infrastructures is still a research challenge due to the complexity of each individual infrastructure but also due to the complexity of the interoperability between them.

Environmental science applications

Environmental science is a multidisciplinary field that integrates physical, biological and information sciences to study together the systems, the problems and the solutions of the environment. In the beginning of environmental science in the 1960s, the scientific community was more focused on disciplines, trying to develop knowledge in particular fields (such as geology, ecosystems, hydrology, etc.) but in the 1980s it became more and more obvious that these disciplines are strongly connected and the scientific community started to study them as interacting elements in a single big system (Dozier 2009). After this shift, it was easier to understand complex, system-oriented phenomena that link concepts from different fields (climate change involves atmospheric science, biology, human behavior, etc.) but also to understand and make a better use of the collected data (such as these coming from satellite observations). The growing understanding of these complex processes lead also to the development of new models. The knowledge gathered mainly for scientific understanding, begins to be used more to support practical decisions and actions, redirecting the environmental science to environmental applications. The role between basic science and applications is emphasized by the societal needs. After collecting and analyzing the gathered information, the community needs also a more fundamental, process-based understanding of the phenomena—a science of environmental applications. This science is guided more by societal needs than by scientific curiosity, focusing more on specific actions as well as on their consequences (Dozier 2009).

In environmental science there is data that is considered “independent” and data considered “dependent” . The “independent” variables are the ones being manipulated and selected to determine its relationship to an observed phenomena. These are normally the input variables that are observed in its naturally occurring variation. The “dependent” variables are the observed results of the independent variables and are usually the output variables that cannot be directly controlled. The distinction of dependent and independent data is done by the researcher and by the context in which it is applied. Now considering the form of the response (dependent) environmental data, we can specify several types of data: continuous data (such as temperature, mass, distance), counts (simple—the number of plants infected by a disease, or categorical—the number of infected plants classified into tree species and town), proportions (such as:

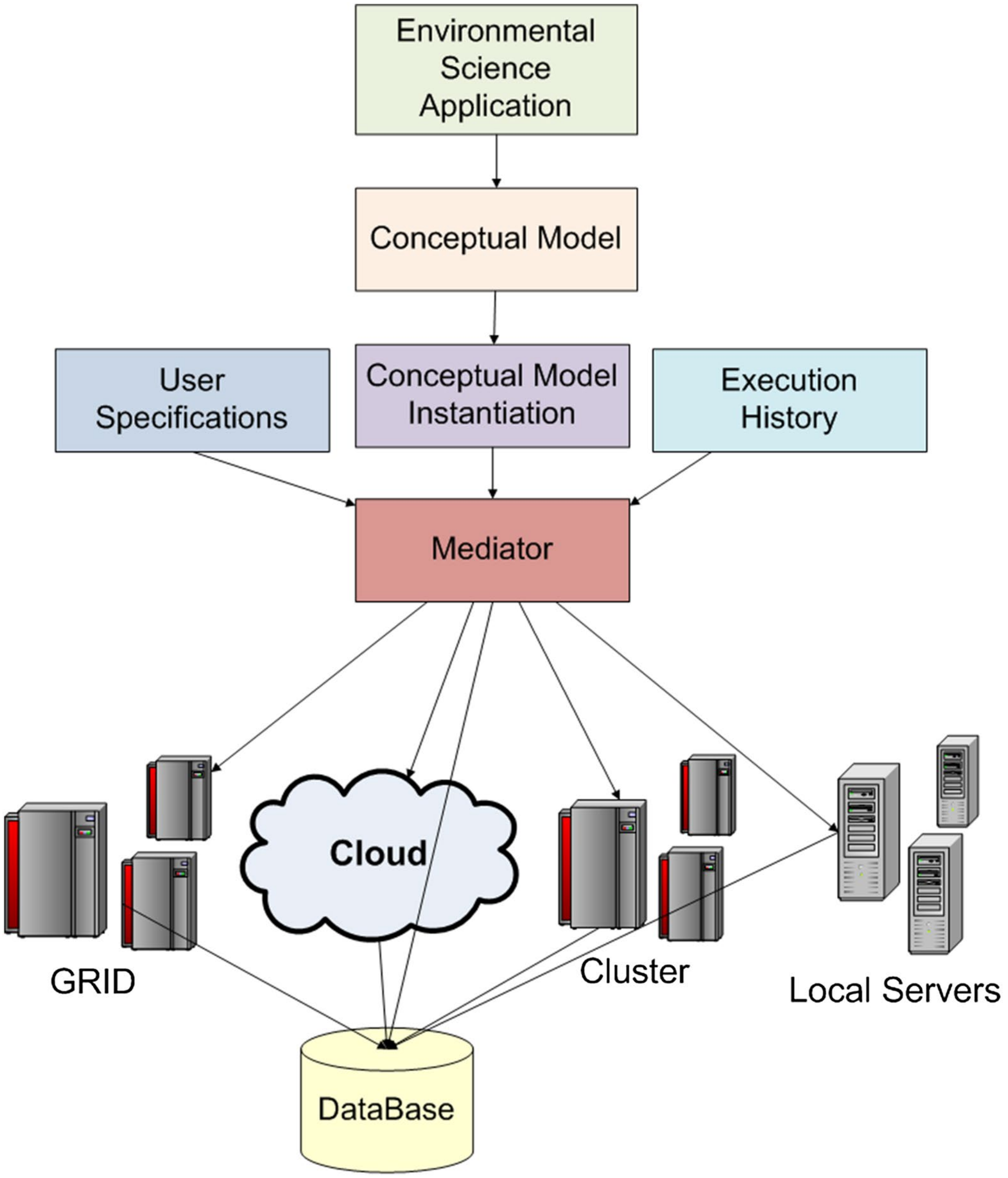

Fig. 1 Conceptual model architectural context

percent mortality, sex ratio), binary data (ex: alive or dead, present or absent), time to death/failure (ex. the time it takes juveniles to disperse out of the study area), time series (such as temperature data measured at fixed intervals, river discharged measured over time) and circular (ex: day of the year). A detailed description of all these types is done by Piegorsh and Bailer (2005), and in (Environmental Data Analysis 2005). There are also many de-facto standards for delivering environmental data such as: HDF (Hierarchical Data Format), HDF-EOS, NetCDF (network Common Data Form), NetCDF-4, XML with initiatives such as GML (Geography Markup Language), CSML (Climate Science Modeling Language), ESML (Earth Science Markup Language), etc.

Conceptual modeling

Conceptual modeling is the activity of formally describing properties and actions of the physical and social world, with the purpose of better understanding, communicating and visualizing these properties and actions. The descriptions that arise from conceptual modeling are meant to be used both by humans and machines. The approach of conceptual modeling was first associated to semantic data modeling, but it soon found applications in many other fields such as modeling organizational environments, modeling software development processes or even modeling different parts of the world for better human communication and understanding (Mylopoulos 1992).

Conceptual models, mostly graphical, are used to represent both static and dynamic phenomena and they usually play an important role in communication between developers and users, for example for understanding of a new domain, providing a good documentation or providing input during the design process. High quality conceptual models also enable early detection and correction of errors (Wand and Weber 2002). A conceptual model is always an approximation, with different levels of details, of the real world system being modeled. It is a physical, mathematical or logical representation of a system, phenomenon or process and serves as a representation of an event/thing that is real or deliberately created. A model is thus produced by abstracting from reality a description of the system, with the observation that not all aspects of the system are represented, as this would be typically too time-consuming, complex and expensive.

Considering that environmental science is a complex and interdisciplinary domain, conceptual models are useful methods for meeting the challenges of deep understanding of the studied environmental phenomena. Conceptual models are useful in improving the coherence and analyzing the environmental issues and integrating knowledge. They can help the user not only to understand the complexity of an environmental system, but also to comprehend the variety of existing scientific approaches used to formulate and solve environmental problems (Fortuin et al. 2011). Not much research work has been reported, to our knowledge, on the usage of conceptual models for describing environmental applications. We review below what is found in the literature on that subject.

ISO191xxx (2003) is a series of standards defining and managing geographic information that is based on conceptual modeling. The main goal of ISO 191xx series is to facilitate the interoperability of geographic information systems by providing abilities to discover, access, understand and use the information and tools independently from the platform supporting them. ISO 18101 also defines a fundamental concept of geographic data—the feature—that is an abstraction of the real world phenomena. The research presented in (Fortuin et al. 2011) highlights the usage of conceptual modeling in facing the challenges given by the complexity and interdisciplinary character of the environmental science curricula. However, the usage of the conceptual models is oriented there more on solving the problems at the academic level instead of actually providing a deep understanding of environmental science applications and their interactions with different computational environments like we intend to do.

Nativi et al. (2013) present the concept of Model Web—a Model as a Service approach which will increase environmental model access and sharing, facilitate modeler to modeler and interdisciplinary interaction and reduce reinvention. The final idea is to have a wide network of interconnected models, data, and tools accessible via websites that are available as a resource for decision makers, researchers and the general public. In describing the Model Web conceptual framework, the manuscript introduces an entity (Model), which represents the conceptual and mathematical structure of an environmental model. Together with this entity, it also introduces other concepts, all part of the conceptual framework: Application, ModelRepresentation, ConfigurationParameter, ModelParameter, ModelRun, ModelEngine, Dataset, Service, InputData, ModelOutput. All these entities are elements in a procedural representation of an environmental model.

Wand and Weber (2002) and Davies et al. (2006) present a detailed description and a framework on conceptual modeling. Mylopoulos (1992) presents the process of conceptual modeling through the existence of four different kinds of knowledge: subject world, usage world, development world and system world. With these, conceptual models can reach a very high degree of complexity while trying to integrate as many aspects as possible from the simulated process/studied phenomena. However, the purpose of our study is to keep things as simple as possible and to highlight only the necessary details, allowing a more flexible execution of environmental science applications on different computational environments.

In (Modeling and Simulations Fundamental, Sokolowski and Banks 1970), the authors discuss about the degree of uncertainty that each real world system has got and about the way in which this uncertainty and variability can be included in conceptual modeling through random variables and random processes. Parekh (2005) proposes the use of ontologies and Semantic Web technologies to tackle the complexity and diversity of knowledge and data within the environmental sciences and engineering with the purpose of enabling efficient data sharing. "Ontologies provide shared domain models that are understandable to both humans as well as machines." Ontologies provide an abstract conceptualization of information by defining basic concepts in specific domains together with their relations. The advantage is that all these definitions are both human and machine interpretable, leading to efficient automated mechanisms for information sharing and integration. The goal of their research work is to use ontologies to provide semantic interoperability among heterogeneous data, semantic descriptions of the datasets, as well as a common conceptual model for those datasets. In environmental science, the modeling activity can be complex and difficult, even for a specialist: acquiring knowledge of individual computational models, searching, gathering and analyzing raw data, ensuring high quality of data, transforming the data into formats compatible with the computational models, and then finally performing the modeling. This process typically takes several days to months. The ultimate vision of Parekh (2005) is to build intelligent and powerful environmental information systems that will enable efficient data sharing and integration mechanisms.

GC3Pie (Maffioletti and Murri 2012; GC3Pie 2012) is a Python framework that aims to orchestrate the execution of external commands over different computing resources (such as a Sun/Oracle/Open Grid Engine cluster, the Swiss National Distributed Computing Infrastructure SMSCG, OpenStack Cloud, ARC-based computational grid, etc.). It is a flexible framework that allows the implementation of command line driver scripts (in the form of Python object classes) that can be customized easily by overriding specific object methods. GC3Pie also conceptualizes the executed applications but using plain programming language (i.e., you describe your application using a set of Python classes which can be extended and specialized). The tool was designed to coordinate the execution of independent applications meaning that it is used to steer the computation, not to perform it. The description of application in a programming manner offers many advantages, but with the drawback of a certain complexity as not all users have programming capabilities. Our solution tries to simplify things as much as possible for the user, by allowing non-specialists in programming to create a simple conceptual model of the executed application.

Klischewski and Wetzel (2012) present an interesting approach in workflow management area by introducing a flexible vision for heterogeneous workflow networks. The idea is to redefine the workflow management to meet today's challenges. The process execution realized based on predefined process patterns and resource relations ("processing by definition") is replaced by a process execution driven by recurrent process evaluation and service contracting ("process by contract"). This approach supports decentralized resource management through dynamically interrelating services and contracting resources as services during workflow execution.

Based on our experience in environmental science applications and on our research done in this area, and also taking into account the previous discussed works done around the topics of conceptual modeling of a phenomena and execution of environmental applications on parallel and distributed infrastructure, we are formulating in the following section a proposition for a simple and efficient conceptual model of an environmental application in general. The proposed model can be applied to any type of application but the parser used to extract the information was developed specifically for environmental science applications, that is it takes into account the specific environmental input and output data, as well as the algorithms used in this field.

Conceptual modeling of an environmental application

The development of our conceptual model was mainly based on the experience that we have gathered in executing different environmental applications (see below) on different computing infrastructures (Clouds, Clusters and Grids). After analyzing the characteristics and the behavior of environmental applications, while executing them on several computing infrastructures, we came up with a solution to flexibly describe, in a conceptual way, a general environmental application. This conceptual model will be later integrated in a methodology that will allow scientists to easily map their applications on different computing infrastructures.

To narrow a little the large area of environmental science applications, we have started our research mainly through hydrological modeling but this does not limit the applicability of the proposed methodology to this research area. Flood and drought forecasting, water management, and prediction of the impact of natural and human induced changes in hydrological cycle are just a few examples in which distributed hydrological models can be very useful. As many other environmental applications, these models have to simulate a large variety of physical processes that lead not only to a high complexity but also to a high degree of parameterization (Silvestro et al. 2013). Hydrological models have evolved a lot in the past decade, both because of the exponential development of computational capacities and because of the progress of Earth observation techniques that allow one to access large amounts of data readily available.

In the future we will consider also global and regional climate models (Dai et al. 2001; Wang et al. 2015) in our experiments, as they also pose diverse challenges regarding the storage and the computational resources.

In what follows we briefly present the hydrological models used in our study.

Continuum (Silvestro et al. 2013) is a distributed and continuous hydrological model that aims at balancing the necessity for a complete description of physical processes with the goal of avoiding over-parameterization. This means that special attention is given to reducing as much as possible the parameterization of the physical processes (so that land information can be extensively used as a constraint to parameter calibration) but at the same time, the model indents to maintain the necessary details of all the terms of the hydrological cycle. The model was designed to be implemented in different contexts but especially on data-scarce environments (with no stream flow data). It has been used notably in the context of the Global Flood model for the UNISDR/UNEP Global Assessment Report (see Global Assessment Report on Disaster Risk Reduction—GAR 2013).

SWAT (Soil Water and Assessment Tool, SWAT 2009) is a physically-based hydrological model used for simulating different physical processes and predicting the impact of land management in large, complex watersheds, with varying soils, land uses, and management conditions. Like most of the other hydrological models, SWAT has to be calibrated first for obtaining meaningful results. The execution of SWAT hydrological models usually involves a large set of input and output data and a large number of simulations for performing model calibration on many parameters. This implies the necessity of large storage and computational resources.

Experiments

We performed several experiments with the presented applications:

Execution of a test SWAT hydrological model calibration on different types of parallel and distributed infrastructures: Grid (gLite middleware), Cluster and Cloud (different instances of OpenStack and Windows Azure)

SWAT execution on cloud: OpenStack and Windows Azure

The testing steps on these infrastructures are as follows:

- Prepare the necessary SWAT input files and pack them in an archive
- Upload the input archive in the Cloud storage
- Launch the necessary number of virtual machines (VMs) (depending on the performed SWAT use case), with a predefined image and Linux flavour
- On each instantiated VM, execute a script that:
    - copies the input archive locally, from the Cloud storage
    - executes the SWAT model on this input
    - retrieves and copies the results back into the storage

For Windows Azure execution, we have developed a program that starts automatically a given number of Linux VMs on Azure. Upon starting, each VM runs a script that starts the execution as described above. On OpenStack, we have executed the tests on two instances. The particularity of one instance, in executing the above mentioned SWAT calibration steps, was that the input data was stored on a proxy machine and we have copied the input data, to each launched VM, using multicast (UDPCast software). This approach reduced significantly the download time. On the second instance, the execution was performed using the boto library [A Python interface to Amazon Web Service—boto (2015)] to access the data from and to S3 (2006)—Amazon Simple Storage Service.

SWAT execution on the "Baobab Cluster" of University of Geneva

The execution on this infrastructure was done using SLURM (2003) workload manager. The steps are quite similar with the ones performed in the Cloud except that the input files were placed in the common file system and instead of launching VMs, we have launched jobs on individual nodes in the cluster.

SWAT execution on Grid running gLite middleware

Tests for executing the calibration of different instances of SWAT model have been performed in the framework of the enviroGRIDS (2009) project. In our case, the execution steps differ from those performed in the Cloud in that the input data is uploaded on a Storage Element instead of a Cloud Storage and the jobs are launched in the Grid using utilities such as Ganga [Gaudi/Athena and Grid Alliance—Ganga (2009)—CERN] and DIANE [Distributed Analysis Environment—DIANE (2007)] instead of starting VMs. Apart form this, the execution flow remains the same. A detailed description of these experiments is found in (Rodila et al. (2012)).

Execution of the global flood model for the UNISDR/UNEP Global Assessment Report

The Global Flood model contains two procedures: downscaling the data and execution of the Continuum hydrological model, procedures which were executed on 13 out of 30 geographic areas (domains) covering the entire surface of the Earth. The execution was done on the Baobab cluster, provided by University of Geneva, using the SLURM workload manager, and the tests were performed using different distribution techniques.

Gridification of OGC web services

The OGC [Open Geospatial Consortium OGC (1994)] Web services (OWS) are Geospatial services used to exchange information in an interoperable and efficient way over a distributed environment. We have made several test on these services executed over the Grid infrastructure [Grid Middleware—gLite (2002)], with a varying number of features in the database (amount of data) and a varying request complexity (number of performed service requests). These tests were done during the enviroGRIDS project and they proved the existence of a complexity boundary for the execution on each computing background. Depending on the level of complexity of the model, the efficiency of the execution varies on different computational platforms. For a detailed description of the experiments and obtained results, see (Rodila and Gorgan 2012).

The performed practical experiments and the knowledge gathered after reviewing the scientific literature in this area formed together the starting point and the foundation on which our conceptual model, for describing a general environmental application, was built. The conceptual description contains specific details of the mapped application, such as: name, description, initial and intermediary inputs, outputs, executable processes, cost associated with each execution, etc. All these details are stored in a file and are used not only in determining the structure of the application but also the execution flow (i.e. control flow, specifying the order of the activities to be executed) and the data flow (specifying the input and the output data for each activity/task to be executed). Having this information in a common standard way is a step forward to automatize the mapping of applications on different computing infrastructures.

The execution flow of the application actually describes a workflow that is composed by connecting multiple tasks according to their dependencies. In general, a workflow can be represented as a Directed Acyclic Graph (DAG) or a non-DAG in which the nodes are execution tasks and the edges represent the communications lines between these tasks. In DAG workflows, the tasks can be structured as sequential tasks, parallel tasks or choice tasks (Costan 2010). The sequential tasks (or sequence) can be seen as an ordered series of tasks in which a task starts only after the previous one has completed. Parallel tasks are performed concurrently, while choice tasks are executed at runtime only when certain conditions are fulfilled. Using our experience in environmental applications, there are a lot of cases in which a set of tasks have to be executed several times, such as the calibration of a hydrological model for example, in which the process is executed in a large number of iterations but with different input parameters. In non-DAG workflows, besides the above-mentioned structures, we can also have iterations structures in which a section of tasks in the workflow can be repeated in an iteration block (i.e. loop). Using these entire structure types we can compose/decompose very complex workflows for an application. The proposed conceptual model has to be general enough to cover all the execution use cases of an application. As this goal is quite hard to achieve as one cannot foresee all the possibilities that might appear, the conceptual model has to be flexible enough to allow the extensions of new flows if this is the case for certain particular applications. It must allow the execution of arbitrary control flows through dependency graphs, taking into account conditional executions, looping, error handling and recovering, etc.

Execution flows

Possible executions flows based on the identified workflow structure types:

Simple execution

The execution flow in this case (Fig. 2) is a simple one in which the user defines the input(s) that are entries for a single execution point producing some output(s). The inputs and the outputs can be of different types and can be specified in different formats.

Sequential execution

In this case (Fig. 3) the execution flow is modeled as a sequence of several executions. The execution of a step normally depends on the results of a previous execution. That is why synchronization has to be taken into consideration. Simple Execution is a particular case of Sequential Execution in which we only have one execution node.

Parallel execution

The execution flow in this case (Fig. 4) is composed of several executions that are run in parallel. Each step is independent and can be executed concurrently with the others. A "Simple Execution" is also a particular case in which we only have one execution node.

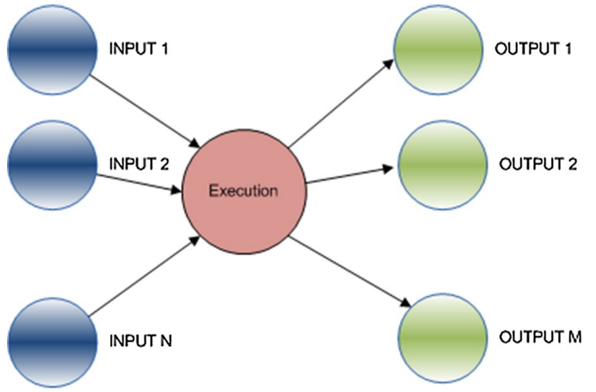

Fig. 2 Conceptual model—simple execution

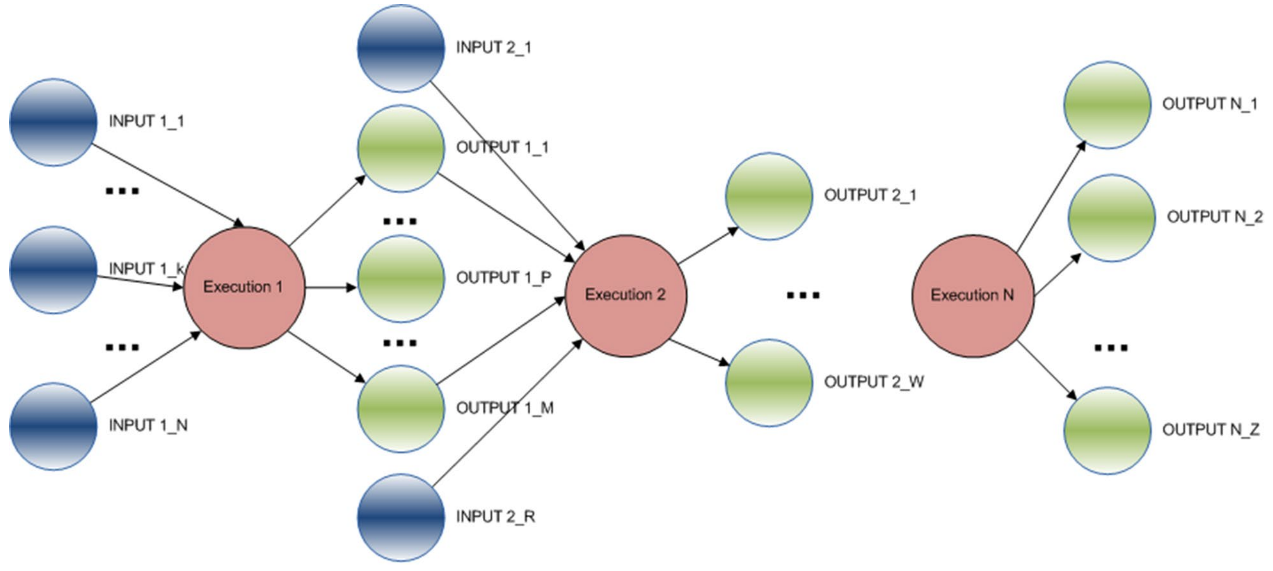

Fig. 3 Conceptual model—sequential execution

Composed execution

The composed execution (Fig. 5) has a complex structure in which one can include different types of executions: simple, sequential or parallel. This is useful if a user wants to save/use previously computed executions without having to define them again. All other mentioned cases: Simple Execution, Sequential Execution and Parallel Execution can be considered particular use cases of this type.

Loop execution

The execution flow in this case (Fig. 6) consists in executing the same module several times. The module can be composed of several types of executions or it can be one of the already presented types. This is useful when the same set of executions has to be repeated several times. An example of this case can be the calibration of a climatic model. The inputs can be the same set of parameters or a slightly different one, but the outputs are usually different.

The proposed conceptual model covers all the above use cases. Using this model, a user can easily describe the structure and the execution flow (workflow) of his/her application. The actual execution of this model can be done through instantiation, i.e. binding the workflow tasks to specific resources (different for each application and for each execution use case). The conceptual model provides a flexible way of specifying an environmental application without being concerned with the low-level implementation details. The tasks in the conceptual model can be mapped on any executable platform at run time using mapping mechanism. In a concrete model, the specific resources of the applications are bind to the tasks. At this level, new tasks may also appear, related to data movement between tasks and/or repositories. The concrete model can be generated (either full or partial) either before or during the execution.

The steps to complete the conceptual model for a specific application are the following ones:

1. Define all the inputs of the application. Here the user has to specify for each input what is its type, how it can be accessed and to which execution task it belongs. The inputs can either be initial inputs or they can also be outputs from other executions.
2. Define all the outputs of the application by specifying as well their type, where should they be stored and from what execution they come from.
3. Define the executions tasks within the applications. Depending on what inputs are associated with a specific task, we can decide if the task will be executed in parallel or sequentially with other tasks. The loop executions are specifically described within a loop tag in the file in which the user has to specify what

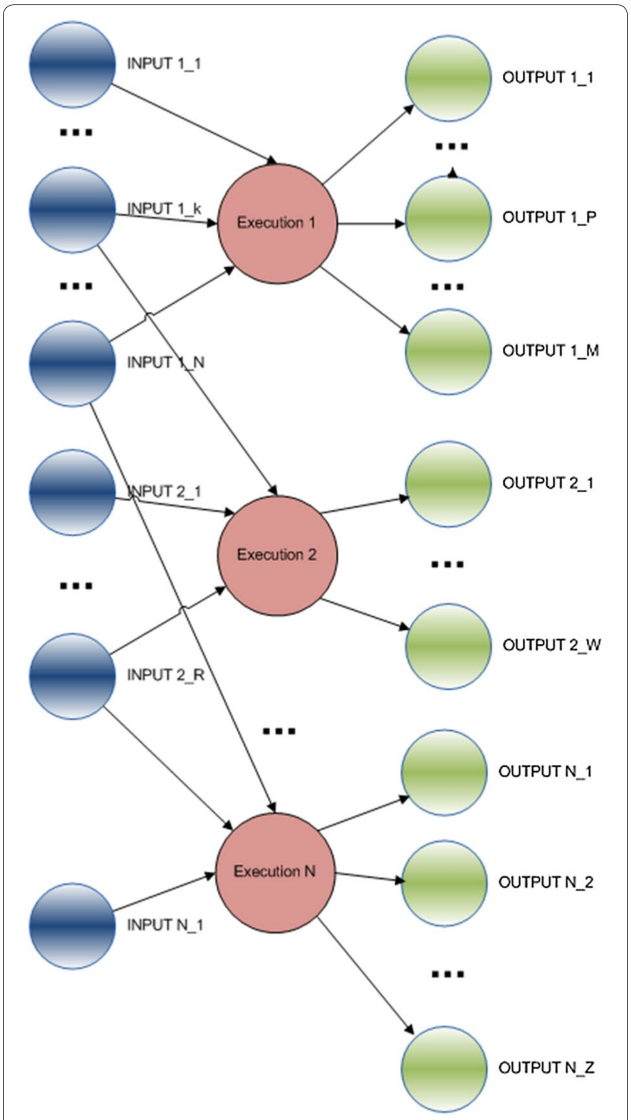

Fig. 4 Conceptual model—parallel execution

executions are part of the loop and how many times the loop is repeated.

4. Define the Composed Executions if any. At this point the user can specify the path to an already defined conceptual model of a previous application.

The parser intended to process the defined conceptual model will explore all these information. The conceptual description can be used in general for any type of application so far but the specificity of the environmental science field will be modeled in the parser component, as this is the level where the differences appear concerning especially the input and output data, as well as the algorithms used to handle environmental data.

SWAT use case: conceptual instantiation

The SWAT hydrological model allows a number of different physical processes to be simulated in a watershed. The inputs of the SWAT model are specific information about weather, soil properties, vegetation, topography, and land management practices of the watershed. To apply and successfully use hydrological models, both good calibration and good prediction analysis are required. Calibration is the process of estimating the model parameters to obtain a better system that closely resembles the system that the model intends to represent. SWAT calibration involves:

- large set of input and output data
- high number of simulations
- time constrain (in certain cases): decision makers may need to obtain near real time output from the SWAT model to be able to make reliable and meaningful predictions and to deal with emergency environmental disasters.

- The parallelization of SWAT calibration, using SUFI2 (Sequential Uncertainty Fitting) algorithm (Abbaspour et al. 2007), is accomplished at the simulation level by simply executing several SWAT runs with different parameters. Each simulation runs the same SWAT model but with different input parameter values. The execution of an iteration consists in performing three important phases:

Pre-processing phase is executed only once for each iteration. In this phase, the input parameter values are generated randomly but within a specific range for each simulation, based on the parameters intervals and using Latin hypercube sampling. A combination of parameters is generated for each simulation. Each simulation in an iteration represents one task which is executed on a node in a certain computing platform. Depending on

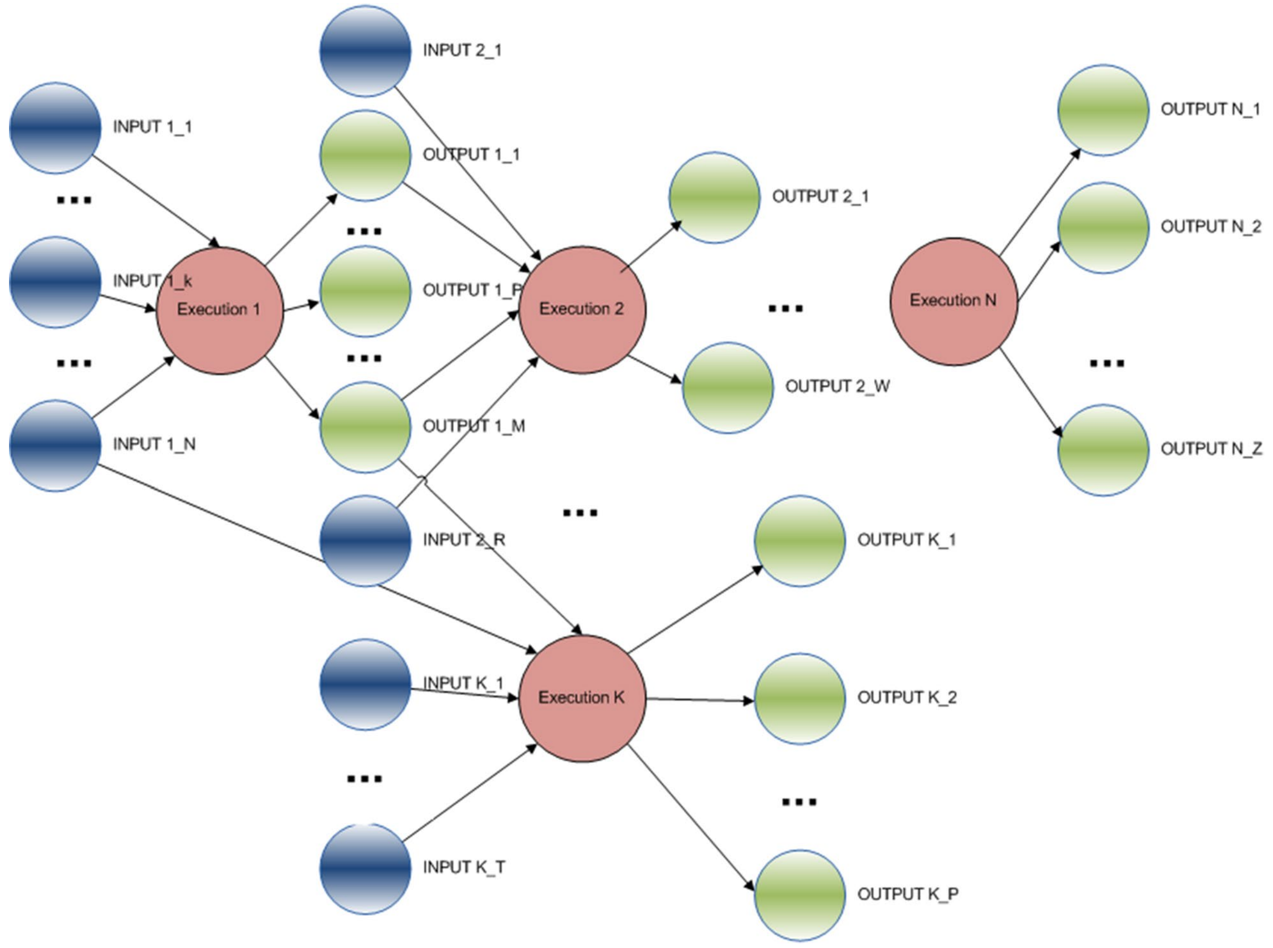

Fig. 5 Conceptual model—composed execution

the available resources, a node can execute one or more tasks (i.e. simulations) from an iteration.

Execution phase each simulation is run on different nodes.

Post-processing phase The output of each simulation is retrieved and processed after all the simulations have finished.

The SWAT model instance used in our experiments was developed in the EU/FP7 enviroGRIDS (2009) (Black Sea Catchment Observation and Assessment System supporting Sustainable Development) project and uses high-resolution data to model the Black Sea catchment. This large hydrological model was built using the SWAT2009 program (SWAT 2009) and covers the Danube River Basin. The Danube River flows for a distance of 2826 km and the model covers an area of 801,093 km^2. The region was divided into 1224 smaller sub-basins and the simulation period was set to 5 years. A detail description of this model and a comparative execution on Multicore and Grid infrastructure can be found in (Rodila et al. 2012).

The application flow of this hydrological model can be described using the proposed conceptual model in the following way:

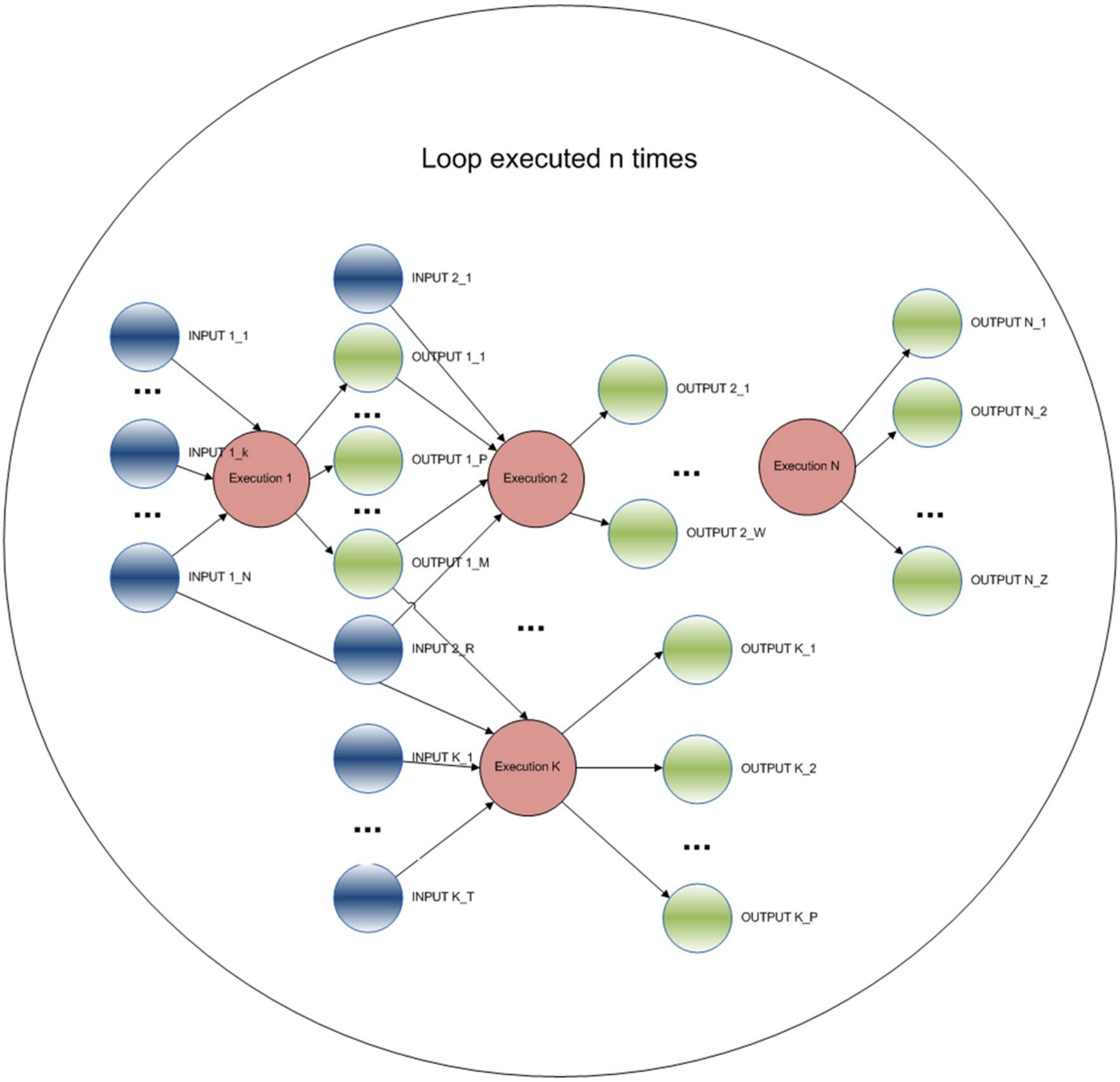

Fig. 6 Conceptual model—loop execution

```
<EnvironmentalApplication>
        <Input idDB="1" name="BSInst11" description="initial IO
            directory of the SWAT model, instance 11">
                <PreConditions>
                </PreConditions>
                <LocalPath path="/home/denisa/SWAT/BSInst11/InOut/" />
        </Input>
        <Execution idDB="1" name="pre-processingBSInst11" description=
            "preprocessing phase of the SWAT model, instance 11">
                <PreConditions>
                        <Input idDB="1"/>
                </PreConditions>
                <URLPath="/home/denisa/SWAT/BSInst11/Executable/Pre-
                    processing/"/>
                <PostConditions>
                        <Output idDB="1", name="BSInst11.tgz"
                            description="generated calibration archive
                             of the SWAT instance 11" />
                </PostConditions>
        </Execution>
        <Execution idDB="2" name="iterationBSInst11" description="one
            iteration of the SWAT model, instance 11">
                <PreConditions>
                        <Input idDB="1"/>
                </PreConditions>
                <URL path="/home/denisa/SWAT/ BSInst11/Executable/
                    Iteration/"/>
                <PostConditions>
                        <Output idDB="X", name="BSInst11-ItX.tgz"
                            description="Iteration X result archive" /
                            >
                </PostConditions>
        </Execution>
                <Loop idDB="1" name="BCInst11Calibration" description=
                    "calibration execution phase of the SWAT model,
                    instance 11">
                <Execution idDB="2"\>
                <Iterations nr="100"/>
                <PostConditions>
                        <Output idDB="Y", name="BSInst11-OUT.tgz"
                            description="Calibration result archive" /
                            >
                </PostConditions>
        </Loop>
        <Execution idDB="4" name="post-processingBSInst11" description
            ="post-processing phase of the SWAT model, instance 11">
                <PreConditions>
                        <Input idDB="Y"/>
                </PreConditions>
                <PostConditions>
                        <Output idDB="", name="BSInst11-final.tgz"
                            description="final results of the SWAT
                            instance 11" />
                </PostConditions>
        </Execution>
</EnvironmentalApplication>
```

Listing 1 Conceptual Model Instantiation for SWAT

Conclusions and future work

The challenges that the environmental science community is facing are immense in light of the current global environmental changes occurring at various scales. Part of the solution is to be able to efficiently analyze the growing set of available environmental "Big Data", and this has become possible recently both due to the increasing capabilities of computational resources (and hardware advances) and to the availability of tools, algorithms and techniques used to take advantages of these resources. But it often remains difficult to easily integrate environmental applications with high performance computing resources. To ease that step, we have introduced there a solution to easily model environmental applications and to facilitate their integration with different parallel and distributed infrastructures.

Taking into account the growing need for computational speed, storage and scalability that environmental applications demand, the users usually tend to use—or to switch between—more than one execution platform for obtaining the necessary resources. To be able to easily switch between these platforms we have proposed an application conceptual model that hides the complexity of different types of environmental applications and that provides an easy and flexible way to map an environmental application to an execution platform. Using this model, a user can describe the structure, the data flow, as well as the execution flow (workflow) of the application. The model is built to cover different execution flows, such as: simple, sequential and parallel executions, as well as composed and loop executions. It also allows the definition of a new type of execution if necessary and the re-usage of an existing one. The actual execution of the model is done through instantiation, i.e. binding the described concepts to specific application resources, which are different for each application and for each execution use case. Using this approach, we managed to conceptualize an application and to disconnect it from the low-level implementation details of an execution environment.

As specified before, the proposed conceptual model is a key component in a general methodology for easily and efficiently porting environmental applications on different parallel and distributed infrastructures.

Following the availability of this conceptual model, the next step in our future work would be to develop a scheduling and execution component that would allow the user to easily submit an instantiated conceptual model (a specific environmental application bind to the conceptual model) to one or more available computing infrastructures. This component will also estimate which of the available computing infrastructure is more appropriate for execution based on several criteria such as: number of parallel jobs, user preferences, history, etc.

To be able to evaluate and verify the correctness and the interoperability of the proposed conceptual model as well as the efficiency of an application execution based on the conceptual model, we also have to develop a set of metrics and a validation component. The development of this methodology is still a work in progress but we have the confidence that it will bring important contributions to the urgent need of environmental community in using parallel and distributed infrastructures for better processing and analyzing the large amounts of data that is collected daily. The proposed conceptual model and its mapping to different computational infrastructures will allow many environmental applications to be more efficiently used. The hope is therefore that better-informed decision-making will follow, responding more effectively to the changes that are threatening our environment and the society at large.

Authors' contributions

The work presented here was carried out in collaboration between all authors. DR designed and conducted the research work including data preparation, concept development, results analysis and drafted the manuscript. Both NR and DG contributed during the conception and design of the proposal and revised it critically for important intellectual content. They have been supervising the progress of the research and providing guidance: NR on the environmental issues and DG on the technical side. All authors read and approved the final manuscript.

Author details

[1] Institute for Environmental Science, enviroSPACE Lab, University of Geneva, Bd Carl-Vogt 66, 1211 Geneva, Switzerland. [2] CGIS Lab, Computer Science Department, Technical University of Cluj-Napoca, G. Baritiu 28, 400027 Cluj-Napoca, Romania.

Acknowledgements

This research was partially founded by the SCIEX (Scientific Exchange Programme NMS.CH) scholarship (http://www.sciex.ch) through the project 12.206.

Competing interests

The authors declare that they have no competing interests.

Received: 28 August 2015 Accepted: 12 November 2015
Published online: 25 November 2015

References

Abbaspour KC, Vejdani M, Haghighat S (2007) SWAT-CUP calibration and uncertainty programs for SWAT. In: Oxley L, Kulasiri D (eds) PMODSIM 2007 International Congress on Modelling and Simulation. Modelling and Simulation Society of Australia and New Zealand: December 2007, pp 1596–1602

Assuncao MD, Calheiros RN, Bianchi S, Netto MAS, Buyya R (2015) Big Data computing and Clouds: trends and future directions. Special Issue on Scalable Systems for Big Data Management and Analytics. J Parallel Distrib Comp 79–80:3–15

Boto (2015) A python interface to amazon web service. https://boto.readthedocs.org

Copernicus. Big Data workshop, Bruxelles, 2014. http://www.copernicus.eu/library/detail/212

Costan A (2010) Autonomic Behavior of Large Scale Distributed Systems based on Monitoring Information. PhD thesis, Polytechnic University of Bucharest, Computer Science Department

DIANE. Distributed analysis environment, 2007. http://it-proj-diane.web.cern.ch/it-proj-diane/

Dai X, Meehl GA, Washington WM, Wigley TM, Arblaster JM (2001) Ensemble simulation of twenty-first century climate changes: business-as-usual versus co2 stabilization. Bull Am Meteorol Soc 82(11):2377–2388

Davies I, Green P, Rosemann M, Indulska M, Gallo S (2006) How do practitioners use conceptual modeling in practice? Data Knowl Eng 58:358–380

Demchenko Y, Zhao Z, Grosso P, Wibisono A, De Laat C (2012) Addressing Big Data challenges for scientific data infrastructure. In: IEEE Computing Society, editor, IEEE 4th International Conference on Cloud Computing Technology and Science (CloudCom 2012), pp 614–617

Dozier J (2009) The fourth paradigm: data intensive scientific discovery. In: Tolle K, Tansley S, Hey T (eds) The emerging science of environmental applications. Microsoft Research, Redmond, pp 13–19

ESFRI. e-irg report on data management, 2009. http://ec.europa.eu/research/infrastructures/pdf/esfri/publications/esfri_e_irg_report_data_management_december_2009_en.pdf

Environmental Data Analysis. Course, 2005. http://www.umass.edu/landeco/teaching/ecodata/schedule/environmental.data.pdf

enviroGRIDS. European project, 2009. http://www.envirogrids.net/

Fortuin KPJ, Van Koppen CSA, Leemand R (2011) The value of conceptual models in cooping with complexity and interdisciplinarity. Environ Sci Educ Biosci 61(10):802–814

GAR. Global assessment report on disaster risk reduction, 2013. http://www.preventionweb.net/english/hyogo/gar/2013/en/home/GAR_2013/GAR_2013_2.html

GC3Pie. Software, 2012. https://code.google.com/p/gc3pie/

GEO (2005) Geoss: 10-year implementation plan reference document. ESA publications division, 2005. https://www.earthobservations.org/documents/10-Year%20Plan%20Reference%20Document.pdf

Ganga. Gaudi/Athena and Grid alliance-CERN, 2009. http://ganga.web.cern.ch/ganga/

Giuliani G, Nativi S, Lehmann A, Ray N (2012) WPS mediation: an approach to process geospatial data on different computing backends. Comp Geosci 47:20–33

Giuliani G, Ray N, Schwarzer S, De Bono A, Dao H, Peduzzi P, Beniston M, Van Woerden J, Witt R, Lehmann A (2011) Sharing environmental data through GEOSS. Int J Appl Geospatial Res 2(1):1–17

gLite. Grid middleware, 2002. http://grid-deployment.web.cern.ch/grid-deployment/glite-web/

GreenLand. Software, 2011. http://cgis.utcluj.ro/applications/greenland

gSWAT. Software, 2011. http://cgis.utcluj.ro/applications/gswat

IPCC. Climate change 2014 synthesis report, 2014. http://www.ipcc.ch/pdf/assessment-report/ar5/syr/SYR_AR5_LONGERREPORT_Corr2.pdf

ISO191xxx. Series of geographic information standards, 2003. www.wmo.int/pages/prog/www/TEM/ET-WISC-I/ISO_191xx.doc

Klischewski R, Wetzel I (2012) Processing by contract; turning the wheel within heterogeneous workflow networks. Business Process Manag J 11(3):237–354

Kumar A, Bawa S (2012) Distributed and Big Data storage management in Grid computing. Int J Grid Comp Appl 3(2):19

Maffioletti S, Murri R (2012) GC3PIE: a python framework for high-throughput computing. In: Proceedings of EGI Community Forum 2012, EMI Second Technical Conference, Munich

Maity S, Bonthu SR, Sasmal K, Warrior H (2013) Role of parallel computing in numerical weather forecasting models. In: IJCA Special Issue on International Conference on Computing, Communication and Sensor Network CCSN2012, Vol 4, pp 22–27

Mylopoulos J (1992) Conceptual modeling, databases, and case: an integrated view on information systems development. In: Loucopoulos P, Zicari R (eds) Conceptual modeling and Telos. McGraw Hill, New York. pp 49–68

Nativi S, Mazzetti P, Geller G (2013) Environmental model access and interoperability: the GEO model web initiative. Environ Model Softw 39:214–228

Nativi S, Mazzetti P, Santoro M, Papeschi F, Craglia M, Ochiai O (2015) Big Data challenges in building the global earth observation system of systems. Environ Model Softw 68:1–26

OGC. Open geospatial consortium, 1994. http://www.opengeospatial.org/

Parekh V (2005) Applying Ontologies and Semantic Web technologies to Environmental Sciences and Engineering, Master of Science. PhD thesis, University of Maryland, Baltimore County, 2005

Piegorsh WW, Bailer AJ (2005) Analyzing environmental data. John Wiley and Sons, England

Rodila D, Bacu V, Gorgan D (2012) Comparative parallel execution of SWAT hydrological model on multicore and Grid architectures. Int J Web Grid Serv 8(3):304–320

Rodila D, Gorgan D (2012) Geospatial and Grid interoperability through OGC services gridification. Int J Select Topics Appl Earth Observ Remote Sens 5(6):1650–1659

S3. Amazon simple storage service, 2006. http://aws.amazon.com/s3/

SLURM. Tool, 2003. http://slurm.schedmd.com/

SWAT. Soil and water assessment tool, 2009. http://swat.tamu.edu/

Silvestro F, Gabellani S, Delogu F, Rudari R, Boni G (2013) Exploiting remote sensing land surface temperature in distributed hydrological modelling: the example of the continuum model. Hydrol Earth Syst Sci 17(1):39–62

Sokolowski JA, Banks CM (1970) Modeling and simulations fundamental. John Wiley and Sons INC. Publication, Suffolk

Vrugt JA, Nualláin BÓ, Robinson BA, Bouten W, Dekker SC, Sloot PMA (2006) Application of parallel computing to stochastic parameter estimation in environmental models. Comp Geosci 32(8):1139–1155

Wand Y, Weber R (2002) Research commentary: Information systems and conceptual modeling—a research agenda. Inform Syst Res 13(4):363–376

Wang X, Huang G, Liu J, Li Z, Zhao S (2015) Rensemble projections of regional climatic changes over ontario, Canada. J Climat 28:7327–7346

Yang C, Huang Q (2013) Spatial Cloud computing: a practical approach. CRC Press, Boca Raton