

メール通知とネットワーク共有機能を付加した HyperSchedule について

福永 哲也* 鱈目 正志† 今城 一夫†

On the HyperSchedule with a Notification using E-Mail and a Commonage on Network

Tetsuya FUKUNAGA* Masashi SAWARAME† Kazuo IMAJO†

Synopsis

The HyperSchedule is a very useful and simple group schedule application and a free software which we can modify. We consider that a notification using e-mail and a commonage on network are necessary on a group schedule system. Accordingly, we add these functions to the HyperSchedule. So it is shown that we can use the modified HyperSchedule as a group schedule system or a billboard.

1 まえがき

現在、グループウェアを導入する機関が多くなってきた。グループウェアは、スケジュール管理、ToDo、メッセージ通知、掲示板、回覧版、ファイル共有、施設予約、在室・不在、メール等の多種多様な機能を有している。しかし、それらの機能を有効に利用できているか疑問の残るところである。

我々は、インストールが簡単で、操作が非常にシンプルで、フリーなスケジュール管理ソフトである HyperSchedule を 1997 年から利用してきた。この間、2000 年対応や祝日、国民の休日、ハッピーマンデーへの対応等を行った。そして、スケジュールのメール通知機能とネットワークを介したスケジュールの共有機能を実現できたので、ここに報告する。更に、HyperSchedule の利用例を紹介する。

2 HyperSchedule の概要

HyperSchedule¹⁾ は、Software Design 1997 年 9 月号で紹介された Windows あるいは UNIX で動作する WWW を使ったグループスケジュール管理ソフトである。当時はグループウェアが出始めた時代

であったこともあり、多くの場所で利用された。そして、HyperSchedule は Perl²⁾ で記述されたフリーソフトウェアで、利用、複製、変更、再配布に関してなら制限を行っていないこともあり、各所で変更がなされながら現在も利用されている。

ここでは、HyperSchedule の概要を文献 1) に従って述べる。

まず、WWW でスケジュール管理する利点としては、以下の項目が挙げられる。

- スケジュールを多くの人と共有できる
- どこからでもスケジュールをリアルタイムで確認、変更ができる
- 色を変えたり、資料にリンクを設定できる
- 記録がファイルとして残る

そして、HyperSchedule の特徴を以下に示す。

非常にシンプル 操作は非常にシンプルで、誰でもすぐに使うことができる。基本操作は、見る、書く、消すの 3 つである。スケジュール中に URL が書いてあれば自動的にリンクを張って表示し、HTML を使わなくても色を変えたりすることができる機能がある。

認証機能も装備 ユーザ名およびパスワードによる

*電子工学科

†情報工学科

認証と、IP アドレスによる制御を行うことができる。パスワードは WWW ブラウザから変更可能で、投稿する権利と削除する権利を別々に設定できる。そして、投稿した端末からはそのスケジュールを自由に消すことができる設定も可能である。

過去 / 未来のスケジュール確認が簡単 年と月を入れればそのカレンダーを表示する。会議やゼミの内容を少し書いておけば議事録としても使用でき、会議毎に資料をリンクさせておけば過去の資料を閲覧できる。

改良が簡単 Perl で作られているため、各グループに都合よく改良できる。表示レイアウトの変更や欲しい機能の追加を自由に行え、Windows でも UNIX でも動作させることができる。

以上のように、HyperSchedule は本来の目的であるグループスケジュールの登録・確認だけでなく、相互にコミュニケーションをとる掲示板としても利用できる非常に有用なイントラネット・アプリケーションである。

HyperSchedule のカレンダー表示を図 1 に例示する。当日をはさんだ約 2 ヶ月間が、対応する日付の欄にサブジェクトとともに時間順で表示される。そのスケジュールをクリックすれば、スケジュール詳細が図 2 のように表示される。

スケジュール登録における入力項目は、年月日と時間、サブジェクト、そして詳細である(後述の図 3 を

参照されたい)。日には、複数の日をまたいで入力することが可能で、例えば、その月の 3 日、4 日、5 日であれば”3,4,5”あるいは”3-5”と、その月の 28 日から翌月の 2 日まで(年はまたげない)であれば”28-2”と入力することができる。時間も開始時刻と終了時刻を入力できる。こうして投稿されたスケジュールはカレンダーのすべての日付欄に時間とサブジェクトが表示されることになる。また、以前に投稿したスケジュールを引用して同じようなスケジュールを登録する Copy 機能や、以前に投稿したスケジュールにコメントを付加して登録する Quote 機能も有している。そして、詳細は改行や空白が認識される Text 形式か、タグを使った HTML 形式で記述できる。

スケジュールを見ることができる権限は、WWW サーバのアクセス制御を利用する。このことにより、各サーバに備わっている高度なセキュリティ機能が利用できる。

スケジュールを書き込むことができる権限は、IP アドレスによる制限か、もしくはユーザ名およびパスワードによる制限で行うことができる。

スケジュールを消去可能にする権限は、書き込み権限と同様な形で設定できるが、書き込み権限とは独立して設定することができる。また、消そうとするスケジュールが書き込まれたときに使われた端末だけから、消すことができるような設定も可能である。この機能は特に 1 人 1 台のコンピュータを使える環境では、非常に有効な機能となる。



図 1 カレンダーの表示例

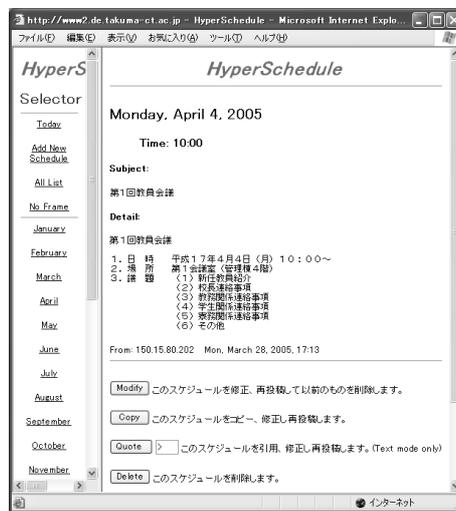


図 2 詳細の表示例

3 メール通知機能の付加

スケジュール管理においてスケジュールの登録通知や変更通知,そして,指定時刻での予告通知は必要であると考えられる。そこで,HyperSchedule にメール通知機能を加えた。ここで,スケジュールと連動したメール通知は以下の4つを行うことにした。

- スケジュール登録時のメール通知
- スケジュール変更時のメール通知
- スケジュール削除時のメール通知
- 指定時刻でのスケジュールメール通知

スケジュールデータは,サブジェクト項目(日時を含む)と詳細項目を”\0”を区切りとして,サブジェクト項目内ではバージョン番号(HPSC_VER),データベース・キー(key),年(year),月(month),日(day),時間(time),サブジェクト(subject)を”\n”区切りとしていた。そして,詳細項目は詳細(detail)の1つだけである。このため,メール通知関係データのスケジュールデータへの追加は,メール通知項目を”\0”で区切り,その内部はメール通知先(mailto)と時刻指定(atmail)を”\n”で区切ることにした。

```
HPSC_VER \n key \n year \n month \n day \n
time \n subject \0
detail \0
mailto \n atmail \0
```

メール通知先は hpsc_user.pl で以下のように任意に設定できる。

```
#通知先メールアドレスのリスト(追加可能)
$MAILTO[0] = '電子教職員';
$MAILTO_ADD[0] = 'all@xx.takuma-ct.ac.jp';

$MAILTO[1] = '教員全員';
$MAILTO_ADD[1] = 'staff@xx.takuma-ct.ac.jp';

$MAILTO[2] = 'aaaa';
$MAILTO_ADD[2] = 'aaaa@xx.takuma-ct.ac.jp';

$MAILTO[3] = 'bbbb';
$MAILTO_ADD[3] = 'bbbb@xx.takuma-ct.ac.jp';
```

そして,時刻指定も同様に以下のように設定でき,日時分を”.”で区切り,正の値は日時分を,”0”はスケジュール開始の日時分を,負の値はスケジュール開始の日時分より前を表す。但し,”0:0:0”はスケジュール登録時に登録された旨のメールを送るのみとしており,スケジュールの開始日の0時0分でのメール通知,そして,スケジュール開始の日時分でのメール通知はできない。

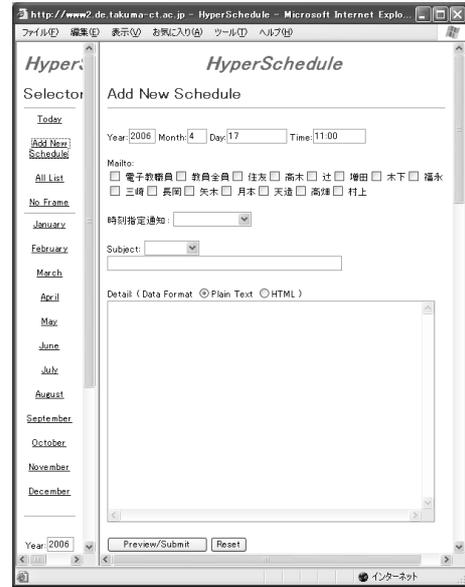


図 3 入力画面の例

```
#時刻指定のリスト(追加可能)
$ATMAIL[0] = '登録通知のみ';
$ATMAIL_TIME[0] = '0:0:0';

$ATMAIL[2] = '30 分前';
$ATMAIL_TIME[2] = '0:0:-30';

$ATMAIL[3] = '1 時間前';
$ATMAIL_TIME[3] = '0:-1:0';

$ATMAIL[5] = '当日の朝';
$ATMAIL_TIME[5] = '0:0:0';

$ATMAIL[7] = '前日の朝';
$ATMAIL_TIME[7] = '-1:7:0';
```

スケジュール入力画面を図3に示す。メール通知関係のメール通知先項目(Mailto)と時刻指定通知項目を追加している。メール通知先はチェックボックス,時刻指定通知はプルダウンメニューとなっている。

メール通知を行わねばならないスケジュールの検索を早くするため,メール通知用データはスケジュールとは別ディレクトリ(data-atmail)に保存している。また,ファイル名はスケジュール(data ディレクトリに保存されている)と同じ名前とし,データはmailto,atmailに加えて送信時刻(send.time)を”\n”区切りとして付加している。

hpsc_atmail.pl(付録1を参照)はdata-atmail内のファイルを調査し,送信すべきメールがあるなら送信し,ファイルを削除する。本プログラムをcronにより一定時間毎に実行すれば,指定時間がくれば通知メールを送ることができる。実用上は,1分刻み

でスケジュールを管理することはなく、5分程度で十分と考えられる。そこで、我々は5分毎に本プログラムを実行している。

一方、メールはsmtpで送ることもできるが、我々は直接sendmailにパイプすることでメール配送を実現している(付録1を参照)。

4 ネットワーク共有機能の付加

スケジュールは階層構造となっており、各学科のスケジュールには学校行事等の全体スケジュールが含まれ、各個人のスケジュールは所属学科、学校のスケジュールが含まれる。各個人が学校行事や所属学科のスケジュールを登録するのは効率が悪く、共有化することは不可欠と思われる。

一方、ある人はいろいろなグループに所属する。例えば、電子工学科に所属、入学試験委員会に所属するとき、自分だけのスケジュールを登録しているHyperSchedule(グループスケジュールとして利用していない)、電子工学科のスケジュール用HyperSchedule、入学試験委員会のスケジュール用HyperScheduleの3つが用意されていれば、自分だけのスケジュールに、自分が所属するグループのスケジュールを重ねて表示できれば非常に便利になる。

そこで、スケジュールのネットワーク共有を実現する。プロトコルとしてhttpを用いたソケット通信で実現することにした。共有するスケジュール先はhpsc_user.plで以下のように任意に設定できる。

```
#他のスケジュールサーバのリスト(追加可能)
$SERVERS[0] = '電子';
$SERVERS_URL[0] =
'http://www2.de.takuma-ct.ac.jp/cgi-bin/hpsc/';
```

図4のServersチェックボックスでサーバ選択を行った後、Submitをクリックすると、図5のように電子工学科スケジュールは先頭に”電子:”(上記の\$SERVERS)が付加されて表示される。これを実現するため、スケジュール表示プログラムhpsc_refreshcalendar.plに他サーバからスケジュールを取得する部分を、更に、他サーバからの取得要求を受けて指定された日付間のスケジュールを返すプログラムhpsc_inform.plを作成した(付録2参照)。ここで、返すデータ構造は

```
html ヘッダ \0 \0 スケジュールデータ \0 \0 html フッタ
```

のように、スケジュールデータが”\0”区切りである



図4 入学試験委員会の例



図5 電子工学科スケジュールを共有した例

ことから、各項目を”\0 \0”区切りとしている。

5 HyperSchedule の利用例

最も一般的なグループスケジュールとしての利用例は、図1の電子工学科スケジュールである。

また、情報交換を目的とした利用としては図6に示すネットワーク関係のお知らせが挙げられる。ネットワーク停止のお知らせをスケジュールに登録すると同時に、教職員全員にメール通知することもできる。また、外部からある部署へ不審なアクセスがあった場合には、スケジュールとして掲示するとともにその部署の責任者にメール通知することもできる。そして、ネットワーク管理者はスケジュール登録可能としており、ここを利用した情報交換が可能であ

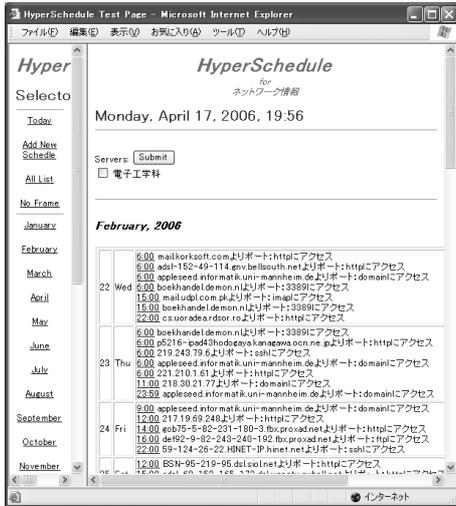


図 6 ネットワーク関係のお知らせ



図 7 大学編入学スケジュールの提供

る。更には、その内容を広く知らせることもできる。

一方、メール通知機能は利用しないが、大学編入学日程の情報提供例が図 7 である。各大学の出願期間、受験日、合格発表日、誓約書提出期限等の情報が取得できる。

6 むすび

スケジュール管理ソフトにはメール通知機能とネットワーク共有機能は不可欠と考え、フリーで変更も自由な HyperSchedule にこれらの機能を付加した。そして、利用した結果、グループスケジュールとして十分に利用できること、情報交換用の掲示板としても利用できること等が明らかとなった。

今後は、自分だけしか見えない非公開スケジュー

ルを実現して、個人スケジュールとしての利用についても検討したい。

参考文献

- 1) 中屋秀彦：HyperSchedule はいかが？ WWW を使ったグループスケジュール管理，Software Design 1997 年 9 月号，pp.136-145 (1997).
- 2) 三島俊司：CGI のための実践入門 Perl，技術評論社 (1998).

付録 1：メールの送信について

以下に、メール通知を実行する hpsc_atmail.pl を示す。

```
#!/usr/bin/perl
#-----#
# HyperSchedule for workgroup schedule management
# Copyright (c) 1996 by Hidehiko Nakaya
# Tetsuya Fukunaga
# hpsc_atmail.pl
#-----#
require 'hpsc_rc.pl';
require 'jcode.pl';
require 'hpsc_lib.pl';
while ($atmail_file = <$DIR_ATMAIL*>)
{
    open(F_ATMAIL, "$atmail_file");
    @data = <F_ATMAIL>;
    close(F_ATMAIL);
    $data = join(' ', @data);
    ( $subject, $detail, $mailto) = split(/\0,$data);
    ( $hpsc_ver,$key,$year,$month,$day,$time,$subject) =
        split(/\n/, $subject, 7);
    ( $mailto, $atmail, $send_time) = split(/\n/, $mailto, 3);
    $time_now = time;
    if($send_time <= $time_now)
    {
        $data = "$year\n$month\n$day\n$time\n$subject\0
            $detail\0
            $mailto\n$atmail\0";
        &SendMail("atmail", $data) ||
            print STDERR "Can not send atmail\n";
        unlink("$atmail_file");
    }
}
}
```

更に、メール送信の主要部分のみを以下に示す。

```
sub SendMail
{
    package SendMail;
    open(F_MAIL, "| $main'SENDMAIL $mailto_add");
    &jcode'convert(*subject, 'jis');
    &jcode'convert(*detail, 'jis');
    &jcode'convert(*mailto, 'jis');
    &jcode'convert(*atmail, 'jis');
    &jcode'convert(*pre_message, 'jis');
    $date_now = "$main'WDAY_STR[$yday_now], $mday_now
        $main'MON_STR[$mon_now] $year_now ".
        sprintf("%02d:%02d:%02d", $hour_now,
            $min_now, $sec_now)." +0900";
    print F_MAIL "Date: $date_now\n";
    print F_MAIL "To: $mailto_add\n";
    print F_MAIL "Subject: $subject\n";
    print F_MAIL "Reply-To: $main'HPSCADMIN\n";
    print F_MAIL "From: $main'HPSCADMIN\n";
    print F_MAIL "X-Mailer: HyperSchedule Mailer Ver 0.1\n";
    print F_MAIL "Content-Type: ";
    print F_MAIL "text/plain; charset=iso-2022-jp\n";
    print F_MAIL "Content-Transfer-Encoding: 7bit\n";
    print F_MAIL "\n";
    print F_MAIL "=====\n";
}
```

```

print F_MAIL "$pre_message\n";
print F_MAIL "-----\n\n";
print F_MAIL "$year / $month / $day ";
print F_MAIL "( $main'WDAY_STR[$wday] ) $time\n";
if($atmail)
{
    $msg = '時刻指定通知';
    &jcode'convert(*msg, 'jis');
    print F_MAIL "$msg : $atmail\n";
}
print F_MAIL "\n$detail\n";
print F_MAIL "-----\n";
close(F_MAIL);
return (1);
}

```

付録 2 : ネットワーク共有について

以下に , スケジュール表示プログラム hpsc_refreshcalendar.pl のネットワーク共有を実現する部分のみと , ネットワークを介してスケジュールを取得する関数 GetData を示す。

```

#---- 他サーバ ----
if( $servers_url )
{
    local($data,$subject,@subject);
    local($hostname,$url,@fservers_url,$fservers_url,
        $fservers);
    @fservers_url = split('&', $servers_url);
    foreach $fservers_url ( @fservers_url )
    {
        $data = &GetData($fservers_url,$StartTime,$EndTime);
        for( $i = $!; $i <= $#SERVERS_URL; $i++)
        {
            if( $fservers_url eq $SERVERS_URL[$i] )
            {
                $fservers = $SERVERS[$i];
            }
        }
        @subject = split(/\0/, $data);
        foreach $subject( @subject )
        {
            local($year,$yday,$time,$key,$view_url,$fsubject) =
                split(/\n/, $subject, 6);
            $dbmkey = sprintf("%04d%03d", $year, $yday);
            $serversdbm{$dbmkey} .= "$time\n$key\n$view_url\n
                $fservers : $fsubject\n";
        }
    }
}

#ネットワークを介してスケジュールを取得
sub GetData
{
    package GetData;
    local($url,$stime,$setime) = @_;
    $SIG{ALRM} = 'onalarm';
    $hostname = ( split('/', $url) )[2];
    $url = ( split($hostname, $url) )[1];
    ($hostname, $port) = split(':', $hostname);
    $port = 80 unless $port;
    ($addr) = (gethostbyname($hostname))[4];
    ($addr1,$addr2,$addr3,$addr4) = unpack('C4', $addr);
    return if( $addr1 == 0 );
    $PF_INET = 2;
    $SOCK_STREAM = 1;
    socket(SOCK, $PF_INET, $SOCK_STREAM, 0);
    $sockaddr = pack('S n a4 x8', $PF_INET, $port, $addr);
    connect(SOCK, $sockaddr);
    $oldhandle = select(SOCK);
    $| = 1;
    select($oldhandle);
    print SOCK "GET $url$main'S_INFORM$stime,$setime HTTP/1.0
        \r\n\r\n";
    alarm(10);
    $return = join("", <SOCK>);
    $SIG{ALRM} = '';
}

```

```

$return = ( split(/\0/, $return) )[1];
$return;
}

sub onalarm
{
    package GetData;
    exit;
}

```

そして , 要求に応じて指定期間内のスケジュールを
応答するプログラム hpsc_inform.pl を以下に示す。

```

#!/usr/bin/perl
#-----#
# HyperSchedule for workgroup schedule management
# Copyright (c) 1996 by Hidehiko Nakaya
# Copyright (c) 2000 by Fukunaga, Sawarame and Imajo
#
# hpsc_inform.pl
#-----#
require 'hpsc_rc.pl';
require 'timelocal.pl';
require 'hpsc_lib.pl';

@QSTR = split(/,/,$ENV{'QUERY_STRING'});
print <<"_HTML";
<head><title>Inform Data</title></head>
<html>
<body>
_HTML

if ($#QSTR != 1)
{
    print "\0Argument error\n\0";
    print "</body></html>\n";
    exit(-1);
}

$StartTime = $QSTR[0];
$EndTime = $QSTR[1];
print "\0\0"; #Header and Data separator
$data = "";
dbmopen(%DBM_HPSC, "$F_DBM", 0644) ||
    &ErrorMessage("Can not open DBM file:$F_DBM\n");

for ( $i=$StartTime ; $i < $EndTime; $i+=86400 )
{
    local(@subject,$subject,$day,$mon,$year,$yday,$yday);
    ( $day,$mon,$year,$yday,$yday ) = (localtime($i))[3..7];
    $year = 1900 + $year;
    $dbmkey = sprintf("%04d%03d", $year, $yday);
    $dbmdata = $DBM_HPSC{$dbmkey};
    @dbmdata = split(/\0/, $dbmdata);
    foreach (@dbmdata)
    {
        local( $key,$time,$subject ) =
            ( &DevideDBMValue( $ ) )[1,5,6];
        $subject = join( "\n", $year,$yday,$time,$key,
            $S_VIEWSCH,$subject );
        push( @subject, $subject);
    }
    @subject = sort( @subject );
    foreach $subject ( @subject )
    {
        $data .= "$subject\n\0";
    }
}

chop($data);
print "$data";
print "\0\0"; #Data and Footer separator
dbmclose(%DBM_HPSC);
print <<"_HTML";
</body>
</html>
_HTML

```