DATAGRAV: A FRAMEWORK FOR KNOWLEDGE SHARING USING TRANSCLUSION ENABLED COLLABORATION MEDIA

Sergey Kochuguev
Datagrav Ltd., sergey.kochuguev@datagrav.com

Andrey Maslov
Gazprom Oil and Gas Nigeria Ltd., a.maslov@gazprom-international.com

Follow this and additional works at: http://aisel.aisnet.org/ecis2016_rp

Recommended Citation
http://aisel.aisnet.org/ecis2016_rp/18
DATAGRAV: A FRAMEWORK FOR KNOWLEDGE SHARING USING TRANSCLUSION ENABLED COLLABORATION MEDIA

Research
Kochuguev, Sergey, Datagrav Ltd., Russia, sergey.kochuguev@datagrav.com
Maslov, Andrey, Gazprom Oil and Gas Nigeria Ltd., a.maslov@gazprom-international.com

Abstract
Quoting document parts by cut and paste is the most popular method of aggregating complex data into an expert review or a business intelligence data stream. But doing so breaks the connection between the selected quote and the original document. The second possible origin of missing links between the data source and data use is the document access restriction. In this study we introduce and advocate a novel knowledge management framework with fine-grained text cross-referencing and transclusion features. On top of this, we apply a sticky access control scheme, which enables the permission profile to be applied uniformly across the data use points. Application of the Datagrav framework to the real life enterprise business intelligence data stream is discussed. The observed system behaviour is in good alignment with generic models of social change, which gives additional credibility to the proposed approach to building transclusion-enabled cooperative information systems.

Keywords: access control, collaborative work, knowledge management, permissions, social web applications

1 Introduction
Reusing parts of existing documents is an important step in producing new documents. Quoting document parts by cut and paste is the most straightforward way to do it. Defined as “reuse of document fragments with duplication” (Nelson, 1995), cut and paste is a very common task in the everyday work of a knowledge worker. In particular, in this paper we consider the industries of risk assessment, economic and political project evaluation, and decision making. At the same time, it is obvious that virtually any modern knowledge-based enterprise is subject to the same practices and problems.

The typical knowledge base of an organization contains multiple text documents with overlapping scopes. The most valuable data is being referenced multiple times and in a number of roles. For example, the statement “The political situation [in country N] will remain stable in the short-term. The probability of force-majeure will remain significantly below 5% during the coming 12 months” would be relevant for assessing all projects involving country N. The same applies to macroeconomic indicators and their assessment, e.g. “The current debt/GDP ratio is 20%, which means it will not affect macroeconomic stability in the mid-term”.

The first and most obvious problem arises when these valuable and ubiquitous statements have to be reconsidered on a regular basis. As soon as the statement is altered, the change must be propagated to all the relevant documents. In turn, that implies the need to keep a connection between the citation and its source, which is lost in the usual cut-and-paste scheme.

Moreover, if such a connection is bidirectional, it should be possible to identify the use count of text fragments (statements), favouring the most productive authors and giving additional credibility to the statements themselves. Further implications of smarter quotation schemes are numerous. The discussion on this topic was initiated in 1960 by Ted Nelson, but continues and intensifies with every decade (Deechow and Struppa, 2015). Since then, the common term for the fine-grained document part reuse
technique has been “Transclusion”. While other terms such as “Mashup” have gained some popularity recently.
However, an attempt to incorporate an access permission scheme with a statement (or document part) reuse framework, immediately leads to the situation where a link between documents is cut by a denied access permission. This problem could be worked around by bringing access rules from document level down to the document part (i.e. the basic element of transclusion) level. At the same time, it appears that the implications of this design decision are numerous and complex. The rules of individual user access level identification, application of the access profile to the given transclusion element or to the new text elements produced as a result of the user’s work are difficult to balance and prone to logical conflicts.
The rest of the paper is structured as follows. In the main body we describe our approach to building a complete transclusion-based collaboration framework which incorporates both fine-grained transclusion and access permission management. In the final section, we discuss application of the Datagrav framework to the real life enterprise business intelligence data stream and specific patterns which were observed in the data life cycle. Of special interest is the pattern of the data access profile for the given user and its evolution in time. This pattern is matched to the generic model of trust and knowledge-sharing in the lifecycle of an organization proposed by Ling (2011).

2 Design decisions behind the Datagrav framework

Datagrav is a cloud-based collaboration framework aimed at knowledge workers whose primary objective is the development of text documents. This is quite a broad scope, as “text document development” covers businesses activities in multiple industries as well as scientific, governmental or educational institutions. The novelty of the design concept is in the combination of the fine-grained transclusion mechanism, access control scheme defined on the same basic objects, record lifecycle management and search engine.

2.1 Basic Elements of the Framework Transclusion Engine

The basic minimal element of the Datagrav framework is a “record”. The record is a text element of arbitrary size, though the user interface and main usage scenario push the upper limit of its size towards several sentences, i.e. a paragraph of text. The one and only way to view the record is to look at it on a page. A page, in turn, is a sequence of one or more records, with the first one being the page title. The rest are arranged to form a sequence of text paragraphs. The opposite is also true, i.e. each record has a page associated with the given title. We call it a “ground page”. Other records could be employed on these pages. The page could be left empty, which is the default state.

With these simple rules, we’ve effectively defined a directed graph of records, which is the main notion to keep links between records in the Datagrav framework. Figure 1 illustrates a simple case with six records A · · · F and two non-empty pages grounding records A and D. It is worth noting that record C is referenced by both pages.

Despite the fact that the record link graph is directed, the Datagrav framework allows effective bidirectional traversal of this graph. In the case of Figure 1, symbols △2▽0 which could be seen next to record C indicate that the record is employed by two pages, namely A and D, while its own ground page is empty. Both “up” and “down” links are traversable with a single click.

With that, we’ve defined both an atomic element of the transclusion engine, i.e. the “record”, and the isomorphism between the transclusion graph and the text form of the pages of records. Our experience suggests that despite being very straightforward, this framework allows effective reuse of the text fragments, i.e. records. This is especially true, if each record is a lucid statement with a single subject.
The whole database is a simple card file of such statements, together with the transclusion graph gluing them together into pages of text.

![Diagram showing records and associated pages.]

**Figure 1. Records and associated pages.**

This approach allows the effective solution of the “record update” problem formulated in the introduction. If the statement of a certain record is updated, then all the documents, i.e. pages referencing this statement, need to be updated. Since the record is not copied, but transcluded to these pages, an update is implicit and fully automatic. If we need to go one level further and reconsider statements referring to the pages referring to the record being changed, bidirectional link traversal allows the user to make these revisions easily.

However, in order to achieve best efficiency of implicit updates, the database should have as few duplicate records as possible. The same applies to records with similar statements but different wording. Duplicate records could appear even in single user mode of operation, as a simple user error, i.e. typing the same text twice instead of finding the proper reference to the previous instance and writing it on the given page via the transclusion mechanism. The case of record duplication is even more common in the multi-user environment. The Datagrav framework provides a tool to merge similar records at the user’s discretion. The tool selects a single “heir” record and allows it to “inherit” an arbitrary number of similar records, replacing them in every use point, but keeping all the branches of the history tree. The tool allows the connectedness of the transclusion graph to be improved. In addition, the concept of record inheritance finds use in the workflow and project management aspects of the framework. These will be covered in the following sections.

### 2.2 Access Control

The Datagrav framework employs a fine-grained record-based access control scheme. Our experience indicates that it is convenient to have a proxy level of permission control in the form of user groups. In this case, 1:1 mapping between user groups and projects that are served by the given instance of the Datagrav framework can be established. Thus, we often refer to user groups as “project groups”. Each user could be a member of an arbitrary number of project groups.
The level of access permission is a binary value, i.e. a given project group either has or doesn’t have access permission to the given record. To summarize, in order to render a page for the given user U we go through the records of this page one by one, and check whether the user is a member of at least one project group associated with the given record. Depending on this condition, we either render the record on the given page, or skip it.

![Diagram](image)

**Figure 2. Access profile propagation**

When a new record is being created on a certain page, it inherits the access profile from the page title as it is seen by the author of the record. The importance of the latter condition is illustrated on Figure 2. Let’s imagine a consulting agency represented by user U. It provides consulting services, in the form of a text document with title A consisting of the records B, C and D. The document is then distributed to customers U1 and U2. The access profile of the records A···D could be set by user U in a manner that U1 and U2 belong to distinct project groups G1 and G2. In this case, they do not know of each other and of the fact that the content is shared with someone else. The comments that they make on the ground page of record D have disjoint access profiles of G1 and G2. This way, they are able to communicate with their consulting provider (user U) in private. At the same time, user U has the full view of aggregated requests from all his customers on a single page. At his discretion, user U can revert the communication scheme to the shared model by editing group membership alone.

It’s worth mentioning another common use case of the per-record access control scheme. It is often necessary to review complex documents (e.g. a commercial contract) with third party or different departments within an organization. At the same time, some parts of such documents often mention sensitive financial, personal or business data. Using the per-record access control, a document could be shared with all the covenanters in its entirety, yet the sensitive parts would have a restricted access profile. All the suggestions, edits or comments would be kept and versioned in a single document tree.

### 2.3 Workflow Elements of the Framework and Search Engine

To illustrate possible workflow based on records and their attributes, let’s consider the case of the “record update” problem again. Let’s assume that together with implicit record update (e.g. current oil price) we need to amend the record which requires an expert review step (e.g. oil price forecast). The user who makes the initial numerical record update then “assigns” the expert review related record to
the responsible employee. The assignment operation notifies an assignee (via email), and fills additional metadata associated with the record (e.g. due date). With that, (assignee - assigned records - due date) pairs could be mined using the search engine, by anyone with proper permission to read these records.

Let’s also consider the proper way to resolve conflicts if user U encounters records with contradictory content. Moreover, user U may lack the knowledge required to judge which record is better. Nevertheless, his duty is to document this issue and start a workflow that will eventually resolve the conflict. This could be done by creating a new record, with a question mark and a problem statement. Then, this record would inherit both contradictory versions. And finally, user U would assign the newly created request record to an employee in possession of the knowledge required. So, he both files a bug report to the responsible employee and puts warning signs around the pit. When the issue is settled by the responsible employee, he edits the requested record and enters the correct information. By means of inheritance, this record has already replaced both defective contradictory records.

As mentioned above, the Datagrav framework provides sophisticated search and indexing capabilities. First of all, the number of record employments (the number of links “up”) is a good clue to a record’s relevance, as the most valuable records are used in many documents. Then, we have a lot of metadata fields which are indexable and searchable. These fields include the record’s owner and authors (those who have contributed comments to the record’s history), record access profile, assignment and due date, file attachment type and more information besides. These metrics could be used by managers to monitor the organizational health, efficiency and advancement of each employee. The search engine is governed by the same access restriction model based on project groups, and prevention of information leakage is taken into account. For example, in Figure 2, record E will have the number of “up” references indexed as 1 for user U1, but as 2 for user U.

It is very important to make sure that members of project groups are in sync with each other with respect to the current progress of the project. A record-based search engine could be helpful here too, as it provides an “update feed” tool, which lists all the recently touched records belonging to the accessible project groups in reverse chronological order.

2.4 Knowledge Pyramid

In previous sections we’ve described the main technical aspects of the Datagrav framework design. The key driver behind these technical decisions was the constant zeal to improve the experience of knowledge workers and, even more important, to improve the quality of the expertise that is being born as a result of their work. The methodical aspect of the arrangement of hierarchical data goes back to the classical model of the DIKW pyramid. The presentation of the relationships between data, information, knowledge, and sometimes wisdom in a hierarchical arrangement has been part of the language of information science for many years (Wallace, 2007). It appeared that the classical knowledge pyramid maps to the record-based transclusion mechanism extremely well. The raw data, in the form of records, finds its way through the knowledge pyramid of the project group, gets transcluded to all the relevant ground pages which use it to support and synthesize information. Information and generalization propagates further “up” towards the “knowledge” level of the pyramid. Ambiguity or conflicts are resolved by means of the record inheritance mechanism. Finally, the real pearls of wisdom receive the due reward by getting high transclusion scores and, in turn, a boost in their search relevance.

2.5 Intelligence while executing

The novelty and relevance of employing transclusion in the generic knowledge management process is illustrated by the “intelligence while executing” usage scenario. Often, BI and Analytics divisions of an organization are managed separately from field offices and divisions involved in the main business
processes. Even the best quality report or recommendation produced by internal intelligence or analytics team will be tested against the real market situation during the execution phase – with multiple opportunities to correct the predictions and the data behind them. Unfortunately, usually field offices are not interested in propagating data “back” to the analytics division. But in the case of the common environment (informational framework) employing the transcuded global records, there will be no need to propagate anything back – as all the “analytical” records which are used, verified and modified during the execution phase are the same entities which are stored, structured and updated in the corporate knowledge base.

3 Use Case Analysis

The Datagrav framework provides numerical insight into certain patterns of data flow within an organization. Certain aspect of employee behaviour and social mobility are also recorded. In particular, using the data of the half yearly activity of circa 50 users collaborating in circa 100 project groups, we observed a specific pattern in the user access profile evolution over time. The typical user participates in many project groups, and the number of these groups most likely exceeds the number of his personal contacts. An ordinary group does not live long. The group has a purpose, which it either fulfils or fails to fulfil in a limited time. For a record, the mean number of permitted project (access) groups grows over time. For a user, the number of project groups he has access to grows faster over time than the total number of groups. However, the number of records produced by a user has an upper limit. Hence, an increase in user productivity in certain groups is due to declining productivity in other groups. And there should be a driver behind the leading increase of the number of accessible groups. Our hypothesis is that this increase is due to the growth of trust within the organization.

In this section, we map this observations and hypothesis to certain aspects of social group dynamics theory.

Due to the fact that effective knowledge-sharing is individual-based, rather than society-based, numerous researchers believe that knowledge-sharing is a type of social dealing between individuals (Riege, 2005). This is why it is crucial for an organization to apprehend the thoughts, minds, and behaviour of its workforce considering the requirement for a culture that facilitates employees to share knowledge as part of their daily work activities. The desire to share is one of the guiding principles of organization survival.

In doing so, there is the issue of trust, which is arguably one of the most crucial success factors for creating a culture that shares knowledge (Ling, 2011). Ribiére (2011) defined culture as the character or identity of an organization – how things are done in an organization. In this context, culture guides day-to-day working relationships, determines how people communicate within an organization, what behaviour is acceptable, and how power and status are allocated.

Sociability is a measure of sincere kindness among workers in a particular organization, where workers are more like friends than co-workers. Solidarity, in contrast, is a measure of the workers’ ability to pursue shared objectives quickly and effectively, in spite of their personal ties and in the best interests of the organization. In order to relate both sociability and solidarity with culture, these two dimensions are plotted against each other, revealing four different elements of culture (see Table 1).

Given the above, we assume that the introduction of a structured knowledge management methodology (in our case, based on the Datagrav framework and DIKW arrangement) pushes the organization’s sharing culture towards the “Communal” quadrant. As a reasonable generalization, it could be said that the increase in the level of trust could be the source of the growth of a certain project group access profile.
<table>
<thead>
<tr>
<th>Networked (Low Solidarity, High Trust)</th>
<th>Communal (High Solidarity, High Trust)</th>
</tr>
</thead>
<tbody>
<tr>
<td>– A lot of talks, possibility of rapid information exchange.</td>
<td>– Communication in every channel.</td>
</tr>
<tr>
<td>– Sharing of relevant information.</td>
<td>– Communication flows easily between different levels.</td>
</tr>
<tr>
<td>– Opportunities for learning and increased creativity.</td>
<td>– Relevant information is shared.</td>
</tr>
<tr>
<td>– Discussions, opinions, and suggestions are solicited and are taken into consideration.</td>
<td>– Discussions, opinions, and suggestions are solicited and are taken into consideration.</td>
</tr>
<tr>
<td>– Little commitment to shared business objectives.</td>
<td>– Equitable sharing of risks and rewards among employees.</td>
</tr>
<tr>
<td>– Management often has trouble getting employees or operating companies to cooperate.</td>
<td>– Teamwork across functions and locations, synergy, opportunity for learning and for creativity.</td>
</tr>
<tr>
<td>– High sociability.</td>
<td>– High commitment and low turnover.</td>
</tr>
<tr>
<td>– People share ideas and information with no immediate expectation of return.</td>
<td>– High consciousness of organizational identity and membership.</td>
</tr>
<tr>
<td>Fragmented (Low Solidarity, Low Trust)</td>
<td>Mercenary (High Solidarity, Low Trust)</td>
</tr>
<tr>
<td>– Selective dissemination of information.</td>
<td>– Communication is swift, direct and work focused.</td>
</tr>
<tr>
<td>– Members don’t share ideas and information with other units.</td>
<td>– Paper and memo driven.</td>
</tr>
<tr>
<td>– Talk is very limited.</td>
<td>– Productivity and performance driven.</td>
</tr>
<tr>
<td>– Documents might not be read.</td>
<td>– High level of commitment to a common purpose.</td>
</tr>
<tr>
<td>– Little commitment to shared business objectives.</td>
<td>– Rarely bastions of loyalty.</td>
</tr>
<tr>
<td>– Management often has trouble making companies cooperate.</td>
<td>– Disinclined to share if busy.</td>
</tr>
<tr>
<td>– Members try to get help without giving anything in return.</td>
<td>– Cooperation between units with different goals is even less likely.</td>
</tr>
<tr>
<td>– Minimal dependence on others.</td>
<td>– Lack of synergy.</td>
</tr>
<tr>
<td>– Few learning opportunities.</td>
<td>– Low tolerance of underperformance and even failure.</td>
</tr>
<tr>
<td>– Individual creativity but not at the group level.</td>
<td>– Minimal dependence on others.</td>
</tr>
<tr>
<td>– No identification with the institution, members might easily leave (high turnover)</td>
<td>– Reciprocity is negotiated.</td>
</tr>
<tr>
<td>– Low sociability</td>
<td>– People protect each other.</td>
</tr>
</tbody>
</table>

Table 1. The description of the four organizational culture types. Source: Ribiére, 2006.
4 Conclusion

We proposed and advocated a novel approach, which employs a balanced combination of the fine-grained transclusion mechanism, an access control scheme defined on the same basic objects, record lifecycle management and search engines. The framework design was strongly influenced by the ideas of DIKW structured information arrangement, and naturally benefits from this methodical approach.

The novelty and relevance of employing transclusion in the generic knowledge management process is illustrated by the “intelligence while executing” usage scenario. The scenario demonstrates a technique which allows the “main” informational process of an organization to coexist effectively with knowledge management, including the reuse, updating and structuring of information in the knowledge base – all based on the same records that appear in the “main” workflow.

Since the framework takes into account and logs not only data connections, but also user connections and project group membership and stats, we were able to map our real-life observations to a few predictions and recommendations of the generic sociological group theory. This mapping could be interpreted as an additional measurable support to certain aspects of the social group dynamics theory. At the same time, this adds additional credibility to the framework design decisions.

Further exploration and the application of measurable results to social group dynamics theory could be a separate and interesting research topic.
References


