SPATIAL DATABASE UPDATE AND VALIDATION – CURRENT ADVANCES IN THEORY AND TECHNOLOGY

P. A. Woodsford

Spatial, Cavendish House, Cambridge Business Park, Cambridge, CB4 0WZ and Dept. of Geomatic Engineering, University College, Gower Street, London, WC1E 6BT, UK - consultancy.woodsford@ntlworld.com

KEY WORDS: Data Models, Data Quality, Knowledge Management, Rules-based Processing, Service Oriented Architecture, System Architecture, Update, Workflow

ABSTRACT:

The paper summarises the importance of spatial data quality, especially as data is used more and more in automated processes and decision-making, and as data is integrated or 'joined-up' from diverse sources. The distinction is made (Busch et al, 2004) between

- logical consistency, i.e. consistency with respect to the data model,
- consistency as regards content, i.e. consistency of data and reality within the scope of the model.

The emphasis in this paper is on logical consistency, although some approaches to consistency of content are also discussed.

Three particular strands of development are discussed:

- the trend to more complex and capable data models, the rationale of these and the consequent implications for data quality and updating
- advances in rules-based processing and formalised semantics
- the trend towards service-oriented architectures and the development and adoption of relevant standards

Radius Studio is used as an example of a modern rules-based processing environment. Key use cases for the application of rules-based processing to ensure data quality in update processes are discussed, including validation and cleaning of existing data, the validation and committal of batches of detected changes (local and remote) and data conflation. Modern developments in workflow management are briefly described including the use of the Business Process Execution Language (BPEL).

1. THE IMPORTANCE OF DATA QUALITY

As the amount of spatial data available increases very rapidly, and tools for locating, viewing and using this data become much more readily available, the need for understandable and reliable measures of data quality becomes more important. So too does the need for tools to validate spatial data and to ensure data quality across the variety of collection and update processes in use. ISO 19113:2002 (ISO, 2002) establishes the principles for describing the quality of geographic data and specifies components for reporting quality information. It also provides an approach to organizing information about data quality. Complementing the progress within International Standards Organisation (ISO), the Open Geospatial Consortium (OGC) has recently established a Data Quality Working Group (OGC, 2007).

Sonnen has recently highlighted the importance of data quality as spatial data moves into enterprise environments (or as mapping moves to spatial information) thus:

> *'Data quality is a problem we need to address if we in the geospatial industry expect to be a part of the enterprise IT picture. Our most pressing need is a simple, reliable way to answer: "Are these data fit for this purpose?" each time spatial data are merged or shared in an enterprise system.'* (Sonnen, 2007)

Sonnen also highlights the fact that data quality issues may be resolved or exacerbated within each data management function. Careful design of data management workflows can minimize problems. This paper focuses on quality issues in update workflows.

2. A NEW GENERATION OF DATA MODELS

Another fundamental factor is the emergence of a new generation of data models, designed to be more capable and to serve multiple purposes, in short to realise the goal of 'store and update once, use many'. A recent EuroSDR Workshop (EuroSDR, 2006) explored this trend, the reasons for it, and the current state-of-the-art. It revealed an encouraging degree of convergence and consensus in the emerging pattern (reliance on standards, feature/object based models, use of persistent unique identifiers, use of UML as a design tool, use of XML/GML as delivery vehicle) and some outstanding research issues, particularly in expressing rules and in formal semantics. These trends are not restricted to Europe and are evident globally.

What is strongly evident is the trend to database-centric architectures. This centralising tendency, together with the implications of more complex data models, has profound implications for update (Woodsford, 2004). Update processes need to be 'datamodel-aware', and to be tightly coupled with validation services, to avoid costly and lengthy error detection-correction cycles. Whilst it is possible to replicate such validation services in each update client (field, photogrammetric systems, imagery change detection systems) as models become more complex, and business rules come to the fore, such an approach becomes more costly and difficult to sustain. Traditional approaches, with several lengthy iterations to detect and correct errors (see Fig. 1) become les and less viable.

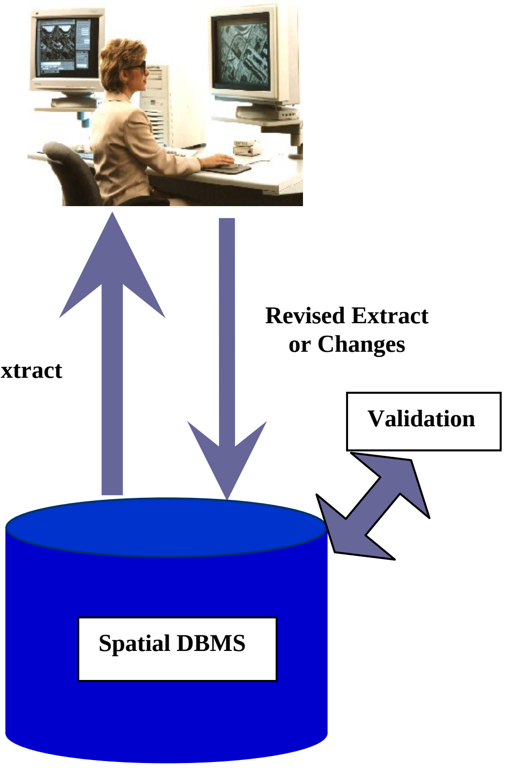


Fig. 1 File-based Data Exchange. Several cycles over extended timescales and very low efficiency.

Good results have been achieved by close-coupling of a photogrammetric client with an object-oriented database (eg BAE Systems SOCETSET with Laser-Scan (now 1Spatial) Gothic (Edwards et al, 2000) or with a geodatabase (SOCETSET with ESRI ArcGIS, (BAE Systems, 2007)). This tightly-coupled approach (see Fig. 2) removes the need for repeated re-validation of changes and lengthy delays.

Very considerable gains in efficiency are achieved, but the rules base used for validation is buried within the Object-Oriented database. This is not ideal from the management and maintenance perspective. Furthermore object-oriented database management systems (OODBMS) have not become part of the Information Technology mainstream.

Modern multi-tier architectures extract the business rules into a middle tier component, for ease of creation, maintenance and scalable access by all client processes (see Fig. 3).

Service-oriented architectures, described in more detail below, take this one step further, to a distributed services architecture.

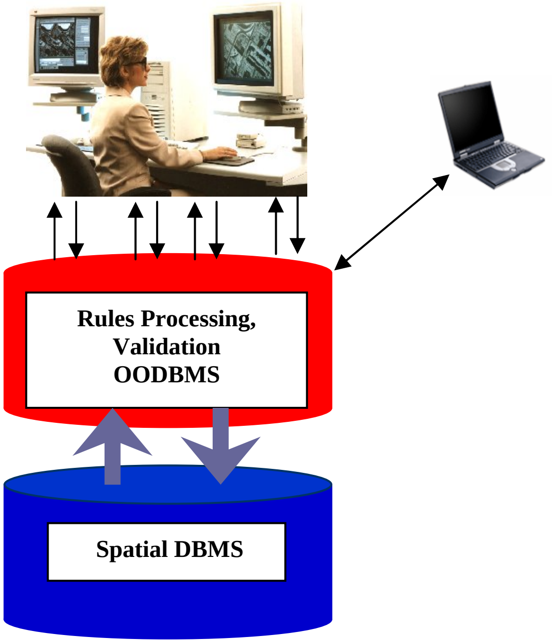


Fig. 2 Direct Link with Active Validation and Underlying Database. Single cycle operation.

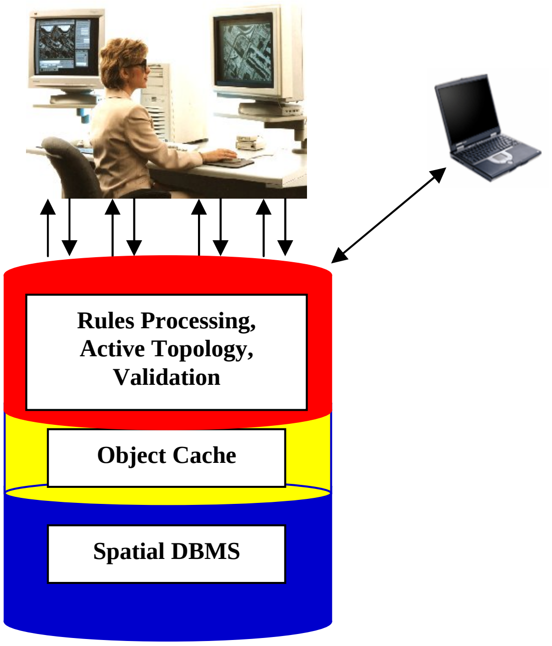


Fig. 3 Direct Link with Multi-tier Architecture.

3. ADVANCES IN KNOWLEDGE MANAGEMENT

Rapid advances in Knowledge Management in Information Technology are beginning to result in benefits in the geospatial realm. Historically, knowledge is buried inside data or in point applications or hidden in people's heads. This results in serious problems in keeping such knowledge up to date. Progress towards rigorous semantics contributes to removing ambiguities and to storing the knowledge/expertise of the organisation where everyone can contribute to it and share it, as enterprise metadata that is portable and independent of specific datasets and systems. A critical component in this development is a rules language to enable logical constraints to be specified. Such a language needs to be unambiguous, logical and portable, compact, intuitive, quantitative, web compatible and declarative and refinable. For a full discussion of these requirements and potential choices of rules languages, see (Watson, 2007). This area has received much attention of late through initiatives such as the Semantic Web community (W3C, 2004b) and rapid progress can be anticipated.

There are currently several candidates to consider as a knowledge representation language (RDF, OWL, XML Rules/SWRL). None as yet cover all the functionality needed in the geospatial domain. Radius Studio (1Spatial, 2007) from 1Spatial (formerly Laser-Scan) is a rules-based processing environment, implemented both as middleware and as a service, that is used for both domain discovery and for conformance checking. Rules-based processing follows the FACT-PATTERN-ACTION dynamic. Given some facts, if they meet any of the patterns/rules, perform the defined action(s). FACTs are a known data source. PATTERNs are the business rules that the data source obeys or should obey. ACTIONs happen as a result of PATTERNs being applied to FACTs as illustrated in Fig. 4

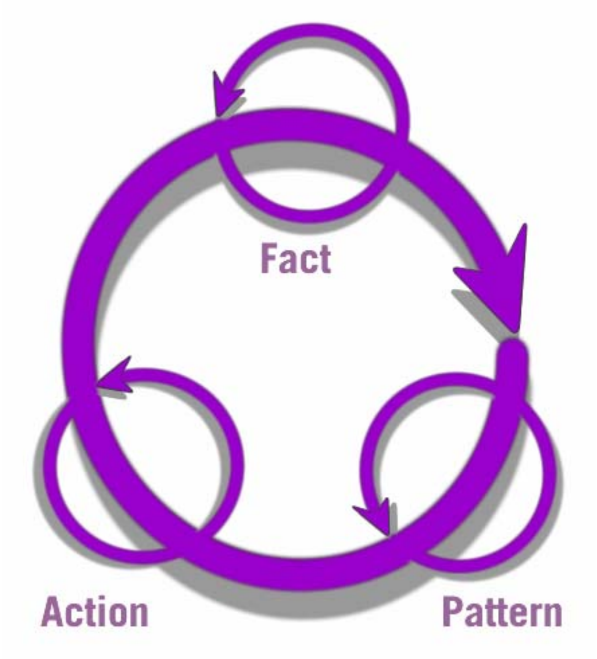


Fig. 4 The Fact-Pattern-Action Dynamic.

4. THE RADIUS STUDIO EXAMPLE

4.1 Architecture

Radius Studio is an implementation of a rules-based processing environment. It can be deployed as an instance the generic multi-tier architecture as shown in Fig. 5 below, or as a Web Service. It is in the process of being deployed on a grid architecture.

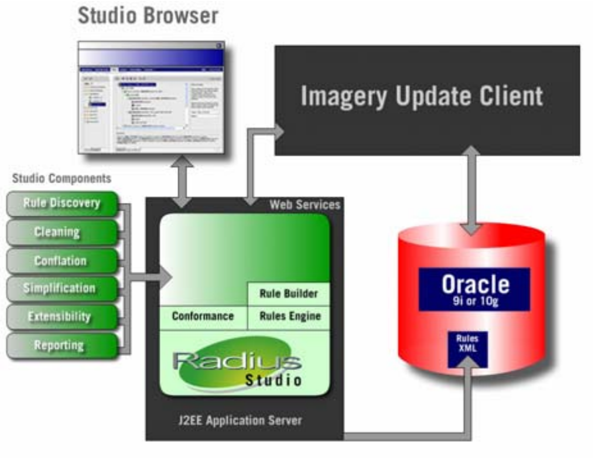


Fig. 5 Radius Studio as a Multi-tier Architecture.

4.2 Overall Workflow

A Radius Studio session is a sequence of processing tasks. The following types of task may be included in a session:

Open Data: Enables access to data from a data source. A session may choose to open data from a number of data sources and then check rules based on relationships between features stored in different locations.

Discover Rules: Analyses data based on a discovery specification to identify candidate rules.

Check Rules: Checks a defined set of rules on the data and reports non-conformances.

Apply Actions: Apply a defined set of actions to the data independently of the business rules.

Apply Action Maps: Apply one or more action maps to the data to identify and resolve non-conformance to business rules.

Commit: Updates a data source with data changed as a result of applying actions or action maps.

Copy To: Copies data from the Radius Studio workspace to a data source.

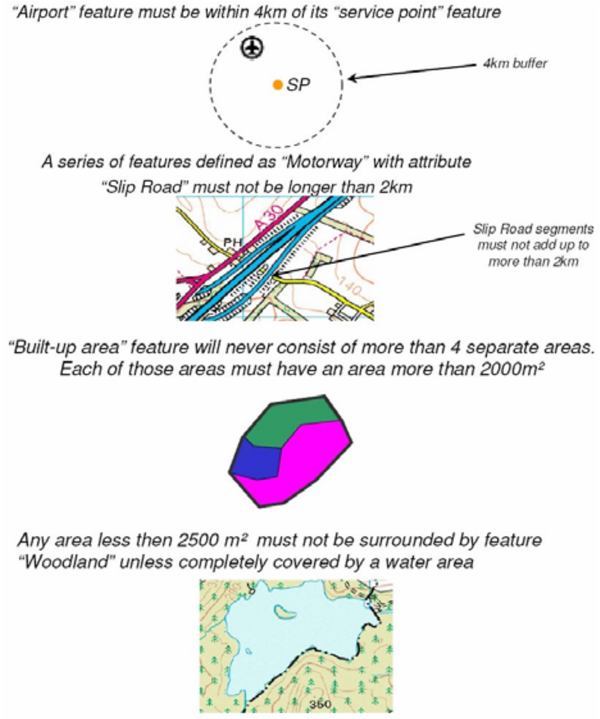
Pause: Requests Radius Studio to suspend execution, to allow results to be examined before processing the next task. When a session completes, all intermediate data is discarded. A Pause Task may be added at the end to retain data in order to retain the data for further processing.

Tasks can be added, deleted and re-ordered. All required parameters can be browsed and updated. Multi-level undo/redo can be used to correct mistakes. Media player style controls are used to control the execution of tasks.

4.3 Defining and Validating the Rules-Base

The rules-base is a set of conditions that objects from the data store should satisfy. A rule in Radius Studio is a tree of predicates against which objects can be tested. Rules are expressed in a form independent of the schema of any particular data store. This means they can easily be re-used with different data sources.

Before formally defining the rules for use within Radius Studio, they must be articulated and understood. A wide range of circumstances are encountered. The rules may be defined in text form, perhaps in conjunction with a logical data model or feature catalogue (see Fig. 6). They may be formally expressed in an ontology language such as Ontology Web Language (OWL, W3C 2004a), in which case they can be directly used (by interfacing with the open source Jena ontology library (Jena, 2007)). More often the rules are not explicit or formalised, but exist in the form of knowledge held by domain experts.



"Airport" feature must be within 4km of its "service point" feature

A series of features defined as "Motorway" with attribute "Slip Road" must not be longer than 2km

"Built-up area" feature will never consist of more than 4 separate areas. Each of those areas must have an area more than 2000m²

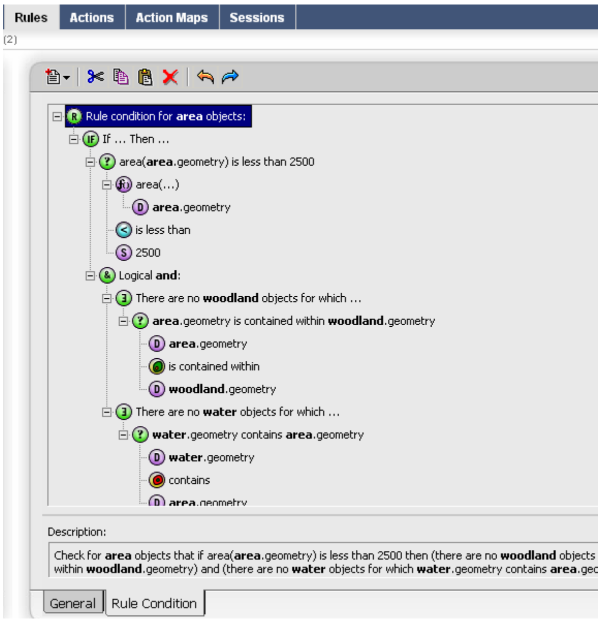
Any area less then 2500 m² must not be surrounded by feature "Woodland" unless completely covered by a water area

Fig. 6 Some Sample Rules Defined by Textual Descriptions

It is not unusual for there to be no explicit rules-base available. For this situation Radius Studio provides a rules-discovery function which analyses data from a data source, looking for patterns in it, and uses statistical techniques to deduce a plausible set of rules that the data appears to satisfy. In all events, the validation of the rules-base by applying it to exhaustive representative datasets is a key stage before using it in earnest. This is particularly so when the 'rules discovery' function is used. Typically validating the rules-base will require extensive evaluation of test cases with the domain experts, and several cycles of rules refinement. The desirability of achieving clarity over the rules before seeking to define them for the rules-based processing engine is highlighted in the Radius Studio training documentation by using the non-trivial example of the offside rule in football.

Radius Studio provides an intuitive web-based interface for defining rules and building up a rules-base. (see Fig. 7) The rules builder allows the definition of potentially complex rules with an easy to use, tree structured browser interface. The rule is expressed as a series of clauses built up using pulldown menus from the bar immediately above the graphical illustration of the rule.

The description at the bottom provides English text representing the currently selected clause; in this case the complete rule. The element details are used to specify the parameters associated with the currently selected rule. This part of the form always includes a description of the information required. In this case, the top-level rule specifies the class to be checked. An optional name label is used when the rule needs to distinguish between two different features of the same class. While editing a rule, it may temporarily be incomplete until a new clause is added or parameters are defined. These problems are highlighted clearly in red and a description of what is required displayed. Multi-level undo/redo is available to recover from mistakes while editing. Drag and drop can be used to reorder clauses of a rule. Cut and paste can be used to transfer all or part of a rule into another rule.



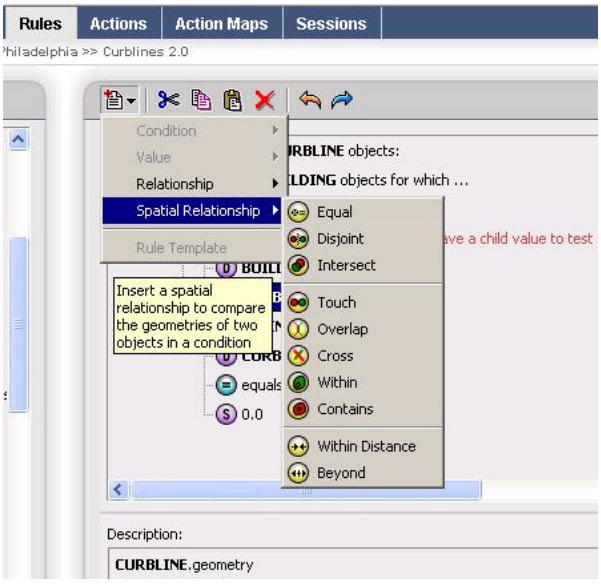
```
Rule condition for area objects:
  If … Then …
    area(area.geometry) is less than 2500
      area(…)
        area.geometry
      is less than
      2500
    Logical and:
      There are no woodland objects for which …
        area.geometry is contained within woodland.geometry
          area.geometry
          is contained within
          woodland.geometry
      There are no water objects for which …
        water.geometry contains area.geometry
          water.geometry
          contains
          area.geometry
```

Description:

Check for **area** objects that if area(**area**.geometry) is less than 2500 then (there are no **woodland** objects within **woodland**.geometry) and (there are no **water** objects for which **water**.geometry contains **area**.gec

Fig 7. The Rules-Definition Interface, defining the fourth rule of Fig. 6

The rule builder uses a syntax that is very much like English, so programming experience is not necessary. A rule is built up as a set of components (also called clauses or nodes) in a tree structure. There are a number of different conditions that may be used in a rule. A comparison takes a value, a relationship and a second value and returns a boolean (true or false) to indicate whether the values fulfil the relationship (scalar or spatial). The data types of values that may be used depends on the relationship. Tests include tests for existence (against conditions), for references between objects and for values (against ranges). The ISO/OGC Simple Feature specification spatial interaction types (ISO 19125-2:2004) are implemented, and take those meanings, together with two distance operators ('within distance' and 'beyond'), see Fig. 8. Chaining can be used to link feature elements together to form a higher-level feature.



Description:

CURBLINE.geometry

Fig. 8. Spatial Relationship and Distance Operators

Radius Studio implements a number of built-in functions, most of which take one or more parameters and all of which return a result. These functions are divided into logical groupings: geometric, conversion, mathematical, bit manipulation and string. Aggregate functions calculate a single result from one or more input values, specified as the result of a sub-condition.

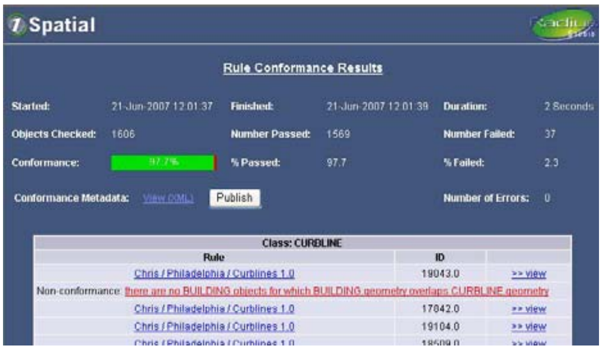
4.4 Actions and Action Maps

An action is a procedure (or set of procedures) to be applied to one or more objects, usually when they are found to violate a particular rule. Actions are expressed in a form independent of the schema of any particular data store, so that they can easily be re-used with different sources of data. Actions are defined using a similar graphical user interface as for defining rules, but can also include operations such as assignment, conditionals and sequencing, object creation and deletion and report generation. Actions can be applied to all the objects from a data store, or in a more targeted manner by use of action maps.

An action map is an ordered list of (rule, action) pairs. When an action map is invoked, objects are checked against the rules. If the object does not conform to that rule, then the associated action is invoked. Action maps are often used to associate fixes to problems with non-conforming objects.

4.5 Measuring and Improving Data Quality

Radius Studio can be used to measure the quality of data in the sense of measuring the degree of conformance of the data to a rules-base. A document or data store can be processed and a report generated with results of the conformance test at both a summary and individual feature level. The summary reports the proportion of compliant objects. Additionally a list of non-compliant objects is generated (as HTML or as an XML-file, with object identifiers). This consists of per-object metadata detailing which object infringed which rule (see Fig. 9) The rule identifier and text is included in each Object element with any unique key attributes and the bounding box (envelope) of the checked feature geometry attribute. Detailed feature level metadata contained within the Object elements can be used in manual reconciliation processes to locate and correct data.



Rule Conformance Results

Started:	21-Jun-2007 12:01:37	Finished:	21-Jun-2007 12:01:39	Duration:	2 Seconds
Objects Checked:	1606	Number Passed:	1569	Number Failed:	37
Conformance:	97.7%	% Passed:	97.7	% Failed:	2.3
Conformance Metadata:	View (XML) Publish			Number of Errors:	0

Class: CURBLINE

Rule	ID	
Chris / Philadelphia / Curblines 1.0	19043.0	>> view
Non-conformance: there are no BUILDING objects for which BUILDING geometry overlaps CURBLINE geometry		
Chris / Philadelphia / Curblines 1.0	17042.0	>> view
Chris / Philadelphia / Curblines 1.0	19104.0	>> view
Chris / Philadelphia / Curblines 1.0	18509.0	>> view

Fig. 9 Report of Conformance Test Results

Of course, the value of such a measure as a measure of quality is dependent of the validity/quality of the rules-base. Quality can be improved by automatically applying ‘fixes’ to cases within allowable tolerances or parameters, as defined by an action map, or by referring the list of non-compliant objects to an interactive client. Tasks can be added, deleted and re-ordered. All required parameters can be browsed and updated. Pauses can be inserted at any stage in processing sequences and multi-level undo/redo can be used to correct mistakes.

When an acceptable level of quality is reached, all objects that have been changed in the Radius Studio workspace are committed back to the main data store.

The final results of the conformance tests are obtained in the form of metadata which is compliant to the conceptual model of ISO 19115 Metadata (ISO 19115:2003) and encoded in the form recommended in ISO 19139. The results are supplied within the DQ_DataQuality metadata element as DQ_Element descriptors. The nameofMeasure and measureIdentification are taken from the corresponding rule or ruleset identifier. The dateTime is taken from the completion time of the conformance check and the results (DQ_Result) are compiled from the appropriate summary statistics within the conformance checking session. The metadata can be published automatically to a compliant OGC Catalogue (OGC, 2005) for long-term archiving and to facilitate discovery of data with appropriate quality characteristics.

5. USE CASES

5.1 Validating and Cleaning Existing Data

A rules-based processing environment provides a very suitable tool for validating and cleaning existing data and for providing a quantitative measure of data quality or conformance. This is important in itself, but also in the context of quality improvement programmes and in the context of update. It is important that update processes do not cause quality impairment or violation of the rules and constraints of data models.

5.2 Validating Detected Changes

Change information arises from a number of different sources including field observations and survey, reports from external sources and changes detected by comparison with up-to-date imagery, either interactively or by automated processes such as in the WIPKA system (Busch et al, 2004). Validation of the logical consistency of such change information, both internally and with respect to the data store as a whole, must be performed before the changes and can be accepted and committed.

5.3 Conflation

An increasingly important use case is that of conflation - a process where two spatial datasets are matched and (usually) merged into one. This may involve applying spatial transformations and/or identifying corresponding objects in the two datasets. The goal may be to align the two datasets (as in the Positional Accuracy Improvement problem) or to generate an added value dataset with enhanced information content. Conflation can be regarded as a superset of Update where some of the modifying information (eg superior attribution) already exists in digital form. Conflation is an ideal application for rules-based processing

5.4 Metadata and Service Levels

Update has to take place within a wider business context, both internal and external. Requirements exist for the traceability of information and update processes. Data providers may also need to prove that their products meet service level standards both of compliance to specifications (rules) and of currency (degree of update). Hence it is very important that update, along with other stages in data production and management, automatically produces requisite metadata and operates within a well-organised workflow. Ideally this will be integrated with the other workflows and processes of the business.

5.5 Quality Measurement and the Reduction of Uncertainty

Chi (Chi, 2007) has recently highlighted the uncertainty of critical data sources and inconsistencies between multiple data sources as key issues affecting the use of geospatial data in decision-making and in analysis. Quantitative metadata concerning data quality is important in aiding discovery of appropriate and usable data. Rules-base processing is very relevant to resolving data inconsistencies.

6. THE TREND TOWARDS SERVICE ORIENTED ARCHITECTURES

OASIS defines Service Oriented Architectures (SOA) as the following:

A paradigm for organizing and utilizing distributed capabilities that may be under the control of different ownership domains. It provides a uniform means to offer, discover, interact with and use capabilities to produce desired effects consistent with measurable preconditions and expectations.

The main drivers for SOA adoption are that it links computational resources and promotes their reuse so as to help businesses respond more quickly and cost-effectively to changing market conditions. SOA may be built on Web services standards (e.g., using SOAP, (W3C 2003)) that have gained broad industry acceptance

Radius Studio can be deployed using SOAP Web Service interfaces for the purposes of validating features remotely. In outline, the Web Services are used to first define a sequence of data processing tasks called a Session. The session is then run and rules are asserted against the data. Progress is monitored and finally, the results of the conformance test at both a summary and individual feature level are obtained (Watson, 2007). The choice to implement the conformance service using standard SOAP RPC/literal bindings and the success of the resulting scenario also shows that the methodology is suitable for integration into much richer and potentially dynamic workflow environments such as are enabled by enterprise workflow technologies like BPEL (see Fig. 9) (OASIS, 2007).

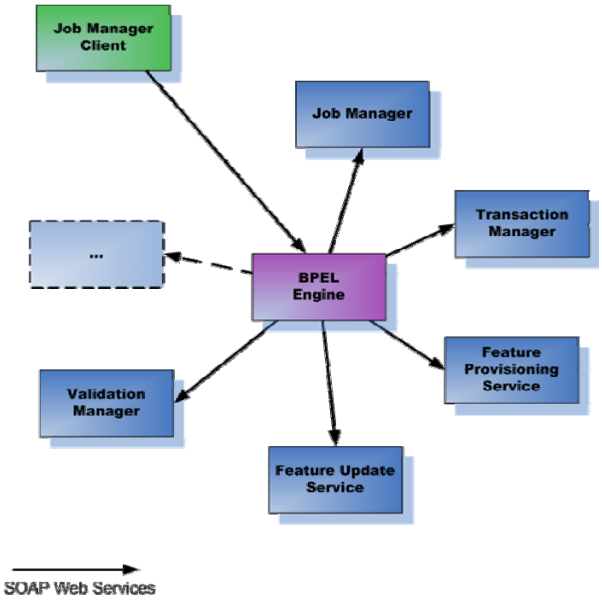


Fig. 9 A BPEL Workflow for Update.

In the 2006 OGC Open Web Services initiative (OGC OWS-4), the Geo Processing Workflow (GPW) thread demonstrated the viability of interconnecting geo-processes through service chaining and orchestration to meet workflow requirements using BPEL. One particular service was a Topology Quality Assessment Service (TQAS), which validated geospatial data against a set of topological rules, returning a conformance level and an exception list. The logical next step is a generic quality assessment service, available as a common service with a standards-base interface to all update processes, from field update to automated change detection from imagery. Such architectures and standards will emerge in the near future, and offer the exciting prospect of uniting both the content and the logical validation aspects of the general update problem.

7. CONCLUSIONS

The detection of change and the corresponding update of data are of increasing importance. The adoption of more capable, more complex data models means that to be efficient update processes must be ‘datamodel aware’. Advances in Knowledge Management technologies are happening at a rapid pace and can be harnessed to this effect. Rules-based processing affords powerful means of measuring and enhancing data quality, which must of course be maintained across all update processes. Standards-based Service Oriented Architectures are rapidly gaining adoption and acceptance and provide the basis for update processes that are both more generic and more efficient.

ACKNOWLEDGMENTS

The author is pleases to acknowledge help from colleagues at 1Spatial in preparing this paper, particularly Paul Watson and Chris Tagg.

REFERENCES

(all WWW url's validated on 21 June 2007)

1Spatial, 2007. Radius Studio Home Page at: http://www.1spatial.com/products/radius_studio/

BAE Systems, 2007. Geospatial eXploitation Products, at: http://socetset.com/about_gxp_products.htm

Busch A, M. Gerke, D. Grünreich, Ch. Heipke, C.-E. Liedtke, S. Müller, 2004. Automated Verification of a Topographic Reference Dataset: System Design and Results, ISPRS Istanbul 2004

Chi, J. 2007. An analysis of research development in spatial data quality and uncertainty modelling. 5th International Symposium on Spatial Data Quality, ISSDQ 2007, Enschede, NL. ISPRS Working Group II / 7, available at: http://www.itc.nl/ISSDQ2007/Documents/keynote_Shi.pdf

Edwards, D., Simpson J., and Woodsford P., 2000. Integration of photogrammetric and spatial information systems, International Archives of Photogrammetry and Remote Sensing, 33(B2): 603–609.

EuroSDR, 2006. Feature/Object Data Models Workshop in EuroSDR Official Publication No. 49 and available from www.eurosdr.net

ISO 19113:2002. Geographic information - Quality principles. Available at: http://www.iso.org/iso/en/CatalogueDetailPage.CatalogueDetail?CSNUMBER=26018

ISO 19115:2003. Geographic Information – Metadata

ISO 19125-2:2004. Geographic information -- Simple feature access -- Part 2: SQL option

Jena, 2007, http://jena.sourceforge.net/

OASIS, 2007. WS-BPEL 2.0 - Web Services Business Process Execution Language. http://www.oasis-open.org/committees/download.php/22036/wsbpel-specification-draft%20candidate%20CD%20Jan%2025%2007.pdf

OGC, 2005. Catalogue Service. http://portal.opengeospatial.org/files/?artifact_id=5929&version=2

OGC, 2006. Open Web Services Initiative - Phase 4 (OWS-4) at: http://www.opengeospatial.org/projects/initiatives/ows-4/#gpw

OGC, 2007. Data Quality Working Group at http://www.opengeospatial.org/projects/groups/dqwg

Sonnen, D. 2007. Emerging Issue: Spatial Data Quality. Directions Magazine, January 2007, available at: http://www.directionsmag.com/article.php?article_id=2372

W3C, 2003. SOAP – Simple Object Access Protocol. http://www.w3.org/TR/soap/

W3C, 2004a. Web Ontology Language (OWL). http://www.w3.org/TR/owl-features/

W3C, 2004b. Semantic Web. http://www.w3.org/2001/sw/

Watson, P. 2007. Formal Languages for Expressing Spatial Data Constraints and Implications for Reporting of Quality Metadata. 5th International Symposium on Spatial Data Quality, ISSDQ 2007, Enschede, NL. ISPRS Working Group II / 7, available at: http://www.itc.nl/ISSDQ2007/proceedings/Session%205%20Dissemination%20and%20Fitness%20for%20Use/paper%20Paul_Watson[1].pdf

Woodsford, P. A. 2004. System Architecture for Integrating GIS and Photogrammetric Data Acquistion. ICWG II/IV, ISPRS Istanbul 2004, The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. 34, Part XXX