A DESIGN BLUEPRINT FOR VIRTUAL KNOWLEDGE ORGANIZATIONS

Wajeeha Khalil
University of Engineering and Technology, Pakistan
wk_rehman1@yahoo.com

Erich Schikuta
University of Vienna, Austria
erich.schikuta@univie.ac.at

Abstract:
United we stand, divided we fall is a well-known saying. We are living in the era of virtual collaborations. Advancement on conceptual and technological level has enhanced the way people communicate. Problem nature has also been changed over the time. Today, e-Collaborations are applied to all domains possible. Extensive data and computing resources are in need and assistance from human experts is also becoming essential. Existing paradigms and technology are used to form Virtual Organization (VO), e.g. DataGrid, XSEDE, Facebook, MySpace, etc, but lack of standards remained a critical issue for last two decades. Our research endeavour focuses on developing a design blueprint for the Virtual Organization building process. The proposed standardization process is a two phase activity. The first phase provides requirement analysis and the second phase presents a Reference Architecture for Virtual Organization (RAVO). This form of standardization is chosen to accommodate both the technological and paradigm shift. For justification of our approach we developed the N2Sky system, a VO for the Computational Intelligence community, as a proof of concept implementation, which caught high attention by the research community.

Keywords: computers and society, virtual knowledge organization, reference architecture, computational intelligence, neural networks

1. INTRODUCTION

“Resource/Service as a utility" once a dream, is now a reality we are living with. Utility computing is not a new concept, but rather it has quite a long history. Among the earliest references is by John McCarthy[1]. The last two decades of Information Technology (IT) development has witnessed the specific efforts done to make this statement of John McCarthy a reality. Utility computing is providing basics for the current day resource utilization. Cluster, grids and now cloud computing have made this vision a reality. E-Collaborations also called virtual organizations have been evolved with the technological and paradigm shift. Cluster computing offered more centralized resource pool, while grid computing remained in need via hardware and computation cycles offerings to the scientific community. Grid computing models observed a deadlock after the introduction of cloud computing concepts. Based on the Pay-as-you-use criteria, cloud computing is still in early stage. Research efforts are going on to establish the basis of cloud computing as Every-thing-as-a-Service paradigm. Infrastructure providing resources as a utility must be dynamic, scalable and reliable. Orchestration of resources across the globe, named as Virtual Organization (VO) / Virtual Enterprise (VE) has been extensively deployed to achieve this target. Change in the hardware and software technology, computing paradigms algorithm and procedures, incorporation of knowledge rather information and data, made the concepts of VO vague.The research community recognizes VO with different names, e.g. collaboratories [1] [2], E-Science or E-Research [1,3], distributed work groups or virtual teams [1,4], virtual environments, virtual enterprise [5] and online communities [1,6]. Lack of standards for VO motivated us to provide a standard vision of E-collaboration incorporating both paradigm and technology shift as a Reference Architecture (RA) to achieve common objective(s) in any domain. Our research efforts also introduced new concepts regarding resources and stakeholder of a VO. To provide a standard for VO, we consider the existing technologies and paradigms. Service Oriented Architecture (SOA), Web 2.0 and Web 3.0 are the underlying technological platform, and computing paradigms include utility computing and cloud computing.

The layout of the paper is as follows: In the next section 2 we introduce the notion of Virtual Organization. In section 3 we present the principal steps for building a virtual organization, which we map to the process of our blueprint design. The generic Reference Architecture for Virtual Organizations, which we call RAVO, is laid out in section 4. For justification of our approach a proof-of-concept implementation based on RAVO, the N2Sky system, is presented in section 5.

2. VIRTUAL ORGANIZATION

A Virtual Organization is a non-physical communication model with the purpose of a common goal. It is built up from people and organizations with respect to geographical limits and nature. Additionally, a Virtual Organization provides typically a collection of logical and physical resources distributed across the globe. From a conceptual point of view a Virtual Organization resembles a detailed non-physical problems solving environment. A Virtual Organization can comprise a group of individuals whose members and resources may be dispersed geographically and institutionally, yet who function as a coherent unit through the use of a Cyber Infrastructure [1]. Virtual Organizations are typically enabled by and provide shared and, in most cases, real-time access to centralized or distributed resources, such as community specific tools, applications, data, instruments, and experimental operations. The different types of Virtual Organizations depend upon mode of operation, goals they achieve and life span for which they exists. Regardless of the objectives, Virtual Organizations possess some common characteristics. Virtual Organizations provide distributed access across space and time. Structures and processes running a Virtual Organization are dynamic. Email, video conferencing, telepresence, awareness, social computing and group management tools are used to enable collaboration among the participants [1]. Operational organizations are supported by simulation, database and analytical services. In daily life we come across many Virtual Organizations in terms of social networks as well (e.g. Facebook, MySpace). It can be phrased that soon every human on this earth will considered to be a part of some Virtual Organization serving its purpose in the said organization.

Generally, a Virtual Organization provides a global problem solving platform. It is difficult to specify or restrict the domain for which they are serving. Some advantageous roles played by Virtual Organizations are facilitator of access (BIRN [7], LEAD [8], nanoHUB [9]), caBig[10], LHC Computing

[1] John McCarthy, speaking at the MIT Centennial in 1961, “Architects of the Information Society, Thirty-Five Years of the Laboratory for Computer Science at MIT", edited by Hal Abelson

Grid, enhancer of Problem Solving Processes (TeraGrid [11]) and key to Competitiveness (GEON [12]). Virtual Organizations have served in the field of earthquake engineering (Southern California Earthquake Center (SCEC)), cancer research (Cancer Biomedical Informatics Grid (caBIG) [10]), climate research (Earth System Grid [13]), high-energy physics (Large Hadron Collider), and computer science. Other communities are now forming Virtual Organizations to study system-level science. These Virtual Organizations are addressing problems that are too large and complex for any individual or institution to tackle alone. It simply is not possible to assemble at a single location all of the expertise required to design a modern accelerator, to understand cancer, or to predict the likelihood of future earthquakes. Virtual Organizations allow humanity to tackle previously intractable problems [1].

3. BUILDING A VIRTUAL ORGANIZATION

The creation of a Virtual Organization is time consuming and should be a well-planned activity. In this section we will discuss Virtual Organization and technology from different perspectives. Both aspects are required to support each other. Technology provides the basic infrastructure for a Virtual Organization to exist. A Virtual Organization in turn places demands on Information Technology and shapes the evolution of technology. For the last decade the Virtual Organization is one of the most discussed collaboration environment; but still no standards exist. From this discussions we assume a step wise approach which is helpful in the creation of Virtual Organization. It can be separated in two phases which are detailed below.

3.1. Phase-1: Questions

The definition of a Virtual Organization starts with a series of questions, which are very critical in order to proceed. These questions (Qx) are listed in the following:

- Q1: Why to form a Virtual Organization? What are the reasons of an organization to create a Virtual Organization?
- Q2: What is the motivation behind participation? Why should other persons, institutes, service providers, etc. want to participate in a Virtual Organization?
- Q3: What services are offered by a Virtual Organization?
- Q4: How are these services fared? What is the type of the resources/business model?
- Q5: Who are the intended users? Who will eventually use and get benefited from this Virtual Organization?
- Q6: What is the life of (membership of) a Virtual Organization? Are temporal alliance or permanent participation expected?

3.2. Phase-2: Identification of Components

Based on these Q&A activity it is necessary to identify the building blocks of a Virtual Organization. Gannon [14] has identified main components of a Virtual Organization. These components are

- Common interest. The reason to form a Virtual Organization,
- Users. the participants of a Virtual Organization,
- Tools and services. This is a crucial part of a Virtual Organization, which maintains the overall working environment and saves the existing patterns to be reused in order to reduce time to solve similar problems. A Virtual Organization requires a collection of shared analysis tools (e.g. visualization tools and provenance tools).
- Data. A Virtual Organization contains two types of data, generally categorized as meta data and operational data that is being operated by tools.

The component identification process provides the basic building blocks. The designer of a Virtual Organization can decide what to be improved and further included in the design process. Also, each component can be given a unique definition by the designer in context of a Virtual Organization being created. Detailed information about creation, management and application area of Virtual Organizations is available in [1].

4. RAVO: A Reference Architecture for Virtual Organizations

According to Gerrit Muller [15] there are two simultaneous trends,

- Increasing complexity, scope and size of the system of interest, its context and the organizations creating system [15].
- Increasing dynamics and integration: shorter time market, more interoperability, rapid changes and adaption in the field, in a highly competitive market, for example cost and performance pressure [15].

These trends form basis for our proposed RA as well. VOs are developed as distributed system at multiple locations, by multiple entities, consisting of multiple applications by multiple vendors, merging multiple domains for providing solutions to multiple problems. RA comes in scene where the multiplicity reaches a critical mass triggering a need to facilitate product creation and life cycle support in this distributed open world [15]. We detail the RAVO in the subsequent sections. We define RAVO as “an open source template that does not only depict the architectural patterns and terminology, but also defines the boundaries where heterogeneous resources from different domains merge collaboratively into a common framework". A RA has a life span and is dependent on the target architecture and possibly other RAs. As guideline for our effort we closely analyzed the RA presented by SHAMAN (European Commission, ICT-216736), GERRIT MULLER [15] and NEXOF[16]. RAVO provides

- A common lexicon and taxonomy.
- A common (Architectural) vision.
- Modularization and complementary context.
- A layered approach (bottom-up).

4.1. Goal

We aim for developing a RA which allows for new forms of IT infrastructure coping with new collaborative processing paradigms, as grid computing and cloud computing. Thus we have to deliver an environment to allow for the new Internet of Services and Things accommodating the novel service stack, as IaaS, PaaS and SaaS. Architecture is classified into different layers according to the service each layer provides. Layered architecture is chosen because it helps to group different components (logical and physical) according to the degree of relatedness and required functionality.

4.2. Components and SPI based Framework

RAVO is based on SPI model. Layered approach is used to achieve the goal of providing all the resources as a service. Layers are distributed into 3 broad categories, IaaS, PaaS, SaaS. Picture 1 presents the framework for VO using the SPI model. The layers are distributed into 3 broad categories, IaaS, PaaS and SaaS thus resulting in XaaS.

Software as a Service Layer

In context of RAVO, SaaS is composed of a Service layer. It contains Domain Specific Applications (DSA) accessible by all users. DSAs are the combination of several user interfaces and Business Models found in the VO layer. Users, who only use the platform to solve their domain specific problems and do not contribute to the VO, find an entry point at this layer.

- *Service Layer*: It has open source, downloadable software, categorized in domains. The Service layer packages several services provided by the VO layer to be subscribe-able entities.

Platform as a Service Layer

In RAVO two layers, namely VO layer and Abstract layer, cover PaaS.

- *Abstract Layer*: This layer is composed of essential tools which enable the whole framework to be exploited in a dynamic manner. Each tool provides its own functionality, its own user interface description [17], as well as an abstract API (identical for each tool) to access the resource in Factory layer.

- *VO Layer*: This layer is the entry point for user. It provides the realization of the user interface description and defines a business model on top of the Abstract layer to set usage cost according to usage statistics.

Infrastructure as a Service Layer

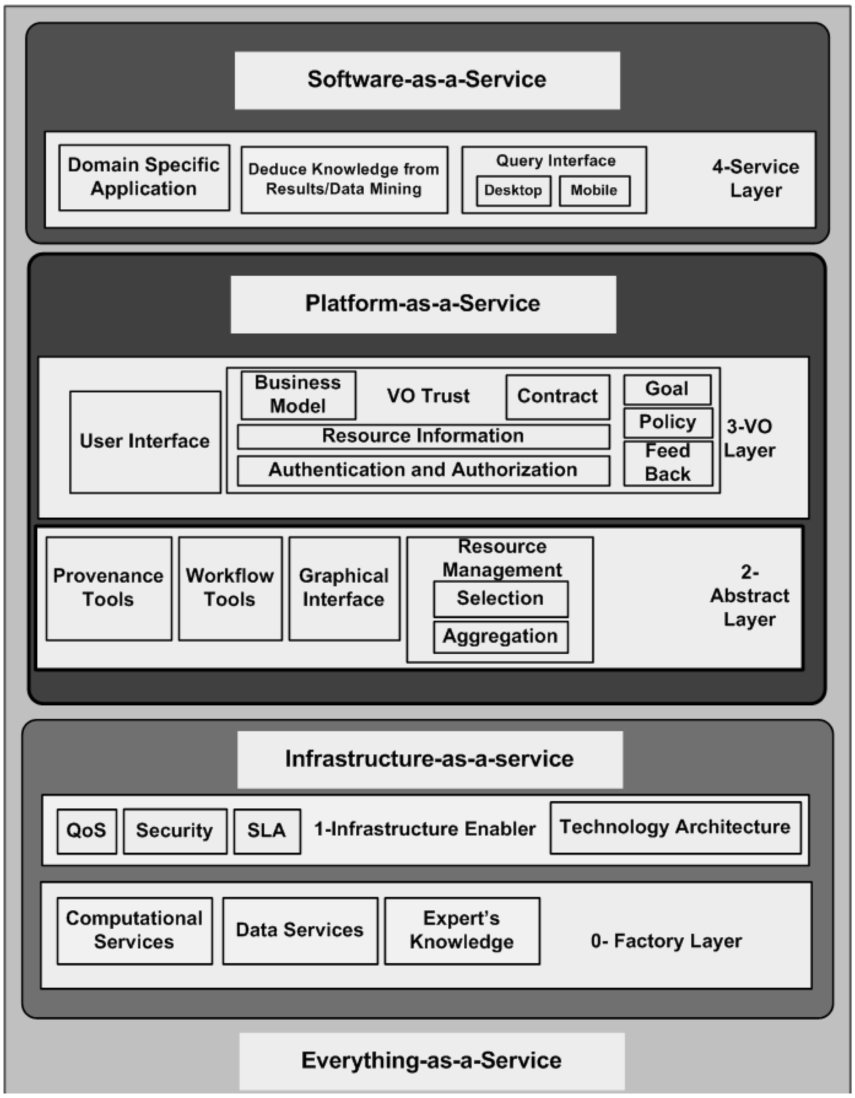
In RAVO logical and physical resources are considered to be the part of IaaS. This part consists of two sub-layers in RAVO: Factory layer and Infrastructure Enabler layer. Only users with administrative rights have access to this layer.

- *Factory Layer*: Belongs to the IaaS category and contains resources for RAVO. Resources are described as physical and logical resources. Physical resources comprise of hardware devices for storage and computation cycles in a distributed manner. Logical resources contain expert's knowledge that supports the problem solution activity.
- *Infrastructure Enabler Layer*: Allows access to the resources provided by the Factory layer. It consists of protocols, procedures and methods to contact the desired resources for a problem solving activity.

Everything as a Service Layer

All layers are providing their functionality in a pure Service-Oriented manner so we can say that RAVO is XaaS.

Picture 1: Generic RAVO Layered Architecture



5. PROOF-OF-CONCEPT: THE N2SKY SYSTEM

As proof-of-concept of our approach we used RAVO as a design blueprint for implementing a cloud based VO for Neural Network Research, namely, N2Sky. N2Sky caught high attention by the research community as stated in [20] and [21].

We based the development of N2Sky on the blueprint provided by RAVO and produced a concrete instance out of our proposed standard [18. We divide this comparison in 3 levels. First, Requirement Analysis Phase that defined boundaries of N2Sky. Second, Component Identification Phase which made it easy to identify the components of N2Sky and also choose between optional and mandatory components. Third, Implementation Phase that reveals how technology independence, XaaS and layered distribution of components made it helpful to implement the system.

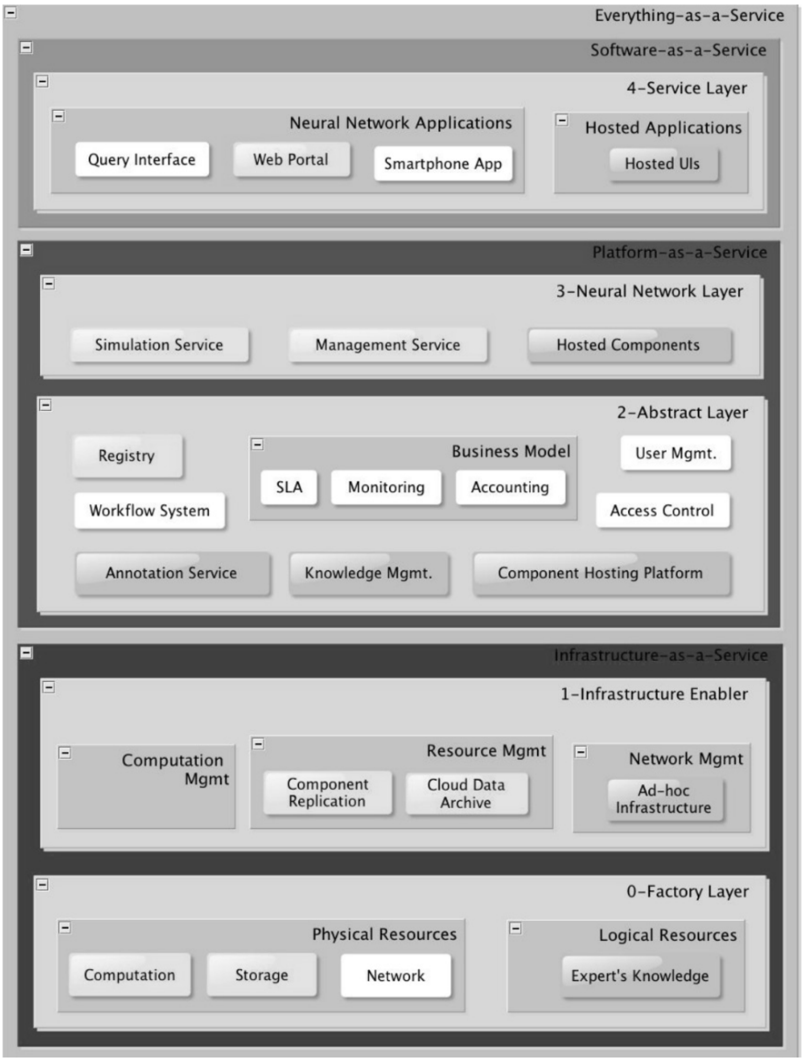
5.1. Requirement Analysis in terms of RAVO

In the previous section we detailed a series of questions which must be answered by the responsible authorities for creating a VO. N2Sky utilizes this pattern for defining the requirements boundary of the system.

5.2. Component Identification in terms of RAVO

N2Sky is a layered architecture instantiated from RAVO. The N2Sky layers are shown in Picture 2. It consists of 3 layers, namely SaaS, PaaS and IaaS. These layers have sublayers similar to RAVO. Each layer has some components which are either mandatory or optional.

Picture 2: Specific N2Sky Layered Architecture



5.3. Interface Specification in terms of RAVO

Section 4 presented interface specification for components. Here, we analyze how these interface specifications were used in N2Sky. We compare the underlying framework RAVO with its instantiation as N2Sky, in a top-down fashion. We start with SaaS layer.

SaaS Layer Comparison

SaaS layer of RAVO consists of optional and mandatory components. Choice of components and decision on their status (mandatory and optional) is open for the developers. The inclusion of components is dependent on the requirement definition by the stakeholders. SaaS Layer has one layer, named Service layer. Here tables are included for the sake of comparison.

- Query Interface: RAVO proposes Query Interface as a mandatory component at Service Layer. In N2Sky, Query Interface is also included as a mandatory component.
- Domain Specific Application (DSA): DSA is a mandatory component. N2Sky has a simulation service but at Neural Network layer (sub layer of PaaS). N2Sky includes DSA as NN specific applications. N2Sky is planned to include NN specific applications. The Simulation Service provides the creation, training and simulation of neural objects which in turn are instances of NN paradigms. Currently, Simulation Services are provided at NN Layer of N2Sky.
- Data Mining Tools: Data mining tools are an optional component of RAVO. N2Sky has not included this option.

N2Sky also has one layer, named Service Layer (similar to RAVO). Extended components included at Service Layer in N2Sky are:

- Web Portal: N2Sky Web Portal is a mandatory component.
- Smartphone APP.

- Hosted UI.

PaaS Layer Comparison

PaaS layer is composed of two layers, namely VO Layer and 2-Abstract Layer. Component Specification is detailed below. In N2Sky PaaS consists of 3-Neural Network Layer and 2-Abstract Layer.
VO Layer comparison. In RAVO VO layer has the following components:

- VO Trust: Mandatory component of VO, which is responsible for enabling resources, defining policies to achieve a goal. It has several components and is extendable according to the need and requirement of stakeholders. N2Sky has distributed Trust component into different modules. In N2Sky, Neural Network Layer has a Management Service component to serve the purpose. Other components are available at Abstract layer namely, Business Model with SLAs and Accounting.
- User Interface: User Interface is a mandatory component for solving problem utilizing VO PaaS utility. It provides an interface to interact with the VO. N2Sky also realizes this component as a part of Web Portal.

Extended Component of N2Sky:

- Hosted Component: Provides and interface for components hosting platform.
- Simulation Service: Already described in Service Layer comparison. It is a mandatory component that is part of Neural Network Layer of N2Sky.

Abstract Layer Comparison. RAVO and N2Sky both have this sub layer named 2-Abstract Layer. Components of these layer in RAVO and N2Sky are compared.

- Resource Management: Resource Management is a mandatory component of Abstract Layer. It provides a mechanism to select and aggregate resources for a problem solving activity. Depending upon the underlying technology, VO developers can deploy different resource management tools. In N2Sky resource management is achieved via mandatory Registry component.
- Workflow Tools: N2Sky also has a Workflow System under development.
- Provenance Tools: Provenance Tools are proposed in RAVO but they are not included in N2Sky.
- Graphical Interface: A mandatory components which facilitates interaction with VO easier and helps user to get results in an understandable format. It also assists user in formulating queries and browsing in VO environment. In N2Sky Graphical Interface is implemented as a Web portal described earlier.

Extended Components supporting VO Trust (as proposed in RAVO) Functionality:

- Controlling and Accounting: This component along with SLA component serves as a Business Model. In RAVO Business Model is optional.
- User management.
- Access Control.
- SLA.
- Annotation Service.
- Knowledge Management: It refers to Expert's Knowledge of RAVO defined at Factory level.
- Component Hosting Platform.

IaaS Layer Comparison

IaaS layer is composed of 1-Infrastructure Enabler Layer and 0-Factory Layer. This layer forms the fabric of RAVO. All the resources are available in Factory Layer and are exploited through Infrastructure Enabler Layer, which brings an open choice for the developers for underlying technology. QoS, Service Level Agreement (SLA), Security, Fault tolerance and Disaster management are aspects to be considered in particular. Further extension can be done by developers.

N2Sky also has an Infrastructure Enabler Layer. It contains following components.

- Data Archive: Implemented as a mandatory component of N2Sky.
- Component Replication Service.

Factory Level of RAVO is also instantiated in N2Sky. It has following components in RAVO

- Resource Catalogue: Resource Catalogue module is an extension of Resource Management Component. It is a mandatory components. It keeps information about resources which is of interest to VO. In N2Sky this task is achieved by Registry component.
- Computational Services: RAVO offers Computational Services as a mandatory component. In N2Sky this component is realized by Component Replication Service. It is a mandatory component which act as N2Sky Paradigm Archive Service.
- Data Services: This component of RAVO is realized by N2Sky as a part of Infrastructure Enabler Layer.
- Expert's Knowledge: N2Sky implements this component of RAVO as Knowledge Management as a subcomponent of Abstract Layer.

For more information on the evaluation of the RAVO process building N2Sky see [19].

6. CONCLUSION

We developed a service-stack pattern to start a Virtual Organization from scratch and also presented a design blueprint for collaborative environments. Emphasis is on flexible and simple interface for interaction from the user perspective. It supports addition of tools and application to the Virtual Organization environment. We give guidelines for building a domain specific Virtual Organization for Computational Intelligence community and can extend it according to our requirements.

We justified our approach by designing the N2Sky system, which realizes the neural network as a service paradigm. N2Sky is a prototype system with quite some room for further enhancement. Ongoing research is done in the following areas:

- We are working on an enhancement of the neural network paradigm description language ViNNSL [20] to allow for easier sharing of resources between the paradigm provider and the customers. We are also aiming to build a generalized semantic description of resources for exchanging data.
- Parallelization of neural network training is a further key for increasing the overall performance. Based on our research on neural network parallelization [21, 22, 23] we envision an automatically definition and usage of parallelization patterns for specific paradigms.
- Key for fostering of cloud resources are service level agreements (SLAs) which give guarantees on quality of the delivered services. We are working on the embedment of our research findings on SLAs [24, 25, 26] into N2Sky to allow for novel business models [27, 28, 29, 30] on the selection and usage of neural network resources based on quality of service attributes [31].

REFERENCE LIST

1. C. Kesselman, I. Foster, J. Cummings, K. A. Lawrence, and T. Finholt, „Beyond being there: A blueprint for advancing the design, development, and evaluation of virtual organizations.," tech. rep., May 2008.
2. W. Wulf, „The collaboratory opportunity," Science, vol. 261, no. 5123, pp. 854{855, 1993.
3. T. Hey and A. E. Trefethen, „E-science, cyberinfrastructure and grid middleware services," Science, vol. 308, pp. 817{812, 2005.
4. M. O'Leary and J. Cummings, „The spatial, temporal, and con_gurational characteristics of geographic dispersion in teams," MIS Quarterly, vol. 31, pp. 433{452, March 2007.
5. B. Katzy, C. Zhang, and H. Lh, „Reference models for virtual organisations," in Virtual Organizations (L. M. Camarinha-Matos, H. Afsarmanesh, and M. Ollus, eds.), pp. 45{58, Springer US, 2005.
6. P. Jennifer, Online communities: Designing usability and supporting sociability. Chichester, England: John Wiley & Sons Ltd, 2000.
7. I. Foster and C. Kesselman, The grid: blueprint for a new computing infrastructure. Morgan Kaufmann, 2004.
8. Accessible at: http://portal.leadproject.org/gridsphere/gridsphere.
9. G. Klimeck, M. McLennan, M. Mannino, M. Korkusinski, C. Heitzinger, R. Kennell, and S. Clark, „NEMO 3-D and nanoHUB: Bridging Research and Education," in Nanotechnology, 2006. IEEE-NANO 2006. Sixth IEEE Conference on, vol. 2, pp. 441{444, IEEE, 2006.

10. A. C. von Eschenbach and K. Buetow, „Cancer Informatics Vision: caBIG," Cancer informatics, vol. 2, p. 22, 2006.
11. C. Catlett, W. Allcock, P. Andrews, R. Aydt, R. Bair, N. Balac, B. Banister, T. Barker, M. Bartelt, P. Beckman, et al., „Teragrid: Analysis of organization, system architecture, and middleware enabling new types of applications," HPC and Grids in Action, Amsterdam, 2007.
12. U. Nambiar, B. Ludaescher, K. Lin, and C. Baru, „The GEON portal: accelerating knowledge discovery in the geosciences," in Proceedings of the 8th annual ACM international workshop on Web information and data management, pp. 83{90, ACM, 2006.
13. I. Foster, „Service-oriented science: Scaling E-Science impact," in IAT '06: Proceedings of the IEEE/WIC/ACM international conference on Intelligent Agent Technology, (Washington, DC, USA), pp. 9{10, IEEE Computer Society, 2006.
14. D. Gannon, „Building virtual organizations around super computing grids and clouds." Indiana University and Tera Grid Infrastructure Group, 2008.
15. G. Muller, „A Reference Architecture Primer," Eindhoven Univ. of Techn., Eindhoven, see http://www.gaudisite.nl/referencearchitectureprimerslides.pdf.
16. Nexof RA." Accessible at: http://www.nexof-ra.eu/.
17. A. A. Huqqani, P. Beran, X. Li, and E. Schikuta ., “N2Cloud: Cloud based neural network simulation application," in In Proceedings of the International Joint Conference on Neural Networks 2010 (WCCI 2010), Barcelona, Spain, 2010.
18. E. Mann, “N2Sky - a cloud-based artificial neural network resource," master thesis, Faculty of Computer Science, university of Vienna, A-1090 Vienna, Austria, 2013.
19. W. Khalil, Reference Architecture for Virtual Organization. Ph.D. thesis, Faculty of Computer Science, university of Vienna, A-1090 Vienna, Austria, 2013.
20. Peter Paul Beran, Elisabeth Vinek, Erich Schikuta, and Thomas Weish¨aupl. ViNNSL - the Vienna Neural Network Specification Language. In Proceedings of the International Joint Conference on Neural Networks, IJCNN 2008, part of the IEEE World Congress on Computational Intelligence, WCCI 2008, pages 1872–1879, 2008.
21. Thomas Weishäupl and Erich Schikuta. Cellular Neural Network Parallelization Rules. In CNNA ’04: Proceedings of the 8th IEEE International Biannual Workshop on Cellular Neural Networks and their Applications, Los Alamitos, CA, USA, 2004. IEEE Computer Society
22. Hannes Schabauer, Erich Schikuta, and Thomas Weishäupl. Solving very large traveling salesman problems by SOM parallelization on cluster architectures. In Sixth International Conference on Parallel and Distributed Computing Applications and Technologies (PDCAT05), page 954–958, Dalian, China, 2005. IEEE Computer Society.
23. Erich Schikuta and Claus Weidmann. Data parallel simulation of self-organizing maps on hypercube architectures. In Workshop on Self-Organizing Maps (WSOM’97), page 4—6, Helsinki, Finland, 1997.
24. Irfan Ul Haq, Ivona Brandic, and Erich Schikuta. SLA validation in layered cloud infrastructures. In Economics of Grids, Clouds, Systems, and Services, 7th International Workshop, GECON’10, volume 6296 of LNCS, page 153–164, Ischia, Italy, 2010. Springer.
25. Irfan Ul Haq and Erich Schikuta. Aggregation patterns of service level agreements. In Frontiers of Information Technology (FIT’10), Islamabad, Pakistan, 2010. ACM.
26. Elisabeth Vinek, Peter Paul Beran, and Erich Schikuta. Classification and composition of QoS attributes in distributed, heterogeneous systems. In 11th IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing (CCGrid 2011), Newport Beach, CA, USA, May 2011. IEEE Computer Society Press.
27. Thomas Weishäupl, Flavia Donno, Erich Schikuta, Heinz Stockinger, and Helmut Wanek. Business in the grid: BIG project. In Grid Economics & Business Models (GECON 2005) of Global Grid Forum, volume 13, Seoul, Korea, 2005. GGF.
28. Erich Schikuta, Flavia Donno, Heinz Stockinger, Helmut Wanek, Thomas Weishäupl, Elisabeth Vinek, and Christoph Witzany. Business in the grid: Project results. In 1st Austrian Grid Symposium, Hagenberg, Austria, 2005. OCG.
29. Thomas Weishäupl and Erich Schikuta. Towards the merger of grid and economy. In 3rd International Conference on Grid and Cooperative Computing (GCC04), volume 3252/2004 of LNCS, page 563–570, Wuhan, China, 2004. Springer.
30. Thomas Weishäupl, Christoph Witzany, and Erich Schikuta. gSET: trust management and secure accounting for business in the grid. In 6th IEEE International Symposium on Cluster Computing and the Grid (CCGrid’06), page 349–356, Singapore, 2006. IEEE Computer Society.

31. Irfan Ul Haq, Rehab Alnemr, Adrian Paschke, Erich Schikuta, Harold Boley, and Christoph Meinel. Distributed trust management for validating sla choreographies. In Grids and Service-Oriented Architectures for Service Level Agreements, pages 45–55. Springer US, 2010.