Vol.2012-GN-82 No.21
Vol.2012-CDS-3 No.21
2012/1/20

クラウドサービスによる知識継承システムの実装

森口俊幸† 荒井大輔† 金井敦† 斉藤典明††

現状の情報共有システムにおいて，蓄積ファイルの活用によりノウハウを含めた情報の伝承をおこなうことが容易ではない．そこで人間の自然な記憶パターンに即した知識の蓄積・継承を実現するシステムの構築を検討した．このシステムにおいて過去の作業内容と共有すべきファイルを関連付けるため，記憶パターンの中でも特に「時間」に着目した．情報活用の具体化として，ネットワーク上に蓄積されているファイルを「スケジューラ」を用いて活用する手法を実装し，その機能確認を行った．

Implementation of the knowledge succession system by the Cloud service

TOSHIYUKI MORIGUCI† DAISUKE ARAI†
ATSUSHI KANAI† NORIAKI SAITO††

it is difficult for the present information sharing system to follow the information which includes the know-how, using the storing file. Therefore, we exploited the system that enables it to store and success the knowledge, which focus on the natural memorizing pattern of human. To correlate the past contents of works and the sharing file, we paid attention to the "time," among other memorizing patterns. As an embodiment of the information application, we took up "scheduler." Then we implement the technique which use the file stored on the network with the scheduler and inspected the function of it.

1. はじめに

昨今の Facebook や Twitter 等の SNS の普及により，これまで以上に多くの場面で情報共有が行われている．特に企業における組織活動にはメンバー同士の情報共有は必須であり，SNS 以上に情報を管理することのできるサイボウズ，アイポなどに代表されるグループウェアによりデータやスケジュールの共有が活発に行われている．このことからも分かるように現在の情報共有の必要性は疑いようがないものとなっている．

しかし，現在多くの組織において，情報が大量になりすぎて情報を探し出すことが難しくなっており，さらには組織を取り巻く環境の変化により情報が埋もれてしまい情報を探し出すことがますます難しくなっている．例えば，組織内ではジョブローテーションにより 2 年から 5 年ほどの間で部署を移動することが珍しくない．また大学等の研究機関においても毎年新入生が入り，上級生が卒業をし，メンバーの入れ替わりが生じる．その中でそれまでの仕事や研究の内容，手順など，積み重ねてきた情報を伝承する時間は限られており，通常は多くの情報が情報共有システムに残されているだけの状態となる．新しく組織に入ってきたメンバーは，その情報の存在そのものを知らなかったり，知っていてもその利用方法がわからなかったりすることが少なくなく，定期的な活動さえもが難しくなる．このように多くの組織においてノウハウを含めた情報の伝承は非常に重要な課題となっている．

情報共有を用いて知識の伝承を行うためには，常に状況が変化することを前提に情報を蓄積し活用できる仕組みが必要である．普段，人は「いつ」，「どこで」，「誰と」，「何を」したのかという情報を自然な記憶のパターンとしており[1]，我々はこれらの情報を活用し人間の自然な記憶パターンに則した情報検索を実現することにより，組織における知識の蓄積と警鐘継承を実現する方法を検討している[1]．具体的には，「時間」，「場所」，「人」の 3 種類の情報（ここでは“３W 情報と呼ぶことにする”）を活用する方式に注目している．

この論文ではこれら 3 種類の方式のうち「時間」を中心に 3W 情報を扱うために「スケジューラ」の活用を取り上げる．ここでは，蓄積ファイルの活用により作業の流れの習得や，様々な判断を助けることのできる知識継承の実現を目指している．具体的には，メタデータとクラウドサーバーを利用したファイルの管理と，ファイルの保存から蓄積ファイルの活用までの一連の流れをカレンダー表現のスケジューラを介してサポートできるクラウド上のシステム実装を行ったので報告する．このシステムによ

† 法政大学
Hosei University
†† NTT 情報流通プラットフォーム研究所
NTT information Platform Laboratories

Vol.2012-GN-82 No.21
Vol.2012-CDS-3 No.21
2012/1/20

り，過去の作業内容を視覚的に認識し，必要なファイルを容易に抽出できることが期待できる．

2. 方式

2.1 全体の方式

自然な記憶パターンに則した知識の蓄積・継承を実現する方法として「時間」，「場所」，「人」の３種類の情報活用のうち，特に過去の作業内容と共有するべきファイルを関連づけることが期待できる「時間」による方式から着手することとした．「時間」に関する情報活用の具体化として「スケジューラ」を取り上げ，ネットワーク上に蓄積されているファイルをスケジューラを用いて活用する手法を実現した．

実装において，スケジュールは常に参照する必要があるためネットワーク上のどこからでも利用できる必要がある．また，高速な動作とスケーラビリティを重視するため，本研究では図１のようにメタデータとクラウドサーバーを利用したファイルの管理と，ファイルの保存から蓄積ファイルの活用までの一連の流れをカレンダー形式のスケジューラを介してサポートできるシステムをクラウド上で実装を行った．

組織における知識の継承を支援するシステムの利用においては，まず通常活動において生産される様々なファイルをスケジューラを介してクラウド上のストレージに蓄積する．次に，現在の活動を遂行するために過去の活動情報を参照する際には，スケジューラの中から該当の活動を選び過去のファイルを探し出し利用することを想定している．この時，一般的にはファイルはディレクトリ構造から取り出すことになるが，本システムでは図２のようなスケジューラ上のリンクからファイルを取り出すことが特徴である．スケジューラ上のリンク先は全て実ファイルストレージでありカレンダー上からファイルに直接アクセスすることでファイルを取り出す．

図３に本方式の全体図を MVC モデルとして示す．ビューとして保存ビュー，検索ビューの二つ，コントローラとして保存コントローラ，マイニングコントローラの二つ，モデルとしてメタストレージモデル，実ファイルストレージモデルの二つをそれぞれ作成した．全体の流れとしては，まず保存ビューから保存コントローラにファイルとメタデータを送信し，実ファイルをファイルストレージモデルに保存し，その保存場所を保存コントローラに渡す．保存コントローラは実ファイルの保存場所を含めたメタデータをメタデータストレージモデルに保存する．検索では検索ビューからキーワードや範囲での検索をかけ，マイニングコントローラを通してメタデータストレージからマイニングモジュールへと渡される．渡されたメタデータはマイニングモジュールによりスケジューラ型に変換され検索・結果ビューにスケジューラに実ファイルとリンクされて表示される．

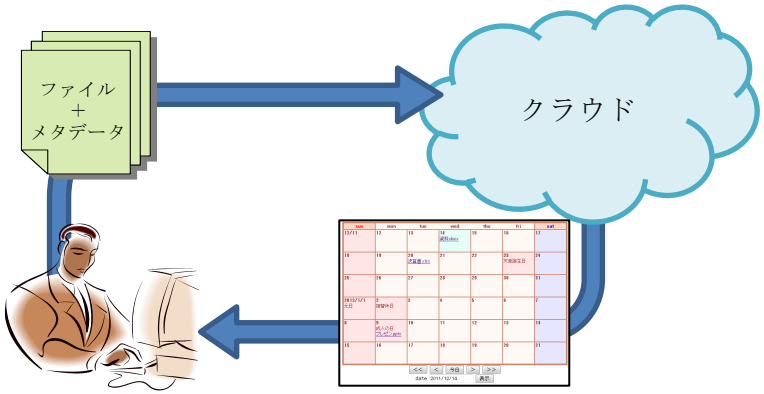

図１．システム全体の概念図

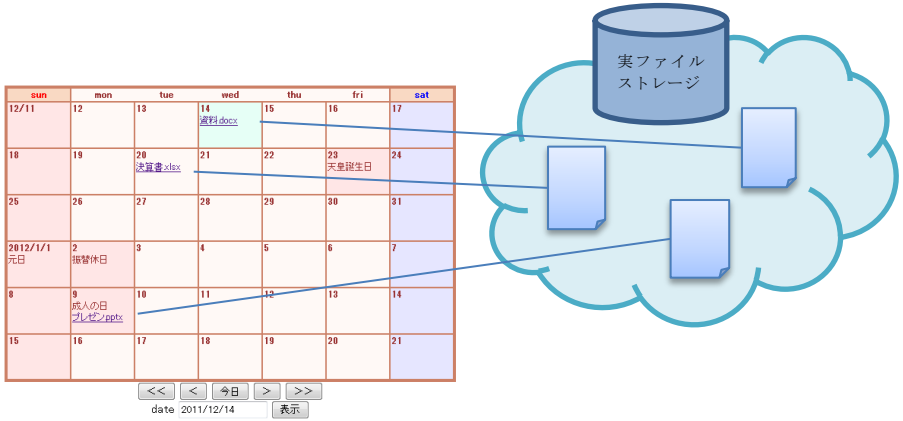

図２．スケジュール表示とファイル実体とのリンク

Vol.2012-GN-82 No.21
Vol.2012-CDS-3 No.21
2012/1/20

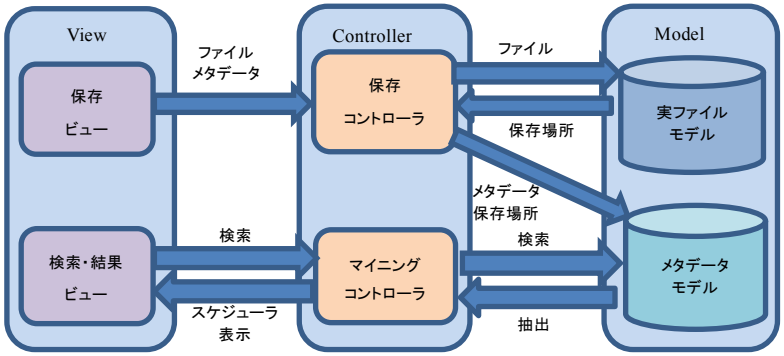

図３．全体の MVC モデル図

2.2 方式細分化

(1) メタデータ

ファイルの種類は非常に多く，様々なファイルを一元管理する為，共通したメタデータを設定する．メタデータは情報共有システムにファイルを保存する際に入力してファイルとともにサーバーに送る．本研究ではあらかじめ多様な検索条件で抽出するためのメタデータの種類を考案する．

また，より高速な検索を実現するため，アップロードを行う際にデータは実ファイルとメタデータとに分離してアップロードされる．さらに分離された実ファイルは実ファイルストレージに保存され，実ファイルの保存場所がメタデータに追記される．実ファイルの保存場所を追記されたメタデータはメタデータストレージにデータベースとして保存される．

(2) データマイニング

検索を行う場合は端末上で検索ビューを開き，そこにキーワード等を入力し，サーバーに送る．サーバーは送られてきたキーワードをもとにメタデータストレージから抽出を行い，抽出されたファイルをスケジューラ表示に変換して端末側に表示する．

(3) スケジューラ表示

視覚的に定期的な活動の情報を把握できるようにするため，表示はカレンダー形式にて行う．スケジュール上では各日付のスペースに予定やファイル名が表示させ，リンクにより実ファイルに直接アクセスすることが出来るようにする．

3. 実装

3.1 全体構成

全体の実装は主に Google App Engine (GAE)，Slim3 を用いて行った．プログラミングに使用した言語は Java 言語と JavaScript である．クライアント端末側のビュー部分はブラウザ上から GAE のビューを表示し，コントローラ，モデルとなるサーバー側部分は主に GAE と Slim3 を用いて実装した．全体の構成は図４のようになる．

3.2 データの流れ

(1) ファイルの蓄積

ファイルの蓄積は JSP で作成した保存ビューを通して行う．保存ビュー上にて保存するファイルを選択し，フォームにメタデータの必須項目と任意項目を入力し，実ファイルとメタデータをセットにして GAE 上のウェブアプリケーションに送信する．送信されたデータはウェブアプリケーション上で保存コントローラが受け取る．さらに実ファイルは Blob 型のデータとして Blobstore に保存し，Blob キーとして保存場所をメタデータとともに Datastore に保存する．

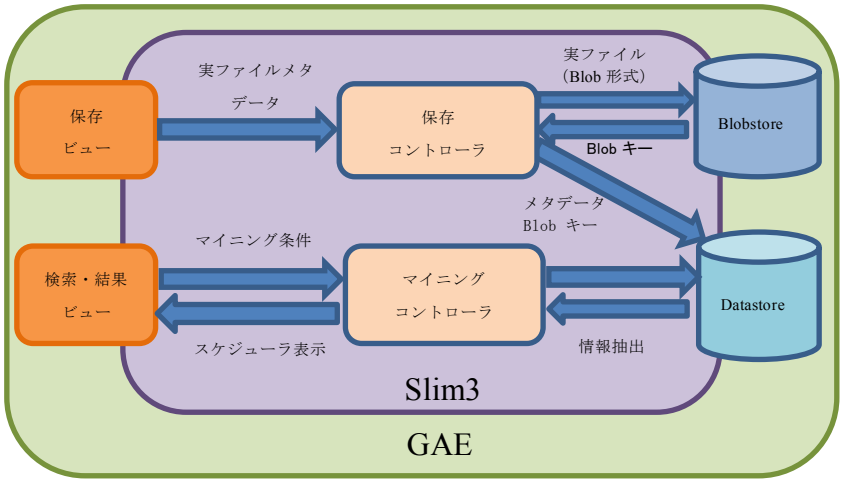

図４．全体構

Vol.2012-GN-82 No.21
Vol.2012-CDS-3 No.21
2012/1/20

(2) ファイルの検索

初めに検索・結果ビューから検索条件を入力し，それをウェブアプリケーション上のマイニングコントローラに送信する．マイニングコントローラはそれらの検索条件をもとに Datasotre からメタデータとそれに関連するファイルの Blob キーを抽出し，JSP でリストとカレンダーを用いて表示する．

(3) データベース管理

データベースの管理はまず消去したいデータを検索・結果ビューから検索を行い，リスト及びカレンダーで表示させる．消去の場合はリストの中から消去したいデータを選択し消去ボタンを押すことで Datastore 及び Blobstore から消去される．一度登録したデータの編集を行う場合は消去同様リストの中から編集を行いたいデータを選択し，編集ボタンを押すことで編集ビューへと移動し，そこで行われた編集が Datastore に上書きされる．Blobstore の制約上 Blob データそのものは更新することが出来ない．このため現状では更新機能はついていないが，Blob データそのものを更新するのではなく新しい Blob データをアップロードし，Blob キーを入れ替えることにより今後実装を行う予定である．

3.3 メタデータの構成

はじめに，検索をする際に必要となるメタデータの構成を表１のように設定した．入力するメタデータは大きく２種類に分かれる．まず，「予定開始日時」，「予定終了日時」等，スケジュール管理に必須となる項目．そしてその他３W 情報に関連する「予定の作成者」や，「ファイル名」，「ファイルの作成日時」等ファイルにかかわるもの，検索に利用できる「キーワード」等の任意入力項目である．それぞれのメタデータの項目は予めウェブアプリケーション上で定義しており，それに基づいてファイル保存時に入力を求めるように入力フォームを設計する．また初期状態からファイルに付随するメタデータに関しては自動で取り込めるように実装を行う．

表１．メタデータ

必須項目（スケジュール）		任意項目	
予定開始日時	予定件名	ファイル名	ファイル属性
予定終了日時	予定内容	ファイルの種類	ファイル所有者
予定作成日時	予定場所	フォルダーパス	ファイル作成コンピュータ
予定最終更新日時	予定任意の ID	ファイルサイズ	GPS 緯度
予定更新回数	予定プライバシー	ファイル作成日時	GPS 経度
ファイルのリンク	予定外部向け表示	ファイル更新日時	キーワード
ステータス		予定作成者	

3.4 各種モジュール

(1) データモデル設計

データモデルの設計は主に Slim3 の機能を用いて行った．あらかじめモデルの各項目に対してセッター及びゲッターの２つの Slim3 のメソッドをそれぞれ用意することでクエリの実行を容易にすることができる．

(2) 保存ビュー

ファイルの保存ビュー実行イメージを図５に示す．実装は GAE のサーブレットと JSP を用いる．サーブレット側において JSP を用いて動的な表示画面を表示させる．今後は Adobe AIR を用いることにより，ブラウザを必要とせずに実装することと，ドラッグ・アンド・ドロップで保存するファイルを直接選択できる機能の実装を行う予定である．

図５．保存ビュー

Vol.2012-GN-82 No.21
Vol.2012-CDS-3 No.21
2012/1/20

(3) 検索・結果ビュー

検索・結果ビューの実行イメージを図 6 に示す．実装は保存ビュー同様 GAE のサーブレット及び JSP を用いる．結果ビューは動的なものとし，検索の都度抽出された予定を全てリストおよびカレンダー上に書き込み表示する．カレンダー上の各日付にできるようにした．また，リストのチェックボックスから任意で選択して削除およびはファイル名が表示され，リンクにより Blob 上に保存されたファイルに直接アクセス編集も可能とした． その他，データマイニング後のリスト表示においても表示するプロパティの項目を任意に選択し必要とする情報のみの表示を可能とした．今後はカレンダー上の予定をクリックすることで詳細な予定を動的にポップアップ表示させ，その中にファイルのリンク埋め込む予定である．

(4) 保存コントローラ

手入力のメタデータに加え，もともとファイルに付随するメタデータも自動的に取得してデータベースに保存できるようにした．

(5) マイニングコントローラ

マイニングコントローラもデータモデル設計同様 Slim3 の機能を用いて行う．GAE はビッグテーブルのインデックスの仕様上，クエリの操作は前方一致しか利用が出来ない．そこで Slim3 に備わっているフィルタインメモリ，ソートインメモリ機能を用いてマイニングを行う．図 7 に示すようにまず，Datastore から一度メモリ上に抽出を行う，そのうえで各検索に合わせソートを行い，フィルタをかけることで必要とするデータの抽出を行う．

(6) データベース管理コントローラ

データベース管理コントローラは 2 種類に分けて作成した．まず一つは削除コントローラである．データベースの削除を行う場合検索・結果ビュー上から削除コントローラに移動して行う．検索した結果は前述のとおりリスト表示とカレンダー表示による出力が同時に行われる．このうちリスト表示にチェックボックスを実装し，任意のエンティティを選択することでデータストア内のエンティティと Blobstore 内の実ファイルの削除行うことが出来る．

二つ目はデータベース編集コントローラである．データベースの編集を行う場合はデータベースの削除同様に検索･結果ビュー上から編集コントローラに移動して行う．チェックボックスで任意のプロパティ選択し，編集ボタンを押すことでデータベース編集コントローラに移動しデータベース編集ビューが表示される．ここで編集を行うことでデータベースが更新される．ただし，現状では Blobstore 内の実ファイルの編集は実現できていない．データベース削除コントローラ及びデータベース編集コントローラの実装イメージを図 8 に示す．

図 6．検索・結果ビュー

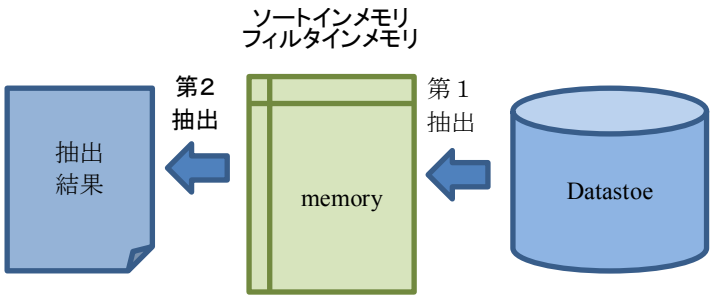

図 7．メモリ内ソート・フィルタ

Vol.2012-GN-82 No.21
Vol.2012-CDS-3 No.21
2012/1/20

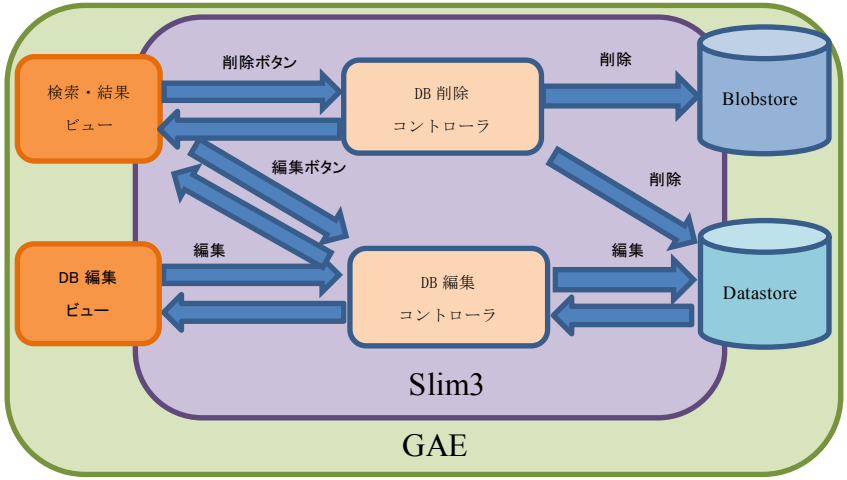

図８．データベース管理

3.5 実装環境

(1) Google App Engine

情報共有におけるメタデータは非常に膨大な量になることが予想される．そこで実行速度とスケーラビリティ観点からクラウドサーバーを利用する．

本研究ではクラウドサーバーの中でも特に高速な検索とスケーラビリティに優れたキーバリューストア型データベースが利用できる Google App Engine (GAE) を採用した． GAE を使用するにあたり Datastore をメタデータストレージ，Blobstore を実ファイルストレージとして使用する．これらはどちらも GAE のビッグテーブル上に存在しており，キーバリューストア型のデータベースのため，検索条件や，応答時間などに制限がかかる．例えば，同じカラムについて２つ以上の大小比較を行うことができないといった問題や, WEB アプリケーションとして稼働するサーブレットの実行時間が 30 秒に限られるといった問題が生じる．これらの問題は Slim3 を利用して解決した．

(2) Slim3

GAE のデータベース管理には標準の API より高速動作が可能な Slim3 を採用した．Slim3 とは Google App Engine/Java 用に最適化されたフルスタック MVC フレームワークである[2]. これにより MVC モデルでの設計を容易に行うことが可能になる他, 様々な制限のある Datastore において RDB ライクな操作が可能になる．GAE はビッグテーブルのインデックスの仕様上，クエリの操作は前方一致しか利用が出来ないものであったが，メモリ内でフィルタをかけることにより，あいまい検索や，同じプロパティ内での複数回の大小比較を行えるようになる．

4. 評価

(1) GAE の各種制限

GAE によるプログラム作成には多数の制限が存在する．まずアプリケーションがウェブ要求の処理のために呼び出された場合，30 秒以内に応答を発行する必要があるという制限である．これは Slim3 を利用することで高速な検索を行うことで解決を計った．次に，不等式フィルタが使用できるのが 1 つのプロパティに限られるという制限や他の並び替えより先に，不等式フィルタのプロパティの並び替えが必要であるという制限である．これも同様に Slim3 の機能を利用することで一度抽出したデータをメモリ上で，ソートし，フィルタをかけることで解決した．

Blobstore を直接更新が出来ない問題に関しても 前述のとおり，一度 Blob データを削除して再度アップロードし，その Blob キーを Datastore 上のプロパティを上書きすることで更新可能にする予定である．

(2) Google Calendar の API による制限

当初は検索結果を iCalendar 形式にし，プログラム上から Google Calendar に読み込ませる設計を試みたが，Google Calendar では iCalendar を読み込ませる API が存在していない. 今回は iCalendar 形式の URL を表示するところまでを実装し, 表示した URL は各個人のスケジュールと重ね合わせて利用することが可能になった．今後はプログラム本体から他の様々なスケジューラに自動で取り込める機能の追加を検討中である．

(3) 検索方式の課題

多様な検索方法に対応する為メタデータを提案した．現時点ではタイトルと日付における検索で動作検証を行った．しかしマイニングコントローラは本来，メタデータを用いて様々な知識を抽出する機能であり，今後はさらに検索ではなくマイニング機能として充実させていく必要がある．

(4) ファイルの保存場所

現状のシステムでは GAE 上の Blobstore のみに保存ができるように実装した．しかし，企業内で扱うファイルなどは機密情報が含まれていることが少なくなく，セキュリティの観点からクラウドサーバーではなく別のサーバーへの保存も可能にしていく予定である．その際に，利用者が保存サーバー選べるようにする機能と，セキュリティレベルに応じて自動でファイルの保存場所を振り分ける機能の実装を現在検討中である．

Vol.2012-GN-82 No.21
Vol.2012-CDS-3 No.21
2012/1/20

(5) 操作性

Webアクセス機能を持ったアプリケーションを作成するにあたってWebブラウザの枠にとらわれないアプリケーション開発のできる，Adobe AIR による実装を検討した．今回はインターフェースの確認を行った．この Adobe AIR を用いることで Web ブラウザからのアクセスとは違う，全く別の独立したアプリケーションとして使うことができる．具体的には保存ビュー及び検索・結果ビューといった端末側での表示はすべて Adobe AIR で実装を行う予定である．

しかし，現状では Adobe AIR の実装は未完成であるが，今後はシステムの操作性の向上を目指し，今回作成したシステムにおいて Adobe AIR による実装を検討している．

5. まとめ

本研究では，メタデータとクラウドサーバーを利用したファイルの管理とカレンダー表現を用いることにより，３W 情報を扱える情報共有システムの実装を行った．

このシステムを利用することで，過去の作業内容を視覚的に認識して必要なファイルの選択と抽出を可能とした．

現状では検索機能と操作性が不十分であるため，これらを改善していくことが今後の課題である．

参考文献

［1］ 斉藤,金井:“スケジューラを用いた知識の蓄積・継承の提案”，情報処理学会 GN ワークショップ 2011，論文集 P.1-8，2011

［2］ Slim3, https://sites.google.com/site/slim3appengine/

［3］ Internet Calendaring and Scheduling Core Object Specification (iCalendar), RFC2445, 1998.Nov., http://www.ietf.org/rfc/rfc2445.txt