Journal of
Computer Science & Systems Biology

Petrenko, J Comput Sci Syst Biol 2015, 8:5
DOI: 10.4172/jcsb.1000201

Review Article Open Access

Collaborative Complex Computing Environment (*Com-Com*)

Petrenko AI*

Department of System Design, Institute of Applied System Analysis, National Technical University of Ukraine, Kyiv Polytechnic Institute, Kiev, Ukraine

Abstract

The *Com-Com* is the user-centric environment which provides researchers with tailored frameworks to support their computational needs. It addresses existing and new user communities in both research and commercial fields. Technically the *Com-Com* provides dynamic infrastructure, dynamic service provision and user-driven application development across the domains. End users can create new applications for solving their computational tasks easily by combining ready-made interdisciplinary services available in the networked Repository and incorporate their own functionalities. Since services may be offered by different enterprises and communicate over the network, they provide an advanced distributed computing infrastructure for both intra- and cross-enterprise application integration and collaboration. The approach in hands potentially opens a door to rapid creating applied software for Exaflops HPC and Exabytes data.

Nowadays the *Com-Com* can provide applications developing in the life science, environment, engineering, physics, computational chemistry, medicine, data mining research by collecting already existing web-services been developed by different research communities EGI, Flatworld, FI-WARE, SAP, ESRC. The goal of the *Com-Com* is to present an open environment of applied computing services and to encourage researchers across Europe to participate in its extending, interchanging or improving.

The *Com-Com* stack presents flexibility enabling users to form dynamic teams, dynamic collections of cross domain services and dynamic infrastructure to run the services on. The *Com-Com* may enhance the capabilities of research organizations who lack resource both in human and technical terms by better integrating researches across international scientific communities with the final aim to strengthen the EU research base.

Keywords: SOC (service-oriented computing); SOA (service-oriented architecture); Web-services; Services composition; User-driven applications' development; Application design platform; Distributed modeling; Services workflow management.

Introduction

Collaborative Complex Computing arises from and is intended to address the specific requirements of a large amount of research and industrial enterprises who are engaged in processing scientific data and performing time-consuming mathematical experiments during their scientific and applied research. Many scientific and engineering fields need powerful tools that meet the needs of a quite wide range of customers in the means of mathematical modeling and collective computing research support, enabling collaboration of distributed group of partners – the providers and consumers of computing resources and data processing solutions. Providing effective ways for the distributed user groups to compose distributed workflows representing the sequence of data processing procedures needed to solve their problems – this is what Collaborative Complex Computing is about.

The aim of the *Com-Com* is to provide an integrated environment that supports the collaborating engineering research and allows its users to create and debug the structure of mathematical experiments or data processing workflows that are selected for execution on Grid resources. *Com-Com* concept is the available online intelligent multidisciplinary research gateway combining.

A) Inhabited information space where both open and private user communities can easily communicate and develop their domain-specific expert knowledge on the base of new emerging design paradigms and best practices.

B) User-driven adaptive tools and methods for distributed data processing and mathematical experiments, their modeling and optimization in a user-friendly environment using the free resources of e- infrastructure. End users can create new applications for solving their tasks easily by combining ready-made services available in the networked Repository and incorporate their own functionalities.

C) Web-services Repository with Task Solving Supporting (Application Specific) Services, which are corresponding to loosely coupled stages and procedures for complex tasks of data processing and modeling, and Environment Supporting (Generic) Services, which are responsible for service management and hosting (Figure 1). The list of offered Task Solving Supporting (Specific) Services covers a significant share of the possible user needs in scientific and applied research, such as: experimental data search and access, collection and management, data analysis, remote modeling of processes (objects) of different physical nature, etc.

D) Semantics-aware mechanism to find proper web-services and target execution resources for the best integration solution of the specific user-defined problem with respect to a quality-of-service.

E) The truly open environment and a set of open services that will allow researchers, service providers, small and medium engineering

***Corresponding author:** Anatoly I. Petrenko, Department of System Design, Institute of Applied System Analysis, National Technical University of Ukraine, Kyiv Polytechnic Institute, Kiev, Ukraine, Tel: +38044-2364166; E-mail: tolja.petrenko@gmail.com

Received August 11, 2015; **Accepted** August 25, 2015; **Published** August 27, 2015

Citation: Petrenko AI (2015) Collaborative Complex Computing Environment (*Com-Com*). J Comput Sci Syst Biol 8: 278-284. doi:10.4172/jcsb.1000201

enterprises and other organizations to develop their custom application software satisfying their needs while still being open and innovative.

Today there is no well recognized user-driven applied platforms with support of arbitrary mathematical experiments during scientific and applied research that can offer all of mentioned above. *Com-Com* stands for a new technology and methodology for planning and modeling of mathematical experiments, and it can offer the following features, Figure 1.

- Execution of composite computing tasks of arbitrary complexity to support collective research via the Internet.
- Promote a high scientific and technical level of research with open knowledge base.
- Literative optimization of results obtained during calculations.
- Reducing terms of scientific and applied research and subsequent development work with intensive workflows, tools, data and knowledge reuse in mind.
- Improving the quality of scientific and technical documents while productivity growth in scientific organizations and SMEs.
- System integrating stages of scientific and applied research, development and technological preparation of production.

Com-Com enhances future competitiveness by strengthening its scientific and technological base in the area of Experimenting and Data Processing, makes public service infrastructures and simulation processes smarter i.e., more intelligent, more efficient, more adaptive and sustainable. It can create extended new inter-disciplinary collaborations, new research alliances with European researchers in order to combine joint knowledge and experience and exploit synergies in user-driven Applications development. *Com-Com* utilizes services as constructs to support the rapid, low-cost and easy composition of distributed applications by end-users. The computing is divided into separate loosely coupled stages and procedures for their subsequent transfer to the form of standardized specific (application support) services, (ASS) at infrastructure and data / user federation level. The offered list of such services covers a diverse range of application domains and the project establishes a point on which this range can be expanded upon.

Activity Overview

The *Com-Com* concept is based on SOC (Service - Oriented Computing) distributed applications development by means of the composition of services [1-11]. Service-Oriented Computing (SOC) is the paradigm for distributed computing that utilizes services as fundamental elements (services) for application development. It represents a new approach in application development moving away from tightly coupled monolithic software towards software of loosely coupled, dynamically bound services. End-users need the support to build new systems easily by incorporating functionalities of available systems and services. Computing procedures, being used in different branches of science and technology, are invariant in their nature. That why they can be used by different customers in their particular needs.

Services implement functions that can range from answering simple requests to executing sophisticated research processes requiring peer-to-peer relationships between possibly multiple layers of service consumers and providers. The delivery of software for complex collaborative computing as a set of distributed services can help to solve problems like software reuse, deployment and evolution. The “software as a service model” will open the way to the rapid creation of new value-added composite services based on existing ones. Although service-oriented computing in cloud computing environments presents a new set of research challenges, their combination provides potentially transformative new opportunities.

Pioneering work in mathematical SOC has been done as part of the ADaM (Algorithm Development and Mining) toolkit which was originally developed by the University of Alabama in Huntsville (UAH) with the goal of mining large scientific data sets for geophysical phenomena detection and feature extraction, and has continued to be expanded and improved [12]. ADaM includes not only traditional data mining capabilities such as pattern recognition, but also image processing and optimization capabilities, and many supporting data preparation algorithms that are useful in the mining process. ADaM provides technology that allows users to locally define analysis workflows that can be executed on data residing in online repositories. The NASA project, called Mining Web Services (MWS), is enabling ADaM capabilities for use in a distributed web service environment. This redesign also allows the algorithms in ADaM to be easily packaged as grid or web services and is being extensively used by different research

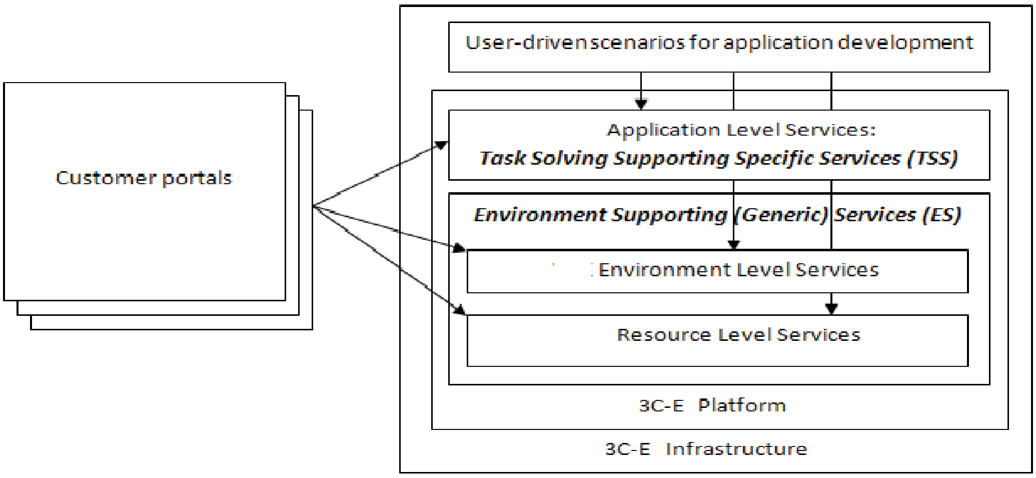
User-driven scenarios for application development
Application Level Services:
Task Solving Supporting Specific Services (TSS)
Environment Supporting (Generic) Services (ES)
Environment Level Services
Resource Level Services
3C-E Platform
3C-E Infrastructure
Customer portals

Figure 1: *Com-Com* general structure

groups [13].The rest of this paper is organized as follows: Section 5 gives an overview of main elements SOC of the computational platform, web-services as the best component of SOC; Section 6 describes computational web-service examples; Section 7 presents Web-services Management; Section 8 describes the prototyping example; Section 9 presents the performance of applied services in the SOC prototype and concludes this paper.

Main Elements of Service-Oriented Architecture of the Computational Platform

The above approach is incorporated in the service-oriented computational platform consisting (Figure 2).

This architecture characterized in that: its web-accessible, its functionality is distributed across the ecosystem of both web services from the *Com-Com* Repository and grid/cloud services (from e-infrastructure); it is compatible with adopted standards and protocols; it supports custom user analysis scenario development and execution; it hides the complexity of web-service interaction from user with abstract workflow concept and graphical workflow editor.

User interface provides the following functionality: authorization, graphical workflow editor, project artifacts browsing (input and output files management, simulation results visualizers etc.), task execution monitoring and others (Figure 3). The server-side part of the architecture has several layers to reflect the abstract workflow concept described above. First tier is the portal which organizes user environment: holds user data and preferences, controls user access, provides information support, organizes user interface. Its modules are also responsible for: abstract workflow description generation according to user inputs, passing this task description to lower architecture layers for execution, retrieving finished task results and storing all the project artifacts in the database.

The next tier is the workflow manager running on the execution server. It is responsible for mapping (with the help of service registry) the abstract workflow description to the concrete web services orchestration scenario expressed in the orchestrator-specific input language (like WS-BPEL for BPEL engines). It also initiates the execution of the concrete workflow with the external orchestrator, monitors its state and fetches the results. Concrete workflow operates with SOAP / REST web services representing the basic building blocks of system's functionality: data preparation and adaptation, simulation, optimization, results processing etc. Compute-intensive steps are implemented as grid/cloud services interacting with grid / cloud resources to run computations as grid / cloud jobs. Introduction of the new functionality to the system is accomplished through the registration of the new web or grid/cloud services.

The overall sequence of user scenario execution is as follows. User passes login procedure on the portal and accesses workflow editor. He may choose and setup the activities available in repository to compose (manually or automatically) the scenario workflow he wants to execute (Figure 3).

Then the execution phase is initiated by user. User task description is passed to workflow management service on execution server, where this abstract workflow is translated to the concrete one. Workflow manager parses the description and checks for errors, requests metadata from service registry and performs mapping from activity sequence to web service invocations sequence, described in one of the standard orchestration languages. Mapper unit of the workflow manager should arrange web services in correct invocation order according to abstract workflow, organize XML messages and variables initializations and assignments between calls, and provide the ways for run-time control (workflow monitoring, canceling, intermediate results retrieving etc.). Then this concrete scenario is executed by orchestrator.

When an orchestrator invokes service the latter initiates the

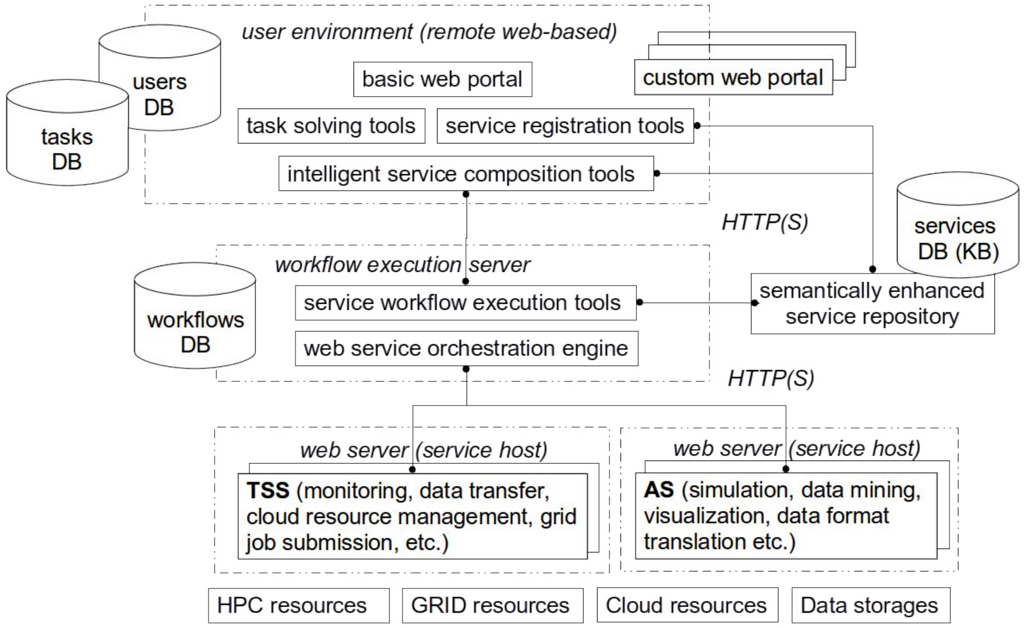
user environment (remote web-based)
users DB
tasks DB
basic web portal
custom web portal
task solving tools
service registration tools
intelligent service composition tools
HTTP(S)
services DB (KB)
workflow execution server
workflows DB
service workflow execution tools
semantically enhanced service repository
web service orchestration engine
HTTP(S)
web server (service host)
TSS (monitoring, data transfer, cloud resource management, grid job submission, etc.)
web server (service host)
AS (simulation, data mining, visualization, data format translation etc.)
HPC resources
GRID resources
Cloud resources
Data storages

Figure 2: Main elements of service-oriented architecture of the computational platform.

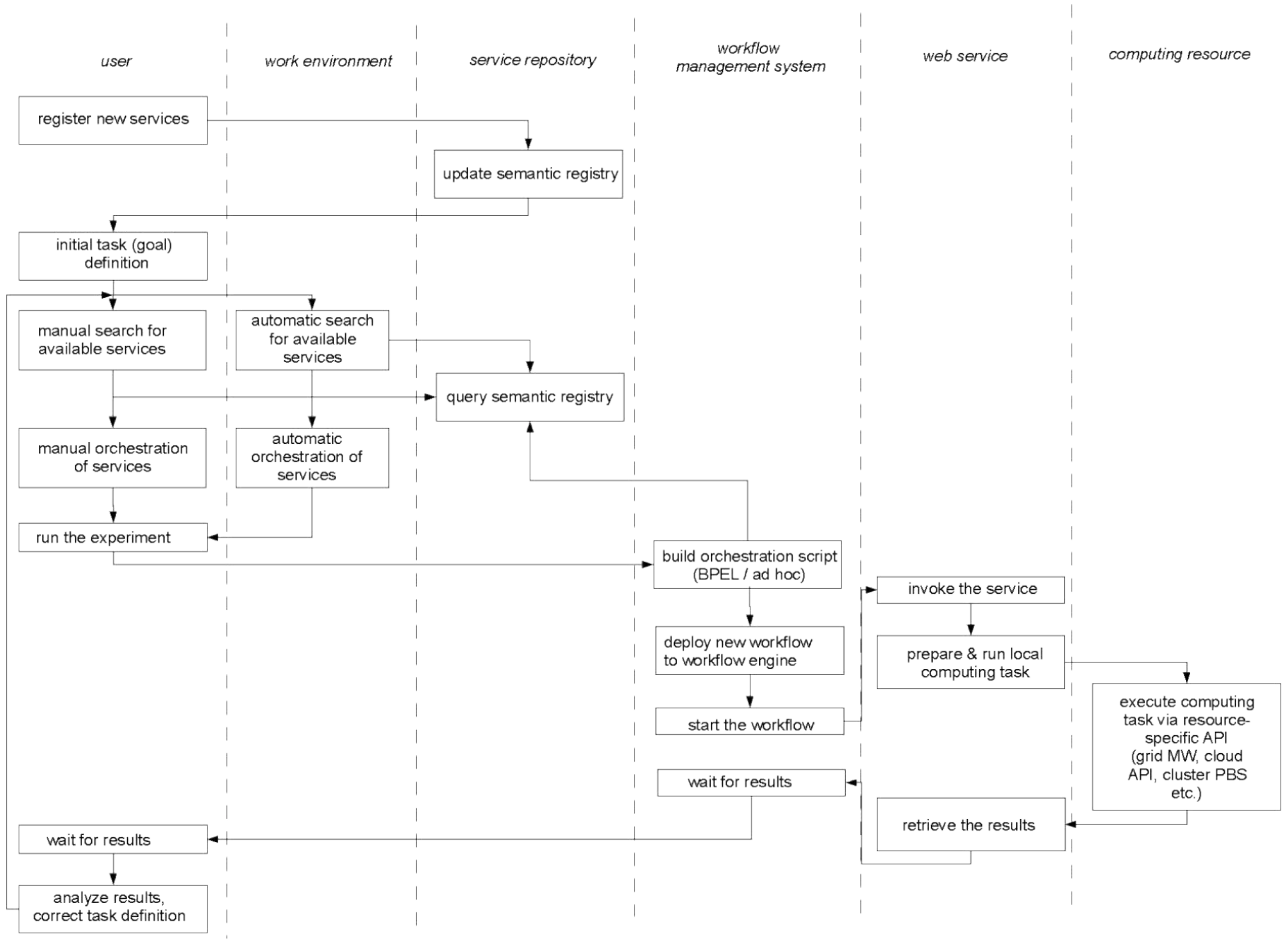
user
work environment
service repository
workflow management system
web service
computing resource
register new services
update semantic registry
initial task (goal) definition
manual search for available services
automatic search for available services
query semantic registry
manual orchestration of services
automatic orchestration of services
run the experiment
build orchestration script (BPEL / ad hoc)
invoke the service
deploy new workflow to workflow engine
prepare & run local computing task
execute computing task via resource-specific API (grid MW, cloud API, cluster PBS etc.)
start the workflow
wait for results
retrieve the results
wait for results
analyze results, correct task definition

Figure 3: The general procedure of user-driven building flexible computational routes.

submission of a job to resource: it prepares a job description and communicates with grid middleware to schedule and execute a job. The similar behavior is for HPC or cloud services (task preparation and execution via the specific API). User is informed about the progress of the workflow execution by monitoring unit communicating with workflow manager. When execution is finished user can retrieve the results, browse and analyze them and repeat this sequence if needed.

Computational Web Service Examples

If a large multidisciplinary and multinational Repository of Application support services is created, the end- users can tailor the services to their own personal requirements and expectations by incorporating functionalities of available services into large-scale Internet-based distributed application software. Typical scheme of a computational modeling experiment in many fields of science and technology has an invariant character and includes the following steps:

- Definition of the mathematical description of the experiment tasks (mathematical model). This is often done manually by researcher. However, it is possible to automate this process as a separate web-service. This service forms a mathematical model usually in the form of a system of nonlinear differential-algebraic equations based on a block diagram (structure) of a computational experiment in hand [14,15].
- The dimension of such model can be very large (some thousands of equations), and its structure is highly sparse. In its formation the library of individual blocks (procedures) descriptions are used. For example, it is possible to generate automatically a mathematical model of the selected data processing using its block diagram.
- If investigated processes (objects) have distributed nature and are governed by partial differential equations (PDE), it is possible to assemble their models also in the form of systems of first order ordinary differential equations (ODE). Otherwise the PDE can be numeral solved, applying the methods of finite differences (FDM) or finite elements (FEM) [16].
- The solution of mathematical model equations for stationary regime when its equations are transformed into a system of nonlinear algebraic equations. It is important to ensure the convergence of the solution, despite the ill conditional nature of the task.
- Solution of the mathematical model equations for dynamic regime in time domain with regard to the possible stiffness of these equations.

- The solution of the linearized mathematical model equations in frequency domain.
- Automatically detecting solution output parameters in the form of extreme values, consuming power, time delay and rise time (fall) of selected variables (for the time domain) or the transfer coefficients, bandwidth, resonant frequencies and quality factors (frequency domain).
- Determination of the solution output parameters sensitivity to value change of internal parameters associated with the description of the individual procedures (blocks), or the parameters of the environment in which objects been built on the results of the experiment are planned to explore.
- Multi-criteria optimization of the task solution output parameters in conditions of functional and parametric constraints.
- Statistical analysis and histogram building for solution output parameters taking into account the laws of distribution of internal parameter values.
- Estimates of the deviations of solution output parameters due to variations of internal parameter values: the worst case for the boundary values of the internal parameter deviations and statistical evaluation, taking into account the laws of distribution of these deviations.
- Inverse problem: determination of optimal tolerance of the internal parameter values for a given deviations of solution output parameters. The problem is solved by optimization procedures (deterministic or statistical by maximizing the ratio of output).
- Determination of the spectral composition of output variables of the experiment (the project) and assessment of their degree of distortion.
- Support for experiments that require repeated execution of the same procedures (steps) with different values of the internal parameters.
- The unified and efficient access to data stored in organizationally distributed environments.
- Visualization of calculation results in a graphical form.
- Search for the required input data and descriptions of individual procedures, scattered across multiple databases.
- These stages of computational modeling experiments are used in different science and technology branches, where investigated objects are composed of discrete blocks (components);
- Aeronautical (involving study, design, and manufacturing of air flight-capable machines, and the techniques of operating aircraft and rockets within the atmosphere).
- Architectural (utilizing current industry technology for both the process and the product of planning, designing, and constructing buildings and other physical structures).
- Chemical (studying chemical structure, bonding and reactivity in chemical systems using mathematical and computational methods or the development of such methods).
- Civil (creating drawings for the civil engineering industry, including areas of land development, transportation, public works, environmental, landscaping, surveying, design visualization and many others).
- Electronic (designing circuits using pre-manufactured building blocks such as power supplies, semiconductors (such as transistors), and integrated circuits).
- Robotic (robotics covers the engineering elements of robotics, automation and autonomy, incorporating robot control, which may be based on Artificial intelligence).
- Industrial Manufacturing (including all intermediate processes required for the production and integration of a product's components).
- Materials (understanding, modeling and processing of metals and alloys with respect to the properties and material behaviour and development of novel materials).
- Mechanical (utilizing CAD software to plan and prepare documents and technical graphics appropriate to the mechanical engineering industry).
- Medical Engineering (research and development of new and existing medical imaging instruments and signals for therapeutic, monitoring and diagnostic purposes).
- Microsystems (this research area captures a broad spectrum of underpinning micro-engineering research aimed at developing a diverse range of novel miniaturized micro-structured devices).

For different branch applications sequences and combination of mentioned steps may vary so as their algorithmic and program implementations. These alternative realizations organizationally can be presented in the form of unified web-services with standardized interfaces.

There is other type of computational experiments in which distributed web service technologies for science data analysis solutions are used. The basic procedures (stages) in these cases for execution of user scenarios against large data stores are:

- Curve fitting and Approximation for estimating the relationships among variables (Linear regression, Simple regression, Ordinary least squares, Polynomial regression, Logistic regression, Nonlinear regression, Nonparametric regression, Semi-parametric regression, Least angle Local, Segmented regression, Interpolation, Fourier Approximation, etc.).
- Classification Techniques for categorizing different data into various folders (Naïve Bayes Classifier, Bayes Network Classifier, CBEA and SEA Classifiers, Decision Tree Classifier, Back Propagation Neural Network, k-Nearest Neighbor Classifier, Multiple Prototype Minimum Distance Classifier, Recursively Splitting Neural Network, etc.).
- Clustering Techniques for grouping a set of objects in such a way that objects in the same group (cluster) are more similar to each other than to those in other groups (Isodata, K-Means, Maximin, Feature Selection/Reduction Techniques, Backward Elimination, Forward Selection, Principal Components, RELIEF (filter-based feature selection), Remove Attributes).
- Pattern Recognition Utilities (Accuracy Measures, Range Filter, K-Fold Cross, Validation, Vector Magnitude, Merge Patterns, Normalization, Sample, Subset, Statics, Cleaning Outliers, Comparing Image File, Discretization);
- Image processing (Collaging, Cropping, Image Difference, Image Normalization, Image Moments, Equalization, Inverse,

Quantization, Relative Level Quantization, Resampling, Rotation, Scaling, Statistics, Thresholding, Vector Plot, Polygon Circumscription, Marking Region, GLCM (Gray Level Concurrence Matrix), GLRL (Gray Level Run Length)).

- Filtering (Dilation, Energy Erosion, Fast Fourier Transfer, Median and Mode Filters, Pulse Coupled Neural Network, Spatial Filter, Gabor Filter, etc.).
- Optimization Techniques (Genetic Algorithm, Multi-Objective Genetic Algorithm, Principal component analysis).

These computational web-services for data proceeding are used in different science and technology branches during data collection, data cleansing, data management, data analytics and data visualisation, where there are very large datasets.

The *Com-Com* supports an end-user in the distributed web-service environment by collection and unification of different computational web services. It also investigates ways to compose and orchestrate these services into a task solution, which end-users can create easily as new applications by combining ready-made services available on the network and incorporating their functionalities. End- users are provided with mechanisms to re-engineer the already available monolithic solutions as sets of services in the Federal or National clouds. Services may be offered by different enterprises and communicate over the Com-Com, that why they provide a distributed computing infrastructure for both intra- and cross-enterprise application integration and collaboration. Very often end-users start to solve their science or technology tasks using web-services of data processing and then transfer to web- services of computational modeling.

Web-services Management

Implementation of the SOC concept means generating end-user applications based on dynamic composition and orchestration of web services workflows. A workflow describes how tasks are orchestrated, what components performs them, what their relative order is, how they are synchronized, how information flows to support the tasks and how tasks are being tracked. Currently, the industry standard for service orchestration is the Business Process Execution Language (BPEL) and 3C-E will use it. BPEL provides a standard XML schema for workflow composition of web services that are based on SOAP. There are other workflow composition tools that create workflow descriptions for a set of web- services execution; however, the tools are not standardized yet. This standardized composition description is eventually deployed on a BPEL engines. The Active BPEL Designer requires too much in- depth knowledge of BPEL definitions to be useful for computing users. To assist users in composing the workflows, 3C-E will adapt a graphical composition tool to work in this environment.

Modern offerings go beyond simple services, including full platforms, complex compositions and whole infrastructures. This leads to a significant complexity in mapping the different modules of these solutions on the large variety of available hardware options. To cope with the challenge to optimize the mapping of services to a variety of different resources, both hardware and software related (e.g., high bandwidth demands), requires topology-aware mapping. This mapping needs to consider placement of the services across geographically distributed centers and demands new intelligent and cross-domain integration of actual and historical usage data. The underpinning idea is based on the assumption that cloud applications can be described and analyzed in terms of workload behaviour, potentially split into segments representing different classes of workload and that an optimised placement of the application elements is feasible relying on rich resource descriptions providing the necessary information from server node capabilities over cluster and data centre topology up to environmental data collected by sensors from the facility management system and business data such as actual power costs.

Prototyping

The Institute of Applied System Analysis (IASA) of NTUU “Kiev Polytechnic Institute” (Ukraine) has developed the prototype of the Engineering Design environment based on SOC [14-16]. It is designed for modeling and optimization of Nonlinear Dynamic Systems, based on components of different physical nature and being widely spread in different scientific and engineering fields. It is the cross-disciplinary application for distributed computing in the form of service compositions functioning within or across organization borders. For example, in cases of electronics, mechanics, hydraulics, control systems, heat, energy, environment tasks selected web-services can provide the following important computational procedures: operations with large-scale mathematical models, steady state analysis, transient and frequency domain analysis, sensitivity and statistical analysis, parametric optimization and optimal tolerances assignment, solution centering, etc.), and supporting procedures (cross-domain mathematical model description translation, data formats translation etc.) based on innovative original numerical methods. Algorithms proposed for many design web-services are novel and unique (multi-criteria optimization, optimal tolerances assignment, yield maximization, stiff- and ill-conditional tasks solving, etc.). The proposed approach to application design is completely different from present attempts to use the whole indivisible applied software in the grid / cloud infrastructure as it is done in TINACloud, PartSim, RT-LAB, FineSimPro and CloudSME. Prototype of this Optimal Engineering Designer was used for micro-electromechanical systems development [16].

Conclusions

Nowadays SOA technology is becoming more and more widespread in many fields of IT industry due to the main advantage: capacity to offer effective approach to the solution of one of the most complicated and actual problems – problem of integration of the information resources (l mathematical procedures and their implementations in our case). Joining the advantages of SOA with the capacities of Grid/ Cloud technology allows providing integration not only of local but of geographically remote applied web- services.

The implementation of any IT service-oriented software system requires performing a number of different steps in order to produce all the required artifacts (either internal or deliverable). In our case the first step is to formalize the research domain concepts and their computing similarities in order to obtain a common interdisciplinary set of applied computing services agreed by all the stakeholders involved in Collaborative Complex Computing. The next step is to select mechanisms that enable story, discovery, selection, mediation, invocation and compose of applied web-services. The development process for *Com-Com* includes also defining web-service semantics, developing the architecture of the framework, designing the supporting software and building a working implementation of whole framework. Presentation of the SOA components as services with standard interfaces ensure their re-use for the creation of new applications and to enhance the capacity of existing ones, rather than re-programming of the same functions. Service capabilities are described using languages such as Description Language Web-services (Web Services Description Language, WSDL).

Service-oriented architecture in the *Com-Com* is the emergence of a new paradigm, which is a response to the increasing complexity of computing distributed software. In other words, it is the dynamic architecture, where the structure and behaviour of the software is changed during its execution, as well as the location where the software components are stored and executed. Penetrating computing environment also introduces new, non-functional requirements for interoperability, heterogeneity, mobility and adaptability collaborative research supported by complex computing.

Com-Com after full realization can enhance Europe's future competitiveness by strengthening its scientific and technological base in the area of Experimenting and Data Processing, makes public service infrastructures and simulation processes smarter i.e., more intelligent, more efficient, more adaptive and sustainable.

References

1. Huhns MN, Singh MP (2005) Service-Oriented Computing: Key Concepts and Principles. IEEE Internet Computing 9: 75-81.
2. Yinong Chen, Wei-Tek Tsai (2008) Distributed Service-Oriented Software Development, Kendall Hunt Publishing, United States.
3. Yinong Chen, Wei-Tek Tsai (2014) Service-Oriented Computing and Web Software Integration. 4th edn, Kendall Hunt Publishing, United States.
4. Yi Wei, Brian Blake M (2010) Service-Oriented Computing and Cloud Computing: Challenges and Opportunities. IEEE Internet Computing 14: 72-75.
5. Michael PP, Traverso P, Dustdar S, Leymann F (2008) Service-Oriented Computing: A Research Roadmap, International Journal of Cooperative Information Systems 17: 223-255.
6. Michael PP, Willem-Jan van den H (2006) Service-Oriented Design and Development Methodology, Int J Web Engineering and Technology 2: 1-17.
7. G Rains (2009) Cloud Computing and SOA, MITRE, White paper.
8. Petrenko AI (2014) Service-oriented computing (SOC) in a cloud computing environment. Computer Science and Applications 1: 349-358.
9. Tsai WT, Sun X, Chen Y, Huang Q, Bitter G, et al. (2008) Teaching Service-Oriented Computing and STEM Topics via Robotic Games. Proc. of IEEE International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing 11: 131-137.
10. Blake MB (2007) Decomposing Composition: Service-Oriented Software Engineers. IEEE Software 24: 68-77.
11. Chen Y, Wei-Tek T (2011) Service-orientation in computing curriculum. Proc. of IEEE 6th International Symposium on Service Oriented System Engineering (SOSE), Irvine, CA, 122-132.
12. Hinke T, Rushing J, Ranganath HS, Graves SJ (2000) Techniques and Experience in Mining Remotely Sensed Satellite Data. Artificial Intelligence Review 14: 503-531.
13. Rushing J, Ramachandran R, Nair U, Graves S, Welch R, et al. (2005) ADaM: A Data Mining Toolkit for Scientists and Engineers. Computers & Geosciences 31: 607-618.
14. Zgurovsky M, Petrenko A, Ladogubets V, Finogenov O, Bulakh B (2013) WebALLTED: Interdisciplinary Simulation in Grid and Cloud. Computer Science 14: 295-306.
15. Petrenko A, Ladogubets V, Tchkalov V, Pudlowski Z (1997) ALLTED-a Computer-Aided System for Electronic Circuit Design, UICEE, Melbourne 205.
16. Petrenko A (2012) Macromodelsof Micro-Electro-Mechanical Systems (MEMS), Microelectro- mechanical Systems and Devices. NazmulIslam (Edt), InTech, US.