Workflow Collaboration with Constraint Solving Capabilities

Yun-Heh Chen-Burger[1], Kit-Ying Hui[2]
Alun D. Preece[3], Peter M. D. Gray[3], Austin Tate[1]

[1] AIAI, CISA, Informatics, The University of Edinburgh, UK
[2] School of Computing, The Robert Gordon University, UK
[3] Computing Science Department, University of Aberdeen, UK

Abstract. This paper describes our efforts to provide a collaborative problem solving architecture driven by semantic-based workflow orchestration and constraint problem solving. These technologies are based on shared ontologies that allows two systems of very different natures to communicate, perform specialised tasks and achieve common goals. We give an account of our approach for the workflow assisted collaboration with constraint solving capabilities. We found that systems built with semantic (web) based technologies is useful for collaboration and flexible to enhance the system with specialised capabilities. However, much care must be exercised before correct semantics may be exchanged and collaborations occur smoothly.

Keywords: Virtual Organisation, Constraint Satisfaction, Business Process Modelling, Business Modelling, IDEF3, Ontology, NIST PSL, Semantic Web, Semantic Grid.

1 Introduction

Modern organisations are virtual entities composed of heterogeneous resources that span across geographical space. People working in organisations may locate in different places but need to work collaboratively to achieve common organisational goals. The tasks they must accomplish are often non-trivial, requiring specialised expertise and resources that are distributed across the organisation. The ability to collect and utilise these distributed knowledge and resources to assist effective collaboration and achieve common goals often requires more than simple information exchange, but more structured communication and coordination methods are needed.

Workflow and Business Process Modelling are well-recognised techniques for promoting and achieving effectiveness and efficiency in the co-ordination of distributed organisational operations [13]. Workflow technology originated from data (control) flow diagrams. These technologies have been widely deployed and refined in the fields of electrical and manufacturing engineering where formal processes are commonly available. It was only recently recognised that informal business processes do share many common characteristics and may be formalised and described using similar process technologies [14]. This recognition and advancement in this area explains the popularity of business process re-engineering and change management [21][18]. Until recent years, most workflow systems lacked an explicit representation of an underlying process model (first-generation workflow systems). Newer workflow systems use process model based design and manipulation and are the second generation of workflow systems [5]. We argue that a third generation workflow system may be one that understands and manipulates semantic rich information and processes and can operate in a distributed agent based environment.

On the other hand, while coordination is important within an organisation, it represents only one side of collaborative problem-solving. As sub-tasks are progressively created and organised, we need problem solvers that find solutions to specific problems. To achieve a meaningful task, one also has to exchange task-specific semantic knowledge between workflow systems and problem solvers. This requirement of semantic knowledge exchange between knowledge-utilising components becomes more important as tasks get semantically richer and interactions more sophisticated.

As Semantic Web (SW) technologies advances, it provides a flexible infrastructure for the exchange of semantic rich knowledge. We make use of ontological technologies and one of the standard SW languages, RDF, for communication. Based on this, we hope to make our integrated system more open and knowledge more interoperable. In this paper, we demonstrate our work through an example: distributed agent-based workflow systems are linked to a constraint problem solver that enables different organisational departments to carry out tasks collaboratively.

2 A Motivating Scenario

Consider a virtual organisation that builds and sells PCs based on customers' individual requirements. It has several departments each may locate in different places. Each may also have certain overlapping domain knowledge with one another and may possess specific non-overlapping local expertise – that may be data and/or work procedure related. They need to collaborate and rely on each other to achieve common organisational goals - i.e. to build and sell customer-tailored PCs.

Our example contains two departments: sales and technical. The sales department handles user orders and has marketing and cost controlling knowledge. The technical department is responsible for building PCs. It possesses information about PC hardware components and has the expertise of fitting different components to construct a workable PC, based on given constraints.

We use this example to demonstrate how the different technologies involved may assist the cooperation of the two departments. Three technologies are used: FBPML [3] provides process modelling and workflow technologies, KRAFT system [12] enables specialised support for constraint problem solving and I-X system [20] facilities a user front-end to manage workflow execution. A brief introduction for each technology, the overall bridging conceptual framework and the connecting architecture for the systems involved are given below.

3 Background Technologies

3.1 Formal Business Process Modelling Language (FBPML)

FBPML adapts and merges two recognised process modelling standards: IDEF3 [15] and NIST PSL (the Process Specification Language) [19]. IDEF3 originated from concurrent engineering disciplines and is one of the richest methods available for process modelling. It provides relatively comprehensive visual notations, modelling method and model-building guidelines. These characteristics make IDEF3 a suitable candidate for capturing processes. Nevertheless, its semantics is informal and its models therefore are open to interpretation.

On the other hand, NIST PSL provides formal semantics for commonly shared process modelling concepts as well as theories that support temporal reasoning on activities. FBPML combines the two different methods by adapting IDEF3's rich visual and modelling methods and mapping those modelling concepts to the formal semantics and theories of PSL. FBPML has a formal representation that is based on First Order Predicate Logic and has a direct mapping to PSL. As a result, FBPML includes relatively standard visual notation and process modelling concepts and is applicable to theories that carries out formal analysis on its process models.

Precise process execution logic has also been added so that virtual workflow machines may be generated and enacted at run-time based on FBPML descriptions, which is not possible for conventional IDEF3 or PSL models. In addition, FBPML makes use of a formal data language, FBPML-DL, that provides descriptions for data constructs and becomes an integral part of a FBPML process description. This explicit representation of data manipulated by a process is also not presented in IDEF3 nor PSL. [3] gives more details about FBPML.

To collaborate with the KRAFT system, it is necessary that the FBPML data description for the PC configuration domain is mapped to the corresponding KRAFT domain representation. In addition, as the KRAFT system is based on constraint satisfaction technologies, constraints described in FBPML must be interpreted so that KRAFT system can process them. Figure 1 shows an example FBPML static constraint giving basic requirements for a PC configuration, i.e. identification and allocation of a suitable processor, disk_controller and IO device. These requirements will be interpreted by KRAFT as a request for constructing a standard PC and KRAFT will incorporate its local technical constraints (described in Subsection 3.3) to find a suitable solution.

It is also possible to specify the preferred capabilities of cards, or a specific type of device. Each requirement is described in either a static or dynamic constraint statement in FBPML. As the above example is a hard constraint, it is represented in a static constraint in FBPML and may not be modified at processing time. Soft or dynamic constraints are often customers' preferences that are more flexible and may be altered during the iteration of the process.

3.2 I-X technology

I-X [20] has a rich systems integration architecture. It stores process models and supports dynamic instantiation and monitoring of process execution. Different communication strategies can be deployed to enable it to communicate with heterogeneous systems.

```
static_constraint(
  [ instance_of(processor_x, processor),
    instance_of(disk_controller_x,
                disk_controller),
    instance_of(io_x, io) ],
  [ allocate(processor_x,  slot1),
    allocate(disk_controller_x,  slot2),
    allocate(io_x,  slot3),
    instance_of(slot1, slot),
    instance_of(slot2, slot),
    instance_of(slot3, slot),
    total_cost(_M)] )
```

Fig. 1. *Example FBPML Constraint*

Some of those communication methods are flexible and modifiable at run time. Various work has been proposed and carried out in different application areas which will seek to create generic approaches (I-Tools) for the various types of tasks in which users may engage. Example components are:

- I-DE - A Multi-Perspective Domain Editor providing a modelling environment that allows the user to use different methodologies and tools via plug-ins.
- I-P^2 - I-X Process Panels to support dynamic process change and management of activity enactment.
- I-Plan - An Intelligent Planning System, which will be used as a workflow process planning aid in the overall approach.

As the I-P^2 component supports hierarchical process decomposition, it is suitable for our experiment and has been used to provide a user front-end for managing process execution and storing of process models. It has also been used as a communication medium between different I-X workflow agents and a constraint solver, the KRAFT system. As I-X is based upon a conceptual framework of <*I-N-C-A*>, conceptual mapping between FBPML, <I-N-C-A> and KRAFT thus has been carried out and will be discussed in Section 4.1.

3.3 KRAFT and Constraint Solving

KRAFT (Knowledge Reuse And Fusion/Transformation) is a distributed information system that emphasizes the use of mobile constraint knowledge to dynamically compose problem instances and tailor them to suit problem solvers [12] [16]. It uses constraints as a uniform formalism to represent domain-specific knowledge, partially solved solutions and intermediate results.

The KRAFT architecture contains “wrappers” that map constraints and data from heterogeneous resources onto a common shared ontology, named *integration schema*. When expressed against a KRAFT domain-wide *integration schema*, these mobile constraints become self-contained abstract knowledge objects which can move within a KRAFT-aware agent network.

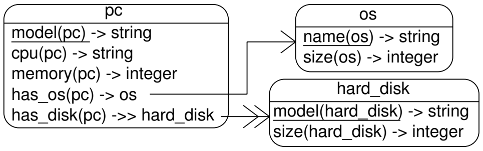


Fig. 2. *This schema shows three entity classes. The single arrow indicates each* `pc` *may have only one* `os` *installed. A double arrow indicates a* `pc` *may have multiple* `hard-disk`.

Figure 2 shows an example schema of a PC configuration domain, in which components are put together to form a workable PC. In KRAFT, domain knowledge is captured as database integrity constraints expressed against the *integration schema* using the CoLan constraint language [1]. CoLan constraints have evolved from database state restrictors in the P/FDM database system[1] into mobile problem specifications [7]. Figure 3 shows example CoLan constraints that captures local expertise in the PC configuration domain. The second constraint is a "*small print constraint*". "*Small print constraints*" captures the semantics of instructions attached to class descriptor for data objects in a product catalogue database. When a data object is retrieved, these attached instructions must also be extracted to ensure that the data is properly used. Thus the attached constraint becomes mobile knowledge which is transported, transformed and processed in a distributed environment. This approach differs from a conventional distributed database system where only database queries and data objects are exchanged.

```
constrain each p in pc
  to have size(has_os(p))
     =< size(has_disk(p))

constrain each p in pc
  such that name(has_os(p))="WinXP"
  to have memory(p)>=128
```

Fig. 3. *The first constraint captures the requirement that "*the size of the hard disk must be large enough to store the OS*". The second one is an example of "*small print*" constraint that conditionally applies only when the installed OS is "WinXP". Semantically, this constraint attaches to the WinXP OS.*

Knowledge processing components in KRAFT are realised as software agents that express a subset of KQML [6]. The underlying language, Colan, has evolved into an RDF-based Constraint Interchange Format (CIF) [8] [9]. At the heart of the system is a *constraint fusing mediator* that combines constraint fragments from distributed sources (Figure 4). The composed *constraint satisfaction problem* (CSP) instance is then analysed and compiled into a combination of distributed database queries and a constraint logic programming (CLP) program. This approach enables the system to cope with the

---

[1] http://www.csd.abdn.ac.uk/∼pfdm/

dynamic nature of both data and constraint knowledge in the distributed environment. A detailed explanation of the constraint problem formulation and compilation into CLP code can be found in [11].

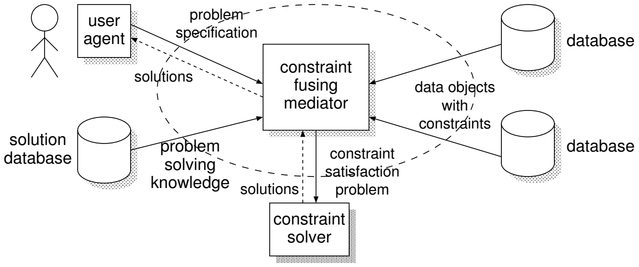

user agent
problem specification
solutions
constraint fusing mediator
database
data objects with constraints
database
solution database
problem solving knowledge
solutions
constraint satisfaction problem
constraint solver

Fig. 4. *Constraint fragments from different sources are fused by the constraint fusing mediator.*

4 Connecting I-X and KRAFT

4.1 Mapping Knowledge between FBPML/I-X and KRAFT

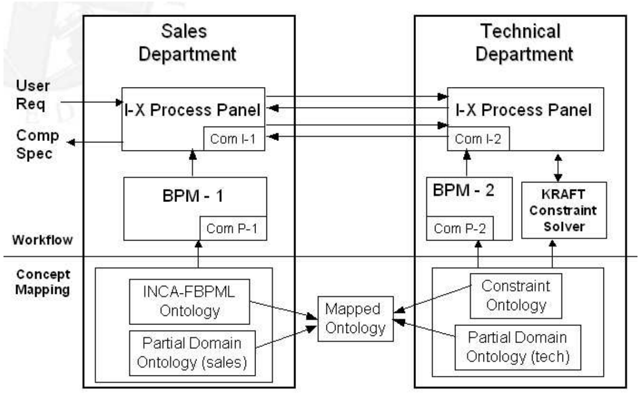


Fig. 5. *Conceptual architecture of collaboration between two (sub)organisations in the PC configuration domain*

Figure 5 gives an overview for a conceptual architecture that enables collaborative problem solving using semantic-based workflow techniques. A horizontal line that divides the two departments has been used to distinguish our work on the actual realisation of workflow and the underlying conceptual mapping. As described previously, three technologies are involved, FBPML, I-X and KRAFT, each underpinned by their own method ontologies. In this particular case, I-X process panels are used to serve two functions: to provide a process-aware interface for user support and to provide a communication mechanism between two I-X agents and between an I-X agent and the KRAFT system. As I-X is based upon the conceptual framework of <I-N-C-A> that provides a human-machine interaction interface for FBPML, FBPML is firstly mapped to <I-N-C-A>, as indicated in *INCA-FBPML ontology* in Figure 5. This enables FBPML business process models *(BPM)* to be translated and managed through *I-X process panels*.

On the other hand, the *constraint ontology* that underpins the KRAFT system is mapped with the INCA-FBPML ontology that allows communication between FBPML, I-X and KRAFT constraint solver. The process of conceptual mapping also indicates patterns needed for correspondence that form the bodies of communication between systems. The communication processes, indicated by *(Com P-i)*, are a recognised type of process in FBPML and are clearly labelled in its models.

Domain knowledge in the PC configuration is divided and stored in different departments of the organisation by their functionalities. This domain knowledge is based upon individual ontologies: the sales and costing ontology and the technical ontology. As the two departments overlap in their operations, their ontologies are partially shared. This shared knowledge assists the collaboration between departments of very different natures. This mimics real-life situations where specialised expertise centres are often geographically disperse yet collaboration is required between them. The mapping of the underlying ontologies provides a rich and sound foundation towards exchange of precise execution semantics as well as ensuring smooth cooperation.

4.2 A RDF-based Collaboration

To exchange semantics between I-X and KRAFT, several types of knowledge have to be transported:

- Two partially overlapping domain models describing the semantics and relationships of objects in the application domain,
- I-X *issues* being passed from the workflow system, representing problems to be solved,
- Constraints representing requirements to be satisfied for a specific task,
- Problem solving results giving specified requirements.

The semantics of a constraint is expressed against a data model in FBPML and the corresponding translation in a functional data model of the KRAFT system. An I-X *issue* provides a construct to include a problem description that is passed to the KRAFT CSP solver for solutions in which requirements are expressed in terms of constraints. After the execution of the KRAFT constraint solver, solutions are passed back to the I-X system[2], otherwise “fail” is returned if no solution is found.

---

[2] I.e. via the technical to the sales department.

Domain Models In the original KRAFT system, we model a domain by a database schema based on a functional data model, which effectively serves as an ontology that captures knowledge of classes, attributes and subclass relationships in the domain. The functional data model is an extended ER model, part of which is mapped into a RDF Schema (RDFS) specification [2]. An interpreter reads meta-data from the database and *generates* a corresponding RDFS description, making meta-knowledge web-accessible. The RDFS fragment in Figure 6 refers to the P/FDM database schema in Figure 2.

```
<rdfs:Class rdf:ID="pc">
  <rdfs:subClassOf rdf:resource=
    "http://www.w3.org/2000/01/rdf-schema#Resource"/>
</rdfs:Class>

<rdfs:Class rdf:ID="os">
  <rdfs:subClassOf rdf:resource=
    "http://www.w3.org/2000/01/rdf-schema#Resource"/>
</rdfs:Class>

<rdf:Property rdf:ID="has_os">
  <rdfs:domain rdf:resource="#pc"/>
  <rdfs:range rdf:resource="#os"/>
</rdf:Property>
```

Fig. 6. *RDFS description of a P/FDM database schema*

Mapping a P/FDM schema into RDFS has the advantage of making the domain model available to RDFS-ready software. As we will see in the next section, the domain model expressed in RDFS plays an important role in specifying the semantics of objects in the domain. A detailed discussion can be found in [8].

Constraints In practice, human-readable CoLan constraints such as those in Figure 3 are compiled into an intermediate format, *Constraint Interchange Format* (CIF). CIF expressions are syntactically Prolog terms, which are easier to process by software components. To make CIF portable, we encode CIF constraints into RDF by defining a schema in RDFS for the CIF language, serving as a meta-schema [8]. One satisfying feature of this constraint interchange format in RDF is that the (name) tags used make a clean separation between information about *logical formulae* with the usual connectives, and information about *Expressions* denoting objects in the data model. Effectively CIF preserves a layer of rich semantic information while providing the processing convenience of RDF.

Expressions in the CIF language store meta-knowledge about entities, their subtypes, attributes and relationships whose instances are expressed in RDF. This enables a rich data model that is independent of the underlying manipulation mechanism that is suitable for different paradigms such as relational, flat files and object-oriented storage.

This is advantageous for interoperability across different platforms and systems, as well as integration of data from different sources over the Web [8][17]. The full RDF Schema and example constraints are available at: http://www.csd.abdn.ac.uk/∼schalmer/schema/.

4.3 Communication Architecture

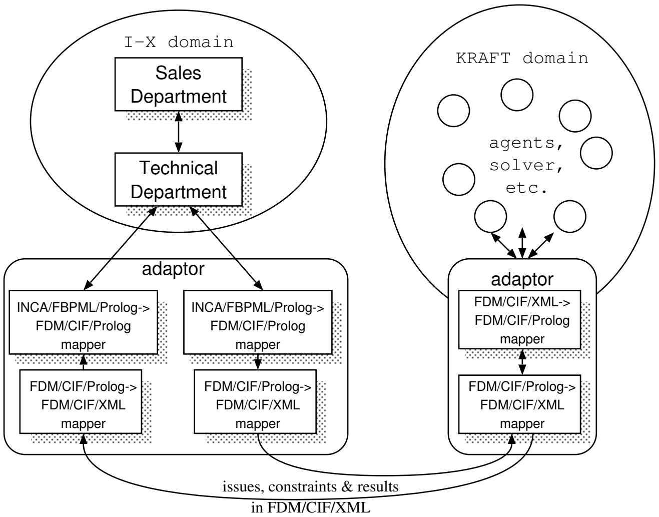

I-X domain
Sales Department
Technical Department
KRAFT domain
agents, solver, etc.
adaptor
INCA/FBPML/Prolog-> FDM/CIF/Prolog mapper
INCA/FBPML/Prolog-> FDM/CIF/Prolog mapper
FDM/CIF/Prolog-> FDM/CIF/XML mapper
FDM/CIF/Prolog-> FDM/CIF/XML mapper
adaptor
FDM/CIF/XML-> FDM/CIF/Prolog mapper
FDM/CIF/Prolog-> FDM/CIF/XML mapper
issues, constraints & results in FDM/CIF/XML

Fig. 7. *I-X and KRAFT are connected by two bi-directional AKT-Bus-compliant adaptors. Knowledge transported between the two systems is translated as it goes through the adaptors.*

While Figure 5 illustrates a conceptual overview of the integrated system, Figure 7 shows the system architecture in which I-X and KRAFT are connected by two *AKT-Bus-compliant* communication gateways.[3] The simplest form of interaction takes a client-server model where I-X sends an "*issue*" (that contains a problem description) to KRAFT to resolve and KRAFT sends back the answer. Such a simple interaction assumes that all required knowledge is readily available and the "*issue*" can be resolved in one interaction.

The same architecture, however, can support sophisticated argumentative communication languages in which the two systems are free to exchange messages of constraint specifications, partially solved problems and solutions to achieve a more complex task. Under this arrangement, the two peer systems can be viewed as software agents that propose constraints, counter-propose constraints and partially solve a CSP, thus modifying the initial problem specification. This is especially useful if two departments in a virtual organisation disagree and/or wish to negotiate new possibilities. This is an example of the power of Agent Mediated Knowledge Management [9].

---

[3] *AKT-Bus* is a HTTP and RDF-based communication protocol developed in the AKT project to support the integration of heterogeneous systems.

The inability to fully exchange knowledge in a virtual organisation is a common phenomena. One possibility is that different departments do not share a common but only use a partially overlapping ontology. As a result they can only exchange semantic knowledge that is commonly understood. For instance, the *Technical Department* has the technical details of hardware components, but may not (nor care to) have knowledge of costing and sales. The *Sales Department*, on the contrary, may have some general knowledge about PC components, but is really only specialised in cost calculation and market prices. One common cause of only partially sharing information may be the unwillingness to disclose local knowledge. A department may wish to keep its information private for commercial confidentiality reasons, e.g. calculation methods for product market price based on cost, or protection for advanced and competitive technologies.

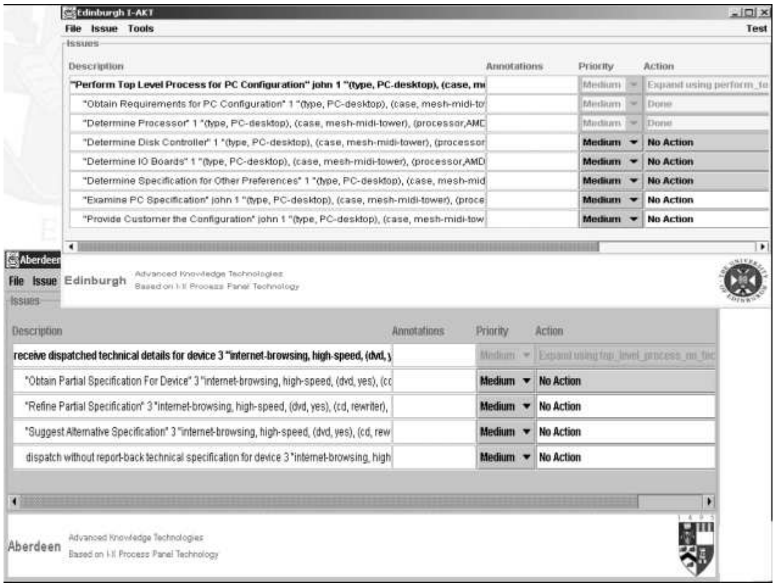
Our system copes with this by passing object IDs and constraints referring to entity types declared in the shared part of the ontology, but it encapsulates in different domains (see Figure 7) the processes that reason about them or access specific object properties. Thus we only pass between domains information that the other end ”needs to know”. The mapping of entity types and constraints between the different representational spaces (I-X/FBPML and KRAFT/CIF) takes place in the mappers shown in Figure 7.

4.4 Implementing the Workflow

In this experiment, two I-X Process Panels have been used. This enables work items to be described and transferred between organisations and assists the collaboration between them. The sales and technical units are each represented by the ’Edinburgh’ and ’Aberdeen’ panels indicating their site locations. One of the tasks that needs to be resolved on the Edinburgh site requires technical abilities in PC configuration. The sales unit of Edinburgh naturally passes this task to its technical counter-part in Aberdeen for support. As this problem may be resolved using Constraint Satisfaction Problem (CSP) solving techniques, the Aberdeen site makes uses of its local CSP solver, the KRAFT system, provided with the necessary details about the problem. After execution, the KRAFT system returns the solution (or acknowledge of failure) to the Aberdeen I-X panel, which returns the solution to the Edinburgh site. If a satisfactory solution was not found, the sales department may decide to find alternative answers through new enquiries. Figure 8 gives a screen shot of the two I-X Process Panels, where [10] stores a live recording of a system run. Although in this experiment, we have used two I-X agents to manage our workflow for simplicity, it is not a limitation and more than one heterogeneous workflow agents may be involved in the process of collaboration, provided that appropriate mapping between processes has been carried out.

5 Conclusion

One vision of the Semantic Web is to enable principled and heterogeneous machine processing abilities on knowledge rich web resources, and thereby automate execution of users’ tasks on the web. Our work demonstrates a collaboration between two systems that makes use of semantic rich technologies and uses a semantic web compliant language for their communication medium.


Edinburgh I-AKT
File Issue Tools
Test
Issues
Edinburgh
Advanced Knowledge Technologies
Based on I-X Process Panel Technology
Aberdeen
File Issue
Issues
Aberdeen
Advanced Knowledge Technologies
Based on I-X Process Panel Technology

Description	Annotations	Priority	Action
"Perform Top Level Process for PC Configuration" john 1 "(type, PC-desktop), (case, m		Medium	Expand using perform_to
"Obtain Requirements for PC Configuration" 1 "(type, PC-desktop), (case, mesh-midi-to		Medium	Done
"Determine Processor" 1 "(type, PC-desktop), (case, mesh-midi-tower), (processor,AMD		Medium	Done
"Determine Disk Controller" 1 "(type, PC-desktop), (case, mesh-midi-tower), (processor		**Medium**	**No Action**
"Determine IO Boards" 1 "(type, PC-desktop), (case, mesh-midi-tower), (processor,AMD		**Medium**	**No Action**
"Determine Specification for Other Preferences" 1 "(type, PC-desktop), (case, mesh-mid		**Medium**	**No Action**
"Examine PC Specification" john 1 "(type, PC-desktop), (case, mesh-midi-tower), (proce		**Medium**	**No Action**
"Provide Customer the Configuration" john 1 "(type, PC-desktop), (case, mesh-midi-tow		**Medium**	**No Action**

Description	Annotations	Priority	Action
receive dispatched technical details for device 3 "internet-browsing, high-speed, (dvd, y		Medium	Expand using top_level_process_on_tec
"Obtain Partial Specification For Device" 3 "internet-browsing, high-speed, (dvd, yes), (cc		**Medium**	**No Action**
"Refine Partial Specification" 3 "internet-browsing, high-speed, (dvd, yes), (cd, rewriter),		**Medium**	**No Action**
"Suggest Alternative Specification" 3 "internet-browsing, high-speed, (dvd, yes), (cd, rew		**Medium**	**No Action**
dispatch without report-back technical specification for device 3 "internet-browsing, high		**Medium**	**No Action**

Fig. 8. *Collaboration through I-X Process Panels*

The two connected systems are of very different natures. The I-X system is workflow and process modelling based, whereas KRAFT has a constraint solving representation and problem solving capabilities. The underlying models of the technologies of the two systems as well as their domain ontologies are firstly mapped onto each other to enable them to cooperate in a complementary manner. In addition, our chosen communication interface, an RDF based representation, between the systems is relatively open. It enables us more easily to collaborate with new systems, thus making our framework extendable with additional specialised functionalities.

Our work has been successful in achieving its objectives, but much mapping effort was needed in the earlier stages of the project since various concepts could not be mapped easily. Practical solutions, such as extending each of the concept sets had to be found. This echoes existing knowledge sharing and interoperability problems between any two or more potentially very different but partially overlapping systems that are well-known in the knowledge systems community [4].

The ultimate goal is to enable a easily and fully interoperable and collaborative Semantic Web, but there are still issues remain to be resolved before this may be achieved. They are in particular related to semantics that are deeply embedded in a system and its specific operation/reasoning mechanism that may not necessarily be fully understood or translated by another system. Our work demonstrates a small step towards this goal.

6 Acknowledgement

Research reported in this paper regarding workflow management and process modelling technologies are devised and carried out by Yun-Heh Chen-Burger. Constraint Problem Solving research is carried out by Kit-Hui. Austin Tate leads design of I-X system. Other authors provide valuable conceptual and practical advice for this work.

This work is supported under the Advanced Knowledge Technologies (AKT) Interdisciplinary Research Collaboration (IRC), which is sponsored by the UK Engineering and Physical Sciences Research Council (EPSRC) under grant number GR/N15764/01. The AKT IRC comprises the Universities of Aberdeen, Edinburgh, Open, Sheffield and Southampton. I-X research is supported by the Defense Advanced Research Projects Agency (DARPA) and US Air Force Research Laboratory under agreement number F30602-03-2-0014. The constraint fusing services were developed in the context of the KRAFT project, funded by the EPSRC and British Telecom. Kit Hui's work was performed while at The Aberdeen University as part of the AKT project.

The AKT IRC research partners and sponsors are authorized to reproduce and distribute reprints and on-line copies for their purposes notwithstanding any copyright annotation hereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of other parties.

References

1. Nick Bassiliades and Peter M.D Gray. CoLan: a Functional Constraint Language and Its Implementation. *Data and Knowledge Engineering*, 14:203–249, 1994.
2. Dan Brickley and Ramanathan V. Guha. Resource description framework (RDF) schema specification 1.0, March 2000. http://www.w3.org/TR/2000/CR-rdf-schema-20000327/.
3. Yun-Heh Chen-Burger and Jussi Stader. Formal support for adaptive workflow systems in a distributed environment. *Workflow Handbook 2003*, April 2003.
4. Flavio Correa da Silva, Wamberto Vasconcelos, and David Robertson. Cooperation between knowledge based systems. *Proceedings of the Fourth World Congress on Expert Systems*, 1998.
5. Delphi Group. *BPM 2002: Market Milestone Report*, February 2002.
6. Tim Finin, Richard Fritzson, Don McKay, and Robin McEntire. KQML as an Agent Communication Language. In *Proceedings of Third International Conference on Information and Knowledge Management (CIKM'94)*. ACM Press, 1994.
7. Peter M. D. Gray, Suzanne M. Embury, Kit-Ying Hui, and Graham J. L. Kemp. The evolving role of constraints in the functional data model. *Journal of Intelligent Information Systems*, 12:113–137, 1999.
8. Peter M. D. Gray, Kit-Ying Hui, and Alun D. Preece. An expressive constraint language for semantic web applications. In Alun Preece and D. O'Leary, editors, *E-Business and the Intelligent Web: Papers from the IJCAI-01 Workshop*, pages 46–53. AAAI Press, 2001.
9. Kit-Ying Hui, Stuart Chalmers, Peter M D Gray, and Alun Preece. Experience in using rdf in agent-mediated knowledge architectures. Technical report, Germany, 2004.
10. Kit-Ying Hui, Yun-Heh (Jessica) Chen-Burger, Steve Potter Peter Gray, Alun Preece, and Austin Tate. Kraft-i-x live demo, 2003. http://www.aktors.org/technologies/kraft-ix/movie/tie-all.html.

11. Kit-Ying Hui and Peter M. D. Gray. Developing finite domain constraints – a data model approach. In *Proceedings of the 1st Internatinal Conference on Computational Logic (CL2000)*, pages 448–462. Springer-Verlag, 2000.
12. Kit-Ying Hui, Peter M. D. Gray, Graham J. L. Kemp, and Alun D. Preece. Constraints as mobile specifications in e-commerce applications. In *9th IFIP 2.6 Working Conference on Database Semantics (DS-9), Semantic Issues in e-Commerce Systems*, pages 357–379, 2001.
13. Peter Lawrence, editor. *Workflow Handbook 1997*. John Wiley and Sons Ltd. in Ass. with the Workflow Management Coalition, 1997.
14. Thomas W. Malone, John Quimby, Kevin Crowston, Abraham Bernstein, Jintae Lee, George A. Herman, Brian T. Pentland, Mark Klein, Chrysanthos Dellarocas, Charles S. Osborn, George M. Wyner, and Elisa O'Donnell. Tools for inventing organizations: Toward a handbook of organizational processes. *The MIT Process Handbook*, pages 13–37, 2003. http://ccs.mit.edu/ph/.
15. Richard Mayer, Christopher Menzel, Michael Painter, Paula Witte, Thomas Blinn, and Benjamin Perakath. *Information Integration for Concurrent Engineering (IICE) IDEF3 Process Description Capture Method Report*. Knowledge Based Systems Inc. (KBSI), September 1995. http://www.idef.com/overviews/idef3.htm.
16. Alun Preece, Kit-Ying Hui, Alex Gray, Philippe Marti, Trevor Bench-Capon, Zhan Cui, and Dean Jones. KRAFT: An agent architecture for knowledge fusion. *International Journal of Cooperative Information Systems*, 10(1 & 2):171–195, 2001.
17. Alun Preece, Kit-Ying Hui, and Peter Gray. An FDM-based constraint language. In Peter M. D. Gray, Larry Kerschberg, Peter J. H. King, and Alexandra Poulovassilis, editors, *The Functional Approach to Data Management*, pages 417–434. Springer, 2003.
18. David Profozich. *Managing Change with Business Process Simulation*. Prentice Hall PTR, 1998.
19. Craig Schlenoff, Michael Gruninger, Florence Tissot, John Valois, Joshua Lubell, and Jintae Lee. The process specification language (psl): Overview and version 1.0 specification. *IS-TIR 6459, National Institute of Standards and Technology, Gaithersburg, MD (2000)*, 2000. http://www.nist.gov/psl/.
20. Austin Tate. I-X: Technology for intelligent systems. *www.i-x.info, AIAI, The University of Edinburgh*, 2002.
21. U.S. Department of Defense. *Leading Change in a New Era*, May 1997.