C^2 NIWA – THE CENTRE OF COMPETENCE FOR NOVEL INFRASTRUCTURE OF WORKABLE APPLICATIONS

HENRYK KRAWCZYK

Department of Computer Systems Architecture
Gdansk University of Technology, Poland

Abstract: Scientific and cloud computing are discussed in the paper. The paradigm of EaaS (Everything as a Service) is presented and the design methodology of service oriented applications is proposed. Implementation of the SOSE (Software Oriented Software Engineering) approach is analyzed and the roles of developing teams to obtain software products satisfying business requirements are also shown. The above listed activities create a basic functionally of the C^2 NIWA centre recently established in the Gdansk University of Technology.

Keywords: scientific computations, cloud and service computing paradigms, service-oriented applications, service-oriented software engineering, C^2 NIWA functionality

1. Scientific computations in supercomputer centres

Scientific computations are concerned with using modern computer systems to model and analyze scientific problems. The idea is to create mathematical models of different natural phenomena, develop computing programs and run them with various sets of parameters. In that way scientists may check if their theory is consistent with experimental results obtained in research laboratories. In case of real-world problems these models are very complex and require very often high performance computer clusters (supercomputers or grids). To facilitate such research activities special computing centres have been created. In Poland there are five supercomputing sites, in Poznan, Warsaw, Cracow, Wroclaw and Gdansk. They offer not only computational resources to run scientific programs for given sets of data, but also deliver professional scientific software to do various kinds of simulations using numerical, quantitative or symbolic methods. It leads to decreased research costs, if many scientific teams are cooperating with those centres. Moreover, further possibilities are developed to integrate computer resources with different centres to obtain higher possibilities

of modelling and simulation. Besides, new visualization techniques for presenting research results are proposed to better understand the analyzed phenomena. An example of such a solution is PL-Grid [1] which offers many domain oriented applications responsible for using mathematical and computational methods in science, engineering and humanities. All users of PL-Grid are able to archive their work results in backup archive systems. In many cases interaction between users and the system is based on a batch solution *i.e.* on a set of commands which initiates a computation process. No particular knowledge about clusters is required. Eventually handy tools are necessary for accessing large computing resources. To support interactive access to clusters, users either run their command lines of applications or compile their own code themselves and run it. Other solutions provide the most efficient and powerful multi-user access to the tasks management and advance reservation of computation and storage resources distributed on many sites. Other capabilities such as allocation of software tasks to resources, choice of communication topologies, asynchronous notification, cross cluster execution of tasks or executed task monitoring are often added. Suitable Web portals having some part of the above functionality are offered (see Figure 1). They also allow building bridges across a multitude of disciplines and promote the best user cases of solutions. Currently, users of the PL-Grid infrastructure are integrating with the Platon infrastructure and in consequence, some scientific computations can be delivered as services on demand in a cloud environment. It is developed taking into account solutions coming from the EGI (European Grid Infrastructure) [2], where basic services are based on the following components:

- compute and storage facilities, including user access management, user training and incident handling;
- user database, storing user data, account setups and roles;
- a service monitor and help desk system for providing service status information and for handling user requests.

The information concerning users may be shared by all Polish centres, and all computing and storage resources available at a site may constitute a consistent pool and may be delivered to users in a uniform way.

The Computer Centre of the Tri-City Academic Computer Network at the Gdansk University of Technology has participated in developing the PL-Grid and Platon systems. Moreover, another project devoted to the Centre of Competence NIWA (Novel Infrastructures of Workable Applications) at the Gdansk University of Technology is under development. It cooperates strongly with the Computer Centre of the Tri-City Academic Computer Network. Due to the project, a new supercomputer has been purchased. Its name is Tryton and it has 1308 nodes with 2 processors in each of them and cooperates with the 56 Gb/s infiniband network. The total possible cluster performance is 1.2 / Peta Flops. Moreover, new dedicated platforms are created and used in the Centre of Competence NIWA. The platforms offer new possibilities for general users as well as researches
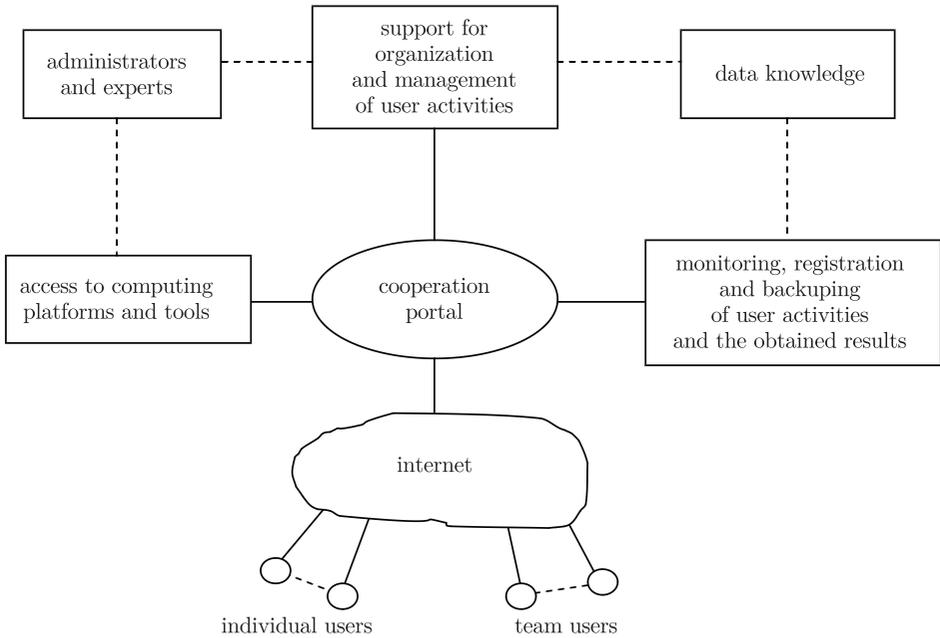
Figure 1. Idea of utilization of Centre of Competence NIWA platforms

and businesspeople. It may create a bridge between science, industry and the market, where the majority of solved problems are strictly related to some real needs.

Scientific computation and cloud computing are discussed in the paper. The offered platforms are briefly presented. Everything as a service (EaaS) is analyzed and a development process of applications consisting of such available services is proposed. The Service Oriented Software Engineering (SOSE) methods are presented and its practical relevance is discussed. Some software products implemented in the C^2 NIWA are described. These show initial results after one year of the centre's operation. Other issues are presented in other papers in this journal.

2. Main activities of C^2 NIWA

A centre of competence is an organization devoted to research and development in multidisciplinary and strategic areas, for example, designing systems for combining medicine and ICT. It creates conditions for close cooperation between industry and science, and offers and runs special training and courses, *i.e.* improvement of human creativity, creation of innovation processes, designing business plans and making their feasibility studies. All of these activities should conform to the highest international standards. In many cases such centres are members of extended international R & D networks.

The C^2 NIWA is one of such centres, operating at the Gdansk University of Technology offering the following activities:

1. Scientific computations based on a high performance supercomputer (Tryton) which allows advanced modelling and simulation of different natural or artificial phenomena.
2. Efficient access to different computing platforms oriented on human-centric distributed processing, such as wide multimedia analysis, mobile services development, business process refinement.
3. An option to develop tools supporting design and step-by-step improvement of large use information processing services and Internet applications using SOSE (Service Oriented Software Engineering).
4. Entry to different digital document repositories storing program source and binary codes, project reports and research publications all available in an Open Science Standard.
5. The possibility to create real and virtual teams to investigate and solve either self-defined practical problems or vital procedures required for industrial endeavours.
6. Research in improvement of the centre's functionality including software development methods, optimizing team performance, rebuilding existing platforms and extending used technologies.

The above various possibilities of the C^2 NIWA are illustrated in Table 1. It features 8 different platforms supporting user activities such as: creating algorithms for data stream analysis (KASKADA), designing some frameworks for mobile devices (Wiki-WS), making some simulations to confirm results of real experiments (Beesy Cluster), modelling real business processes (BPM/BPEL), creating and testing Internet applications (REDMINE), preparing some e-lessons about effective use of offered environments (MOODLE), gathering some data about designed processes and their teams (SAWA), or developing open repositories for publications and project documentation (DSPACE). In consequence, some products of user activities have been proposed and implemented (see Table 1). In other words, the centre encourages out-of-the-box thinking (*i.e.* original and innovative) so that new and exiting propositions, and of course concepts, may be born healthy and strong into the arms of one or more of specialist working teams.

As always, competitions are constantly organized for students to test their ingenuity and originality, to spontaneously cross new boundaries of our rapidly changing world. Consequently, they are challenged to create innovative IT services or IT applications. These activities may be curricular, strictly connected with university courses – or extra-curricular activities funded by industry or business. In these cases student teams should be organized autonomously and select leaders for optimum communication with teachers or external managers. These leaders are responsible for management, work organization, and they may change over time. In some cases virtual teams should be created, and they can operate in the Internet using the available platforms. Their work can be monitored, and based

on the collected data, analyzed in detail, quality assessment of teams or their products can be performed, and also some suggestions to improve development processes can be formulated.

3. EaaS – Everything as a Service

Services in software engineering, means autonomous platform independent entities that can be described, published, discovered and used to perform the required user functions, offered by service providers to service consumers. Two similar approaches: service oriented computing (SOC) and service oriented architecture (SOA) define the concept of development of efficient, low-cost, and interoperable user services [3]. By adapting SOA and SOC we create new opportunities to design, run, monitor and manage Web-services-enabled applications. In other words – with different kinds of services offered by various vendors we can develop also domain-specific applications essential and useful for many human activities, and in consequence increasing the quality standards of our life. All kinds of such applications can be named simply as user applications.

Nowadays cloud computing is also increasingly popular because it offers new possibilities. All the available resources become both dynamically scalable virtualized and provided runtimely as services over the Internet. It is assumed that in the future information access to different digital devices will be as ubiquitous as electricity. In consequence new development trends lead designers to design and implement cloud computing systems that deliver – Everything as a Service (EaaS) [4] – from infrastructure resources to personal interactions. Therefore, layers of cloud systems become sets of many kinds of services. We have Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS). Moreover, these three layers are still under development and are becoming more complex and more flexible.

Table 2 illustrates the differences between SOA and EaaS oriented computing clouds [5]. The IaaS layer, instead of typical computing and storage resources, contains large-scale data centres with millions of users accessing, and thousands of services. Moreover IaaS for EaaS should be energy efficient, scalable, flexible and extensible coordinate, carefully designed to minimize the need for global information exchange and central arbitration.

The PaaS layer, instead of static configuration and classical development tools, is oriented on high quality services which scale its resources dynamically according to the changing workload. Those services are isolated from infrastructure management operations. The user may declare the desired topology of virtual machines and computation and storage capacities in it. The infrastructure service creates an instance of such topology with a set of quality of service constraints. For this reason, computation units, storage volumes, and subnets called virtual cells create a virtual infrastructure, with can be managed and configured according to suitable policies.

Table 1. Basic C²NIWA activities and products

	C²NIWA SOCIETY							
social relations								
products	immerse 3D visualization lab, fall detection, plagiarism detection, pulsar waves analysis	smartphone applications, recommender of services, moving robot	library of graph algorithms KOALA	modelling of business processes 1. production of a simple good 2. banking operations	platform components applications services	e-lessons on team creativity and team motivation	reports about questioners project development process	open science repositories to store and offer specialized knowledge
main function	multimedia stream data	pervasive & mobile service	scientific applications	business workflows analysis	designed applications and services	distance learning	data acquisition of team activities	publications moodle lectures, project documentation source codes
platforms	KASKADA	Wiki-WS	Beesy Cluster	BPM/BPMN	REDMINE SOSE	Moodle Power Point	SAWA Warehouse	DSpace
super-computer infra-structures	TRYTON Linux, Apache, MySQL, PHP							

Table 2. Comparison of SOA and EaaS types of clouds

		SOA services – hosted and managed by third parties (desktop Internet)	EaaS – pervasive and ambiguous services (mobile Internet)
	SaaS	Different kinds of user applications: related to, *e.g.* Web Business and industry areas	Mobile user applications referred to different industry and business areas as enterprise services with personal access and available anywhere and anytime
	PaaS	Virtual runtime machines, operation systems, middlewares and database development tools	Scalable, secure, economical cloud middleware with tools and environment platform services
IaaS		CPU, memory and networking storage services	Energy efficient infrastructure with virtual services and IT management services and large scale data centre services

The SaaS layer creates various mobile applications oriented at the different kinds of users, and also special service tools to collect knowledge about the occurring context, and user behaviour, to evaluate in the long-term not only the popularity of applications, but also estimate the occurring trends in the user behaviour, and user relationships. In that way, collective intelligence for the variety of users of different services can be captured, and new recommendations for a development strategy can be defined.

In Table 1 we present all platforms built by teams using the open source solutions as basic support C²NIWA functionality. It is a very tedious time consuming task, sometimes (up to now) impossible to transform them into a cloud with the EaaS concept. Therefore, we have decided to make this operation only for the KASKADA platform.

4. Implementation of SOSE Approach

As has been shown, modern computing paradigms share two basic approaches: cloud and service computing. Cloud platforms offer virtual computing cells to execute either providing services or realizing application scenarios consistent with the available services. These two paradigms impact on changes in software engineering, creating a new concept called SOSE (Service Oriented Software Engineering) [6]. Each user application can be described as a set of service-oriented scenarios. The idea of the SOSE Methodology is illustrated in Figure 2.

To prepare a final user application, in other words, its service oriented scenarios, six continuous steps can be made. We start from the problem definition, its transformation into a model which allows us to create general scenarios of the application. Next, this general scenario could be adapted to an available computing platform, and be optimized and tested in order to obtain the final scenarios.
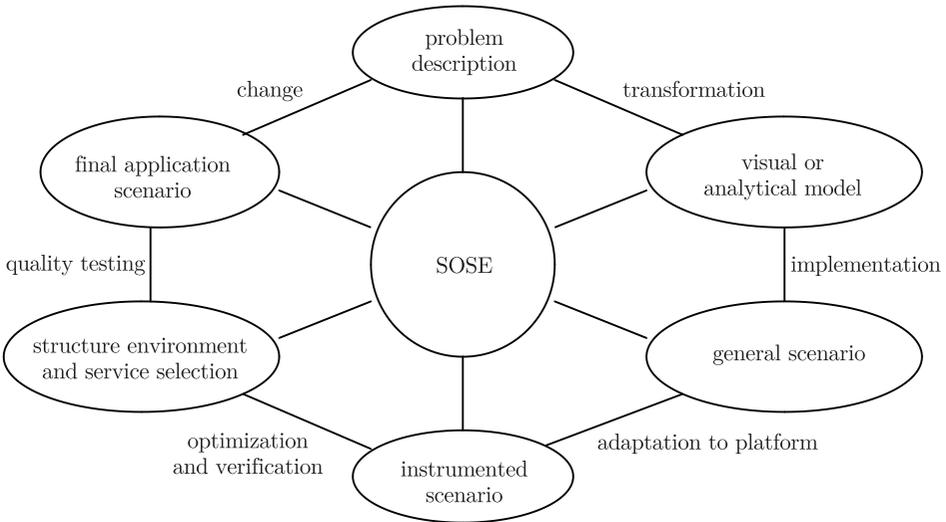
Figure 2. SOSE Methodology for developing of service oriented application scenarios

In this way we undertake the effort to bring agility and flexibility of application into market-oriented scenarios of available services. In consequence we obtain a service based application.

The problem description does not have to consist of a requirement specification document only, it may contains a contract which specifies the project data and budget, legislative regulatory requirements, and also often some suggestions about service recommendation. Moreover, the visual or analytical model can show the most important aspects of the scenarios in a readable and understandable way. As our methodology is standard agnostic, it is only the notation that we do not improve. Depending on the category of applications, it may be a graph, or a UML diagram or a SOA-dedicated BPMN diagram. What is important is that the assumed model should be easily turned into executable scenarios which is still understandable not only to designers, but also by the future users.

To obtain a general scenario we should define its main elements in an extensible way, *i.e.* using suitable notation to describe local instructions for data manipulations, controlling the flow and handling errors, and add some mechanisms to support invocation of remote available services. In consequence we obtain a scenario which should be next adapted to a concrete cloud platform to be executable. Moreover, instrumentation adds also a monitor service and the logic of events logging. It is fully transparent to end users, and also allows you to monitor all service invocation operations.

Verification means a process, where the suitability of the scenario has been proven. First of all, an optimization procedure is enhancing the scenarios structure, where lengths of loops are reduced, the sequence of service invocation is reordered. After this, more detailed evaluation of service suitability is done. From the set of alternative services only the best ones are selected. These operations

require additional development time because alternative sets of scenarios must be designed, implemented and evaluated. It is likely that additional cloud resources can be required for the best scenario, as well.

To obtain a final scenario, extra quality tests can be carried out. They take into account many performance, dependability, and usability metrics. Moreover, the application or a platform should offer an alert mechanism to warn software engineers or end users, when the computed scores are unsatisfactory. If some external factor, like changes in the application requirements, or new offers of better services, may cause the whole development methodology to start again, then we come back to the description step and repeat the whole development process from the beginning.

The development process engages many actors, as is shown in Figure 3. They need strong cooperations not only to prepare services-based applications, but also to run them in a cloud environment. Moreover, C^2 NIWA provides support to collect very important information about development processes, and also about the activities of all actors. The gathered knowledge allows us to improve some important aspects of the SOSE methodology.
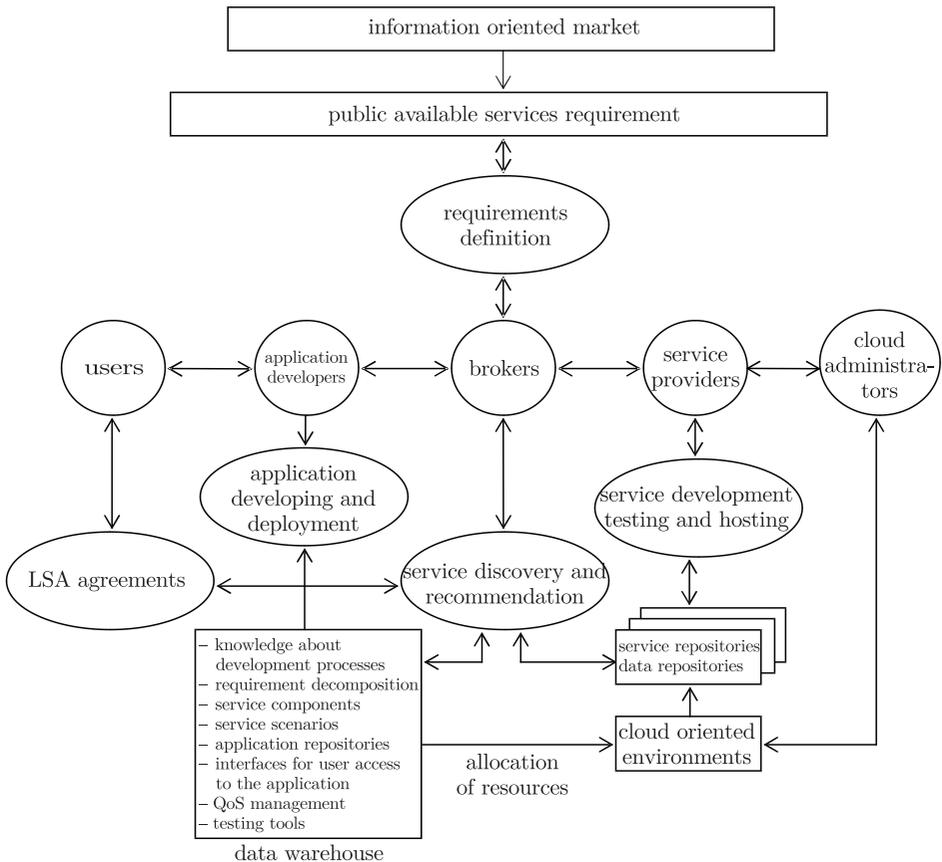
Figure 3. Basic roles of main actors in SOSE

5. Software development teams

The goal of developing software is to deliver a software product of value to all customers who need it. To do this at least three attributes can be taken into account (see Figure 4). They correspond to the available development methods and tools, largely-used development platforms supporting designer activities, and also the quality of the teams engaged in the development processes. To make the software product in an efficient and effective way, the right balance among these attributes should can be achieved. It is important to find a common way of description, understanding and combining various software development techniques, instead of setting them up in competition with each other [7].

Figure 4. Main attributes for product development

All these main attributes for software development on the lower level determine (in case of the available methods and tools) a proper recommendation of developing the methodology and point out the most useful procedures. In the case of developing platform and processes it concentrates on the suitable activities for creating a software product step by step. In the case of engaged teams it shows the necessary competence of teams and their motivation for success, full finalization of the work.

The most popular developing method is an agile approach which prefers flexibility and adaptability in case of changing requirements. This is often done by developing software in small components (increments), obtaining feedback in iterations, and continually adjusting as necessary. Such an approach is used

Table 3. Primary products of C^{2}NIWA

products	descriptions
service recommender	It allows you to recommend, according to the given criteria, the most suitable services for service oriented applications.
graph algorithms as services	It offers a library of graph algorithms available as services through a web browser.
multimedia stream service repository	It consists of a set of services supporting multimedia processing, *e.g.* analysis of single frames, detection of some objects or events occurring in the sequence of frames.
open science repository	It contains different kinds of digital documents, such as source codes, e-lectures on creativity and team organization, scientific papers, *etc.*
example of supply chain analysis in BPM (Business Process Modelling)	It describes the system of organizations, the people, their activities, information and resources involved, a product from supplier to customer.

to produce Web Services using Wiki-WS platform (see Table 3). The SOSE methodology is strongly recommended in case of large scale applications.

Largely used and quite universal development practices can always be discussed and then adapted by teams as appropriate for the new approaches. The special tools for measuring the progress of the development process should be continuously used to determine the best further activities. There are many tools such as Firefox, Subversion, Bugzilla or Redmine which allow you to collect software engineering data and next to create important knowledge about development processes. In our case such roles are offered by the SAWA platform (see Table 1).

To carry out all such activities, the development team members should be experts (not only IT specialists) in areas corresponding to the problems to be solved. Only in this case will the results of team works, and their final products, achieve the required level of quality [8].

We can look at software development processes from two points of view: market and science, or more precisely, from market rules and research requirements. Figure 5 illustrates how to bridge the gap between the market developer community and the pure scientist community. Let us consider a situation where these communities work separately. Each of them produce different products: either market goods or publications. In many cases a market product is not an innovative one, and a publication is without practical relevance. Another possibility occurs when both communities cooperate with each other. Then, they create a common multidisciplinary team, which generates an integrated solution. It means that it is an obtained compromise which satisfies both kinds of requirements. In some cases when solutions are very promising it generates new circumstances to create new firms. It takes place if either quality owners of the parent company receive equity stakes in the newly spun off company (spin-off),
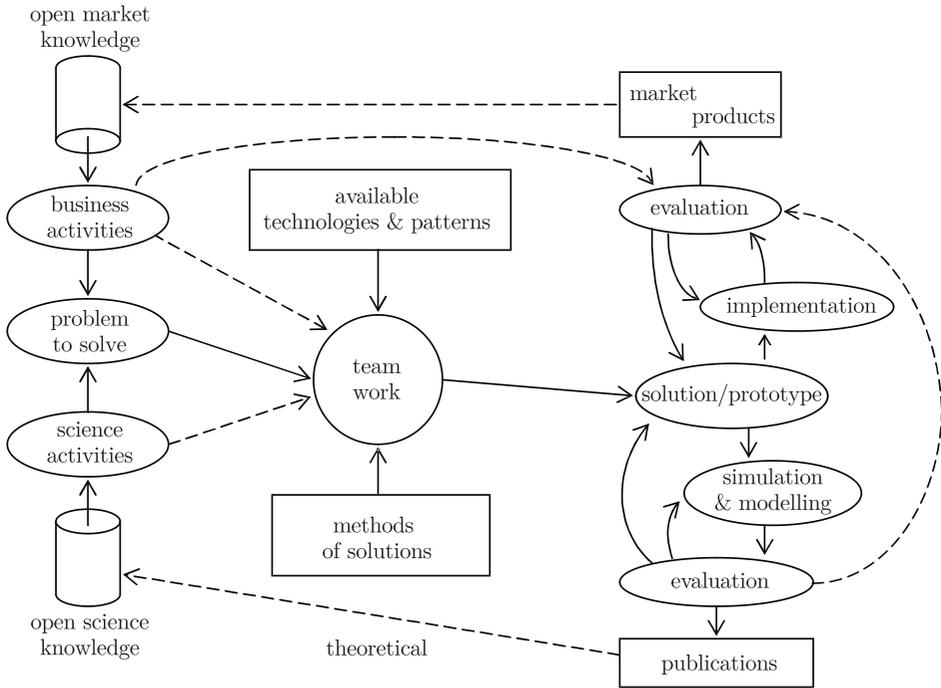
Figure 5. Multidisciplinary teams for solving joint business problems

or a group of employees leaves an existing entity to form an independent start-up firm (spin-out).

In the nearest future the proposed schema of commercialization of software products should be verified. It requires meeting market partners and solving many joint business/science problems. Up to now, we have created some primary products which can be further developed to achieve an acceptable level of maturity and quality. Examples of such products are shown in Table 3. We plan to develop some of them using the rules of the open software approach.

6. Final remarks

The C^2 NIWA was initialized in November 2013 as a direct consequence of Grant POIG.02.03.00-22-059/13 funded by NCBiR (National Centre of Research & Development). The majority of the grant finance was spent to buy the supercomputer. During the opening ceremony it was named Triton which has several meanings. One is the name of the mythological Greek god, son of Poseidon and Amphitrite. Poseidon is the counterpart of the Roman god Neptune. The other meaning of Triton is a class of ships of either the Royal Navy or the US Navy. We hope the C^2 NIWA with Triton will create new opportunities for research and business achievements.

Up to now, Triton has been used for large-scale scientific simulations (such as batch processing) to verify theoretical models with results obtained from various experiments related to natural phenomena. Our main aim was to enhance

the supercomputer utilization to enable interactive and real-time processing, and make possibilities to connect the supercomputer with many Internet devices (according to the "everything as a service" approach). This allows us to adapt the supercomputer into a computing cloud. Coming from cloud and service computing paradigms we develop a SOSE methodology to design and evaluate a new kind of software products called service-oriented applications. We offer also new tools to monitor the behaviour of work teams and show which attributes of development processes play an essential role in the achievement of high quality applications. We propose to create multidisciplinary teams to solve and implement essential practical problems. In consequence, we can implement software products satisfying conditions of business requirements. For this, we use Open Stack software, and up to now the KASKADA platform has been engaged in this cloud. Many other proposed solutions are also based on open software, for example, Dspace, Redmine or Moodle. In consequence we prepare frames for the Open Science approach. It creates a change that real and virtual design teams will be cooperating with C^2 NIWA. This will take more time, when we take new challenges to prepare new kinds of software applications, which will lead to gradual improvement of the quality of human life.

References

[1] Szepieniec T, Tomanek M, Radecki M, Szopa M and Bubak M Implementation of Service Level Management in PL-Grid Infrastructure, *Building a National Distributed e-Infrastructure-PL-Grid* Bubak M, Szepieniec T and Wiatr K Eds, Springer Berlin Heidelberg, **7136** 171

[2] Radecki M, Szepieniec T, Szymocha T, Szopa M and Krakowian M 2012 Towards Professional Service Operations in Grids, *Building a National Distributed e-Infrastructure-PL-Grid* Bubak M, Szepieniec T and Wiatr K Eds, Springer Berlin Heidelberg, **7136** 27

[3] Papazoglou M P, Traverso P, Dustdar S and Leymann F 2007 *Service-Oriented Computing: State of the Art and Research Challenges*, *Computer* **40** (11) 38

[4] Banerjee P, Friedrich R, Bash C, Goldsack P, Huberman B A, Manley J, Patel C, Ranganathan P and Veitch A 2011 Everything as a Service: Powering the New Information Economy, *Computer* **44** (3) 36

[5] di Costanzo A, de Assuncao M D and Buyya R 2009 Harnessing Cloud Technologies for a Virtualized Distributed Computing Infrastructure, *IEEE Internet Computing* **13** (5) 24

[6] Krawczyk H and Budnik Ł 2011 Agile Methodology for Designing and Improving Enterprise Scenarios, *Proc. of the International Conference on Software Eng. Research & Practice*, CSREA Press, USA

[7] Jacobson I and Seidewitz E 2014 A new software engineering, *Communications of the ACM* **57** (12) 49

[8] Li G, Muthusamy V and Jacobsen H A 2010 A Distributed Service Oriented Architecture for Business Process Execution, *ACM Trans. on the Web* **4** (1)