



Metazette

Intellog proposes the development of *Metazette* – a new, open source software product which facilitates the generation and presentation of metadata for document collections using Atom (RSS)¹ feeds. The Atom feeds produced by *Metazette* can be consumed by existing feed readers and aggregators², which enable users to keep up-to-date on additions or other changes to the collection of documents on which the feed is based. In addition, third-parties can consume these same feeds to build either general purpose, industry-specific or other specialized indices and applications which will provide additional levels of functionality and insight the collection of documents.

Scenario: A petroleum well service company decides to publish treatment reports for jobs which have been particular successful in achieving their customers' objectives. They locate these PDF-format documents in an Internet-accessible folder, and each document is addressable through a unique URL. Metazette is implemented over the document collection, and automatically provides an Atom feed consisting of metadata about each document. An item automatically appears in the feed each time a document is added, updated or deleted from the public folder. The service company's customers subscribe to the feed using any popular web browser, allowing them to keep completely up to date on the work of their well service supplier. Simultaneously, Intellog (or other third-party) consumes and parses the same Atom feed to incorporate links to these documents into their integrated well information database.

Metazette currently exists only as a concept and vision along with descriptions of potential features. Students will be expected to provide all services required to design, build and release the product. *Intellog* will act in the role of customer/advisor, and provide input to the student development team. *Intellog* will also provide sample data, and facilitate alpha & beta sites where the software can be tested. *Metazette* is an important, but not mission critical, part of *Intellog's* overall business plan.

For additional descriptive material regarding the scope of the product, see the Pro Forma Product Backlog, on page 5, of this document

Educational Objectives Students participating in this project will have their learning experience enhanced in the following ways;

- *Whole Lifecycle Software Development* In addition to design, implementation and testing, it's intended students will release the software to the general public. As a result, students will have the opportunity to interact with real customers attempting to use the software to solve real business problems. In particular, students will be called to respond to the unexpected, but inevitable issues arising from the release of the software to a user community they do not directly control.
- *'Distributed Agile/SCRUM'* This project will provide experience with the Agile/SCRUM³ methodology where physically distributed team members will be working asynchronously, and where team communications are facilitated through a team blog. (The Agile development methodology is usually based on co-location of teams to facilitate communication between team members.) This reflects an emerging pattern for the development of open source software.
- *Open Source Development* Development tools for this project will be limited to those available with an open source license. Their use will enable the students to assess the impacts, both positive and negative, of the open source approach as opposed to one based on commercial, propriety development tools.
- *Start- Up Experience* This project will include many of the steps students will encounter should they decide to create their own start-up companies subsequent to graduation. Students will be provided with real-world experience with this potential career choice, and help them prepare for a future where traditional career development principles may not apply.

Development Environment Development tools will be limited to mainstream, open source products. While the specific set of tools will be determined by consensus of the project team (subject to final approval by *Intellog*), it is anticipated the work related to this project can be accomplished in Apache, PHP, MySQL and other, similar tools.

Additional Technical Guidelines User-interfaces will preferably be web-based, but at the very least, platform independent so they are equally suitable for either a Windows or Macintosh desktop environment. It is understood the students will provide their own PC/Mac workstation to support their individual development efforts. Either is appropriate so long as a mixture of workstation types does not impede information flow between team members. It's not anticipated any other special equipment or products will be required to complete the development. *Intellog* has already established a specific pattern for its database design, and will require the student development team to continue with this standard when the database design is extended.⁴



Development Methodology and Guidelines A modified version of Agile with SCRUM will be used. In keeping with this methodology, the student development team is expected to organize themselves and their work to accomplish the project objectives. However, a few, high-level guidelines are also provided; 2008-09 would be used to establish project scope, firm up the Product Backlog, finalize the specific development environment and pursue any other 'sprint zero' activities. This will be followed by development sprints in each calendar month 2008-10 through 2009-04, and each will deliver an increment of potentially shippable product functionality. Beta testing may start as early as the conclusion of the first development sprint. Absent a co-located team, the intra-team dialogue will be facilitated through a team blog. A brief (one hour) sprint review meeting will be held at the conclusion of each development sprint, with all team members expected to attend, in person.

Project Documentation Students will be expected to use the team blog as the main repository of project documentation. Contributions to the blog should emulate the level of detail found in current contributions to the *Intellog* Developers Journal⁵. Students will be expected to make reasonable efforts to ensure correct grammar and spelling are used, and to otherwise uphold the professional tone of the blog. In addition, there may be other documentation artifacts required by the CPSC 594 course, and it will be the students' responsibility to see these responsibilities are upheld.

Licensing *Intellog* will retain copyright on the software produced under the terms of this project, but intends to license it under the Creative Commons Attribution-Share Alike⁶ (or similar) license. By implication, students would be free to carry on the work on *Metazette* beyond the term of the CPSC 594 course, should they choose to do so. As is normally the case with open source software, other developers would be invited to contribute to the development effort, but only subsequent to the conclusion of the development period discussed in this document. Because the work of the team will be facilitated through a publically-accessible blog, there is no issue with non-disclosure agreements.

Measurement of Success Success of the project, from *Intellog's* perspective, will be measured in terms of the growth of the number of unique documents whose metadata can be found in a *Metazette* feed. When users download the *Metazette* product, they will be asked to agree to provide high-level, summary statistics of how the product is being used from which the metrics above will be derived.

Also, this project will be deemed a success if the application is sufficiently robust to become an *Intellog* product, freely-downloadable from the *Intellog* website, and



used to contribute to the achievement of *Intellog's* overall business objectives. In other words, success will have been achieved if the resulting product is used by real users, in a real environment to address real business requirements.

* * *

What's in a Name? *Metazette* is an amalgamation of the words metadata and gazette. Metadata is the fundamental output of the *Metazette* software, and gazette, when used as a noun, means newspaper. The British (primarily) also use gazette as a verb, in which case, it's defined as "[t]o announce or publish in an official journal or in a newspaper." The combination of the two concepts succinctly describes the vision of the *Metazette* software. Both *metazette.org* and *metazette.ca* domains have already been reserved by *Intellog*.



Pro Forma Product Backlog

The following are brief descriptions of features likely to be included in the *Metazette* product. These are 'placeholders' for conversations which will be used to develop detailed user stories. Order is not significant.

- **Create/Modify/Delete** Identify the location and nature of the collection of documents which are the subject of the associated feed. This would include URLs to one or more containing structures (ie. folders), and accompanying information which could help to further define the collection of documents. Examples of the latter include items such as file name wildcards, file size, modification or access date and content-type. Establishing a universally-unique name for the document collection and its associated feed is also included as deemed to be part of this story.
- **Validate/Assess/Test** Query the *Metazette* site for information regarding the size and nature of the document collection at any point in time. By doing so, test the validity of the configuration after it has been created or modified, or to routinely gather statistics such as total number of documents, along with break downs by various categories such as content type, metadata template being used, integrity of metadata, etc.
- **Add/Modify/Delete Metadata** Identify the types of metadata to be supplied in the feed, and also identify the underlying characteristics of the metadata, such as whether it's mandatory (or not) what type of access is permitted (and by whom), what values are acceptable for a particular item of metadata, and similar information. *Scenario: There is a defined list of geological formations, and well treatment reports are often related to those formations. The content of formation metadata would be limited to those values found in a table of formation data.*
- **Controlled Document Access** There will be feeds where the document collection is not intended for universal, public access. In such cases, enable the user to limit document to specific users. Furthermore, provide a range of options as to the degree access is limited. At one of the end of the spectrum, it might simply be to deny access to the document once it has been requested, or at the other, it may be to suppress the presence of the document entirely including any appearance in the associated feed. *Scenario: A well service company only wants its 'premier', high-volume customers to be able to access its treatment reports. In order to access the reports, the identity of the user needs to be added to an access control list. Any other*



party will not even be aware the treatment report exists because it never appears in the feed associated with the document collection.

- **Create/Modify/Delete an ‘Audience’** Enable the selection and collection of network identities (likely OpenIDs⁸) into a group, and enable document access to be assigned to an audience, as opposed to an individual user.
- **Document on Request** Provide evidence in the feed of a document existing without necessarily providing access to the document itself. For example, a feed indexer (such as *Intellog*), would have access to the metadata of the document, but requests to view the document are forwarded to the publisher, who can choose how to respond to the request.
- **External, Independent Reference Information** Incorporate access to external, independently-maintained reference lists (eg. the *Intellog* well list) in order to integration between heterogeneous document collections. Allow for the fact the external reference list will change and evolve over time, and that updates will not be under the control of the *Metazette* site. *Scenario: Treatment reports from two different well service companies performing services on a given well need a reference code to ensure both reports get associated with the same well. To address this, both feeds will incorporate the same, freely-downloadable Intellog well identifier⁹ in order to ensure both reports are associated with the same well.*
- **Document Collection Monitoring and Feed Generation** Each time a change is made to the document collection – regardless of whether it’s an addition, change or deletion of a document – action need to be initiated to update the feed and format it correctly.
- **Metadata Review/Approval** Where possible, metadata should be extracted automatically from the documents in the collection. However, there will a number of situations where metadata can only be partially extracted from a document, or not extracted at all. Where this situation exists, an application needs to provide access to whatever metadata has been extracted, allow a user to modify or correct it, and then release the document for use.
- **Standard Metadata Templates** Provide a library of standard metadata templates which can be shared. Use of standard templates will have the added advantage of homogenizing what are fundamentally the same



documents even though they come from different publishers. *Scenario: Fracture treatment reports produced by the well service industry need certain types of information in order to be useful to their readers. Rather than each service company coming up with their own template, allow one to be downloaded by all parties producing this type of document.*

- **Transaction Support** In the scenario where a document is requested as a result of having been found in a third-party index, provide transaction information to both parties in order to support payments between them. *Scenario: A company agrees to pay a small fee to a third-party indexer each time a document request is sourced from their index. When a request such as this occurs, provide sufficient information to both parties so the fee can be paid to the indexer of the data. The reverse may also be true – some indexers may opt to pay a source of documents for access to their information, and this type of transaction should be supported as well.*
- **Automatic Application Update** Enable users to subscribe to an updating service which will automatically determine if application updates are available. If they are, the new code is downloaded and installed on the clients' hardware, and does so in a manner where existing configurations are not disturbed.



Notes, References and Additional Reading

¹ Note that while Atom and RSS are considerably different, they are to be considered synonymous for the purposes of this document. This is not intended to be a comment on either standard.

² All current versions of the popular web browsers have some capability to subscribe and monitor Atom/RSS feeds. Feedburner (<http://www.feedburner.com>) is one example of a feed aggregator, and a reasonably complete list can be found at http://en.wikipedia.org/wiki/List_of_feed_aggregators

³ [http://en.wikipedia.org/wiki/Scrum_\(development\)](http://en.wikipedia.org/wiki/Scrum_(development))

⁴ Intellog database standards are closely aligned with those found SQL Server 2005 database standards found at <http://butzi.ca/tech/?cat=4>.

⁵ The Intellog Blog Developers' Journal can be found at <http://www.intellog.com/blog/?cat=3>

⁶ <http://creativecommons.org/licenses/by-sa/3.0/>

⁷ <http://dictionary.reference.com/browse/gazette>

⁸ <http://www.openid.net>

⁹ See *Well Identification in the 21st Century* at http://www.intellog.com/blog/?page_id=49

