

省電力のための運休スケジュール参照機能を持つ機器監視システムの試作

西垣 桂^{†1} 齊藤 明紀^{†1}

省エネルギーのためには夜間や週末などの不使用時間帯は機器の電源を切るという運用が有用である。しかし電源操作そのもの手間と、ネットワーク監視システムの誤警告を抑制するための設定変更の手間が問題になる。そこで、機器の稼働運休スケジュール定義データベースに従って遠隔電源操作を行う電源管理システムと、同じスケジュールデータを参照して働く死活監視システムを用いる方式を提案する。

提案するスケジュール記述方式は、複雑な運用スケジュールでも容易に定義できるように、対象機器の階層的グループ化や週間スケジュール定義機能を有している。本研究では MySQL を用いてシステムを試作し、提案方式がうまく動くことを確認した。

An alive monitoring system and a power control system sharing a power control schedule

KATSURA NISHIGAKI ^{†1} and AKINORI SAITOH ^{†1}

For electric power saving, it is effective to turn unused equipments off. For example, in night or weekend. But it needs much human resource to turn power switches on or off, and changing configuration file for alive monitoring system. We propose an alive monitoring system and a power control system cooperates by referring single database that stores equipments' operate time schedule definition.

For ease of operation, this system can use hierarchical grouping of target machines. It also enables repeated applying of pre-defined weekly schedule.

We also implemented a experimental system using MySQL and we proved that proposed approach is feasible.

1. はじめに

現在、地球温暖化などの環境問題が注目されており、それに伴って IT 機器の消費電力の節減への要請が強まっている。エネルギー効率のよい機器を導入することもそうであるが、不使用時間帯に電源を切ることができれば効果は高い。たとえば大学の講義棟の情報コンセント用ハブや無線 LAN 基地局などは講義の行われない夜間と週末に運休させることで消費電力を 1/2 以下に減らすことができる。

しかしながら、時間帯によって不使用機器の電源を切る手法にはいくつかの課題がある。まず、手で電源スイッチを操作するには人手がかかりすぎる。また、始業に電源投入を間に合わせるには電源操作要員の早朝出勤が必要になるなど人事管理上の問題も生じる。そこで実質的にはコンピュータシステムによる集中自動管理で遠隔電源操作を行う方式が必要となる。遠隔電源操作に対応していない機器ではリモート電源タップ等の追加機器が必要となるが、場合によっては追加機器のコストが節減できる電力料金を上回ってしまうという問題もある。これに関しては、ネットワーク機器自身が電源制御機能を持つ¹⁾方式や、また追加機器の低価格化や Power on Ether 対応機器の増加などで解決される傾向にある。

もうひとつの問題は機器監視システム²⁾⁻⁴⁾の管理の手間である。死活監視を行うシステムは機器の応答がなくなると警告を発する。設定ファイルを書き換えて計画運休のたびに一部の機器を監視対象からはずすのは手間がかかる。監視システムにアラートを出す/出さないのスケジュールをあらかじめ設定できる^{5),6)}ものもある。しかし、これらは比較的の低頻度の運休を想定したものであり、対象機器が多く稼働スケジュールが一樣ではない場合は入力操作にかなり手間がかかる。

そこで筆者は、機器の稼働スケジュールデータを電源管理機能と監視機能が共用するシステムを提案する。これにより、運休の誤警告を抑制するために監視システム管理に余分な手間がかかることがなくなるはずである。また提案システムでは複雑な運用スケジュールを容易に定義するために、対象機器グループに対してスケジュールを定義する機能や、週間スケジュール定義機能を持たせる。

2. 考察と方式設計

本研究では事前に定めた運転スケジュールに従って機器の電源オン/オフを行うシステムと、その運行予定定義を参照することで、計画停止に対してはアラートを出さない機器監視システムを考える(図1)。ここでいう計画停止は年に1回の受電設備点検や長期休暇の間の

^{†1} 鳥取環境大学環境情報学部情報システム学科
Tottori University of Environmental Studies

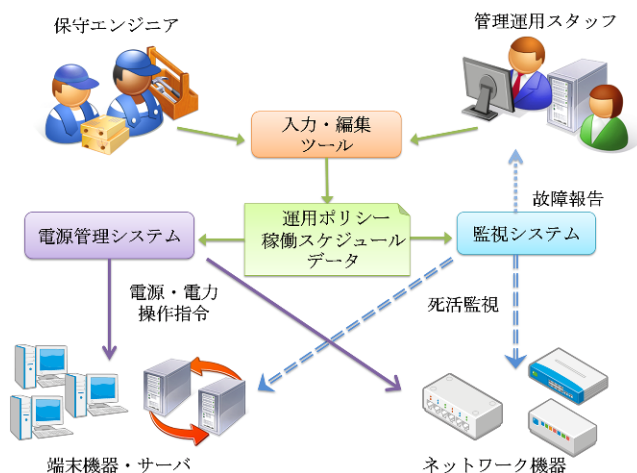


図1 システムの全体図

電源オフだけでなく、平日夜間には停止させるといった程度の頻度のものを考える。

通常の監視システムの死活監視は稼働を正常状態、停止を異常ととらえ、機器の停止（無応答）を見つけてアラートを出す。提案システムでは電源オフ指令に従わない事象も異常とみなすべきである。そこで、計画停止の時間帯に動き続けている場合もアラートを出すことにする。

ここで重要なのは、複雑な運用スケジュールを少ない手間ですべて記述できる方式である。そこで次に述べるような階層的グループ化と優先度を利用した手法を提案する。

2.1 スケジュールの定義手法

同じスケジュールで運転する機器群に対する設定を一度で済ませるために、機器のグループに対してスケジュールを設定する機能はよくみられる。より柔軟な設定を可能にするために、対象機器グループを階層的に定義できるようにする。また、スケジュール設定に優先度を設け、複数のスケジュール設定を合成して最終的な個々の機器のオン/オフ時刻を決定する。

グループの階層構造は図2のような木構造としてとらえることができる。グループ(木の頂点)の階層の深さは根をレベル0とする整数値で表す。図2での研究棟はレベル1、左下端の機器はレベル4となる。同じ時刻に異なるスケジュール設定がある場合は、より階層の

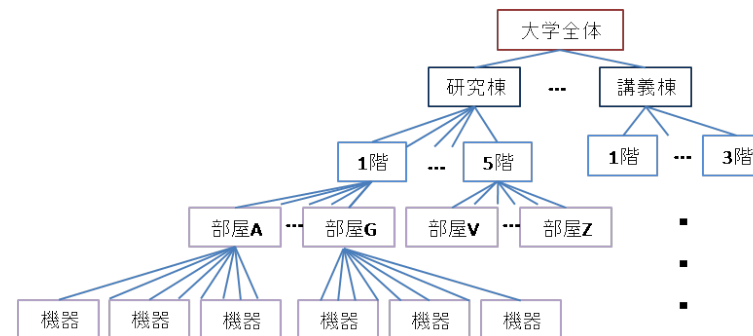


図2 グループ階層の概念図

低い(個別的な)グループに対する設定を優先する。

電源スケジュール設定は、オン時刻/オフ時刻を設定する方式と、オン時間帯/オフ時間帯を設定する方式が考えられる。ここでは、スケジュール合成処理がしやすいように後者の方式をとる。以降、スケジュール定義を(設定対象, 優先度, 起点日時, 終点日時, オンオフ)の5つ組の集合で表現することにする。設定対象にはホスト名やグループ名が入る。

機器の稼働スケジュールは曜日毎に同じスケジュールを繰り返すことが多い。そこで、別に名前をつけて定義した週間スケジュールを呼び出して使う機能を設ける。週間スケジュールの呼び出しは、日単位の期間設定とする。対象グループと優先度も合わせて設定するので週間スケジュールの呼び出しは、(設定対象, 優先度, 起点日付, 終点日付, 週間スケジュール名)の5つ組の集合で表現することにする。

2.2 週間スケジュール機能

週間スケジュールは(曜日, 開始時刻, 終了時刻, オンオフ)の4つ組の集合として定義する。例を図3に示す。これは、「講義期間中講義室」が「平日 8:30 から 18:00 まで稼働」、「休業期間中講義室」が「毎日オフ」を意図している。この週間スケジュールは電源オフというデフォルトのスケジュールが存在することを前提としているため、電源オン区間だけを定義している。

たとえば、(全講義室, 2, 2010年1月7日, 2010年1月24日, 講義期間中講義室)という1つの週間スケジュール参照設定で17日分の動作が定義される。これは図4に示す12項からなるスケジュール設定と同等である。

```

講義期間中講義室={
  (月, 8:30:00, 17:59:59, ON),      (火, 8:30:00, 17:59:59, ON),
  (水, 8:30:00, 17:59:59, ON),      (木, 8:30:00, 17:59:59, ON),
  (金, 8:30:00, 17:59:59, ON),
}

```

```

休暇期間中講義室={ } /* 空集合 */

```

図3 週間スケジュールの例

```

(全講義室, 2, 2010年1月7日 8:30, 2010年1月7日 18:00, ON)
(全講義室, 2, 2010年1月8日 8:30, 2010年1月7日 18:00, ON)
(全講義室, 2, 2010年1月11日 8:30, 2010年1月11日 18:00, ON)
(全講義室, 2, 2010年1月12日 8:30, 2010年1月12日 18:00, ON)
... 6項目省略 ...
(全講義室, 2, 2010年1月21日 8:30, 2010年1月21日 18:00, ON)
(全講義室, 2, 2010年1月22日 8:30, 2010年1月22日 18:00, ON)

```

図4 週間スケジュールを用いない場合の例

2.3 優先度値の利用

週間スケジュールにしたがって運用する期間の中に例外的な日が入ることがある。そのときに、週間スケジュールとの差分だけを記述すれば済むように、優先度値を導入する。個々のスケジュール設定が持つ優先度値は非不正数とし、0が最高優先度とする。

たとえば優先度2の「8時から18時までオン」という設定と、優先度1の「12時から14時までオフ」という設定とが同時に存在する場合は、「8時から12時までオン、12時から14時までオフ、14時から18時までオン」として解釈する。

おなじ優先度のスケジュール設定が複数あった場合は、より階層の深い(個別化した)グループに対するものが優先する。

2.4 優先度処理の説明

提案方式の優先度処理を具体例で説明する。

次の例は以下のような状況を記述している。:

A棟にはいくつかの講義室がある。2010年1月7日が新年最初の講義日である。1月の講義期間は1月20日までとする。講義室のネットワーク機器は講義期間中は平日の8:30~

18:00に稼働させる。1月16~17日はセンター試験のため、前日15日の午後は準備のため講義を行わないので教室のネットワーク機器の電源は切る。ただし試験本部として使用する第1教室はその対象外とし、15日午後と16~17日の7時~19時の間も機器を稼働させる。機器のグループ定義は、A棟全体/全講義室/第1教室/個々の機器という4階層で行われているものとする。

このような状況での1月20日までのスケジュール設定には、図3の週間スケジュール設定に加えて、以下のような設定を行えばよい。

まずデフォルトの電源オフ設定を行うための定義を置く。

```

(A棟全体, 20, 2000/1/1 0:00:00, 2038/3/31 23:59:59, OFF)

```

グループ「A棟講義室」に対しては週間スケジュールを優先度20で用いる。

```

(全講義室, 20, 2010/1/7, 2010/1/20, 講義期間中講義室)

```

以上の設定で、平日8:30~18:00は稼働しそれ以外は運休するという設定になる。

次にA棟の全講義室に対し、15(金)昼以降の機器停止を優先度10で定義する。

```

(全講義室, 10, 2010/1/15 13:00, 2010/1/18 8:00:00, OFF)

```

そして、A棟の第1教室の機器だけは稼働させる設定を行う。

```

(第1教室, 10, 2010/1/15 13:00, 2010/1/15 18:59:59, ON)

```

```

(第1教室, 10, 2010/1/16 7:00, 2010/1/16 18:59:59, ON)

```

```

(第1教室, 10, 2010/1/17 7:00, 2010/1/17 18:59:59, ON)

```

これらのスケジュールが設定された状況の1/13~1/19部分を図で表現したものが図5である。この図で横軸が時間(右が未来)である。一つのスケジュール設定は一つの線分で表される。オン区間は赤の線分、オフ区間は黒の線分で表す。また、週間スケジュール(「講義期間中講義室」)の参照を緑の線分で表している。

図5の週間スケジュールの参照を展開し、スケジュール定義を合成した結果を図6に示す。機器個別の設定はないため、第1教室の各機器は同じスケジュールが適用される。またそれ以外の教室の機器は教室レベルの設定がないため「全教室」に対する設定に従う。

3. 試作と実装

本節では試作したシステムとその実装について述べる。まずスケジュールデータと機器管理データの表現方式の実装について述べ、続いて個々のシステムについて述べる。

本システムのスケジュールデータは、機器監視システムと電源管理システムで共有する。

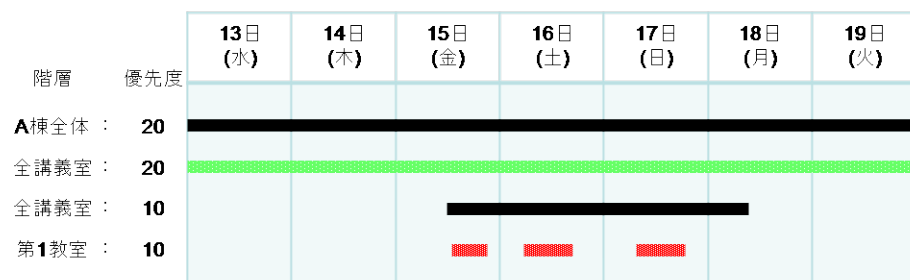


図5 スケジュール入力例

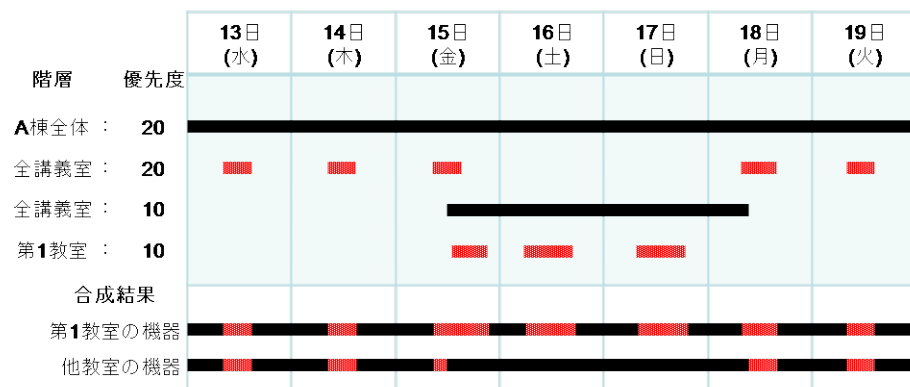


図6 スケジュール展開・合成例

そこでリレーショナルデータベースのテーブルとして表現し、保管することにした。

3.1 スケジュール処理の基本方針

提案システムでは、週間スケジュールをどのように実装するかが大きな課題である。われわれは、人間が入力するデータと、システムが参照するスケジュール解釈用データを別々に保持するというアイデアで実装した。スケジュール解釈用データは入力データから自動生成する。

スケジュールを保管するテーブルには、管理者が入力した週間スケジュールの参照をそのまま保管する。すなわち、入力インターフェース上は指定した日時区間に対して、オンまた

はオフまたは週間スケジュール名が設定できるようにみせる。

そして、週間スケジュールを具体的な日時に変換した結果を電源管理システムや機器監視システムに参照させる。

図6ではスケジュール全体で週間スケジュールを展開したものを例示したが、効率を考慮して、変換は昨日～明日の3日分だけを行うことにする。そして日付が変わるごとにインクリメンタルに変換を行う。また実行済みのスケジュールデータは逐次破棄する。

3.2 データベース設計

3.2.1 機器管理データの設計

運用・監視対象になる機器の情報もデータベースで管理する。対象にアクセスするために必要なデータとしてホスト名、IPアドレス、MACアドレスが挙げられる。MACアドレスは電源管理システムでWakeOnLAN (WOL)⁷⁾を使用する際に必要となる。電源操作の手法は対象機器ごとに違う。そこで、起動手法、シャットダウン手法それぞれの情報も必要である。

IPアドレスとMACアドレスは何らかのディレクトリサービスにすでに登録済みであることが多いと考えられる。今回の実装では、動作実験環境の都合に合わせて、IPアドレスはDNSを参照することに保持していない。

3.2.2 機器グループデータ構造

グループ定義は親子関係を示す(親グループ名、要素グループ名)ペアの集合として記述する。個々の対象機器が直接属するグループ情報は(グループ名、ホスト名)ペアとして記述する。

3.2.3 スケジュールデータの設計

2節での考察から、週間スケジュールを定義するテーブルの主要な項目は、次の通り：

- 週間スケジュール ID, 週間スケジュール名, 曜日,
- 開始時刻, 終了時刻, 動作 (オンなら 1 オフなら 0)

週間スケジュールは連番の ID で管理し、週間スケジュール名は GUI の表示用に利用する。スケジュールのデータ項目は以下のように定めた。

- 操作対象, 優先度, 開始日時, 終了日時, 動作 (オンなら 1 オフなら 0),
- 週間スケジュール名, 開始日, 終了日

このレコードが、具体スケジュールを表すのか週間スケジュールの参照を表すのかの区別は、週間スケジュール名が NULL かどうかで行う。

試作システムのデータベースで用いたテーブルの ER 図を図7に示す。管理対象台帳が

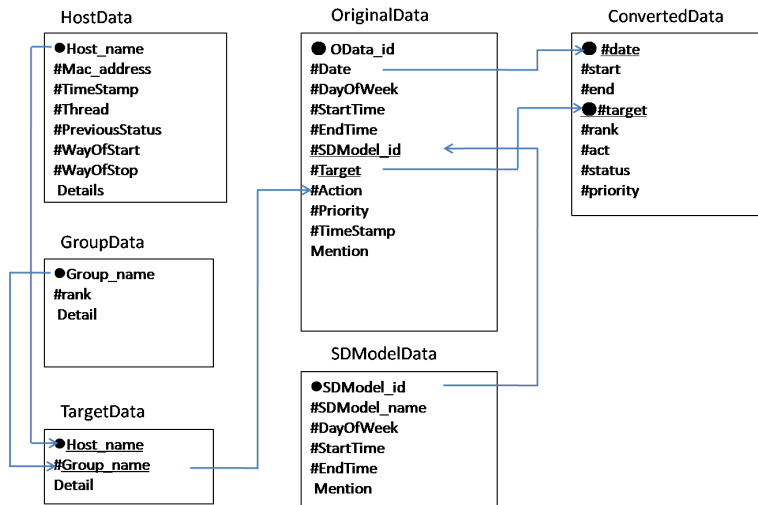


図7 ER図

HostData, グループ定義が GroupData と TargetData, 入力・編集に用いるスケジュール定義が OriginalData, 週間スケジュール定義が SDModelData, OriginalData と SDModelData から抽出/生成された前後3日分のスケジュールデータが ConvertedData である。

3.3 スケジュール処理のアルゴリズム

スケジュール定義データを評価して、各対象機器の「現時点で運転/運休どちらであるべきか」を算出する機能は、試作した電源管理システムと機器監視システムが共通して用いる機能である。

特定の日付 D について週間スケジュールの参照を具体スケジュールに展開するには、週間スケジュールのうち当該の曜日のレコードそれぞれについて Convertdata のレコードを生成すればよい。そのとき、開始日時は D+開始時刻、終了日時は D+終了時刻で算出できる。

展開後 (Convertdata) の個々のスケジュールレコードは適用期間の開始日時と終了日時を持つ。ある日時 DT での機器が稼働か運休かを決めるのは、開始日時 ≤ DT ≤ 終了日時であるレコードである。

そこで、Convertdata のレコードを開始日時順に走査することで逐次「現在の各機器のあるべき稼働状態」を計算する。まず、終了日時 < 現在日時 である Convertdata のレコードは読み飛ばす。そして、開始日時 ≤ 現在日時 ≤ 終了日時 であるレコードをすべて読み出してメモリ上で優先度処理を行い、個々の対象機器のあるべき状態を算出して初期状態とする。

初期化後は Convertdata の次のレコードの開始日時まで待機する。そして、新たに開始日時 現在日時 となったスケジュールレコードの指示を適用する。同じ開始日時のレコードは複数ありえることに注意が必要である。

稼働中にスケジュールデータが変更された場合のアルゴリズムは未策定である。今回の試作ではシステムを再起動して対応している。

3.4 スケジュールデータ入力・編集用 GUI

現状では入力・編集 GUI の実装途中である。システムの動作確認は SQL コマンドによってテーブルを直接入力することで行った。

マルチユーザシステムとして提案システムを用いるためには、スケジュールデータ入力・編集用 GUI がユーザごとの使用可能優先度値や編集可能な対象グループなどの権限管理も行う必要がある。優先度は、スケジュールを入力したユーザの権限に合わせて自動的に入力されるようにすべきである。優先度の高さは例えば、システム管理者 > メンテナンス業者 > 事務職員・教員などの順番で高くすることが考えられる。

3.5 機器監視システムの実装

現在の機器の状態を監視し、その状態をスケジュールデータと照合することで故障停止と運休とを見分けて管理者に警告するシステムである。試作システムでは SNMP host resource MIB⁸⁾ での死活監視のみを行う。

機器管理システムは SNMP リクエストを送信して現在の機器の状態を取得するプログラムと、取得した現在の状態とスケジュールを比較してエラーメッセージを送信するプログラムから構成した。実装は Java で行なった。

3.5.1 SNMP 問い合わせプログラム

SNMP のエージェント機能・マネージャ機能を提供する既存のライブラリ⁹⁾ を使用した。SNMP 問い合わせプログラムは複数のスレッドで分担して処理を行う。各対象機器に SNMP 問い合わせを行い、結果を MySQL のテーブルに書き込む。60 秒以内に応答がなければ稼働していないものとみなす。

3.5.2 スケジュール照合プログラム

SNMP 参照プログラムが更新する MySQL のホスト状態テーブルを参照し、スケジュール通りに機器が動作しているか照合するプログラムである。このプログラムは3分毎に各機器がスケジュール通りに動いている場合の状態を計算し、機器状態テーブルと照合する。

ただし、機器の種類によっては起動やシャットダウンに数分を要するものもある。そこで、対象機器ごとに起動とシャットダウンの所要時間を登録するようにしている。異常がその所要時間を越えて継続して初めて本プログラムはアラートを発行する。

3.6 電源管理システム

電源管理システムは、各機器の「現在あるべき状態」を逐次計算し、オン オフ、オフ オンの変化があったら起動やシャットダウンの操作を行う。しかし、機器によって電源操作方法が異なるため、実際の操作は外部プログラムとした。予め hostdata にその機器に使用する起動方法とシャットダウン方法を登録しておく。登録する文字列は電源操作を行なう外部プログラムの引数にする。例えばもし「WakeOnLAN」という文字列を起動方法に登録した場合、電源管理プログラムが「WakeOnLAN」を引数に起動する。

なお、部屋全体や建物全体を週末や長期休暇の際にまとめて電源を切るというような場合は遠隔電源操作機器は導入せず主幹ブレーカー1箇所を手動操作する手間が許容できる場合もあると考えられる。そのような場合、本システムではシャットダウン/起動手段のプログラムとして管理者に電源操作作業を行う時刻が来たことを知らせるプログラムを（GUI やショートメッセージなど）登録しておけばよい。

3.7 動作実験

以上で述べた2つのシステムの動作確認実験を独立した実験環境で行った。

まず、SNMP 問い合わせプログラムについて検証を行ない、MySQL の機器状態テーブルが逐次更新されることを確認した。次に、スケジュール照合プログラム部分について動作確認を行ない、稼働時間帯の無応答と連休時間帯の稼働を検出することを確認した。

管理者に異常を知らせる部分は、hostdata テーブルで指定された通知方法の文字列に、起動する通知プログラムの名前を登録することで、指定した通知プログラムが動作するようになっている。今回は、コマンドプロンプトに異常のある機器のホスト名と異常があったと表示だけのプログラムを登録して検証を行なった。

次に、電源管理プログラムの動作確認を行った。検証では実機としては WOL で起動する PC だけを用いた。WOL 以外の手法で起動する対象機器に関しては、電源操作プログラムの代わりにコマンドプロンプトに電源操作が行われたことを表示するプログラムを登録す

ることで動作確認を行った。

4. ま と め

本研究では、電源管理システムのための稼働スケジュールデータを死活管理システムも参照するというアイデアにもとづいたシステムを提案し、試作した。これにより、省エネルギーのため不使用時間帯にこまめに機器の電源を切るという運用をした場合に死活監視システムの誤アラートを抑止するための余分な管理作業が発生しない。提案手法により、稼働スケジュールの入力の手間が半分になると考えられる。

提案システムは階層的な対象機器グループ定義やスケジュール指定の優先度処理、また、週間スケジュール定義機能などにより、少ない分量の記述で複雑な稼働スケジュールを記述できるように工夫した。

参 考 文 献

- 1) アラクサラネットワークス株式会社: AX シリーズダイナミック省電力機能活用ガイド, [http://74.125.153.132/search?q=cache:AmSa-GfadKEJ:www.alaxala.com/jp/techinfo/archive/guide/pdf/N09R008_SavingEne_Guide_V1R0.pdf+N09R008_SavingEne_Guide_V1R0.pdf, pp.17-24\(2004\)](http://74.125.153.132/search?q=cache:AmSa-GfadKEJ:www.alaxala.com/jp/techinfo/archive/guide/pdf/N09R008_SavingEne_Guide_V1R0.pdf+N09R008_SavingEne_Guide_V1R0.pdf,pp.17-24(2004)
- 2) 日本ヒューレット・パッカー: HP Network Node Manager i-series software データシート, [https://h10078.www1.hp.com/cda/hpdc/navigation.do?action=downloadPDF&caid=41589&cp=54.4000_306&zn=bto&filename=4AA2-5668JAP.pdf, pp.1-4\(2008\)](https://h10078.www1.hp.com/cda/hpdc/navigation.do?action=downloadPDF&caid=41589&cp=54.4000_306&zn=bto&filename=4AA2-5668JAP.pdf,pp.1-4(2008)
- 3) ZABBIX: Zabbix 1.6 Manual (old release), [http://www.zabbix.com/downloads/ZABBIX%20Manual%20v1.6.pdf\(2009\)](http://www.zabbix.com/downloads/ZABBIX%20Manual%20v1.6.pdf(2009)
- 4) 株式会社 NTT データ: Hinemos オンラインカタログ, [http://www.hinemos.info/Hinemos%1%CA0SC2007Kansai%1%CB.pdf\(2007\)](http://www.hinemos.info/Hinemos%1%CA0SC2007Kansai%1%CB.pdf(2007)
- 5) FXC 株式会社: APS-1000 操作マニュアル (2008)
- 6) NEC: WebSAM System Navigator, <http://www.nec.co.jp/middle/WebSAM/products/sysnavi/kinou.html>
- 7) Advanced Micro Devices: Magic Packet Technology, [http://www.amd.com/us-en/assets/content.type/white_papers_and_tech_docs/20213.pdf\(1995\)](http://www.amd.com/us-en/assets/content.type/white_papers_and_tech_docs/20213.pdf(1995)
- 8) P. Grillo and S. Waldbusser: Host Resources MIB, RFC1514(1993)
- 9) SNMP4J:SNMP4J - Free Open Source SNMP API for Java, [http://www.snmp4j.org/index.html\(2003\)](http://www.snmp4j.org/index.html(2003)