An Epistemological Basis for Alignment in Enterprise Architecture Views

Srinivas Chemboli
Research School of Computer Science
Australian National University
Canberra, Australia
srinivas.chemboli@anu.edu.au

ABSTRACT

In several Enterprise Architectures (EA), the views at the CXO-levels may often be at variance with the views provided to mid-tier and operations levels. As a result of this misalignment, organizations may not be able to fully realize the intended enhancements to business management. Alignment of architecture views with one another, as well as consistency with the overall architecture description of the enterprise can be achieved by developing an epistemological basis for enterprise architecture management. In this paper, an approach to develop such a basis is proposed through use of Omnispective Analysis and Reasoning (OAR), an epistemic framework for managing intellectual concerns. This approach focuses on ensuring that the representation of enterprise concerns captures the necessary information regarding their applicability, correctness and completeness for a given problem scenario, and facilitates development of localized ontologies for describing the behavior of component systems and the interactions between them. An example of architecture view alignment is presented to illustrate the approach.

Categories and Subject Descriptors

H.1.0 [**Models and Principles**]: General; H.4 [**Information Systems Applications**]: Miscellaneous—*enterprise architectures*

General Terms

Theory

Keywords

Enterprise Architecture, Omnispective Analysis and Reasoning, Intellectual Concerns, Architecture Views

1. INTRODUCTION

The main focus of Enterprise Architectures (EA) is to provide views suitable for various levels of management in an enterprise — ranging from overall governance to systems implementation level [10, 6, 1, 9]. However, in practice, the abstract views as presented to the CXO–level may often be at variance with the views provided to mid–tier and operations level personnel [9], resulting in the EA not yielding the intended enhancements to business management [10]. This 'misalignment' in architecture views may result in missed opportunities of development and growth, disaffection, and simmering discontent across ranks in an organization, ultimately affecting productivity and performance, and in extreme cases leading to total breakdown of governance in the enterprise. In addition, the diversity of interacting component systems makes it difficult to develop cohesive enterprise–wide 'ontologies', or fix control and execution contexts, resulting in the EA and associated IT activities being bogged down with outdated and inconsistent models of systems and processes [8].

A novel approach to address this issue by developing an epistemological basis for aligning EA views is proposed in this paper. This approach employs Omnispective Analysis and Reasoning (OAR) [4], a framework for managing intellectual concerns.

This paper is organized as follows. An overview of View Alignment (VA) in EA Management (EAM) is presented in section 2. Application of the OAR framework to develop an epistemological basis for VA is described in section 3. An illustration of the approach for aligning architecture views is presented in section 4. Summary and conclusions are presented in section 5. A brief summary of the OAR framework from earlier publications [2, 4, 3] is given as an appendix.

2. ALIGNMENT OF EA VIEWS

The ISO/IEC/IEEE 42010 standard [7] defines architecture and "addresses the creation, analysis and sustainment of architectures of systems through the use of architecture descriptions." Addressing stakeholder concerns is realized through a number of Architecture Views of the system. The rules and processes for generating the views are prescribed by an Architecture Viewpoint (AVP). Each Architecture View (AV) is designed in such a way as to clearly present an identified and pre–determined set of stakeholder concerns. The Architecture Framework (AF) must clearly spell out which are the concerns, the stakeholders who have these concerns and the viewpoints that capture these concerns. While the standard indicates what an enterprise architecture 'has to be' in order to be effective, it does not specify the means for building an effective architecture, leaving it entirely up to the practitioners. In a comparative analysis of

7th International Conference on Software Engineering (CONSEG),
15th-17th November, 2013 @ Pune, India

six architecture frameworks, Tang, Han, and Chen [12] find that the level of detail required in an architectural model is generally not specified and the architecture rationale is not a mandatory part of the model; consequently, the models cannot be verified or traced.

The exercise of enterprise architecture development consists of four steps: *data collection*, *preliminary view generation*, *review* and *revising and publishing* the views [5]. Building an architecture is steeped in a variety of implementation details. All the 'nouns', 'verbs' and 'connective phrases' of architecture viewpoints, procedures and outcomes are geared to low–level procedural constructs. Epistemological considerations do not figure in architecture frameworks.

Building an architecture description to provide a faithful representation of an enterprise is often a prolonged activity which involves interaction between a large number of internal personnel as well as external experts. By the time all the component systems of the enterprise are modeled, some of them might have changed or could have been retired from the enterprise. Hence a 'snapshot' description of the enterprise architecture has a limited span of usefulness unless continuous revision and update of the models and their descriptions is carried out. Moreover, since the enterprise is not static or fixed but is dynamic and changing, the EA activity often starts out with a 'wrong' ontology with expectation that a 'correct' ontology would evolve over time [6]. Consequently, the architecture description is of uncertain veracity and the views generated therefrom often result in an "architecture view quagmire" [10].

As depicted in fig. 1, these views may not be in alignment to the architecture description, or it may even be possible that the architecture description is 'broken' due to several areas of inconsistencies and gaps. Highlighting the importance of designing EAs for optimal alignment and mutual compatibility of component entities, Lankhorst et al. [9] observe that current practices of EA management do not provide such architecture alignment. They further emphasize that consistency between the models describing the processes, applications and subject domains is rarely achieved and contend that the current practices of architecting methods and notations are "too complex and inflexible to be used in the current business environment," and could be further compounded by inconsistencies between local optimization criteria used in the architectures of component systems and global optimization criteria applicable to the overall business architecture.

Since the processes for specifying the rationale of the models used are not mandated, validation and verification depend indirectly on the strategies, policies, present–state and vision of the enterprise, which are incorporated in the architecture meta–model. This is not an 'open feature', no provision exists to verify that the concerns captured in the specification of the viewpoints are correct, appropriate, or are in alignment with the architecture description. If the enterprise description is left to just evolve and present a 'quagmire of views' not having proper referential validation with respect to the basic entities and key concerns of the enterprise, the resulting state of confusion arising from misalignment of views presented at different levels would be akin to "Buffaloes Wallowing" [11].

3. ALIGNING VIEWS WITH OAR

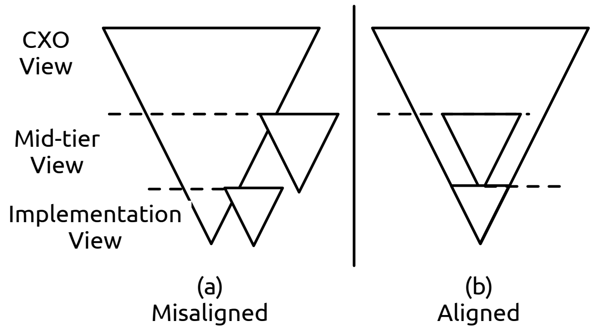

Figure 1: Alignment of views. © 2012 Chemboli, S.

The processes of formulating viewpoints and generating the views can be considered as workflows. The input information for orchestrating the workflow is contained in the architecture description and the ontology of the architecture. The 'scientific foundation' and 'theory' of an enterprise comprise of the appropriate set of principles, rules, conventions, business theories, models etc. used in the various component systems of the enterprise. The expected and desirable aspects of the interactions between the component systems are also to be considered in this.

In applying OAR, the analysis, design and operation of a system is considered as a scientific workflow formulation and solution specification [3]. In the OAR framework, a scientific workflow is considered to be the "representation of any logical, systematic and repeatable inquiry, investigation and corresponding set of actions." The inquiry and investigation covers exploratory as well as in–depth study of the system and has the basis in the Foundations and Theory of a Science. *Science*, in this context, stands for a body of knowledge and information developed from empirical studies and organized with referential transparency. The framework facilitates capturing intellectual concerns and mapping them to the formulation and implementation of a workflow, and can therefore be employed for aligning the outcomes of a system to its design and operation [2]. In a similar manner, the OAR framework can also be applied to enterprise architecture to provide alignment of the views with the architecture description and stakeholder concerns by explicitly capturing the context of the outcomes.

A *concept–level view* of enterprise architecture according to OAR involves collecting data sufficient to provide adequate and clear description of the enterprise (*omnispection*), generating preliminary views and refining them (*formulating models* of the views), and publishing the views providing the outcomes (*implementation*).

First, all identified concerns and entities of the enterprise are abstracted as *recipes*. These recipes are subjected to 'contextualization' to ascertain their well–formedness and relevance to the description of the enterprise and the component systems, and are organized in external shelves which are suitably categorized and labeled for reference. Identifying stakeholders and stakeholder concerns forms part of the activity of managing recipes in shelves.

Developing and maintaining exhaustive enterprise–wide ontologies is both time–consuming and error-prone [8]. This burden is alleviated by considering the enterprise–wide ontology "as an aggregation of numerous localized ontologies, each of which can be further adapted, evolved, retired and replaced as appropriate to changes" [3]. The external shelves encapsulate available recipes in the form of localized ontologies suitable for formulating the models of component systems and stakeholder concerns in the enterprise. Hierarchical representation of recipes at concept, model and implementation levels, their contextualization and management in shelves constitute a deliberate exercise of providing an epistemological basis.

The second step involves formulating the viewpoints that specify the views for the various stakeholders. The localized ontology sufficient for this purpose may be realized from the external shelves described above. This step is the process of building the architecture meta–model.

The third step is to generate the viewpoints for different stakeholders. To obtain a viewpoint appropriate for a given stakeholder, *concern refinement* is carried out as per the prescriptions of the meta–model (which provides constraints) to select relevant recipes into the problem–domain shelf. The solution shelf for the viewpoint is realized through *context refinement*. The viewpoints so generated are considered as recipes in a viewpoints external shelf for further use.

The fourth step of generating the views is analogous to the workflow for generating the viewpoint. The problem–domain shelf contains the relevant viewpoint, tools and constraints as prescribed by the meta–model. The views generated are collected in a views shelf for reuse.

A conceptual representation of the OAR process for realizing the views is illustrated in fig. 2.

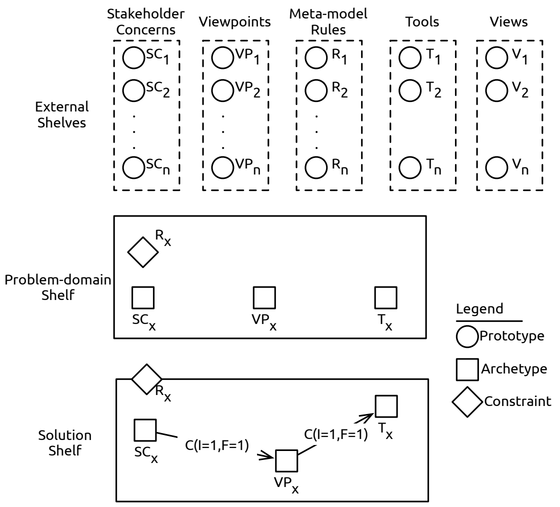

Figure 2: OAR process for generating views. © 2012 Chemboli, S.

The issues of alignment are resolved because of the referential transparency provided by the intellectual concerns that are verified, validated and encapsulated in the recipes. In other words, the epistemological basis in tandem with concern and context refinement provides verification, validation and alignment as default realizations. Contextualizing the recipes representing divergent and conflicting interests reveals their relative importance for the enterprise and hence can form the basis for taking informed management decisions. By considering the influence of alternate scenarios on the key issues of the enterprise, an optimum resolution of conflicts may be arrived at. In this process fixing belief by the Method of Science encourages rational decisions.

4. ILLUSTRATION

An illustration of how misalignment may arise between views at different levels in the enterprise and the use of OAR to develop a solution specification for aligning views is presented in this section.

XYZ Enterprise is a contractor specializing in services to City Council CC_1. XYZ has a well–defined process for assembling and disseminating information packages in suitable formats (currently 'Format RF_1') to prospective clients, which also happens to be the acceptable format for CC_1. In the enterprise architecture for XYZ, the views corresponding to the 'information packages workflow' are presented at the CXO, Business Manager (BM) and Implementation Personnel (IP) levels. Following the OAR methodology, the external shelves for analyzing this workflow may include various stakeholder concerns, report formats and different levels of views. These are illustrated in fig. 3.

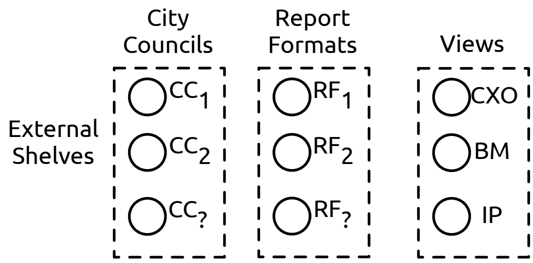

Figure 3: External shelves and recipes for analyzing the current information package workflow. © 2012 Chemboli, S.

In order to analyze the current state of the enterprise, *concern refinement* is carried out to select relevant recipes into a *problem–domain shelf* for the information package workflow views. Since the current processes utilize format RF_1 in dealing with CC_1 at the IP level, the corresponding recipes are selected in the problem–domain shelf. The CXO view primarily deals with ensuring that an 'appropriate format' ($RF_?$) is used in interacting with a corresponding city council ($CC_?$). The resulting problem–domain shelf is illustrated in fig. 4.

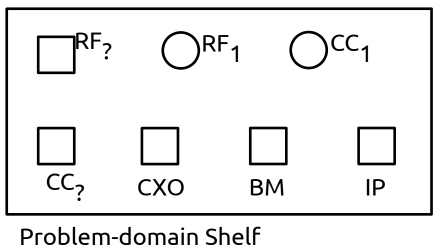

Figure 4: Relevant recipes in the problem–domain shelf for the information package workflow views. © 2012 Chemboli, S.

Context refinement is now performed to define a solution specification for the view at the three levels. In the CXO level view, for a given city council $CC_?$, a suitable format $RF_?$ is to be used. Hence the city council's requirements are considered a constraint in determining the suitable format, i.e., $CC_?\ C(I=1, F=1)\ RF_?$.

At the BM level, the view indicates that format RF_1 is to be used for CC_1, i.e., CC_1 acts as a constraint, fixing the information package format: $CC_1\ C(I=1, F=1)\ RF_1$.

Finally, insofar as the IP view is concerned, the enterprise practice is to only use format RF_1; preferences of the recipient (CC_1) have no influence on the information package format: $RF_1\ C(I=0, F=1)\ CC_1$, and RF_1 is an enterprise–level constraint. The corresponding solution specifications are illustrated in fig. 5.

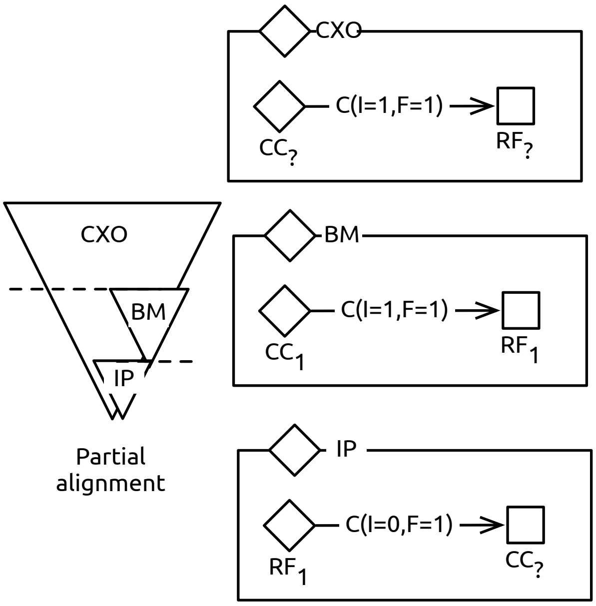

Figure 5: Solution specifications at the three levels. The views are only partially aligned across the enterprise. © 2012 Chemboli, S.

It can be seen that as long as XYZ deals with city councils which accept format RF_1, the business process fulfills the desired aim, irrespective of the misalignment in the views. In the current situation, XYZ does not possess any capability to suitably adapt the information package to a different format. If a city council requires a format different from RF_1, the effects of this misalignment will manifest. The CXO view does not expose this shortcoming; the BM view does not have information about which format to adapt to; and the IP view specifies only RF_1 as the format to be used.

In the absence of the above analysis, this 'gap' in the enterprise capability may not be revealed or discerned until it results in a process failure. For instance, if XYZ decides to expand its client base by sending an information pack to city council CC_2 which requires format RF_2, the package will be sent in format RF_1. Since this is not in accordance with the requirements of CC_2, the information package may be sent back or ignored, resulting in wasted effort. However, formulating the solution specifications using OAR makes explicit this misalignment and mismatch of views across the enterprise and also points out the need to augment the capability for additional formats. By adapting the information package workflow specifications to customize the report format as per city council requirements, alignment of the CXO, BM and IP views is achieved. The evolved solution specifications for aligned views are illustrated in fig. 6.

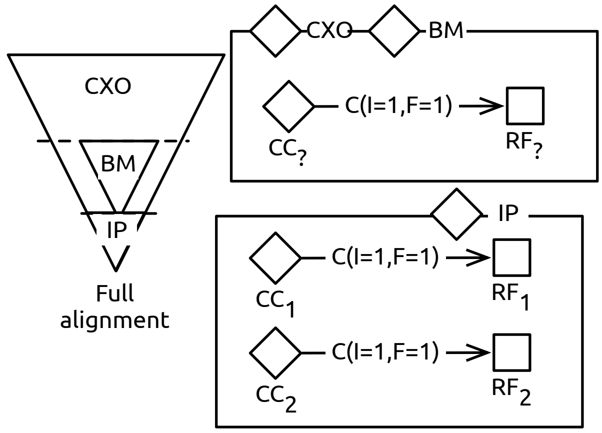

Figure 6: Aligned views. © 2012 Chemboli, S.

5. SUMMARY AND CONCLUSIONS

A novel approach for providing an epistemological basis for managing alignment of views in Enterprise Architectures (EA) has been proposed in this paper. This approach employs Omnispective Analysis and Reasoning (OAR), a framework for managing intellectual concerns.

In enterprise architectures, there is often a variance between the views presented across different tiers. This mismatch and misalignment arises mainly because the development of an EA is in many cases a protracted exercise, while the enterprise itself is dynamic and changing. As a result, the models which form the basis for the Architecture Description may be outdated, incorrect or developed from partially evolved ontologies. The misalignment may be further compounded by discrepancies between local and global optimization criteria of the component systems and the overall business architecture. Since an epistemological basis is not mandated in the architecting process, often the correctness, appropriateness and the alignment of the viewpoints with the AD cannot be verified, and each case of view misalignment may require ad–hoc measures.

The OAR framework facilitates the capture of intellectual concerns and mapping them to the formulation and implementation of a workflow, making it suitable for aligning the outcomes of a system to its design and operation. The hierarchical representation of enterprise concerns in recipes at one or more of the concept, model and implementation levels and contextualizing and managing these recipes across various external, problem–domain and solution shelves forms a deliberate exercise that provides an epistemological basis of the architecture description, viewpoints and views. The epistemological basis, in tandem with the OAR processes of concern and context refinement, ensures verification, validation and alignment of architecture views across multiple tiers in the enterprise.

An example of aligning views across different levels in an enterprise is presented to illustrate the approach. Analysis using the OAR framework is used to reveal an existing gap in business capability which otherwise would have been known only in the instance of a process failure. Formulating the solution specification makes explicit any misalignment in views and also highlights the need to augment enterprise capability. The choice of a simple and concise example to illustrate the application of OAR to view alignment should not be construed as any limitation imposed by the framework. The choice of a simple, pithy and lucid example without involving too many intricacies of the business domain is a deliberate exercise to clearly highlight how misalignment may be recognized and managed using OAR. It is the author's belief that the ideas developed in this paper show promise for enhancing the practice of enterprise architecture management and further studies in relation to large–scale enterprises are desirable to fully evaluate the scope and benefits of applying OAR.

ACKNOWLEDGMENTS

This work is based on initial research supported by the Australian National University, and the Commonwealth of Australia, through the Cooperative Research Centre for Advanced Automotive Technology. The author would like to thank Dr Clive Boughton for his guidance and suggestion to examine Enterprise Architectures through OAR; and to Dr V. Ganesh for his many helpful comments.

APPENDIX: OAR FRAMEWORK

A brief overview of the OAR framework is presented here, based on earlier publications [2, 4, 3].

Omnispective Analysis and Reasoning (OAR) is an epistemic framework for managing intellectual concerns in scientific workflows. Concerns in a problem domain are managed at three levels of abstraction (fig. 7): *Concept*, *Model* and *Implementation* in terms of *ukes* (unit knowledge entities), which are grouped into convenient *recipes*. The concept–level, which is the highest level of abstraction, represents the 'meaning' and 'understanding' of a concern as it exists in relation to the domain of study. The model–level abstraction stands for the way the concern is represented in terms of simpler and related concepts and constructs of the domain. Implementation–level is the lowest level and considers the concern in terms of practical details like measurements and data collection and processing.

The process of abstraction effectively separates low–level implementation details and enables capture and mapping of the underlying science to the workflow processes. Each recipe may now be considered at a level appropriate to the outcome of the problem analysis.

The *context* of a recipe is formally represented by $\mathsf{C(F,I)}$ in terms of two attributes: *firmness* (F) which is native to the recipe and *influence* (I) which is specified in relation to another recipe or the problem situation. A well–formed and explicitly defined recipe has high firmness ($\mathsf{F} = 1$). A 'best practice' recipe for a problem situation is considered to exert a high influence ($\mathsf{I} = 1$). Depending on the context evaluated, OAR recipes are categorized as *prototypes*, *archetypes* and *constraints*. A prototype is any recipe that is available without particular consideration of applicability or robustness; an archetype is a prototype that is identified

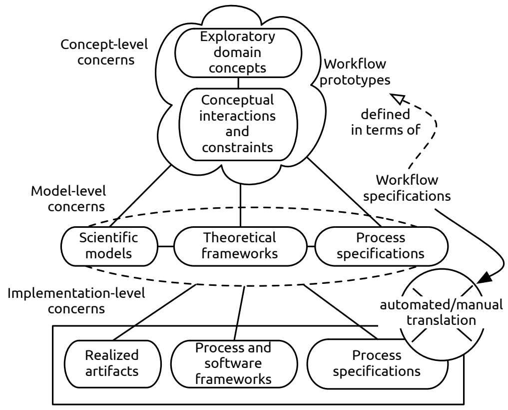

Figure 7: Hierarchy of concerns in OAR (from [3]).

as an exemplar or 'best practice'; an archetype that imposes strict criteria on the OAR specification is regarded as a constraint.

Recipes are managed in unordered collections designated as *shelves* (fig. 8).

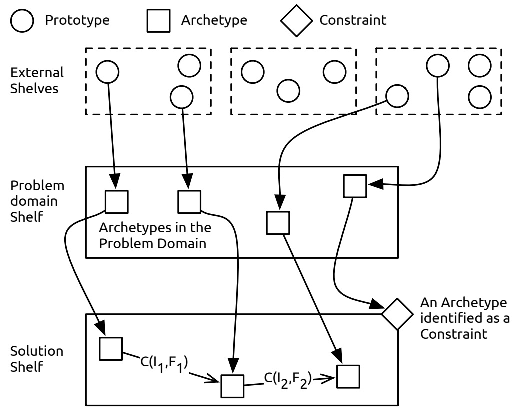

Figure 8: Concern and context refinement (from [3]).

Various *external shelves* hold all known recipes from different domains. Through *concern refinement*, recipes selected from the external shelves that satisfy requisite criteria for a problem situation are held in a *problem–domain shelf.* A context mapping for the recipes in the problem–domain shelf is obtained as a *solution shelf* using *context refinement.* Depending on the context, a solution shelf may either be an executable domain or may require further translation. Shelf management facilitates *localized ontologies* which are applicable to particular instances of a problem situation and assist the process of arriving at shared semantic understanding by reducing the scale and complexity of ontological mapping.

By applying the OAR framework, all identified concerns are evaluated for their firmness and influence for a given problem situation; a solution specification is formulated in terms of recipes that are verified and validated with reference to the scientific basis of the problem domain, providing a mapping between the intellectual concerns and the solution specification.

References

[1] Ahlemann, F. et al., eds. *Strategic Enterprise Architecture Management – Challenges, Best Practices, and Future Developments*. Springer-Verlag, 2012. ISBN: 978-3642242229.

[2] Chemboli, S. and Boughton, C. "Contextual Course Design with Omnispective Analysis and Reasoning". In: *Changing Demands, Changing Directions. Proceedings ascilite*. Ed. by Williams, G. et al. Hobart, 2011, pp. 210–219. URL: http://www.leishman-associates.com.au/ascilite2011/downloads/papers/Chemboli-full.pdf.

[3] Chemboli, S. and Boughton, C. "Managing Large and Complex Systems with Omnispective Analysis and Reasoning". In: *Proceedings of SETE APCOSE 2012*. Brisbane, Australia, 2012. URL: http://hdl.handle.net/1885/9009.

[4] Chemboli, S. and Boughton, C. "Omnispective Analysis and Reasoning: A Framework for Managing Intellectual Concerns in Scientific Workflows". In: *Proceedings of the 5th India Software Engineering Conference*. Kanpur, India, 2012, pp. 143–146. DOI: 10.1145/2134254.2134279.

[5] CIO Council. *A Practical Guide to Federal Enterprise Architecture v1.0*. Tech. rep. P00201. U.S. Government Accountability Office, 2001. URL: http://www.gao.gov/products/P00201.

[6] Holt, J. and Perry, S. *Modelling Enterprise Architectures*. The Institution of Engineering and Technology, 2010. ISBN: 978-1849190770.

[7] ISO/IEC/IEEE. *ISO/IEC/IEEE 42010 - Systems and software engineering – Architecture description*. Standard. Institute of Electrical and Electronics Engineers Inc., 2011.

[8] Kaisler, S. H., Armour, F., and Valivullah, M. "Enterprise architecting: Critical problems". In: *Proceedings of the 38th Hawaii International Conference on System Sciences*. 2005, pp. 1–10. DOI: 10.1109/HICSS.2005.241.

[9] Lankhorst, M. et al. *Enterprise Architecture at Work: Modelling, Communication and Analysis*. 2 Ed. Springer, 2009. ISBN: 978-3642013096.

[10] Op't Land, M. et al. *Enterprise Architecture: Creating Value by Informed Governance*. Springer, 2009. ISBN: 978-3-540-85231-5.

[11] Planning Commission of India. *Scenarios - Shaping India's Future*. Tech. rep. 2012. URL: http://goo.gl/tyr7Y.

[12] Tang, A., Han, J., and Chen, P. "A comparative analysis of architecture frameworks". In: *Proceedings of the 11th Asia-Pacific Software Engineering Conference, 2004*. 2004, 640–647. DOI: 10.1109/APSEC.2004.2.