Chapter 09

Collaborative Complex Computing Environment (*Com-Com*)

Anatoly I Petrenko*

National Technical University of Ukraine "Kyiv Igor Sikorsky Polytechnic Institute", Ukraine

***Corresponding Author:** Anatoly I Petrenko, National Technical University of Ukraine "Kyiv Igor Sikorsky Polytechnic Institute", Kiev 03056, Ukraine, Email: tolja.petrenko@gmail.com

First Published **March 26, 2018**

This Book Chapter is an excerpt from an article published by Anatoly I Petrenko at Journal of Computer Science & Systems Biology in August 2015. (Petrenko AI (2015) Collaborative Complex Computing Environment (Com-Com). J Comput Sci Syst Biol 08:278-284. doi:10.4172/jcsb.1000201)

Abstract

The goal of the Com-Com is to present an open ecosystem of services and infrastructure that can be easily added, interchanged or improved by researchers across the domains. Collaborative Complex Computing is an emerging interdisciplinary field and it embraces physical sciences like electronics, chemistry, physics, biology, environmental sciences, hydrometeorology, engineering and even art and humanities. All these fields need powerful tools that meet the needs of a wide range of customers in the means of mathematical modelling and collective computing research support, enabling collaboration of distributed group of partners - the providers and consumers of computing resources and data processing solutions. The services and infrastructure are semantically described and can be orchestrated automatically according to the end users goals. *Com-Com* will enhance the capabilities of research organizations which lack resource both in human and technical terms by better integrating distributed researches. Dynamic infrastructure is provided by the utilization of both research and commercial Cloud provision via innovative Cloud composition technology.

Keywords

SOC (Service-Oriented Computing); SOA (Service-Oriented Architecture); Web-Services; Services Composition; Cloud Computing; Engineering Design Platform; Distributed Modelling; EDA for Cloud

Introduction

Collaborative Complex Computing arises from and is intended to address the specific requirements of a large amount of research and industrial enterprises which 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/cloud resources. *Com-Com* concept is the available online intelligent multidisciplinary research gateway combining:

1) 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.

2) 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.

3) Web-services Repository with *Task Solving Supporting (Application Specific) Services (TSS),* which are corresponding to loosely coupled stages and procedures for complex tasks of data processing and modeling, and *Environment Supporting (Generic) Services (ES),* 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.

4) 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.

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

- 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;
- Iterative 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 interdisciplinary 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 at infrastructure and data / user federation level.

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 [12] 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. 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 [13] and is being extensively used by different research groups.

The rest of this paper is organized as follows: Section 3 gives an overview of main elements SOC of the computational platform, web-services as the best component of SOC; Section 4 describes computational web-service examples; Section 5 presents web-services Management; Section 6 describes the prototyping example; Section 7 concludes this paper.

Service-Oriented Architecture of the Computational Platform

The typical computation and data management that exists today in most research and industrial organizations is shown on figure 2.

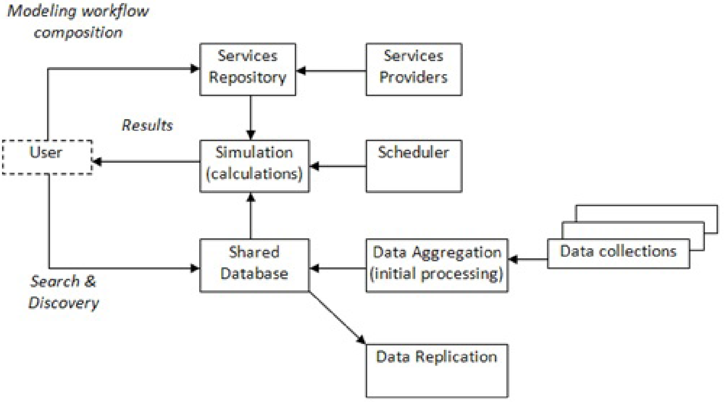

Figure 2: The typical data management in research / industrial enterprises.

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. VRE3C stands for a new technology and methodology for planning and modelling of mathematical experiments, and it can offer the following features:

- ***Execution of composite computing tasks*** of arbitrary complexity to support collective research via the Internet;
- ***Promoting the high scientific and technical level*** of research with open knowledge base;
- ***Iterative 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.

Today humanity comes into the era of Exabyte amount of information that is obtained through high precision measurements in observations and experiments and complex modelling procedures. Humanity starts to use Exaflops HPC for management of large amounts of data (big data), long-term data storage, intelligent data processing, data retrieval in existing sources.

Service-oriented application governance involves knowledge about services, providers and users. Development is divided between a service provider and an application builder. This separation enables application builders to focus on research logic of his application while leaving the technical details to service providers. 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.

The above approaches will be incorporated in the service-oriented computational platform consisting of the following layers (Figure 3). This architecture characterized in that: its web-accessible, its functionality is distributed across the ecosystem of both web services from the VRE3C's 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. 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.

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.

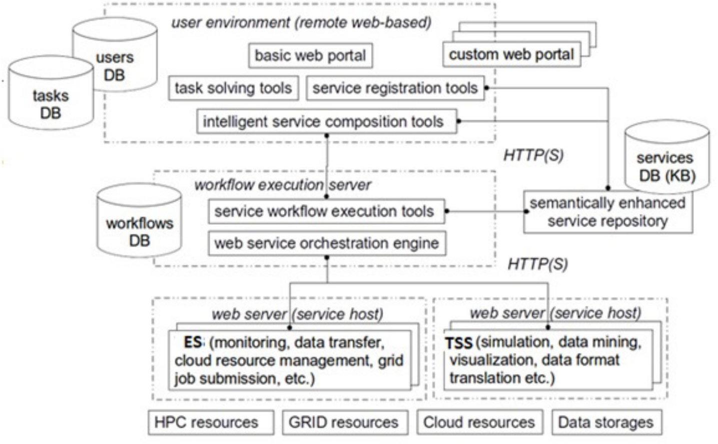

Figure 3: Main elements of service-oriented architecture of the computational platform

When orchestrator invokes grid service the latter initiates the submission of grid job to grid resource: it prepares job description and communicates with grid middleware to schedule and execute grid 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, modelling 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:

1. ***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.);
2. ***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.);
3. ***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);
4. ***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);
5. ***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));
6. ***Filtering*** (Dilation, Energy Erosion, Fast Fourier Transfer, Median and Mode Filters, Pulse Coupled Neural Network, Spatial Filter, Gabor Filter, etc.);

7. ***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 modelling.

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) [17] and *Com-Com* uses 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, *Com-Com* 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 optimise 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 centres 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 analysed in terms of workload behaviour, potentially split into segments representing different classes of workload and that an optimized 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*** [18], ***PartSim*** [19], ***RT-LAB*** [20], ***FineSimPro*** [21] and ***CloudSME*** [22].

The Service Repository is constantly replenished by new services with taking into account recent interest to microservices and containers (Table 1).

It is worthy to mention that IBM has proposed recently SOC-based **Service Sciences** as a new discipline [18] and IBM sponsors universities' educational initiatives and activities in this field because the services sector employs 75% of labour force and SOC is the best technology for the service sector. The SOC paradigm supports the difference between ***software design engineers*** and ***programmers***, which is in the following: software design engineers define software specification to meet the requirements of applications, while programmers take a given specification and write a program to implement the specification. For example, under the traditional programming paradigm students are taught the syntax and semantics of constructs, and are asked to develop small programs. Under the SOC paradigm students learn a repository of reusable services, possibly supplied by third parties and vendors, and then they are asked to compose applications by using the available services. SOC students are taught from the beginning that programming is no longer writing programs, but visual composition and reusing of existing components. SOC programming

Top 10 Contributions on Bioinformatics & Systems Biology

Table 1: The Services Repository developed content.

Task Solving Supporting Specific Services (TSS)		
N	Service Description	Service Impact
1	***Data Processing Services***	Data – it's one of the most valuable assets. Com-Com provides possibilities for users to work with nearly any type of data from almost any source – either on premise or in the cloud. They allow discovering, cleansing, and integrating enterprise data so that they always available to meet research/ development needs.
1.1	Data cleaning and pre-processing	
1.2	Digitize data and documents	
1.3	Aggregating concurrent access to data in many distributed databases located in different countries and available world-wide.	
1.4	Data translation between standard and discipline-specific data exchange formats.	
1.5	Copying (replication) data for improving their localization, increasing availability and reducing the risk of loss	
1.6	Using metadata (data about data) with descriptive information about the origin of the data, describe howdata were generated (measured or calculated).	
1.7	Providing the powerful associative data search and discovery (search by value, not by the location), and automatic parallel access.	
1.6	Data mining technology to detect hidden patterns in the form of significant features, correlations, trends and patterns.	
1.8	Online Data Entry and Remote Verification	
1.9	Context management with Semantic support	
1.10	High level intelligent Services like Semantic Data Aggregation	
1.11	Access, integrate, or move data to generate insights	
1.12	Semantic data inference	
2	***Modelling and Simulating Services***	Together these applied services form a powerful and versatile tool that meet the needs in the means of mathematical modeling and computing support of research, in supporting collective scientific and applied research by distributed group of partners. **More effective collaboration between researchers** is achieved by attracting small actors to build these services and applications, e.g. SMEs, university teams, start-ups whoever.
2.1	Self-explaining user interface even for unskilled end users	
2.2	Automatic forming of mathematical model of a mathematical experiment (or a process) from description of its route and stages' properties.	
2.3	Reducing the dimension of the formed mathematical model of an experiment (or a process) and developing its "macromodel" if it is necessary.	
2.4	Implementing new stages for mathematical experiment by customers themselves (scripting).	
2.5	Executing mathematical experiment (modelling) in statics (DC), using new advanced numerical procedures with the high solution convergence	
2.6	Special procedures for solving **ill-conditional tasks**	
2.7	Automatic determination of output parameters (frequency band, resonance frequencies, etc) during Frequency analysis (AC)	
2.8	Dynamic modelling (TR) in time domain, usingnovel implicit methods of variable order and variable step for **solving stiff tasks**, and automatic determination of design parameters (delay time, rise and fall times, consumed power, etc.)	
2.9	Analysis of sensitivities of output parameters of a mathematical experiment to changes of parameters of internal stages (components) in time and frequency domains	
2.10	Optimization of output parameters of a mathematical experiment with the **novel method of variable order** (1st-4th), which covers the gradient methods of 1st order and the quasi- Newton methods of variable metric of 2nd order as particular cases	
2.11	Statistical analysis (STA) of output parameters and characteristics of a mathematical experiment with possibility to optimize **Yields**	
2.12	Classification, clustering and association rule mining methods that are common to many data mining systems	
2.13	Image processing and recognition	
2.14	Image contouring and feature extraction	
2.15	Visualization of calculation results in a graphical form	
Environment Supporting (Generic) Services (ES)		
N	Service Description	Service Impact
1	***Com-Com Cloud Hosting for computational, network and software resources provision***	**Higher efficiency and creativity in research, higher productivity of researchers** are provided by enabling ***anybody*** (due to the applied web-services Repository) to be either ***as a contributor*** of specific web-services and applications solutions or ***as a user*** reusing available solutions for its own product.
1.1	**IaaS Data Center Resource Management** for self-service provisioning and life cycle management of virtual machines and associated compute, storage and network resources	
1.2	**Monitoring** of virtual machines deployed in any Cloud system.	
1.3	**Objects Storage** for storing and retrieving storage objects accompanied by metadata.	
1.4	**PaaS Management** for hosting of application containers, such as Web container, database instance, etc. It includes the provisioning of the required virtual resources at IaaS and configuration of the whole software stack on those virtual resources.	
1.5	**Software Deployment and Configuration** which allows the application developer to develop and deploy applications on virtual machines.	
1.6	**Interface to Network services and to open networking entities**	
2	***Applications/Services Ecosystems and Delivery Framework***	Providing mechanisms for describing, publishing and discovering available ***Environment Supporting Generic Services***, based on available EGI services and workflows, and developed ***Task Solving Supporting Specific Services*** for Complex Collaborative Computing.
2.1	**Repository** which contains services and applications descriptions and supports for providing services to both consumers and developers of applications.	
2.2	**Application Mashup creator** that offers a composition editor and execution engine that allows end users with little or no programming skills to create and run a composite web application built from web- services	
2.3	**Data Warehouse** offers support to manage open data at large scale and transform it into knowledge	
2.4	**POI (point of interaction) Service Provider**for semantic search services in the Repository and Data Warehouse.	
2.5	**Network Information and Control** for providing open flow network information and control.	
2.6	**Multi-channel-multi-service access** means all the various ways in which users may reach services and support, including phone and email, knowledge bases, communities, and live chat tools.	
3	***Security***	**Increasing take-up of collaborative research and data sharing** by launching and maintaining a user community and using standardized building blocks and workflows, well documented safe interfaces and interoperable components
3.1	**Access Control** – this allows managing authorization policies, and provides authorization decisions for requests to REST APIs of other services in Com-Com	
3.2	**Identity Management**– these ES services cover a number of aspects involving users' access to networks, services and applications, including secure and private authentication from users to networks and services.	
3.3	**Security Monitoring** – this is a suite of services for risk analysis, security visualization, decision making support and technical forensics.	
3.4	Interface to Cloud proxies	
4	***Work-flow engine for calculating purposes***	**Accelerating innovation in research** as result of integrating of web-services from different suppliers into joined application software and user-customizable computing scenarios.
4.1	Definition and execution of workflows	
4.2	Resource discovery in e-infrastructure based on application of formal logic models.	
4.3	History keeping of user actions and data capturing for auditing purposes	
4.5	Common Com-Com Capacity infrastructure for trials	
4.6	User cases – to – workflow translation	
4.7	Automatic **workflow reconfiguration** based on current resources' properties and user goals	
5	***Digital Science in Com-Com***	Mentioned operational means will implement the mainstreaming of Digital science in Com-Com activity.
5.1	Open Access to obtained results and an obligation to ensure that all publications produced will be linked to the Open portal.	
5.2	**Knowledge base support** services	
5.3	Opening services code and algorithms made to make them to be available for and to (re)use by others.	
5.4	Research community-driven development of virtual collaboration platform of specific topics and disciplines	
5.5	Embrace open and collaborative research approaches, by sharing research data, tools and results and collaborating with broader research community	

is just a high-level visual modelling approach that will be easier to understand than traditional command line-based programming constructs.

Conclusions

The original concept of the Complex Collaborative Computing Environment (with computing procedures as web-services) have the following innovative features:

- **Division** of the entire computational **process** into separate loosely coupled stages and procedures for their subsequent transfer to the form of unified web-services with standard interfaces that can launch computations at different grid / cloud sites;
- Creation of a **Repository** of computational web-services which contains components developed by different producers that support collective applications development and globalization of R&D activities, either for free or with certain fee. This Repository provides a systematic formal approach to computerization of science and technology tasks;
- Separation services into **Environment Supporting (Generic) Services** and **Task Solving support (Application) services.** Task Solving support services provide different applied procedures and processes available for a user to consume in own workflow in the particular scientific or engineering field. Environment supporting services may be shared with other producers (for example ***EGI[24], Flatworld [25], FI-WARE [26], SAP[27], ESRC [28],*** etc.). The invariant application support web-services may be provided by so "freely deployable" libraries ***PETSc, Netlib, Math Kernel Library (MKL), CERNLIB, IMSL, BLAS***, etc. which may be linked to the created Repository;

- Unique web-services to enable automatic formation of mathematical models for the solution tasks in the form of equation descriptions or equivalent substituting schemes;
- **Personalized** and **customized** user-centric application design enabling users to build and adjust their design scenario and workflow by selecting the necessary web-services (as calculation procedures) to be executed on grid/cloud resources. A user can also introduce new calculation procedures and their parameters, which are absent in any existing monolithic software solution;
- **Re-composition of multidisciplinary applications can at runtime** because web-services can be further discovered after the application has been deployed;
- ***Service metadata*** creation to allow meaningful definition of information in cloud environments for many service providers which may reside within the same infrastructure by agreement on linked ontology. Third-party software agents operating within a cloud might be able to derive ontological information from the stored data and operations. Thanks to ontology it becomes possible to create service-oriented applications even by orchestrating legacy applications that do not support the web- services specifications.

Service-oriented applications design and development methodology provides sufficient principles and guidelines to specify and construct rapid, low-cost and easy composition of distributed applications choreographed from a set of internal and external web-services. The visionary promise of them is a world-scale network of loosely coupled services that can be assembled with little effort in agile applications that may span organizations and computing platforms. Since services may be offered by different enterprises and communicate over the Internet, they provide a distributed computing infrastructure for both intra- and cross-enterprise application integration and collaboration.

Com-Com enhances 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. MN Huhns, MP Singh. Service-Oriented Computing: Key Concepts and Principles. IEEE Internet Computing. 2005; 9: 75-81.
2. Yinong Chen, Wei-Tek Tsai. Distributed Service-Oriented Software Development. Iowa: Kendall Hunt Publishing. 2008; 467.
3. Yinong Chen, Wei-Tek Tsai. Service-Oriented Computing and Web Software Integration, 4th end.. Iowa: Kendall Hunt Publishing. 2014; 400.
4. Yi Wei, M. Brian Blake. Service-Oriented Computing and Cloud Computing: Challenges and Opportunities. IEEE Internet Computing. 2010; 14: 72-75.
5. Michael P. Papazoglou, Paolo Traverso, Schahram Dustdar, Frank Leymann. Service-Oriented Computing: A Research Roadmap. International Journal of Cooperative Information Systems. 2008; 17: 223–255.
6. Michael P Papazoglou, Willem-Jan van den Heuvel. Service-Oriented Design and Development Methodology. Int. J. of Web Engineering and Technology (IJWET). 2006; 1-17.
7. G Rains. Cloud Computing and SOA. MITRE, white paper. Oct. 2009; http://www.mitre.org/work/tech_papers/tech_papers_09/ 09_074309_0743.pdf

8. AI Petrenko. Service-oriented computing (SOC) in a cloud computing environment. Computer Science and Applications. 2014; 1: 349-358.

9. WT Tsai, X Sun, Y Chen, Q Huang, G Bitter, et al. Teaching Service-Oriented Computing and STEM Topics via Robotic Games. Proc. of IEEE International Symposium on Object/Component/Service- Oriented Real-Time Distributed Computing (ISORC). 2008; 131-137.

10. MB Blake. Decomposing Composition: Service-Oriented Software Engineers. IEEE Software. 2007; 24: 68–77.

11. Yinong Chen, Wei-Tek Tsai. Service-orientation in computing curriculum. Proc. of IEEE 6th International Symposium on Service Oriented System Engineering (SOSE), Irvine, CA, 12-14 Dec. 2011; 122-132.

12. T Hinke, J Rushing, HS Ranganath, SJ Graves. Techniques and Experience in Mining Remotely Sensed Satellite Data. Artificial Intelligence Review: Issues on the Application of Data Mining. 2000; 14: 503-531.

13. J Rushing, R Ramachandran, U Nair, S Graves, R Welch, et al. ADaM: A Data Mining Toolkit for Scientists and Engineers. Computers & Geosciences. 2005; 31: 607-618.

14. Michail Zgurovsky, Anatoly Petrenko, Volodymyr Ladogubets, Oleksii Finogenov, Bogdan Bulakh. WebALLTED: Interdisciplinary Simulation in Grid and Cloud. Computer Science (Cracow). 2013; 14: 295-306

15. Petrenko A, Ladogubets V, Tchkalov V, Pudlowski Z. ALLTED- a Computer-Aided System for Electronic Circuit Design. Melbourne: UICEE.(UNESCO). 1997; 204.

16. A Petrenko. Macromodelsof Micro-Electro-Mechanical Systems (MEMS). In: NazmulIslam, editor. Microelectromechanical Systems and Devices. Croatia: InTech. 2012; 155-190.

17. A Milanović, S Srbljić, D Skrobo, D Čapalija, S Rešković. Coopetition Mechanisms for Service-Oriented Distributed Systems. Proceedings of the CCCT 2005 (The 3rd International Conference on Computing, Communications and Control Technologies), Austin, Texas, USA, July 2005; 1: 118-123.

18. TINA Cloud project home: http://www.tina.com/English/tina

19. PartSim project home: http://www.partsim.com/examples

20. RT-LAB project home: http://www.opal-rt.com/company/company-profile

21. FineSim Pro project home: http://www.automation.com/content/magmas-latest-version-of-finesim-pro-delivers-3x-faster-runtime

22. CloudSME project home: http://cloudsme.eu

23. IBM Service Sciences: http://www.research.ibm.com/ssme

24. EGI: http://www.egi.eu/

25. Flatworld: http://www. flatworldsolutions. com/

26. FI-WARE: http://catalogue.fi-ware.org/enablers

27. SAP: http://www.sap.com/pc/ tech/enterprise- information - management/

28. ESRC: http://ukdataservice.ac.uk/