Making context explicit towards decision support for a flexible scientific workflow system

Xiaoliang Fan (1, 2, xiaoliang.fan@gmail.com), Patrick Brézillon (1, patrick.brezillon@lip6.fr), Ruisheng Zhang (2, zhangrs@lzu.edu.cn) and Lian Li (2, lil@lzu.edu.cn)

(1) LIP6, Box 169, Université Pierre et Marie Curie
4 Place Jussieu, Paris 75005 France

(2) School of Information Science and Engineering, Lanzhou University
222 South Tianshui Road, Lanzhou 730000 P.R.China

Abstract

Scientific workflow (SWF) system is a specific workflow management system applied to science arena. For years, SWF systems are widely applied to many applications, namely in physics, climate modeling, drug discovery process, etc. However, current SWF systems face the challenge to adapt the flexibility and lack of decision support for scientist. We believe the major reason for the failure is due to do not make context explicit. We propose a solution to introduce contextual graphs (CxG) in the four phases of the SWF lifecycle, each of which is expressed in a standard format, including a case study in virtual screening. Contextual graph allows to model scientists' decision making processes as a uniform representation of knowledge, reasoning, and of contexts, so that scientists are closely involved in each phase of SWF lifecycle to maximize the decision support. Finally, we conclude and highlight that using CxG is the key human-centered process for SWF systems.

Introduction

Scientific workflow system liberates the computational scientists from burden of data-centric operations to concentration on their scientific problems (Altintas et al., 2004; Goble et al., 2007). However, it is not yet satisfied, considering that computational science (Roache, 1998) is always reproduced in a flexible and exploratory pattern. Consider virtual screening (Chen & Shoichet, 2009) for example, the choice of one software over others depends much on contextual information that are highly specific of the situation at hand, and where, when, how and by whom the scientific workflow is executed. Thus a strong and sustainable decision support is urged for scientists to transfer hypotheses to discovery.

Workflow flexibility becomes a critical challenge to deal with intermittently available resources, execution failures, and to support human-centric decision-makings. However, identifying how scientists make decisions to address workflow flexibility is a very complicated issue. The ways of scientists make their decision vary from one another: (1) based on their past experience considering successful or failed ones; (2) inherited from the best practices within science communities; (3) from the observed intermediate results; and (4) just follow their own distinguished way. Various approaches (Zhang et al., 2008; Courtney, 2001; Tabak et al., 1985) are proposed to get user involved to describe their decision making processes. Normally in such applications, a decision making (e.g., choose methods, change parameters, re-design the experiment) is measured by a decision node in workflow design accompanying with a numerical value (e.g. IF the variable is greater than 5, THEN execute the activity A, ELSE execute activity B; WAIT for 2 minutes to execute activity C). However, scientific discovery is by nature a knowledge-intensive one (van der Aalst et al., 2005) that scientists' decisions rely not only on data and information available, but also on a learning process in which user's preference, knowledge, and situation are captured to adapt the human-centered processes.

Such challenges mentioned above become an obstacle when scientists are making adaptive decisions to deliver new outcomes with fresh data and its context (Fan et al., 2010). Brézillon and Pomerol (1999) define context as "what constrains the resolution of a problem without explicit intervention in it". We believe that the main reason for this failure is largely due to the lack of context management in an explicit way. In this paper we propose four ways of making context explicit in scientific workflow, by introducing contextual graph to in the four phases of scientific workflow lifecycle. Representing and making "context" explicit in SWF system would provide sustainable decision supports for scientists by formalizing their research, strategies, and customization information, where elements of knowledge, reasoning and contexts are represented in a uniform way.

Hereafter, the paper is organized in the following way. Section 2 introduces the four phases of the scientific workflow lifecycle. Section 3 investigates the possibility of integrating contextual graphs to the four phases of scientific workflow lifecycle through a case study in virtual screening. Section 4 discusses previous works on workflow flexibility in order to point out what is reusable while problems remain to support decision-makings in a flexible scientific workflow system. The general conclusion and future work in Section 5 closes the paper.

Scientific Workflow Lifecycle

Scientific workflow lifecycle is coming from workflow lifecycle (van de Aalst & van Dongen, 2003; Gil et al., 2007; Deelman & Chervenak, 2008). It normally starts from the scientific hypotheses (Beaulah et al., 2008; Tadmor & Tidor, 2005; Claus & Johnson, 2008) to reach a specific experimental goal, which includes four phases (see Figure 1):

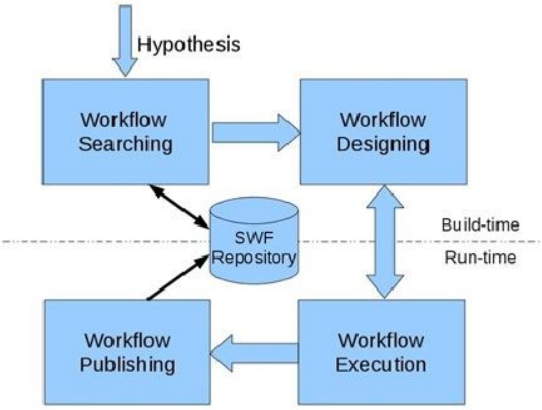


Figure 1: SWF lifecycle

- *Workflow Searching*: before initiating a brand new workflow designing, scientists get used to firstly consult a public SWF repository for searching previously published workflows (Wroe et al., 2007). Once found, it would be easy to reproduce the pre-existing workflow to constitute a new one. Workflow searching results of sharing SWF considered with its context of use. The more shared SWFs are taken place in the SWF repository, the more accurate the searching result would be.
- *Workflow Designing* is then initiated for constructing a workflow model (Ludascher et al., 2009). An abstract workflow model will firstly be designed, in which scientific tasks and their execution orders, as well as data and its dependencies will be described. Secondly, the phase involves the mapping from abstract workflow to concrete/executable workflow where the required resources are selected. By mapping the workflow instance onto the available execution resources, an executable workflow is created for the next phase.
- *Workflow Execution* is the enactment of executable workflow by a *workflow engine* (Deelman & Chervenak, 2008), in which input data is consumed and output data is produced (Tan et al., 2010). Workflow engine follows the order of tasks and their dependencies defined in the workflow model. It is common to re-execute the workflow iteratively, considering the evolutionary changes of workflow model (e.g., in workflow design, adding or skipping tasks, and altering task dependencies) or momentary changes of a running workflow instance (e.g., making local decisions in response to a special situation, alter decision after analysing observed intermediate result, reporting exceptional cases).
- *Workflow Publishing* is a post-execution phase for scientists to interpret workflow results (Tan et al., 2010; Ludascher et al., 2009) and to publish the SWF in its context of use (Wroe et al., 2007; Deelman & Gil, 2006). Depending on the workflow outcomes and analysis results, the original hypotheses or experimental goals may be revised or refined, giving rise to another round of workflow design/execution in an iterative manner. Furthermore, it must then be facilitated to publish the workflow on a repository, so that SWF could be archived for re-use later.

Figure 1 shows the relationship among each phases of scientific workflow lifecycle: hypotheses arrive as keywords to search pre-existing scientific workflow in SWF repository; then scientist begin to design the workflow model and maintain the mapping from an abstract workflow to a concrete one; workflow execution phase enacts the workflow model on available resources according to data and control dependencies; if a change is encountered, there is an iterative process to re-design the workflow model as well as re-execute the workflow instance; if executed successful, scientist will publish the workflow in the SWF repository for the sake of reproduction in the research communities.

Current studies (van de Aalst & van Dongen, 2003; Deelman & Chervenak, 2008) on SWF lifecycle generally result in the weakness to manage the workflow changes and exceptions. We believe that the major failure is due to do not make context explicit in the SWF systems.

Make Context Explicit in SWF Lifecycle

Representing and making context explicit in SWF system is a challenge that could promote a SWF system more flexible and enhance its intelligence to facilitate effective decision-makings. In this section, we discuss managing contexts explicit throughout the four phases of the SWF lifecycle, each of which is described using a standard format including: *motivation*, *realization approach*, *example*, and *discussion*.

The example is represented in the Contextual graphs formalism (Brézillon, 2005) through a case study entitled "Virtual screening research on avian influenza H5N1 virus", which aims to find dozens of drug candidates for H5N1 virus (He et al., 2008), by docking 7.7 million small molecules separately on H5N1 protein (Chen & Shoichet, 2009). Figure 2 shows a docking example, which binds a molecule (ZINC12050767) to a virus protein (H5N1 PA_C Polymerase, known as Bird flu) through the Dock 6.2 software. Virtual screening could be considered as millions of docking procedures on the PA_C protein.

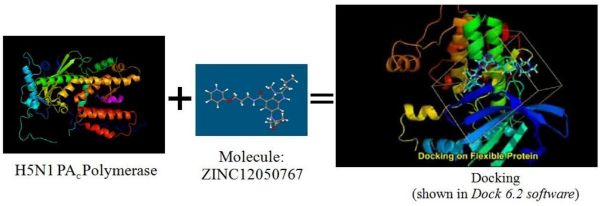


Figure 2: Docking example

The application is not only a time-consuming workflow application in which intensive computing is expected to be performed by docking software, but also a very flexible one that there is no unique solution for each computing because they vary from each other on selecting docking software. For example, scientists should identify the context in which the experiment is organized as a scientific workflow. According to the current focus and context, they link a specific resource (e.g., software, database, and instrument) with the workflow to realize a specific task. The concept of human-centered process is particularly relevant in such domains.

Figure 3 provides the definition of the elements in a contextual graph (actions, contextual elements, sub-graphs, activities and temporal branching). A more complete presentation of this formalism and its implementation can be found in (Brézillon, 2005).

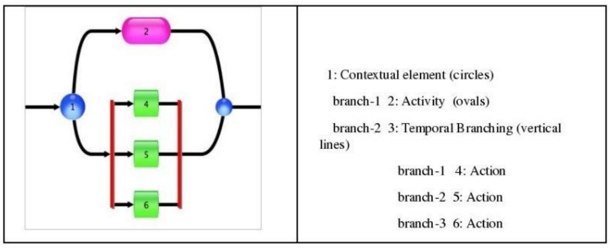


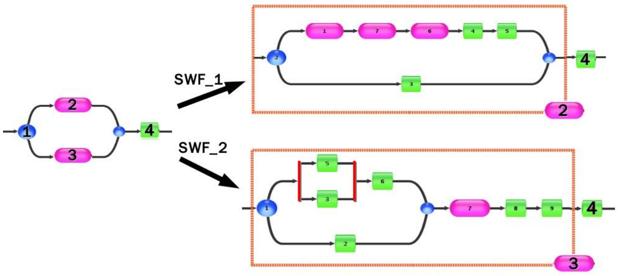
Figure 3: Elements in Contextual graph

Workflow Searching

Motivation: Before the workflow design, context behaves as an interface to determine which SWF should be chosen from a library of SWFs, or a SWF repository. In this case, a scientist plays a role as a context provider to guide the choice of the right SWF model according to current focus and context at hand, so as to largely match what the scientific hypotheses indicate.

Realization approach:

- Scientist firstly searches a SWF from a SWF repository, using keywords which could best describe their hypotheses and are coherent with the context at hand.
- If the pre-existing SWF is exactly what they want, the scientist could skip workflow design phase and just replace with their own parameters for workflow execution directly.
- Otherwise if it is similar to their needs, slight modifications will be carried out shortly in the workflow design.



Context graph: virtual screening on protein PAc
1: Is the protein rigid or flexible?
Rigid 2: Activity: perform first rigid screening
Flexible 3: Activity: perform second flexible screening
4: analyze the result

Figure 4: (Left) Contextual graph of virtual screening on H5N1 protein; (Right) Choosing one SWF from two SWFs (SWF_1 and SWF_2)

Example: In Figure 4 (Left), CE1 is a contextual element (blue circle with number 1). The instantiation of the CE1 (Is the protein rigid or flexible?) leads to the generation of two scientific workflow instances in Figure 4 (Right): one is SWF_1 (i.e. value of CE2= “Rigid”), and the other is SWF_2 (i.e. value of CE2=“Flexible”). In the application, if scientists want to do a rigid virtual screening, “rigid” will become a keyword when performing the searching. Thus, SWF_1 will be selected. Similarly, SWF_2 is chosen when searching for a “flexible” screening. As a result, CxGs act as an interface to make decisions to choose SWF from the SWF repository.

Discussion: It is normal to expect nothing from the repository, scientist could move to the next phase to start workflow design from scratch.

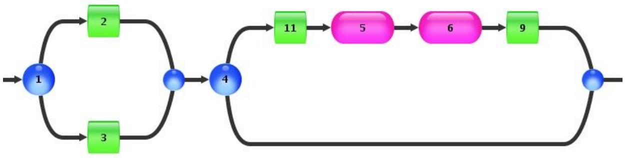
Workflow Designing

Motivation: During workflow design, a certain degree of freedom is given to the user to execute a workflow by offering multiple alternative execution paths. Classical workflow systems reduce the degree of flexibility by offering powerful design constructs (e.g., start, if/else, repeat until, parallel execution, end), in which decision-making is always measured by a decision node accompanying with a numerical value. However, human decision is so complex that a numerical decision is less descriptive than a simple question. As a result, we describe execution paths of workflow in contextual graphs (CxGs) which model contextualized information (CEs) and their dependencies. In a contextual graph, the most appropriate execution path could be selected from those encoded during the execution time to address the context at hand.

Realization approach:

- Firstly, it is necessary to know all the current instances of the CEs at the moment of the application of the workflow. An instantiation is the value that a contextual element can take for a specific instantiation of the focus at hand.
- Then, a group of contextualized information is generalized as a set of CEs.
- CEs are then formalized in a contextual graph by their dependencies. The contextual graph is ready for the workflow execution, when a SWF instance corresponds to a specific execution path under the instantiation of context. In CxG, the execution path is a sequence of actions, connected by the instantiation of the selected contextual elements.

Example: In Figure 5, a scientist designs the workflow of protein preparation as a contextual graph with a set of contextual elements (CE1 and CE4) and their execution dependencies. The possible execution paths are controlled by the value of each contextual element. For example, the instantiation of CE1 (i.e., value of CE1= “Yes”) and CE4 (i.e., value of CE4= “Yes”) leads to the execution path of “1→2→4→11→5→6→9”.



Contextual graph: protein preparation (old)
1: Can you find the protein by yourself?
Yes 2: download it from "Protein Data Bank"
No 3: ask for help until you get the protein
4: Do you need to do "protein preparation"?
Yes 11: enter parameters during "protein preparation"
5: Activity: remove unrelated molecules
6: Activity: add hydrogen and charge
9: store the protein prepared in the database
No

Figure 5: Contextual graph: protein preparation (old)

Discussion: Describing a completely set of all possible execution paths during workflow design might be either undesirable or impossible (Schonenberg et al., 2008). For example, a certain number of possible execution paths are unknown before execution. As a result, late-modelling (Han et al., 1998) could enable to make sub-model dynamically defined during execution.

Workflow Execution

Motivation: Scientists frequently re-execute the scientific workflow by adding or ignoring portions of workflow realized at design time. Context should support the assembling of SWF components, which must be recompiled each time when a new context arrives (i.e., a contextual element takes a new instance). As a result, a new execution path, or even a new contextual graph will be inserted or removed when SWF evolves along with its context.

Realization approach:

- Each time a new instantiation of a CE occurs, the contextual graph is re-executed, and the SWF is recompiled for generating a new SWF instance for execution.
- If the scientist wants to re-design the workflow by adding or ignoring portion of SWF, they first stop the current workflow execution.
- Then, a new group of contextualized information, including the information representing the workflow changes, should be generalized as a new set of contextual elements.
- If a CE with the following activities/actions is added or ignored, a new contextual graph is produced to address the new focus.

Example: Figure 6 is inherited from Figure 5. During the execution phase, the scientist finds something wrong with the intermediate result, because he doesn't take into account whether the protein is flexible or rigid. So he decides to stop the current execution and re-design the experiment. As a result, a new contextual element CE7 (Is it a rigid or flexible screening?) is added. When the value of CE7 is “flexible screening”, Activity13 (Activity: optimize the protein) is invoked as a new SWF component. Furthermore, the contextual graph is updated along with the change of CEs, and it is necessary to record such update in a knowledge base for the sake of workflow sharing, which will be discussed in the next section.

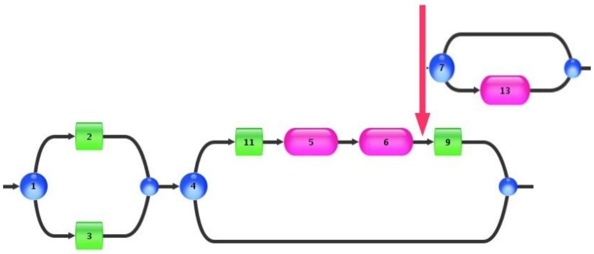


Figure 6: Contextual graph: protein preparation (new)

Discussion: It would be a risk of incoherence between the running workflow instance and results. For example, when you made a decision two minutes ago and the contextual graph chooses an execution path for the workflow. But later, right before the workflow execution, a new context arrives to urge the adaptation of a new contextual graph.

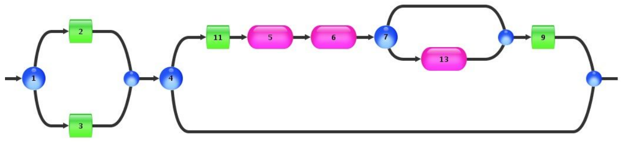
Workflow Publishing

Motivation: If executed successfully, the scientist then try to analyse the results generalized by workflow execution. Type of result analysis includes: 1) evaluate data quality (e.g., does this result make sense?), 2) examine execution traces and data dependencies (e.g., which results were “tainted” by this input dataset?), 3) debug runs (e.g., why did this step fail?), or 4) simply analyse performance (e.g., which steps took the longest time?). After the result analysis process, it is possible to re-design and re-execute the workflow iteratively until the new context is addressed. Incremental knowledge acquisition should be proceeded to make contextual graph growing to be more efficient. Furthermore, one of the motivations what scientists are counting on SWF is the sharing, reproduction, transformation, and evolution of the “old” SWF to be a brand “new” one. It is expected to enable sharing of SWFs according to their contexts of use. In this circumstance, the context defines the status of the knowledge and also maintains the relationship between different kinds of knowledge.

Realization approach:

- A SWF repository is build up to document workflows with their contexts of use.
- When workflow is re-executed, the contextual graph is adapted incrementally to trace the workflow flexibility. Once a new contextual graph is generated, add it as a new scenario to SWF repository.
- Conscientious users might partition the workflow into coherent fragments and publish them.

Example: Once a contextual element is modified, a new CxG is created to address the new focus and its context. Drawn from Figure 6, Figure 7 shows a new contextual graph to be added in a SWF repository for future sharing with other scientists.



Contextual graph: protein preparation (new)

- 1: Can you find the protein by yourself?
  - Yes 2: Download it from "Protein Data Bank"
  - No 3: Ask for help until you get the protein
- 4: Do you need to do "protein preparation"?
  - Yes 11: Enter parameters during "protein preparation"
    - 5: Activity: remove unrelated molecules
    - 6: Activity: add hydrogen and charge
    - 7: Is it a rigid or flexible screening?
      - Rigid
      - Flexible 13: Activity: optimize the protein
    - 9: store the protein prepared in the database
  - No

Figure 7: Contextual graph: protein preparation (new)

Discussion: Encourage sharing of scientific workflow with its context, would make it as a complementary of paper-based publications. In such a case, scientific workflow would be archived along with paper-based publications. However, the quality of sharing data and workflow becomes a new question.

Summary

Contextual graphs are a formalism of representation allowing the description of decision making in which context influences the line of reasoning (e.g. choice of a method for accomplishing a task). The advantage of contextual graphs relies on that: (i) CxGs provide naturally learning and explanation capabilities in the system; and (ii) CxGs allow a learning process for integrating new situations by assimilation and accommodation. In short, the notion of context is made explicit during the four phases of scientific workflow lifecycle by contextual graphs. Contextual Graphs formalism has been already used in different domains such as medicine, incident management on a subway line, road sign interpretation by a driver, computer security, psychology, cognitive ergonomics, etc.

Related Works

Various approaches, such as BPEL (Zhang et al., 2008), UML (Courtney, 2001), Petri-net (Tabak et al., 1985), are proposed to address the issue of workflow flexibility by getting user involved in representing decision-making. Applications (Yu et al., 2005; Hey et al., 2009) have proven the significance of current systems to handle numerical decision-making as control-flow functions, such as "wait 30 second, and then proceed the next task", "if the value is greater than 5 then execute the task_A, else execute the task_B". However, it becomes an obstacle to manage the common but important decisions, such as "are you satisfied with the result?" and "do you need to do the protein preparation again", which is more comprehensive for scientists.

Context has been considered as a key element to support decision making in human centered processes for a long time (Brézillon, 2003; Brézillon, 2010). To address a coherent formalism of context, Sowa (1984) proposes conceptual graphs with their mechanisms of aggregation and expansion. Then, Sowa (2000) introduces a way to manage the context in conceptual graphs. Brézillon (2005) presents a simpler formalism of Contextual Graphs (CxGs) for representing context. Compared with other approaches, CxGs formalism is good at describing decision making in which context influences the line of reasoning.

In the implementation level, a number of applications exist for preparing formal representation of context. McCarthy (1993) formalizes contexts as formal objects, and the basic relation is ist(c,p). It asserts that the proposition p is true in the context c, where c is meant to capture all that is not explicit in p that is required to make p a meaningful statement representing what it is intended to state. Formulas ist(c,p) are always asserted within a context, i.e., something like ist(c', ist(c,p)): c': ist (c, p). Sharma (1995) gives a list of desirable properties for contexts in a formal language and distinguishes four approaches for formalizing contexts: (1) incrementing arity; (2) variation on implication; (3) modal operator forms; and (4) syntactic treatment. Based on McCarthy's work on context logic, Farquhar et al. (1995) present an approach to integrating disparate heterogeneous information sources.

In Table 1, we compare various approaches to model decision making in workflow, as implementation of "Exclusive Choice workflow pattern" (van de Aalst & Hofstede, 2003).

Table 1: Comparison of various implementations of "Exclusive Choice workflow pattern"

Approach	Decision Element	Decision Value	Decision Type
BPEL (Zhang et al., 2008)	<if>, <pick>	Condition	Numerical value
CxG (Brézillon, 2005)	Contextual Element	Value of CE	Any value
UML (Courtney, 2001)	Decision Node	Condition	Numerical value
Petri-net (Tabak et al., 1985)	Exclusive choice	Arc expression	Numerical value

By comparison, Contextual Graphs plays an equivalent role to other approaches for representing decision making. Furthermore, the advantage of contextual graphs embraces: (1) multiple representations of decision making, not only with a numerical value, but also with any kind of answers to questions to get scientists involved in a local decision-making process; (2) it is directly readable (e.g. generally something as "If the contextual element C has the value V1, thus use method M1, and with the value V2 use method M2"); and (3) it is very easy to have an incremental growth of a contextual graph by addition of contextual elements and branches for representing practices developed by users and not yet known by the system.

Conclusion

The human-centered processes must be considered at a global level to deal with the user, the task at hand, and the context in which the task is accomplished. Take a flexible scientific workflow for example, scientists could not handle the transferring from hypotheses to discovery in the SWF system without taking into account the context.

We propose a solution to introduce contextual graphs in the four phases of SWF lifecycle, each of which is expressed in a standard format, including a concrete example in the area of virtual screening. In our application on virtual screening, we use contextual graphs to model the decision making processes of scientists as a uniform representation of knowledge, reasoning, and contexts. As a result, scientists are closely involved in each phase of SWF lifecycle to maximize the decision support received from the system.

We believe that all of data, information and knowledge should be invoked, assembled, organized, structured and situated according to the given focus, and finally be formulated as the chunk of professional knowledge for scientists to maintain their research sustainability.

The extension of our work includes the development of a prototype interface between scientific workflow system and contextual graphs. Representing and making "context" explicit in SWF system by contextual graph would enhance workflow flexibility by formalizing scientists' research, strategies, and customization information, where elements of knowledge, reasoning and contexts are represented in a uniform way.

Acknowledgments

This work is supported by grants from National Natural Science Foundation of China (90912003, 60773108, 90812001, 61011130212), Centre national de la recherche scientifique (Researcher exchange project with NSFC 2010), and Région Ile-de-France (CP10-201), and by scholarships from China Scholarship Council (2008618047), and Égide (690544G).

References

Altintas, I., et al. (2004), Kepler: an extensible system for design and execution of scientific workflows. *Proceedings of 16th International Conference on Scientific and Statistical Database Management* (pp. 423-424). Danvers, MA: IEEE Computer Society.

Beaulah, S., Correll, M., Munro, R., Sheldon, J. (2008). Addressing informatics challenges in Translational Research with workflow technology. *Drug discovery today*, *13(17-18),* 771-777.

Brézillon P., and Pomerol J.C. (1999). Contextual knowledge sharing and cooperation in intelligent assistant systems, *Le Travail Humain*, 62 (3), 223-246.

Brézillon, P. (2003). Focusing on Context in Human-Centered Computing. *IEEE Intelligent Systems*, *3(18)*, 62-66.

Brézillon, P. (2005). Task-realization models in contextual graphs. In Dey, A.K., Kokinov, B., Leake, D., and Turner, R. (Eds), *Proceedings of Modeling and Using Context: 5th International and Interdisciplinary Conference* (pp. 55-68). Berlin :Springer Verlag.

Brézillon, P. and Pomerol, J.-Ch. (2010). Framing decision making at two levels. In Respicio, A., F. Adam, G. Phillips-Wren, C. Teixeira & J. Telhada (Eds.), *Bridging the Socio-Technica Gap in Decision Support Systems- Challenges for the next Decade*. Amsterdam: IOS Press.

Chen, Y., & Shoichet, BK. (2009). Molecular docking and ligand specificity in fragment-based inhibitor discovery, *Nature Chemical Biology, 5 (5)*, 358-364.

Claus, B., and Johnson, S. (2008). Grid computing in large pharmaceutical molecular modeling, *Drug discovery today*, *13(13-14)*, 578-583.

Courtney, J.F. (2001). Decision making and knowledge management in inquiring organizations: toward a new decision-making paradigm for DSS. *Decision Support Systems, 31(1)*, 17–38.

Deelman, E. and Chervenak, A.L. (2008). Data Management Challenges of Data-Intensive Scientific Workflows. *Proceedings of the 8th IEEE International Symposium on Cluster Computing and the Grid* (pp. 687-692), Lyon: IEEE CS Press.

Deelman, E. and Gil, Y. (2006). Managing large-scale scientific workflows in distributed environments: Experiences and challenges. *Proceedings of the Second IEEE International Conference on e-Science and Grid Computing* (pp. 144), Washington: IEEE Computer Society.

Fan, X. et al. (2010). Context-oriented scientific workflow system and its application in virtual screening. In Respicio, A., Adam, F., Phillips-Wren, G., Teixeira C., and Telhada J. (Eds.). *Bridging the Socio-Technical Gap in Decision Support Systems-Challenges for the next Decade* (pp. 335-345). Amsterdam: IOS Press.

Farquhar, A, Dappert, J, Fikes, R and Pratt, W. (1995). *Integrating information sources using context logic* (Tech. Rep. KSL-95-12). Palo Alto, California: Stanford University, Knowledge Systems Laboratories.

Gil, Y., Deelman, E., Ellisman, M., Fahringer, T., Fox G., Gannon, D., Goble, C., Livny, M., Moreau, L., Myers, J. (2007). Examining the Challenges of Scientific Workflows. *Computer*, *12(40)*, 24-32.

Goble, CA., et al. (2007). myExperiment: social networking for workflow-using e-scientists. *Proceedings of 2nd workshop on Workflows in support of large-scale science* (pp. 1-2). NY: ACM.

Han, Y., Sheth, A., Bussler, C. (1998). A Taxonomy of Adaptive Workflow Management. *In Workshop of the 1998 ACM Conference on Computer Supported Cooperative Work*. Seattle, Washington, USA.

He, X., et al. (2008). Crystal structure of the polymerase PA(C)-PB1(N) complex from an avian influenza H5N1 virus. *Nature*, *454*, 1123–1126.

Ludascher, B., Altintas, I., Bowaers, S., Cummings, J., Critchlow, T., Deelman, E., De, R.D., Freire, J., Goble, C., Jones, M., Klasky, S., McPhillips, T., Podhorszki, N., Silva, C., Taylor, I., and Vouk, M. (2009). Scientific process automation and workflow management. In Shoshani, A., and Rotem D. (Eds), *Scientific Data Management: Challenges, Existing Technology, and Deployment, Computational Science Series*, chapter 13. Chapman & Hall/CRC.

McCarthy, J. (1993). Notes on formalizing context. In Bajcsy, R. (Ed.), *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence* (pp. 555-560). San Mateo, California: Morgan Kaufmann.

Roache, P.J. (1998). *Verification and validation in computational science and engineering*. Albuquerque, NM: Hermosa Publishers.

Hey, T., et al. (2009). *The Fourth Paradigm: Data-Intensive Scientific Discovery*. Redmond, Washington: Microsoft research.

Schonenberg, H., Mans, R., Russell, N., Mulyar, N., van der Aalst, W.M.P. (2008). Process flexibility: A survey of contemporary approaches. In: Dietz J., Albani A., and Barjis J. (Eds), *Advances in enterprise engineering I*. Berlin: Springer-Verlag.

Sharma, N (1995). *On formalising and reasoning with contexts* (Tech. Rep. 352). Brisbane, Queensland: University of Queensland, Department of Computer Science.

Sowa, J.F. (1984). *Conceptual Structures: Information Processing in Mind and Machin*e. Reading, MA: Addison Wesley Publishing.

Sowa, J.F. (2000). *Knowledge Representation: Logical, Philosophical, and Computational Foundations*. Pacific Grove, CA: Brooks Cole Publishing.

Tabak, D. and Levis, A.H. (1985). Petri Net Representation of Decision Models. *IEEE Transactions on Systems- Man and Cybernetics*, *15(6)*, 812–818.

Tadmor, B., and Tidor, B. (2005). Interdisciplinary research and education at the biology-engineering-computer science interface: a perspective (reprinted article), *Drug discovery today*, *10(23-24)*, 1706-1712.

Tan, W., Missier, P., Foster, I., Madduri, R., De Roure, D., Goble, C. (2010). A comparison of using Taverna and BPEL in building scientific workflows: the case of caGrid, *Concurrency and Computation: Practice and Experience*, *22(9)*, 1098–1117.

Yu, J. and Buyya, R. (2005). A Taxonomy of Scientific Workflow Systems for Grid Computing. *In special issue on Scientific Workflows, SIGMOD Record*, *34(3)*, 44-49.

van de Aalst, W.M.P., van Dongen, B.F., et al. (2003). Workflow Mining: A Survey of Issues and Approaches. *Data & Knowledge Engineering*, *47*, 237–267.

van de Aalst, W.M.P., Hofstede, A., et al. (2003). Workflow Patterns. *Distributed and Parallel Databases. 14(3)*, 5–51.

van der Aalst, W.M.P., Weske, M., and Grunbauer, D. (2005). Case Handling: A New Paradigm for Business Process Support. *Data and Knowledge Engineering 53(2),* 129–162.

Wroe, C., Goble, C., Goderis, A. et al. (2007). Recycling workflows and services through discovery and reuse. *Concurrency and Computation: Practice and Experience*, *19(2)*, 181-194.

Zhang, H., et al. (2008). Extending BPEL2.0 for Grid-Based Scientific Workflow Systems, *Proceedings of IEEE Asia-Pacific Services Computing Conference* (pp. 757-762), Yilan: IEEE CS press.