Supporting Collaboration through Semantic-based Workflow and Constraint Solving

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 a shared ontology that allows two open systems of very different natures to communicate, perform specialised tasks and achieve common goals. We give an account of our approach and the semantic basis of the workflow and constraint languages. We argue that having open systems built on Semantic Web standards is an effective way to provide inter-operable Web services. 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 be located 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 collaborate and utilise this distributed knowledge and resources to achieve a common goal requires smooth and effective co-ordination.

Workflow and Business Process Modelling are well-recognised techniques for promoting and achieving effectiveness and efficiency in the co-ordination of distributed organisational operations [16]. Workflow technology originated from data (control) flow

* Level 4, Appleton Tower, Crichton Street, Edinburgh EH8 9LE. Tel: + 44 131 650-2756, Fax: +44 131 650-6513, Email: jessicac@inf.ed.ac.uk, a.tate@ed.ac.uk

** St. Andrew Street, Aberdeen AB25 1HG, UK. Tel: +44 1224 262-708, Fax: +44 1224 262-727, Email: khui@comp.rgu.ac.uk

*** King's College, Aberdeen, Scotland, UK, AB24 3UE, Tel: +44 1224 272-291, Fax: +44 1224 273-422, Email: apreece@csd.abdn.ac.uk, pgray@csd.abdn.ac.uk

diagrams. These technologies are also 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 they too may be formalised and described using similar process technologies [17]. This recognition and advancement in this area explains the popularity of business process re-engineering and change management [24, 21]. 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 [8]. We argue that a third generation workflow system may be one that understands and manipulates semantics rich processes and information and can operate in a distributed agent based system architecture.

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 still need specialised problem solvers that find solutions to problems. To achieve a meaningful task, one also has to exchange task-specific semantic knowledge between a workflow system and problem solvers. This requirement of semantic knowledge exchange between knowledge-utilising components becomes more important as tasks get semantically richer and interactions become more sophisticated.

Agent based workflow systems that understand process models at different levels of abstraction simplify managing run-time execution as well as monitoring progress. On the other hand, semantic-based workflow covers internal as well as external organisational operations, enabling them to be grounded on the explicit semantics that typed dataflow provides. The emergence of Semantic Web (SW) technologies provides a flexible infrastructure for the exchange of semantic knowledge that enables systems to communicate, reconfigure and collaborate among themselves [25]. By providing a semantic-web-compliant gateway, a system can be offered as a web service, thus allowing it to openly communicate with any other web services providers and clients.

In this "*Technology Integration Experiment*" (TIE) supported by AKT [1], we show how agent-based workflow management and constraint reasoning systems can be connected using Semantic Web technologies to achieve collaborative tasks. While this paper focuses on the collaboration between the two systems, we briefly describe our semantic-based workflow language that has be mapped to and deployed through agent-based systems. It involves two complimentary aspects of semantics, the data and process layers. These connect to the data and process layers in FDM, a Functional Data Model used in KRAFT (section 4.2) and in database interoperation. FDM data types are modelled as entity types with subclasses and object instances, while constraint knowledge is modelled in first-order logic. This mathematical similarity to the FBPML workflow language (section 3.1) greatly eases communication of semantics. It underlies our first step towards a semantic-based network for non-trivial collaboration between web services.

2 A Motivating Scenario

Consider a computer company that builds and sells PCs to customers, with each department in the company possessing specific and local expertise. The *sales department*, which handles user orders, may have knowledge of budget and cost control. The *technical department*, on the other hand, is responsible for building the ordered PCs and has the expertise of putting hardware components into a workable system. Figure 1 shows a conceptual view of the two departments connected through a workflow system.

While a workflow system coordinates the information flow inside the company, there is a distinctive flow of *"user requirements"* and *"PC configuration"* in the form of semantic knowledge between the two departments. For example, the sales department may pass an order to the technical department with the user requirements as constraints, and the technical department uses its expertise on hardware to configure a usable PC that meets the user's need. The complete scenario requires the collaboration of a workflow system and a constraint solver so that both cost and technical constraints are satisfied.

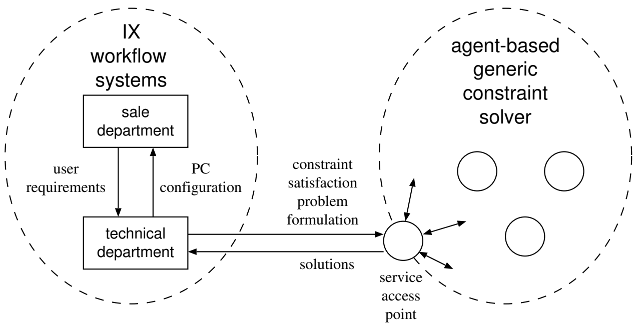


Fig. 1. *An open systems architecture allows participants to benefit one another via collaboration*

This systems architecture allows participants to benefit one another while allowing their full potential to be exploited via collaboration. This system architecture is also an open one. In this example, two workflow systems coordinate information flow between the *sales* and *technical* departments while the *technical* department connects to a problem solver[1]. However, more agents may be added to this architecture when appropriate. In general, there is no limit to the number of systems being connected. One basic problem to resolve, however, is the exchange of semantic knowledge between systems of potentially very different natures. To provide a better understanding for the task at hand, Section 3 describes the enabling technologies.

3 Background Technologies

3.1 Formal Business Process Modelling Language (FBPML)

FBPML [6] adapts and merges two recognised process modelling languages: NIST PSL (the Process Specification Language) [22] and IDEF3 [18]. NIST PSL provides formal semantics for commonly shared process modelling concepts as well as theories, such

[1] Other departments are omitted from the figure for simplicity.

as situation calculus, that support the use of such concepts. As it was designed to be an interchange language between different process languages, it covers the core concepts required for process models, but does not provide visual notations or model development methods.

IDEF3 originated from concurrent engineering disciplines and is one of the richest methods available for process modelling. It provides visual notations, rich modelling method and guidelines. Nevertheless, its semantics is informal and its models therefore are open to interpretation. Combining the two different methods retains IDEF3's rich visual and modelling methods and provides the formal semantics and theories of PSL, so that reasoning mechanisms and formal analyses can be performed on those models. In addition, precise process execution logic has been devised such that a virtual workflow machines may be created and process enacted at run-time, which was not possible before for IDEF3. A FBPML description makes use of a data language, the FBPML-DL, that provides descriptions for data constructs and becomes an integral part of a FBPML (process) description. This data language may be used on its own to describe a domain [5] and has been mapped to the KRAFT representation (See Section 3.3) to describe the PC configuration data model.

FBPML Data Language (FBPML-DL) The FBPML Data Language (FBPML-DL) has a strong basis in logic [4]. It is based on first-order predicate logic and set theory. Its syntax is in Prolog so can be easily manipulated by software. FBPML-DL has four parts:

1. Foundational Model provides concepts, predicates and functions of background theories that are used in the language. The primitive predicates provided here are used to define other predicates in the other parts of FBPML.
2. Core Data Language introduces core predicates and functions for concepts that are common to many applications. Their semantics are defined using constructs from the Foundational Model.
3. Extension Data Language includes predicates and functions that are additional to the Core Data Language and are often application and domain-dependent.
4. Meta-predicates may define axioms of an application model.

Predicates may be used to describe instances of classes such as processors, disk controllers, slots; subclass relations and other relations such as allocate, and attributes such as capability, length and power. Given this, a FBPML process or constraint thus may instantiate its attributes using FBPML-DL constructs. Figure 2 gives a static constraint description in FBPML with two arguments: precondition of the constraint and the constraint content. This constraint restricts the total cost and the allowed allocation of different boards in the slots. This includes instances of the core data language of FBPML-DL as part of predicates that are defined in the FBPML process model.

3.2 I-X technology

I-X [23] is a rich systems integration architecture. It stores process models and supports dynamic instantiation and viewing of processes. Different communication strategies can

```
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. 2. *Example FBPML Constraint*

be installed that enable communication between multiple I-X agents as well as non-I-X (web services compliant) agents. Some communication methods are 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 task 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 the dynamic instantiation and management of process instances as well as storage for process models. It has also been used as a communication medium between I-X and a constraint solver, the KRAFT system. As I-X is based upon a conceptual framework of *<I-N-C-A>* that comprises issues, nodes/activities, processes, constraints and annotations, conceptual mapping between FBPML, <I-N-C-A> and KRAFT therefore 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 [15, 19]. 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.

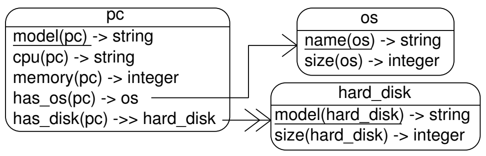


Fig. 3. *This schema shows three entity classes. The single arrow indicates each* `pc` *may have only one* `os` *installed. A double arrow indicates a* `pc` *may have multiple* `hard-disk`.

Figure 3 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 [2]. CoLan constraints have evolved from database state restrictors in the P/FDM database system[2] into mobile problem specifications [10]. Figure 4 shows example CoLan constraints, with the second one being 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.

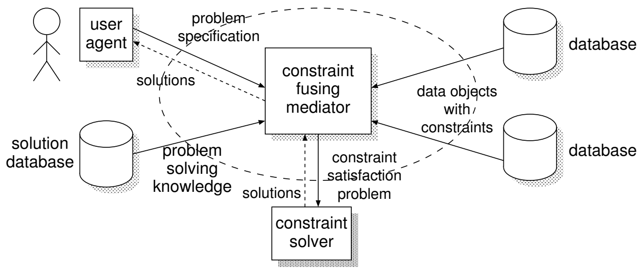
```
constrain each p in pc
  to have size(has_os(p))
     =< size(has_disk(p))

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

Fig. 4. *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 [9]. The underlying language, Colan, has evolved into an

[2] http://www.csd.abdn.ac.uk/∼pfdm/

RDF-based Constraint Interchange Format (CIF) [11, 12]. At the heart of the system is a *constraint fusing mediator* that combines constraint fragments from distributed sources (Figure 5). 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 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 [14].



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. 5. *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

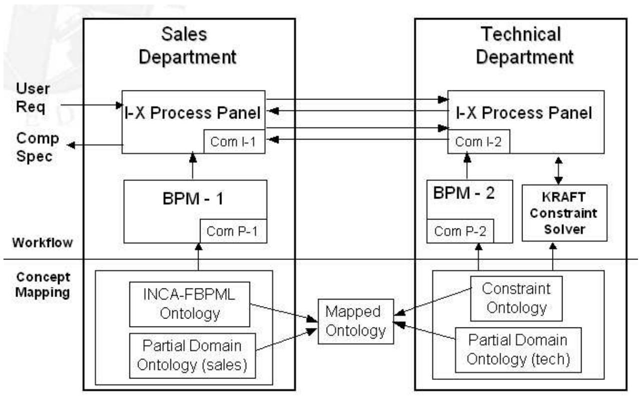


Fig. 6. *Conceptual architecture of collaboration between two (sub)organisations in the PC configuration domain*

Figure 6 gives an overview for a conceptual architecture that enables collaborative problem solving using semantic-based workflow techniques. A horizontal line has been drawn 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 (Figure 1). 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 6. 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/IX and *KRAFT constraint solver*. The process of conceptual mapping also indicates patterns needed for correspondence that form the bodies of communication. The communication processes *(Com P)* are a recognised type of process and are clearly labelled in an FBPML process model.

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[3], otherwise "fail" is returned if no solution is found.

[3] 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 [3]. An interpreter reads meta-data from the database and *generates* a corresponding RDFS description, making meta-knowledge web-accessible. The RDFS fragment in Figure 7 refers to the P/FDM database schema in Figure 3.

```
<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>
```

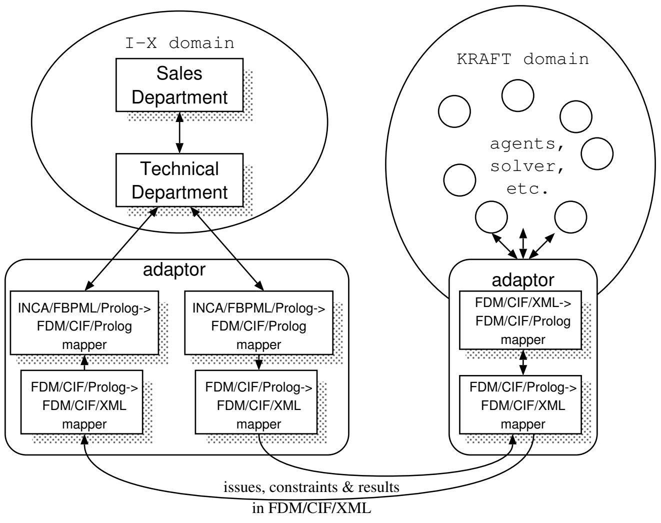
Fig. 7. *RDFS description of a P/FDM database schema*

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 [11].

Constraints In practice, human-readable CoLan constraints such as those in Figure 4 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 [11]. 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 [11,20]. 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. 8. *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 6 illustrates a conceptual overview of the integrated system, Figure 8 shows the system architecture in which I-X and KRAFT are connected by two *AKT-Bus-compliant* communication gateways.[4] 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 [12].

[4] *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 8) the processes that reason about them or access specific object properties. Thus we only pass between domains along the AKT-Bus 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 8.

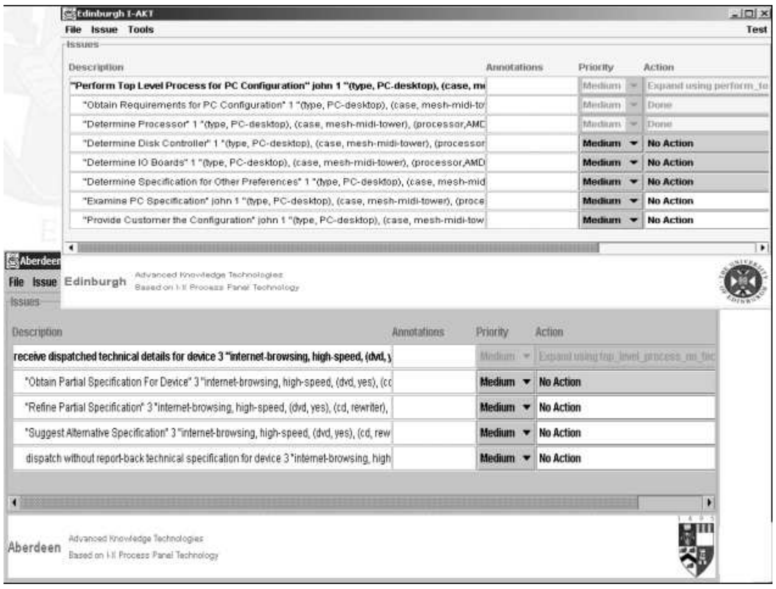
4.4 Implementing the Workflow

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, providing 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 9 gives a screen shot of the two I-X Process Panels, where [13] stores a live recording of a system run.

5 Conclusions and Future Directions

“The Semantic Web is the representation of data on the World Wide Web. It is based on the Resource Description Framework (RDF), which integrates a variety of applications using XML for syntax and URIs for naming.” – W3C Semantic Web working group [25].

The Semantic Web is an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation. – Tim Berners-Lee [25].



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

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

Description	Annotations	Priority	Action
receive dispatched technical details for device 3 "internet-browsing, high-speed, (dvd, y		Medium	Expand using top_level_process_on_tic
"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

Aberdeen
Advanced Knowledge Technologies
Based on I-X Process Panel Technology

Fig. 9. *Collaboration through I-X Process Panels*

One vision of the Semantic Web is to enable machine processing on web information resources, and thereby automate execution of users' tasks on the web. The Semantic Web itself, however, does not create or invent any semantics but provides the infrastructure and mechanisms for the representation, communication and utilisation of semantic knowledge. The semantic knowledge does not come from the Semantic Web but comes from the applications that are built upon the Semantic Web. While much effort is focused on Semantic Web languages research, we believe that applications on the web also play a crucial role in pushing the overall development.

This "*AKT Technology Integration Experiment*" (TIE) demonstrates a collaboration between two systems of very different natures: the I-X and KRAFT systems. Each is based on workflow and constraint solving technologies that are complimentary in terms of collaborative problem solving and task accomplishment. By mapping domain knowledge with Semantic Web compliant languages, such as RDF/RDFS, these specialised systems can be offered as web services. In encoding the semantic knowledge, we have thus deployed a RDF-based approach in representing the domain knowledge in KRAFT as well as its communication language, CIF.

Our work has been successful in the defined task, but much mapping effort was needed in the earlier stages of the project as not all modelling concepts can be mapped and fully translated, so practical solutions must be found. This echoes 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 [7]. We therefore argue that the dream of the Semantic Web vision is real, but there are still issues that require much effort before its maturity. They are in particular related with semantics and operation/reasoning that are used by a system but may not be fully understood, or understood from a different perspective, by another system. The ultimate goal of the Semantic Web is to provide ways of connecting arbitrary open systems to achieve non-trivial tasks using semantically rich knowledge. The *I-X-KRAFT "TIE"* is only a small step towards this goal.

6 Acknowledgements

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. AKT Consortium. *Advanced Knowledge Technologies (AKT) Interdisciplinary Research Collaborations (IRC) project. Partners: University of Southampton, Aberdeen, Edinburgh, Sheffield and Open University, UK, http://www.aktors.org*, October 2001.
2. Nick Bassiliades and Peter M.D Gray. CoLan: a Functional Constraint Language and Its Implementation. *Data and Knowledge Engineering*, 14:203–249, 1994.
3. 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/.
4. Yun-Heh Chen-Burger. Informal semantics for the fbpml data language. *Informatics Report Series: EDI-INF-RR-0154, School of Informatics, the University of Edinburgh*, October 2002.
5. Yun-Heh Chen-Burger. *AKT Research Map, Advanced Knowledge Technologies (AKT) IRC Project technology show case*. http://www.aiai.ed.ac.uk/ jessicac/project/3-akt-res-map-tech-profile-sub/details.html, 2003.
6. Yun-Heh Chen-Burger and Jussi Stader. Formal support for adaptive workflow systems in a distributed environment. *Workflow Handbook 2003*, April 2003. (Workflow Management Coalition is one of the most influential standard organisations for business process modelling and workflow.).
7. Flavio Correa da Silva, Wamberto Vasconcelos, and David Robertson. Cooperation between knowledge based systems. *Proceedings of the Fourth World Congress on Expert Systems*, 1998.
8. Delphi Group. *BPM 2002: Market Milestone Report*, February 2002.
9. 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.

10. 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.
11. 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.
12. Kit-Ying Hui, Stuart Chalmers, Peter M D Gray, and Alun Preece. *Experience in Using RDF in Agent-mediated Knowledge Architectures*, pages 177–192. Springer, Germany, 2004.
13. 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.
14. 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.
15. 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.
16. Peter Lawrence, editor. *Workflow Handbook 1997*. John Wiley and Sons Ltd. in Ass. with the Workflow Management Coalition, 1997.
17. 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/.
18. 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.
19. 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.
20. 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.
21. David Profozich. *Managing Change with Business Process Simulation*. Prentice Hall PTR, 1998.
22. 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/.
23. Austin Tate. I-X: Technology for intelligent systems. *www.i-x.info, AIAI, The University of Edinburgh*, 2002.
24. U.S. Department of Defense. *Leading Change in a New Era*, May 1997.
25. World Wide Web Consortium. *Semantic Web home page*, 2004. http://www.w3.org/2001/sw/.