

JAERI-Data/Code
99-023



JP9950388



原子力発電プラントデータベースインターネット版

1999年4月

山本信夫

日本原子力研究所
Japan Atomic Energy Research Institute

本レポートは、日本原子力研究所が不定期に公開している研究報告書です。
入手の問合わせは、日本原子力研究所研究情報部研究情報課（〒319-1195 茨城県那珂郡東海村）あて、お申し越し下さい。なお、このほかに財団法人原子力弘済会資料センター（〒319-1195 茨城県那珂郡東海村日本原子力研究所内）で複写による実費領布を行っております。

This report is issued irregularly.
Inquiries about availability of the reports should be addressed to Research Information Division, Department of Intellectual Resources, Japan Atomic Energy Research Institute, Tokai-mura, Naka-gun, Ibaraki-ken 319-1195, Japan.

原子力発電プラントデータベースインターネット版

日本原子力研究所東海研究所安全性試験研究センター原子炉安全工学部
山本 信夫

(1999年3月16日受理)

原子力発電プラントデータベースは、原子炉設置許可申請書を情報源として、国内の軽水型原子力発電所 50 基分の設計情報をデータベース化したものであり、平成7年度に日本原子力研究所大型計算機版を公開した。今回構築したインターネット版は、大型計算機版の格納情報のうち、「原子炉設置許可申請書・添付書類八 原子炉施設の安全設計に関する説明」に関する部分について、インターネットを通じて情報を提供できるようにしたものであり、設備や機器の形状、寸法、性能などの64,000点の情報を格納している。

本報告書は、本インターネット版のデータベース構造、プログラム仕様、検索機能についてまとめたものである。

Nuclear Power Plant Database Internet Version

Nobuo YAMAMOTO

Department of Reactor Safety Research
Nuclear Safety Research Center
Tokai Research Establishment
Japan Atomic Energy Research Institute
Tokai-mura, Naka-gun, Ibaraki-ken

(Received March 16, 1999)

The nuclear power plant database contains the design information on 50 domestic light-water reactor nuclear power plants retrieved from the safety analysis reports. Its version for the main frame computer in the Japan Atomic Energy Research Institute (JAERI) was opened to the public in 1995.

In the present Internet version, approximately 64,000 data were retrieved from the main frame version and stored. These data include the shapes, dimensions, performances, etc. of the equipment and components described in "Appendix 8 : Safety design of the reactor facility" of each safety analysis report.

This report describes the database structure, program specification and information retrieval functions of the Internet version.

Keywords: Nuclear Power Plant, Safety Design Information, Database, Internet, User's
Manual

目 次

1. はじめに	1
2. データベースシステムの構築	1
2.1 システム設計	1
2.2 データベース構造	3
2.3 プログラム仕様	5
2.4 情報検索機能	6
2.5 画面構成	6
3. データの移植	20
4. 検索例	22
4.1 初期画面	22
4.2 文字列検索	23
4.3 項目検索	26
5. おわりに	28
参考文献	28
付録 各プログラムのモジュール一覧、モジュール設計概要	29

Contents

1. Introduction	1
2. Construction of Database System	1
2.1 System Design	1
2.2 Database Structure	3
2.3 Program Specification	5
2.4 Information Retrieval Function	6
2.5 Screen Layout	6
3. Transplantation of Data	20
4. Examples of Information Retrieval	22
4.1 Home Page	22
4.2 Character String Retrieval	23
4.3 Information Item Retrieval	26
5. Concluding Remarks	28
References	28
Appendix List of Program Modules and Outline of their Designs	29

This is a blank page.

1. はじめに

「原子力発電プラントデータベース」は、国内の軽水炉原子力発電プラントの設計方針、機器仕様など、プラントの設計に関する情報を系統的に整理し、データベース化して、情報の必要ときに迅速に提供できるようにした計算機システムである。

日本原子力研究所（原研）では、原子力の安全性研究などで必要となる原子力発電プラントの設計情報を所内で共有するため、昭和 58 年度から、原子炉設置許可申請書（申請書）を情報源として、設計情報の抽出、データベース化を開始した。⁽¹⁾ 昭和 63 年度には、東海研究所の大型計算機に当初の設計に基づいたシステムを完成させ、それ以後、所内の利用に供しながら平成 6 年度まで情報の追加更新を継続し、50 基の発電プラント（沸騰水型原子炉（BWR）：28 基、加圧水型原子炉（PWR）：22 基）の設計情報の登録を終了させた。^{(2) (3)}

平成 7 年度には、大型計算機のデータベースシステムを一般に公開し、所外からの利用を可能にしたが、所外から大型計算機を利用する場合、原研ネットワークへの接続や利用者登録に関して、原研から事前に許可を得ておく必要があるなど、使い勝手は良くなかった。

そこで、大型計算機上の情報を UNIX 系ワークステーションに移植し、データベース管理システム（DBMS : DataBase Management System）と WWW（World-Wide Web）サーバとを連携させることによって、インターネット上で検索できるデータベースシステムを構築した。

今回移植した情報は、大型計算機の格納情報 556,000 点のうち、大型計算機で利用の多かった「添付書類八 原子炉施設の安全設計に関する説明書」の設備や機器に関する形状、寸法、性能などの設計仕様 64,000 点である。

本報告書は、今回構築した「インターネット版」のデータベース構造、情報検索機能、検索方法などについてまとめたものであり、利用手引書として、また、設計書として利用できるように意図した。

2. データベースシステムの構築

2.1 システム設計

(1) 処理の流れ

利用者からの検索要求に対して、システムが情報を返すまでの処理の流れは、以下の考え方に基づいて設計した（図 1 参照）。

- ①利用者の検索要求は WWW ブラウザからインターネットを通じて WWW サーバへ送られる。
- ②検索プログラムが CGI（Common Gateway Interface）の機能によって呼び出され、Oracle データベースに対して検索を実行する。その際、利用者の検索条件は CGI を介して検索プログラムに渡される。
- ③検索結果は、HTML（HyperText Markup Language）変換プログラムによって、HTML 文書に変換され、WWW ブラウザに送り返される。
- ④送り返された検索結果は WWW ブラウザに表示される。

(2) システム環境

原子力発電プラントデータベースインターネット版は、以下のハードウェア、基本ソフトウェアを用いて構築した。

ハードウェア

ワークステーション：SUN7/300U モデル 200E

基本ソフトウェア

OS：日本語 Solaris 2.5.1

RDBMS：Oracle7 RDBMS V73L10 (リリース 7.3.2)

WWW サーバ：apache (httpd)

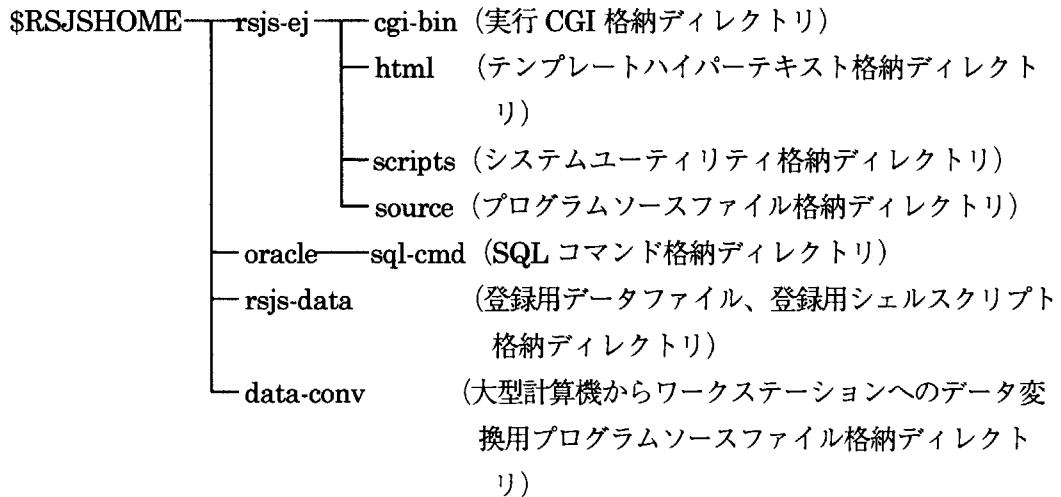
データコンバータ：富士通標準コード変換 1.0.2

コンパイラ：日本語 SPARCompiler C4.0

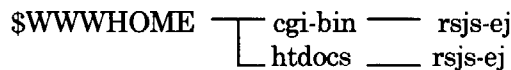
ネットワーク用漢字変換フィルタ：nkf V1.62

(3) 資源の格納ディレクトリ

インターネット版で作成した CGI プログラムなどの資源は、以下のディレクトリに格納することとした。各ディレクトリの内容は表1のとおりである。



但し、\$RSJSHOME はシステムホームディレクトリである。



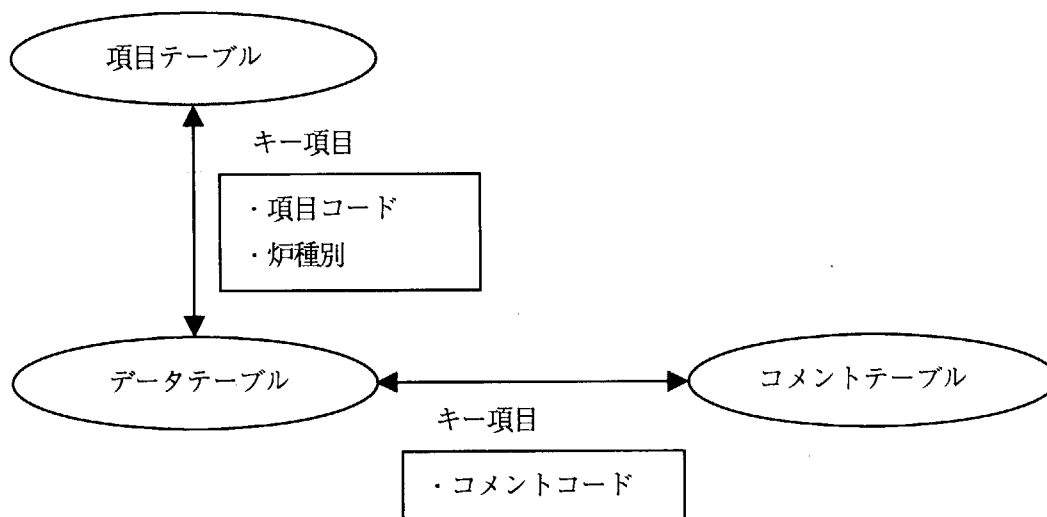
但し、\$WWWHOME は WWW サーバ格納ディレクトリである。

2.2 データベース構造

大型計算機（富士通 GS8400, OS:MSP）では、申請書から抽出した設計項目や設計仕様に、固有のコード番号を付けて体系的に整理し、情報の種類別にファイルに格納して VSAM (Virtual Storage Access Method) ファイル編成を用いて管理している（図2、表2 参照）。⁽³⁾

インターネット版では、情報をより効率的に管理するため、以下の三つのテーブルで構成するデータベースとし、これまでに抽出してきた情報を各テーブルに移植して、キーとなる項目でリンクさせている。

テーブル名称	テーブル名	データベース名	キー項目
項目テーブル	item	rsjs	項目コード、炉種別
データテーブル	pldata	rsjs	項目コード、炉種別、コメントコード
コメントテーブル	cmt	rsjs	コメントコード



各テーブルのテーブルフォーマットと列名の内容は以下のとおりである。

(1) 項目テーブル (item)

テーブルフォーマット

番号	名称	列名	サイズ	
1	項目コード	大項目コード	l i t e m	char(7)
2		中項目コード	m i t e m	char(7)
3		小項目コード	s i t e m	char(7)
4		データ項目コード	d i t e m	char(7)
5	炉種別	r t y p e	char(1)	
6	属性フラグ	a t t r	int	
7	項目フラグ	i t e m f l g	int	
8	データフラグ	d a t a f l g	int	
9	項目名称・英語	e i t e m n m	vchar(400)	
10	項目名称・日本語	j i t e m n m	vchar(400)	

- ・項目コードは、設計項目を識別するためのコードで、4階層に階層化した設計項目を、各コードとも"7桁の数字"で表す。設計仕様は最下層の項目名とリンクさせている。
- ・炉種別は、情報がBWRプラントかPWRプラントかを識別するフラグで、BWRは"B"、PWRは"P"で表す。
- ・属性フラグは、登録していない情報などの項目名を非表示にするためのフラグで、非表示は"1"、表示は"0"で表す。
- ・項目フラグは、項目の階層深さを識別するフラグで、上位から"1"、"2"、"3"、"4"で表す。
- ・データフラグは、設計情報の種類を識別するフラグで、以下の数字で表す。

数値データ "1"
 文字データ "2"
 型式データ "3"
 数式データ "4"
 図データ "5"
 表データ "6"

(2) データテーブル (pldata)

テーブルフォーマット

番号	名称	列名	サイズ
1	大項目コード	l i t e m	char(7)
	中項目コード	m i t e m	char(7)
	小項目コード	s i t e m	char(7)
	データ項目コード	d i t e m	char(7)
2	炉種別	r t y p e	char(1)
3	属性フラグ	a t t r	int
4	プラントコード	p l n t c d	char(4)
5	プラント名・英語	e p l n t n m	vchar(300)
6	プラント名・日本語	j p l n t n m	vchar(200)
7	データ・英語	e d a t a	vchar(300)
8	データ・日本語	j d a t a	vchar(200)
9	文書コード	r e p o r t c d	char(6)
10	単位・英語	e u n i t	vchar(100)
11	単位・日本語	j u n i t	vchar(100)
12	コメントコード	c m t c d	char(4)
13	ページ	p a g e	char(12)
14	日付	d a y	char(4)

- ・項目コード
 - ・炉種別
 - ・属性フラグ
- } 内容は項目テーブルと同じ。

- ・プラントコードはプラントを識別するコードで、表3の英数字で表す。
- ・文書コードは、申請書から抽出した文章データを識別するコードで、"6桁の英数字"で表す。
- ・コメントコードは設計情報に対するコメント文を識別するコードで、"4桁の英数字"で表す。
- ・ページは設計情報が記載されていた申請書のページで、"全角6桁の数字"で表す。
- ・日付は、設計情報を抽出した申請書の発行日で、"4桁の西暦"で表す。

(3) コメントテーブル (cmt)

テーブルフォーマット

番号	名称	列名	サイズ
1	コメントコード	c m t c d	char(4)
2	空き	r e s e r v	char(3)
3	コメント文・英語	e c m t n m	vchar(200)
4	コメント文・日本語	j c m t n m	vchar(200)

- ・コメントコードの内容はデータテーブルと同じ。
- ・コメント文は、設計仕様に対する解説文で、申請書に書かれていた文を抽出した。

2.3 プログラム仕様

インターネット版は、WWWサーバのドキュメントルートと cgi-bin から\$RSJSHOME にリンクをはって、ブラウザからの参照を行っており、WWWサーバと DBMS とを連携させて検索を実行するために以下のプログラムを作成した。また、項目テーブル (item) 及びデータテーブル (pldata) に属性フラグを設定するプログラムや、文字コードを変換するプログラムなどを作成した。各プログラムのモジュール一覧、モジュール設計概要は付録として巻末に示した。

(1) プラント選択画面作成処理 (sjs01)

プラントの選択画面用のハイパーテキストを作成し、ブラウザ側に送信する処理。

(2) 項目選択画面作成処理 (sjs02)

項目の選択画面用のハイパーテキストを作成し、ブラウザ側に送信する処理。

(3) データ表示画面作成処理 (sjs03)

データ表示画面用のハイパーテキストを作成し、ブラウザ側に送信する処理。

(4) 全文検索処理 (sjs04)

検索条件から該当するデータを検索し、検索結果表示画面用のハイパーテキストを作成してブラウザ側に送信する処理。

(5) 属性フラグセット処理 (setflg)

項目テーブル及びデータテーブルの「属性フラグ」列に、表示・非表示フラグのセット及びリセットを行なう処理。

(6) 共通処理 (sjslib)

各プログラムで共通に使用するプログラムライブラリ。

2.4 情報検索機能

検索機能として、今回は①文字列検索機能と②項目検索機能を作成した。

(1)文字列検索機能

文字列検索は、Oracle の構造化照会言語 (SQL : Structured Query Language) を用いたパターンマッチングによって、項目名称及び設計仕様を検索する機能で、入力文字列と部分一致する文字列を検索する。検索には論理演算記号指定を可能にした。

・ 論理演算記号 (半角)

& : 論理積

| : 論理和

文字列検索の検索範囲は、項目テーブルの「項目名称」及びデータテーブルの「データ」とした。なお、プラント選択が設定されている場合は、選択されたプラントに限定して検索を行うこととした。また、検索処理は、それぞれのテーブルを単独で検索し、表示の段階で検索結果を項目順に連結する処理とした。

検索結果の表示は、入力文字列がデータと一致した場合でも、初めに項目名称だけ階層表示させることとし、データを参照する場合は、アンカー付きの項目名称を選択することによって参照することとした。

(2)項目検索機能

項目検索は、画面に表示されている項目名称を順次選択していき、最下層のアンカー付き項目名称を選択したときに設計仕様を表示させる機能である。プラント選択が設定されている場合は、文字列検索と同様に選択されたプラントだけの設計仕様を表示させることとした。

2.5 画面構成

画面設計にあたっては、利用者が Netscape Navigator 3.0 あるいは Internet Explorer 3.0 以上のブラウザを使用していることを前提に設計した。

検索画面は図3に示すように、HTML ファイルとそのファイルから呼び出す CGI プログラムによって構成されており、検索しやすくするために3画面に分割した。

frame4 は、画面制御と文字列検索時の入力画面に使用することとし、文字列入力欄と実行ボタン、ホームページへの戻りボタン、選択項目のリセットクリアボタン、ヘルプボタンを配した。

frame2 は、プラント選択画面に使用することとし、炉型選択ボタン、プラント選択ボタンを配した。

frame3 は、項目検索時や文字列検索時の項目名称の表示画面に使用するとともに、アンカー付き項目名称を選択したときの設計仕様の表示画面に使用することとした。

表1 格納ディレクトリの内容 (cgi-bin) (1/7)

CGIプログラム一覧			CGIプログラム	
	ID	RSJS	ID	CGI-BIN

CGIプログラム名	概要
sjs01.cgi	プラント選択画面作成処理CGIプログラム
sjs02.cgi	項目選択画面作成処理CGIプログラム
sjs03.cgi	データ表示画面作成処理CGIプログラム
sjs04.cgi	全文検索処理CGIプログラム
initenv	CGIプログラム実行環境設定用シェルスクリプト

表1 格納ディレクトリの内容 (scripts) (3/7)

スクリプト一覧			スクリプト	
	ID	RSJS	ID	SCRIPTS
スクリプト名	処理概要			
makerb.sql	ロールバックテーブルスペースを確保する。			
plantdata.sql	プラントデータを表示する。			
res-attr.sql	全てのデータに属性フラグ"1"をセットする。			
set-attr.sql	データフラグが1~4のデータに属性フラグ"0"をセットする。			
setflg-all.sql	全てのデータに属性フラグ"0"をセットする。			
showdbseg.sql	データベースセグメント情報を表示する。			
setallflg	セットフラグ実行用シェルスクリプト			
resetflg	リセットフラグ実行用シェルスクリプト			
setflg	セットフラグ実行用シェルスクリプト			
sql	SQL文実行用シェルスクリプト			

表1 格納ディレクトリの内容 (source) (4/7)

プログラムソース一覧			プログラムソース	
	ID	RSJS	ID	SOURCE

プログラムソース一覧名	概要
include	プログラム共通インクルードファイル格納ディレクトリ
lib	プログラム共通関数ファイル格納ディレクトリ
nkf16	ネットワーク用漢字変換フィルタプログラムソースファイル格納ディレクトリ
setflg	属性フラグセットプログラムソースファイル格納ディレクトリ
sj s 0 1	プラント選択画面作成処理プログラムソースファイル格納ディレクトリ
sj s 0 2	項目選択画面作成処理プログラムソースファイル格納ディレクトリ
sj s 0 3	データ表示画面作成処理プログラムソースファイル格納ディレク
sj s 0 4	全文検索処理プログラムソースファイル格納ディレクトリ

表1 格納ディレクトリの内容 (sql-cmd) (5/7)

SQLコマンド一覧			SQLコマンド	
	ID	RSJS	ID	SQL-CMD

SQLコマンド名	処理概要
createcmt.sql	コメントテーブル作成用コマンド
createindex.sql	インデックス作成用コマンド
createitem.sql	項目テーブル作成用コマンド
createpdata.sql	データテーブル作成用コマンド
createuser.sql	ユーザ作成用コマンド
dropindex.sql	インデックス削除用コマンド
jaeri_allcmd.sql	RSJSデータベース作成用コマンド
jaeri_createcontrol.sql	コントロールファイル作成用コマンド
jaeri_createrbseg.sql	r bセグメント作成用コマンド
jaeri_createrbspace.sql	r bセグメントテーブルスペース作成用コマンド
jaeri_creatersjsdb.sql	データベース作成用コマンド
jaeri_creatersjstable.sql	テーブルスペース作成用コマンド
jaeri_createtemproll.sql	テンポラリロールバック作成用コマンド
jaeri_createtempspace.sql	テンポラリスペース作成用コマンド
jaeri_createtoolspace.sql	ツールスペース作成用コマンド
jaeri_rbsegonline.sql	r bセグメントオンライン用コマンド
startup.sql	データベース起動用コマンド
startupinit.sql	データベース初期起動用コマンド

表1 格納ディレクトリの内容 (data-conv) (7/7)

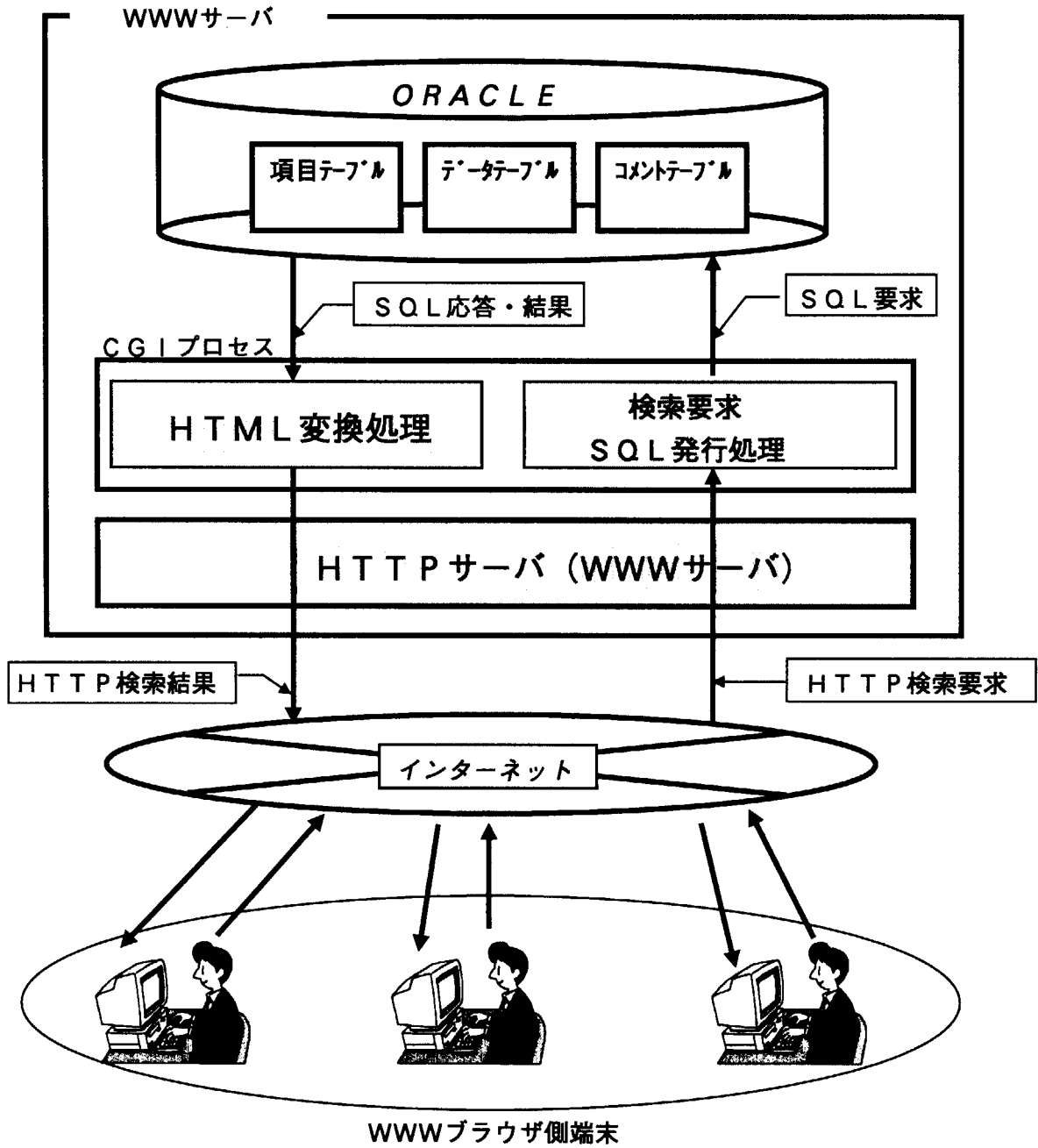
データ変換プログラム ソース一覧		データ変換プログラムソース	
ID	RSJS	ID	DATA-CONV
プログラムソース一覧名	概要		
comment	コメントデータ変換用プログラムソース格納ディレクトリ		
databwr	BWR項目データ変換用プログラムソース格納ディレクトリ		
datapwr	PWR項目データ変換用プログラムソース格納ディレクトリ		
fjconv	fjconv変換テーブル格納ディレクトリ		
itembwr	BWR項目変換用プログラムソース格納ディレクトリ		
itempwr	PWR項目変換用プログラムソース格納ディレクトリ		
mst	マスタファイル変換用プログラムソース格納ディレクトリ		
plant	プラントデータ変換用プログラムソース格納ディレクトリ		
unit	単位データ変換用プログラムソース格納ディレクトリ		

表2 大型計算機上のデータ管理例

原子炉設置許可申請書〔添付書類8〕項目一覧リスト				
項目番号				
大	中	小	データ	
0801000	0000000	0000000	0000000	①安全設計 (添付書類8)
0801000	0001000	0000000	0000000	②安全設計の方針
0801000	0001000	0001000	0000000	③安全設計の基本方針
0801000	0001000	0001000	0010000	安全設計適用法令及び審査指針
0801000	0001000	0001000	0020000	基本方針
0801000	0001000	0002000	0000000	④原子炉固有の安全性
0801000	0001000	0002000	0010000	原子炉固有の安全性
0801000	0001000	0003000	0000000	⑤原子炉施設的设计、製作における安全上の考慮 (原子炉系的设计方針)
0801000	0001000	0003000	0010000	原子炉施設的设计、製作における安全上の考慮 (原子炉系的设计方針)
0801000	0001000	0004000	0000000	⑥炉心の核設計基本方針
0801000	0001000	0004000	0000200	炉心の核設計
0801000	0001000	0004000	0010000	炉心有効高さ対等価直径比
0801000	0001000	0004000	0020000	炉心形状
0801000	0001000	0004000	0030000	燃料集合体数
0801000	0001000	0004000	0040000	燃料濃縮度の決定
0801000	0001000	0004000	0050000	原子炉の反応度制御
0801000	0001000	0004000	0060000	原子炉の出力制御
0801000	0001000	0005000	0000000	⑦炉心の熱水力設計基本方針
0801000	0001000	0005000	0010000	炉心の熱水力設計
0801000	0001000	0005000	0020000	最小限界出力比 (通常運転時、過渡変化時)
0801000	0001000	0005000	0030000	被覆管円周方向の平均歪み率 (通常運転時、過渡変化時)
0801000	0001000	0005000	0040000	最小限界出力比 (各サイクルについて、初期から、末期よりさかのぼって、炉心平均燃焼度で1000Mwd/t手前までの期間)
0801000	0001000	0005000	0050000	最小限界出力比 (各サイクルについて、初期から、末期よりさかのぼって、炉心平均燃焼度で1000Mwd/t手前以外の期間)
0801000	0001000	0005000	0060000	最小限界出力比 (新型8×8燃料が実用されるまでのサイクル)
0801000	0001000	0005000	0070000	最小限界出力比 (通常運転時)
0801000	0001000	0005000	0080000	燃料棒最大出力密度 (通常運転時)
0801000	0001000	0006000	0000000	⑧核分裂生成物放射の防護
0801000	0001000	0006000	0010000	核分裂生成物放射の防護
0801000	0001000	0007000	0000000	⑨平常運転時における核分裂生成物放射の防止及び抑制
0801000	0001000	0007000	0010000	平常運転時における核分裂生成物放射の防止及び抑制
0801000	0001000	0008000	0000000	⑩安全保護系設計の基本方針
0801000	0001000	0008000	0010000	安全保護系設計の基本方針
0801000	0001000	0009000	0000000	⑪反応度制御系設計の方針
0801000	0001000	0009000	0010000	反応度制御系設計の方針
0801000	0001000	0010000	0000000	⑫工学的安全施設設計の基本方針
0801000	0001000	0010000	0010000	工学的安全施設設計の基本方針
0801000	0001000	0011000	0000000	⑬放射性廃棄物の処理施設設計の方針
0801000	0001000	0011000	0010000	放射性廃棄物の処理施設設計の方針
0801000	0001000	0012000	0000000	⑭強度設計の基本方針
0801000	0001000	0012000	0010000	強度設計の基本方針
0801000	0001000	0013000	0000000	⑮品質保証の基本方針
0801000	0001000	0013000	0010000	品質保証の基本方針
0801000	0001000	0014000	0000000	⑯火災に対する設計上の配慮
0801000	0001000	0014000	0010000	火災に対する設計上の配慮
0801000	0001000	0015000	0000000	⑰安全上重要な系統の機械的、電気的分離 (物理的分離)
0801000	0001000	0015000	0010000	機械的、電気的分離 (物理的分離)
0801000	0001000	0016000	0000000	⑱環境条件に対する設計の基本方針
0801000	0001000	0016000	0010000	環境条件に対する設計の基本方針
0801000	0001000	0017000	0000000	⑲放射線に対する設計の基本方針
0801000	0001000	0017000	0010000	放射線に対する設計の基本方針
0801000	0002000	0000000	0000000	⑳放射線管理の方針
0801000	0002000	0001000	0000000	放射線管理の方針
0801000	0002000	0001000	0010000	放射線管理の方針
0801000	0003000	0000000	0000000	㉑発電機水型原子炉施設に関する安全設計審査指針への適合
0801000	0003000	0001000	0000000	①指針1、準拠規格および基準
0801000	0003000	0001000	0010000	適合のための設計方針
0801000	0003000	0001000	0012000	原子炉冷却材圧力バウンダリの準拠している法令、規格、基準
0801000	0003000	0001000	0013000	格納容器バウンダリの準拠している法令、規格、基準
0801000	0003000	0001000	0014000	工学的安全施設設計の準拠している法令、規格、基準
0801000	0003000	0001000	0015000	原子炉停止系の準拠している法令、規格、基準
0801000	0003000	0001000	0016000	原子炉冷却材圧力バウンダリ、格納容器バウンダリ、工学的安全施設、原子炉停止系に関連する安全保護系、電源の法令、規格、基準
0801000	0003000	0001000	0017000	原子炉建屋の準拠している法令、規格、基準
0801000	0003000	0002000	0000000	②指針2、自然現象に対する設計上の考慮
0801000	0003000	0002000	0010000	適合のための設計方針 (1. について)
0801000	0003000	0002000	0015000	適合のための設計方針 (2. について)
0801000	0003000	0002000	0020000	敷地付近で観測された最大瞬間風速
0801000	0003000	0002000	0030000	最深積雪量観測記録
0801000	0003000	0002000	0040000	建築基準法に定める最深積雪量としての設計値
0801000	0003000	0002000	0050000	最低気温観測記録
0801000	0003000	0002000	0060000	月平均最低気温
0801000	0003000	0002000	0070000	日 (平均) 最低気温
0801000	0003000	0002000	0080000	最高潮位観測記録
0801000	0003000	0002000	0090000	最低潮位観測記録
0801000	0003000	0002000	0010000	朔望平均満潮位
0801000	0003000	0002000	0011000	朔望平均干満差
0801000	0003000	0002000	0012000	参照: 添付書類
0801000	0003000	0003000	0000000	③指針3、人為事象に対する設計上の考慮
0801000	0003000	0003000	0010000	適合のための設計方針
0801000	0003000	0003000	0020000	参照: 添付書類
0801000	0003000	0004000	0000000	④指針4、環境条件に対する設計上の考慮
0801000	0003000	0004000	0010000	適合のための設計方針

表3 プラントコード一覧

BWR		PWR	
プラント名	プラントコード	プラント名	プラントコード
東海2号	B021	敦賀2号	P032
敦賀1号	B031	泊1号	P111
女川1号	B211	泊2号	P112
女川2号	B212	美浜1号	P511
福島第1-1号	B311	美浜2号	P512
福島第1-2号	B312	美浜3号	P513
福島第1-3号	B313	高浜1号	P521
福島第1-4号	B314	高浜2号	P522
福島第1-5号	B315	高浜3号	P523
福島第1-6号	B316	高浜4号	P524
福島第2-1号	B321	大飯1号	P531
福島第2-2号	B322	大飯2号	P532
福島第2-3号	B323	大飯3号	P533
福島第2-4号	B324	大飯4号	P534
柏崎・刈羽-1号	B331	伊方1号	P711
柏崎・刈羽-2号	B332	伊方2号	P712
柏崎・刈羽-3号	B333	玄海1号	P811
柏崎・刈羽-4号	B334	玄海2号	P812
柏崎・刈羽-5号	B335	玄海3号	P813
柏崎・刈羽-6号	B336	玄海4号	P814
柏崎・刈羽-7号	B337	川内1号	P821
浜岡1号	B411	川内2号	P822
浜岡2号	B412		
浜岡3号	B413		
浜岡4号	B414		
島根1号	B611		
島根2号	B612		
志賀	B911		



HTTP : HyperText Transfer Protocol
SQL : Structured Query Language
CGI : Common Gateway Interface
HTML : HyperText Markup Language

図1 検索処理の流れ

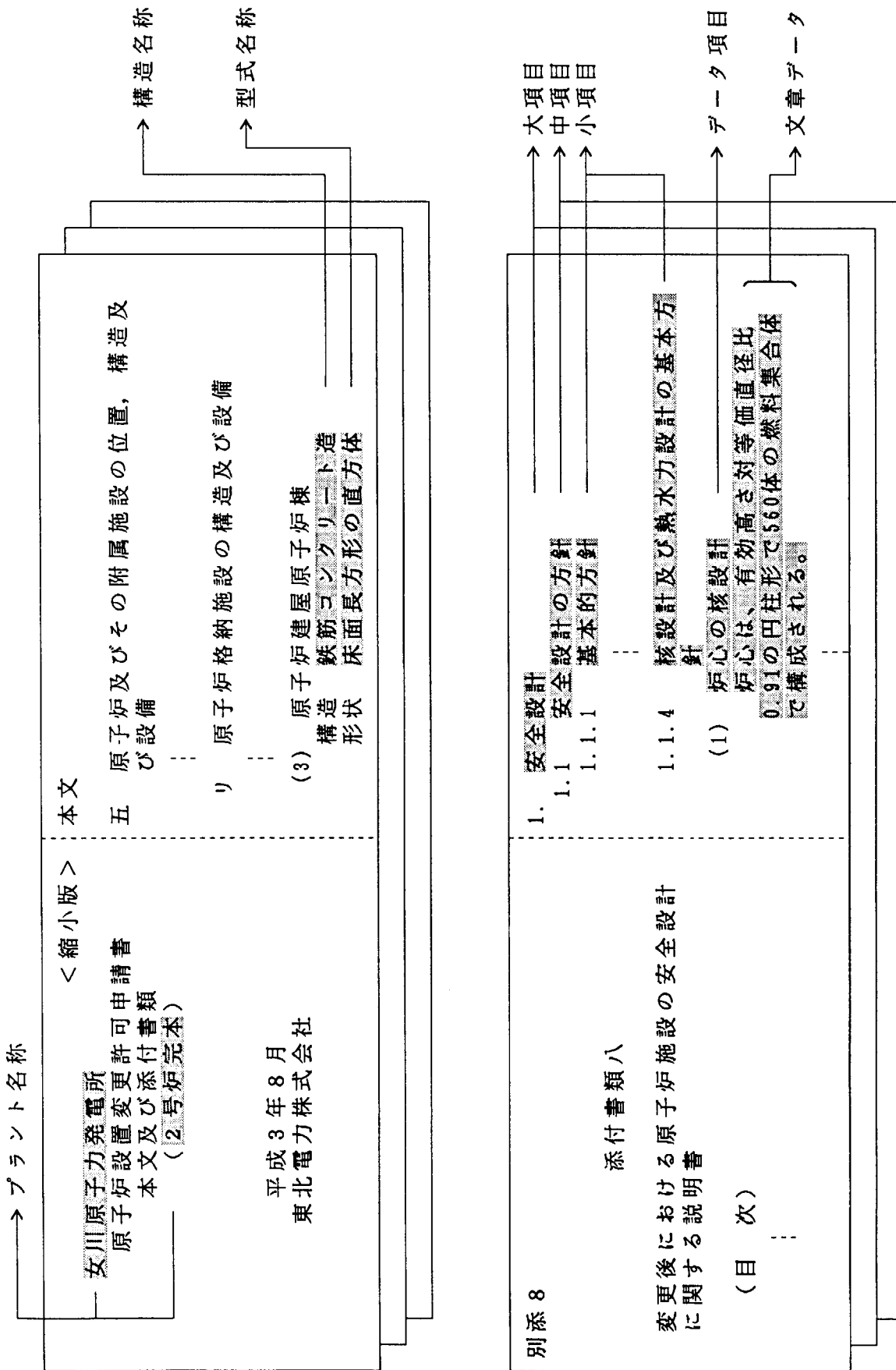


図2 設計情報の抽出例 (1 / 2)

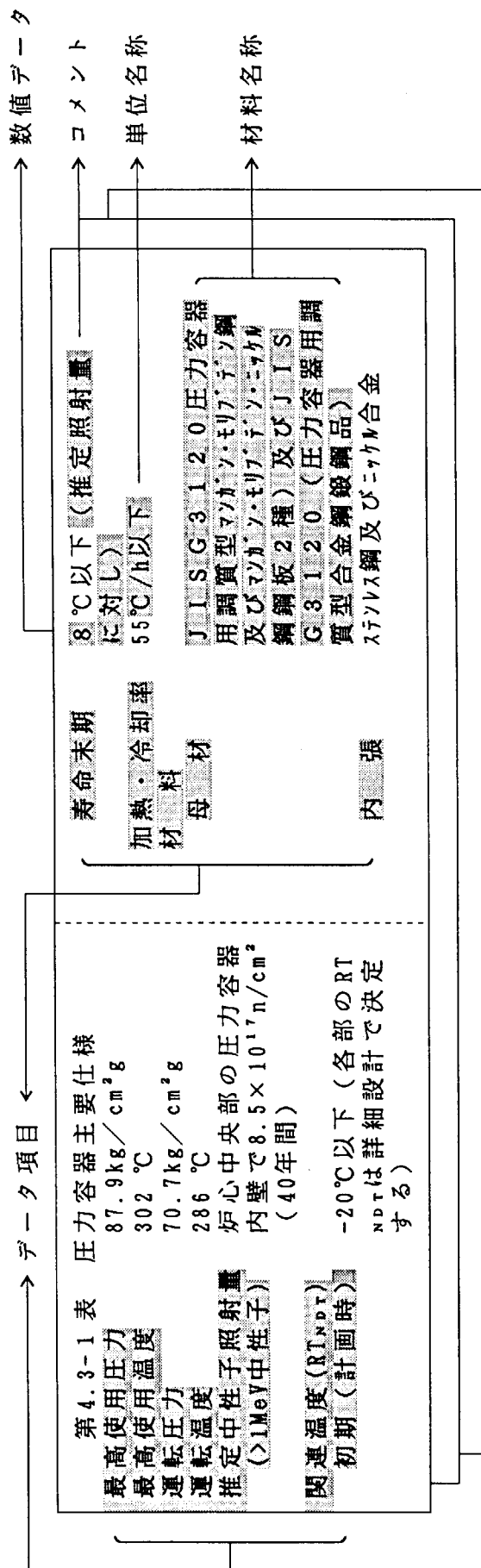


図2 設計情報の抽出例 (2 / 2)

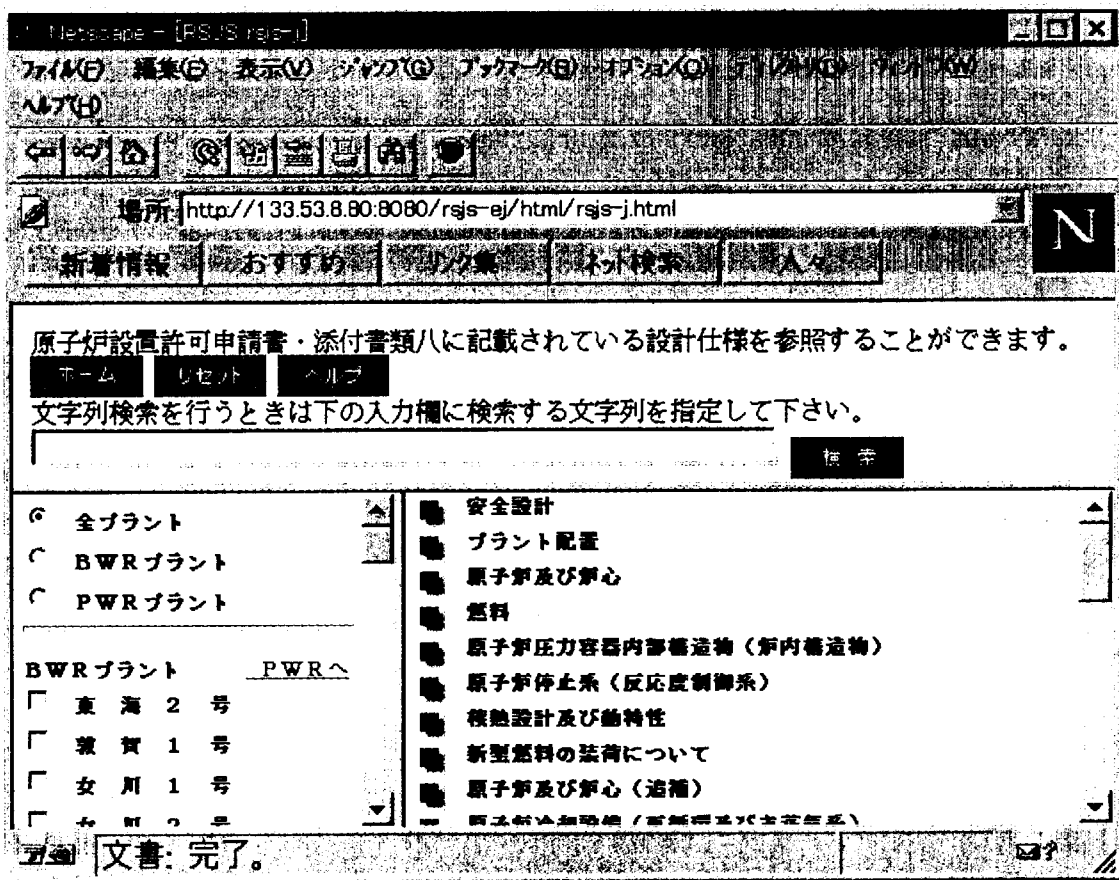
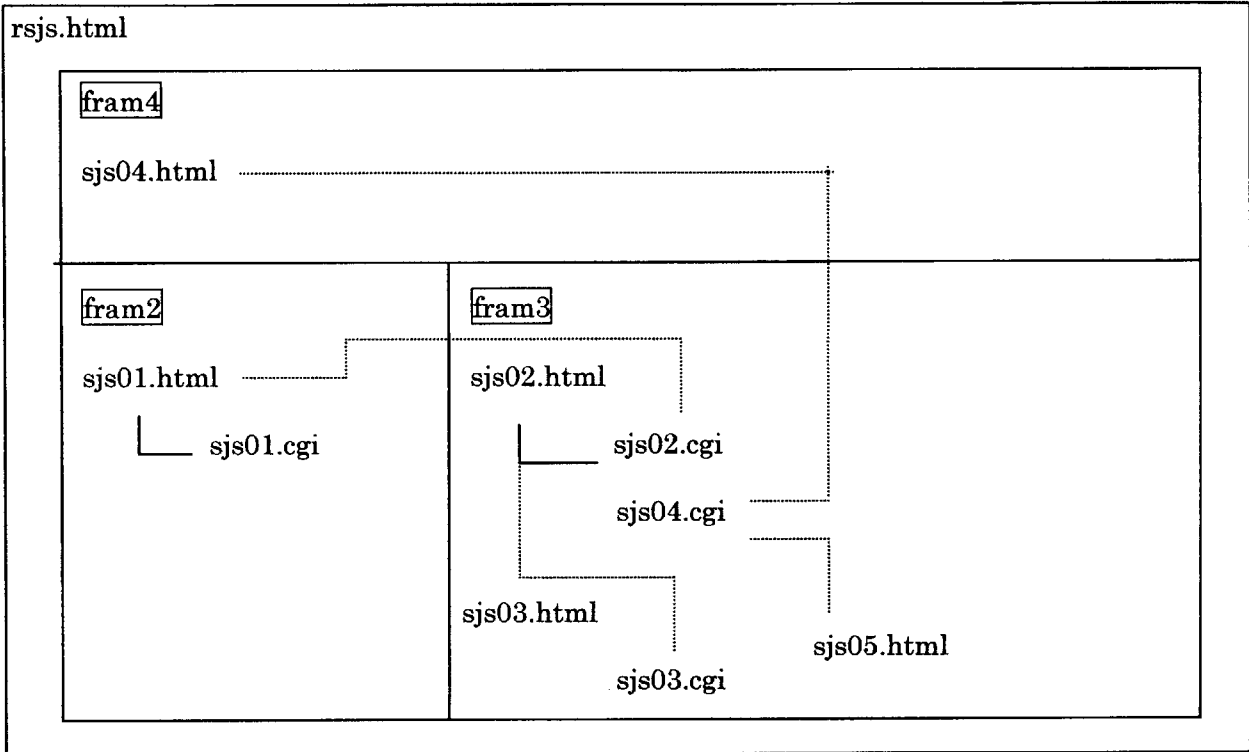


図3 検索画面構成

3. データの移植

大型計算機から必要な情報をワークステーションに移植するためには、ファイル変換と併せて文字コードの変換が必要になることから、今回は、以下に示す手順でデータを移植した(図4参照)。

- ①大型計算機で、REPRO コマンドを用いて、VSAM ファイル (JEF コード) を PS ファイル (JEF コード) に変換する。
- ②大型計算機上の PS ファイル (JEF コード) をワークステーションに転送する。
- ③ワークステーションで、fjconv コマンドを用いて、文字コードを JEF コードから euc コードに変換する。
- ④ワークステーションで、変換プログラムを介して、大型計算機のファイル形式からワークステーションのファイル形式に変換し、ロード用ファイルを作成する。
- ⑤ワークステーションで、SQL コマンドを用いて、ロード用ファイルの各情報を Oracle データベースに格納する。
- ⑥項目名称、設計仕様の表示・非表示属性フラグを設定する。

・データフラグが "1" "2" "3" "4" "5" "6" の情報の属性フラグを全て表示 "0" に設定する場合

```
>setallflg
```

・データフラグが "1" "2" "3" "4" の情報の属性フラグを設定する場合

```
>setflg [フラグ 炉種別 大項目コード [中項目コード] [小項目コード] [データ項目コード]]
```

フラグ	0 : 表示、1 : 非表示 省略時はデータフラグが "1" "2" "3" "4" の情報は全て表示 "0"
炉種別	B : BWR、P : PWR
各項目コード	各項目コードをブランクで区切って指定する

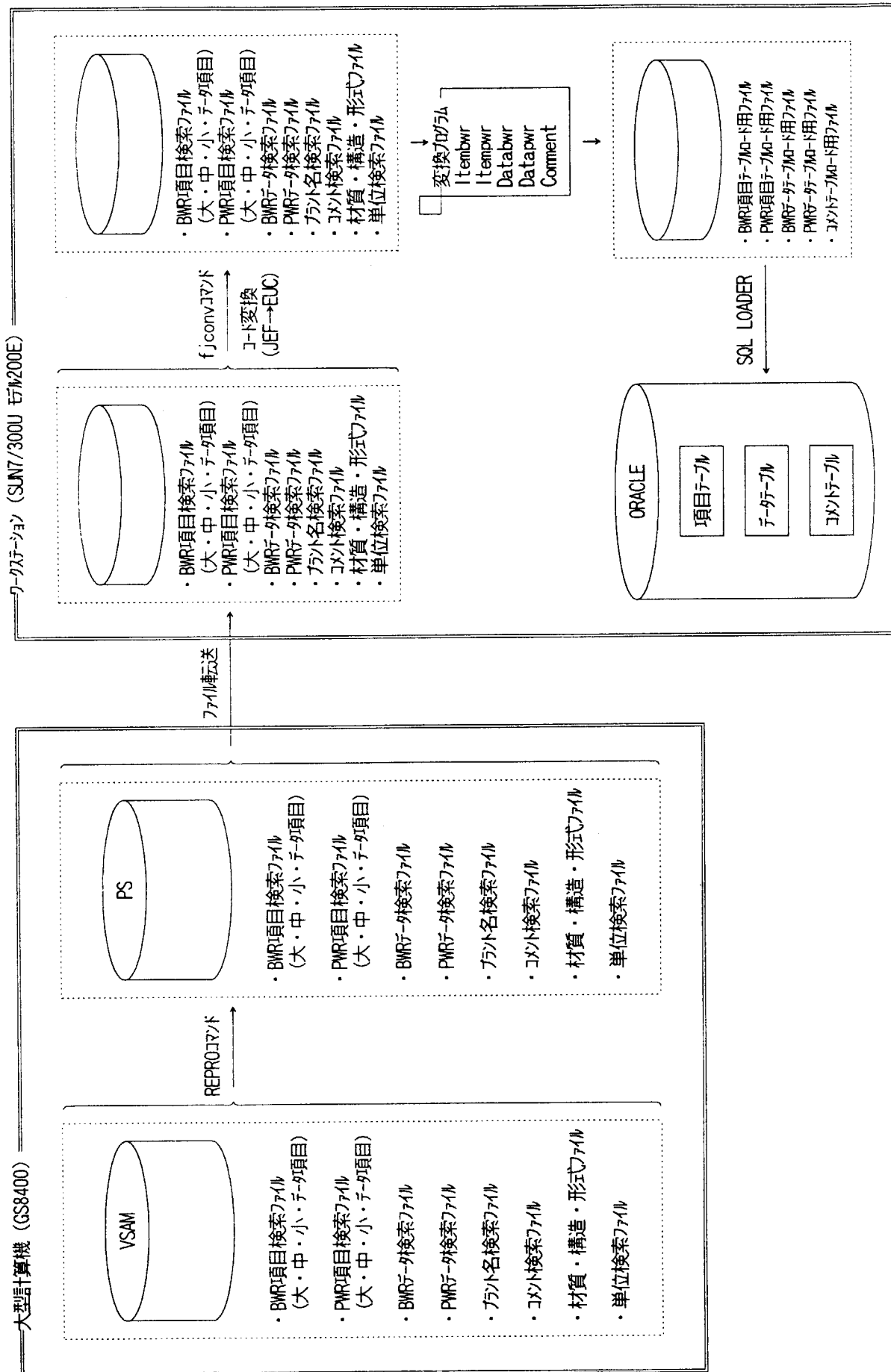


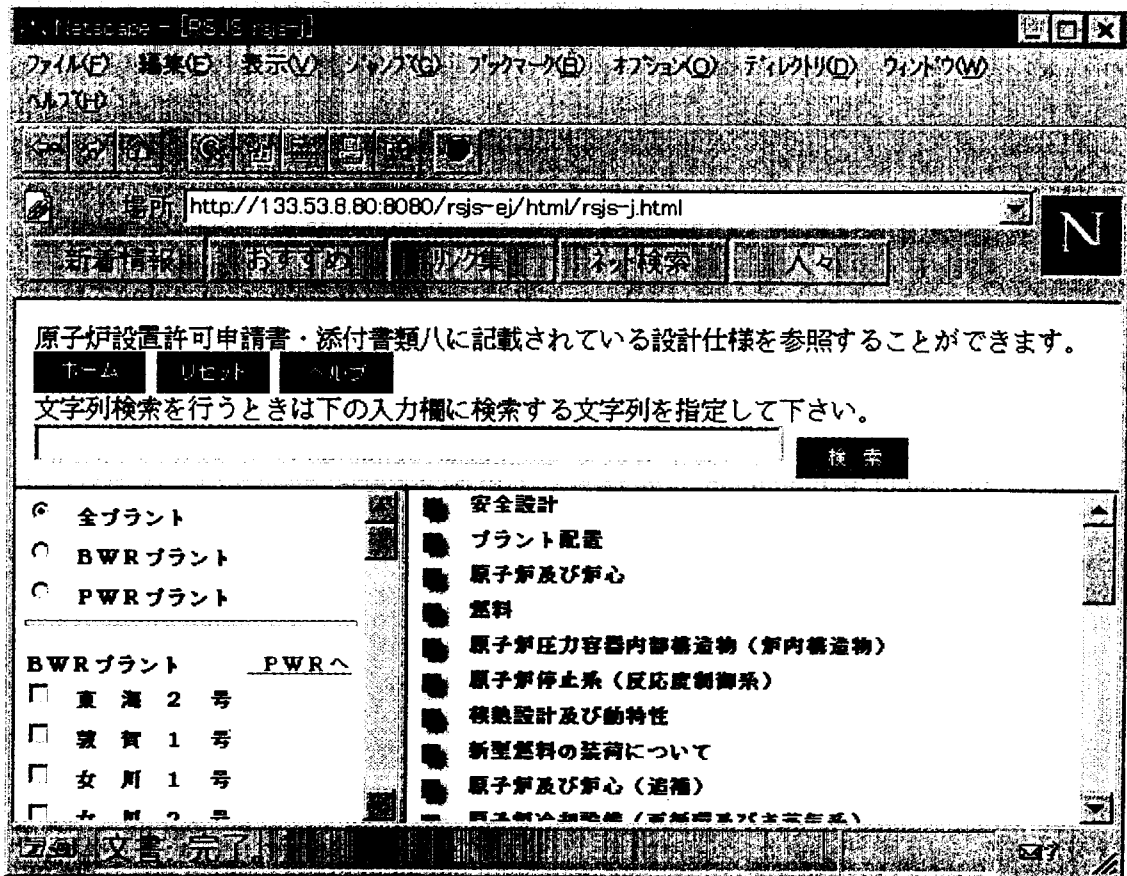
図 4 データ移植概念図

4. 検索例

以下に、WWW ブラウザからの検索例を示す。ここではブラウザに Netscape 3.0 を使用した。

4.1 初期画面

WWW ブラウザに URL を指定し、初期画面を表示させる。



[ボタン操作]

ホーム : ホームページへ戻る。

リセット : 初期画面状態へ戻る。

ヘルプ : 操作方法などが書かれたヘルプ画面を表示する。

検索 : 文字列検索を実行する。

[プラント選択]

BWR プラント : BWR 全プラントが検索対象となる。

PWR プラント : PWR 全プラントが検索対象となる。

個別プラント : 複数選択が可能である。

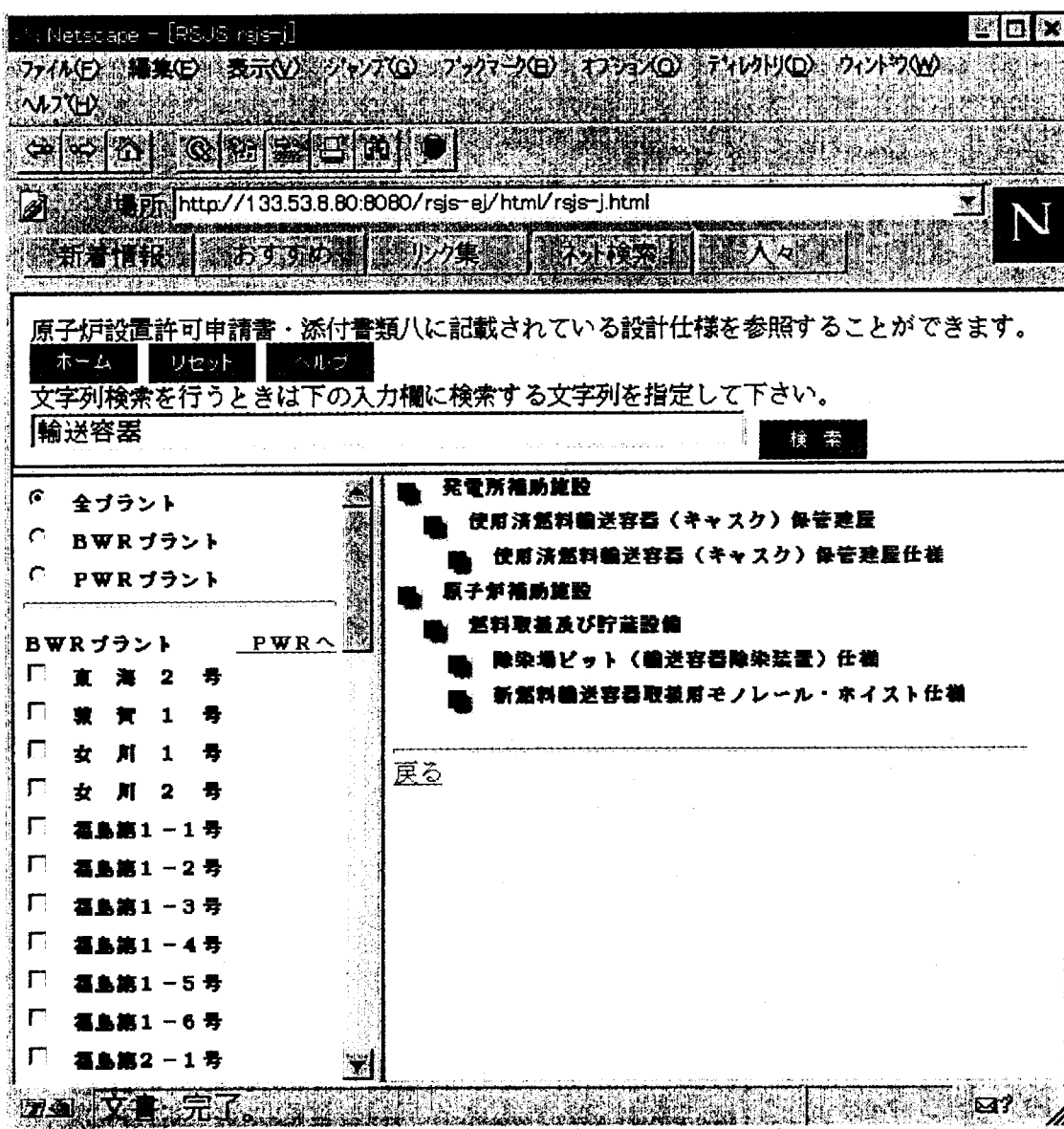
[プラントの移動]

PWR へ : PWR のプラント名の位置にカーソルを移動する。

BWR へ : BWR のプラント名の位置にカーソルを移動する。

4.2 文字列検索

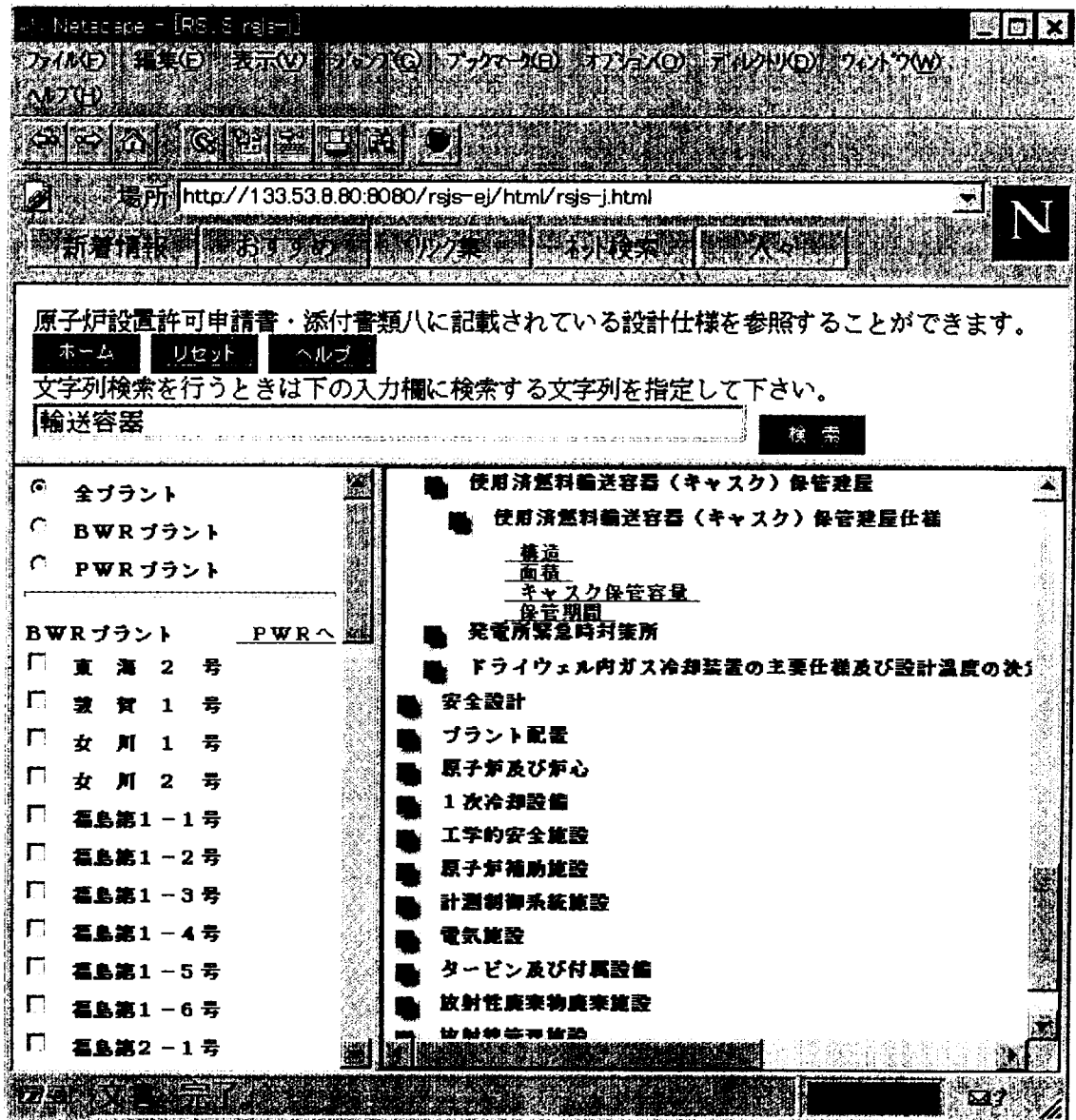
- (1)文字列検索を行う場合は、文字列入力欄に検索したい文字列を入力し、検索ボタンをクリックする。ここでは「輸送容器」と入力した。
- (2)プラントを指定する場合は、指定したいプラントのチェックボックスを選択してから検索ボタンをクリックする。ここでは全プラントを検索対象にしている。



[入力条件]

検索文字列のどれかが含まれている情報を検索したい場合は半角の「|」でつなげる。
 検索文字列が全て含んだ情報を検索したい場合は半角の「&」でつなげる。

- (3)参照したい設計項目アイコンをクリックしていき、アンカー付きのデータ項目を表示させる。
 ここでは「使用済燃料輸送容器（キャスク）保管建屋仕様」を選択した。



(4)アンカー付きのデータ項目をクリックし、設計情報を参照する。ここでは「キャスク保管容量」を選択した。

原子炉設置許可申請書・添付書類八に記載されている設計仕様を参照することができます。
[ホーム](#) [リセット](#) [ヘルプ](#)
 文字列検索を行うときは下の入力欄に検索する文字列を指定して下さい。

検索結果の表示

発電所補助施設
 使用消費材料輸送容器 (キャスク) 保管建屋
 使用消費材料輸送容器 (キャスク) 保管建屋仕様
 キャスク保管容量

福島第1-1号	最大(1号~6号炉共用)	26基
福島第1-2号	最大(1号~6号炉共用)	26基
福島第1-3号	最大(1号~6号炉共用)	26基
福島第1-4号	最大(1号~6号炉共用)	26基
福島第1-5号	最大(1号~6号炉共用)	26基
福島第1-6号	最大(1号~6号炉共用)	26基
浜岡1号	1号、2号、3号及び4号炉共用	10基
浜岡2号	1号、2号、3号及び4号炉共用	10基
浜岡3号	1、2、3号炉共用	10基
浜岡4号	1号、2号、3号及び4号炉共用、既設	10基

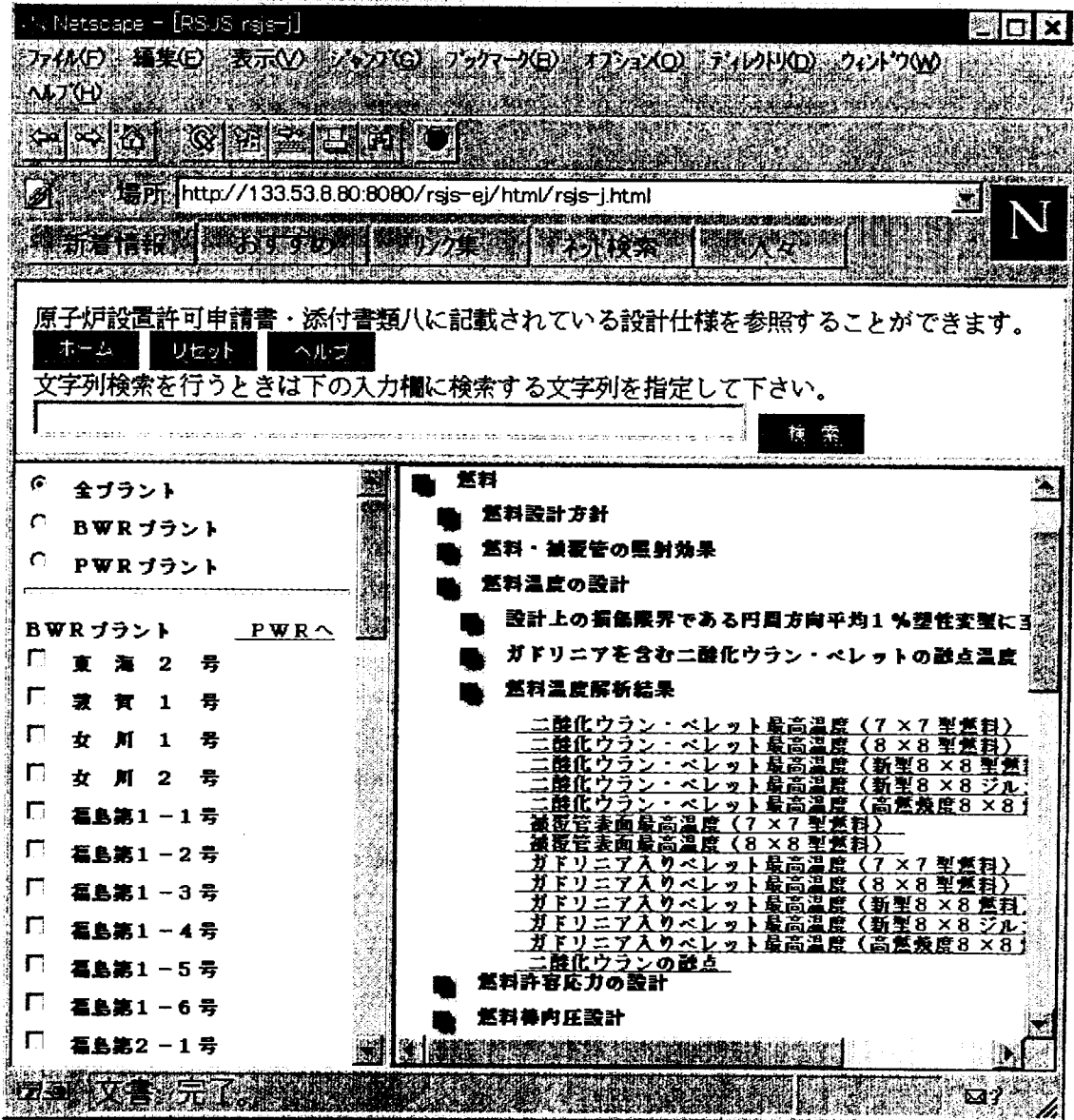
全プラント
 BWRプラント
 PWRプラント

BWRプラント PWRへ

東海2号
 敦賀1号
 女川1号
 女川2号
 福島第1-1号
 福島第1-2号
 福島第1-3号
 福島第1-4号
 福島第1-5号
 福島第1-6号
 福島第2-1号

4.3 項目検索

- (1)項目検索を行う場合は、参照したい設計項目アイコンをクリックしていき、アンカー付きのデータ項目を表示させる。ここでは「燃料」「燃料温度の設計」「燃料温度解析結果」と、順に選択した。
- (2)プラントを指定する場合は、指定したいプラントのチェックボックスを選択する。ここでは全プラントを検索対象にしている。



(3)アンカー付きのデータ項目をクリックし設計情報を参照する。ここでは「被覆管表面最高温度(8×8型燃料)」を選択した。

原子炉設置許可申請書・添付書類八に記載されている設計仕様を参照することができます。

ホーム リセット ヘルプ

文字列検索を行うときは下の入力欄に検索する文字列を指定して下さい。

検索

検索結果の表示

燃料温度の設計
 燃料温度解析結果
 被覆管表面最高温度(8×8型燃料)

敦賀1号	設計出力密度(0.44kW/cm ²)で燃料寿命を満して	300, 380℃
女川1号	設計出力密度(0.44kW/cm ²)で	390℃
福島第1-1号	設計出力密度(0.44kW/cm ²)で	300℃
福島第1-2号	設計出力密度(0.44kW/cm ²)で	300℃
福島第1-3号	設計出力密度(0.44kW/cm ²)で	300℃
浜岡1号	設計出力密度(0.44kW/cm ²)で	300℃
浜岡2号	設計出力密度(0.44kW/cm ²)で	300℃
島根1号	設計出力密度(0.44kW/cm ²)で	300℃

全プラント
 BWRプラント
 PWRプラント

BWRプラント PWRへ

- 東海2号
- 敦賀1号
- 女川1号
- 女川2号
- 福島第1-1号
- 福島第1-2号
- 福島第1-3号
- 福島第1-4号
- 福島第1-5号
- 福島第1-6号
- 福島第2-1号

完了

5. おわりに

原子炉設置許可申請書を情報源として、国内の軽水型原子力発電所 50 基分の設計情報をデータベース化している原子力発電プラントデータベース大型計算機版の格納情報のうち、添付書類八の設備や機器に関する設計仕様をインターネット上で検索できるようにするため、ワークステーションにデータベースを構築し、大型計算機版から情報を移植した。さらに、検索機能として、文字列検索機能、項目検索機能を作成した。これにより、国内の原子力発電プラントの設計情報を、容易に参照できるようになった。

参考文献

- (1) 石川雅章ほか、“原子力発電プラント・データベースシステムの概念設計” JAERI-M 84-051、1984年3月
- (2) 泉 文男ほか、“原子力発電プラント・データベース (PPD) の情報検索システム利用手引書” JAERI-M 90-218、1984年3月
- (3) 山本信夫ほか、“原子力発電プラント・データベースの構造及びファイル様式” JAERI-Data/Code 95-018、1995年12月

付録 各プログラムのモジュール一覧、モジュール設計概要

プラント選択画面作成処理	(sjs01)
項目選択画面作成処理	(sjs02)
データ表示画面作成処理	(sjs03)
全文検索処理	(sjs04)
属性フラグセット処理	(setflg)
共通処理	(sjslib)

モジュール一覧			プラント選択画面HTMLテキスト作成処理	
	ID	RSJS	ID	SJS01

プログラム	プラント選択画面HTMLテキスト作成処理		
プログラムID	SJS01		
モジュール名	モジュール概要	ソースファイル名	
main	メイン処理	main.c	
makehtml	HTMLテキストの作成	makehtml.c	

モジュール設計			プラント選択画面HTMLテキスト作成処理	
	ID	R S J S	ID	S J S 0 1

ソースファイル名	sjs01/main.c
モジュール名	メイン処理
モジュールID	main
処理概要	プラント選択画面HTMLテキスト作成メイン処理。
1. インクルードファイルの定義	
<pre>#include <stdio.h> #include <stdlib.h> #include <string.h> #include "fd.h"</pre>	
2. 関数の一覧	
<p>(1) 関数名称 : メイン処理関数</p> <pre>int main(argc, argv) int argc; char **argv;</pre> <p><処理概要></p> <ul style="list-style-type: none"> ・プラント選択画面HTMLテキスト作成処理を呼び出す。 	

モジュール設計			プラント選択画面HTMLテキスト作成処理	
	ID	RSJS	ID	SJS01

ソースファイル名	sjs01/makehtml.c
モジュール名	HTMLテキストの作成
モジュールID	makehtml
処理概要	プラント選択画面用HTMLテキストを作成する。
1. インクルードファイルの定義	
<pre>#include <stdio.h> #include <stdlib.h> #include <string.h> #include "fd.h"</pre>	
2. 関数の一覧	
<p>(1) 関数名称 : HTMLテキストの編集関数</p> <pre>int sjs_make_html(lang, type) char *lang; /* 言語 (E:英語, J:日本語) */ char *type; /* プラント種別(A, B, P) */</pre> <p><処理概要></p> <ul style="list-style-type: none"> ・テンプレートHTMLファイル名の編集を行なう。 ・テンプレートHTMLファイルのオープンを行なう。 ・テンプレートファイルの読み込みを行なう。 ・HTMLテキストの編集処理を呼び出す。 ・リストボックスの定義のときリストの個数を埋め込む。 ・リストボックスの項目定義のとき、valueと名称を埋め込む。 ・上記以外のHTMLテキストを出力する。 ・テンプレートファイルのクローズを行なう。 <p>(2) 関数名称 : プラント一覧作成処理関数</p> <pre>int sjs_plan_list(lang, type) char *lang; /* 言語 (E:英語, J:日本語) */ char *type; /* プラント種別(A, B, P) */</pre> <p><処理概要></p> <ul style="list-style-type: none"> ・plantファイル名の編集を行なう。 ・plantファイルのオープンを行なう。 ・plantファイルの読み込みを行なう。 ・plantファイルのリワインドを行なう。 ・plantファイルのクローズを行なう。 	

モジュール一覧			項目選択画面HTMLテキスト作成処理	
	ID	RSJS	ID	SJS02

プログラム	項目選択画面HTMLテキスト作成処理	
プログラムID	SJS02	
モジュール名	モジュール概要	ソースファイル名
main	メイン処理	main.c
database	データの検索処理	database.c
makehtml	HTMLテキストの作成	makehtml.c

モジュール設計			項目選択画面HTMLテキスト作成処理	
	ID	R S J S	ID	S J S 0 2

ソースファイル名	sj02/main.c
モジュール名	メイン処理
モジュールID	main
処理概要	項目選択画面HTMLテキスト作成メイン処理。
1. インクルードファイルの定義	
<pre>#include <stdio.h> #include <stdlib.h> #include <string.h> #include "fd.h"</pre>	
2. 関数の一覧	
<p>(1) 関数名称：メイン処理関数</p> <pre>int main(argc, argv) int argc; char **argv;</pre> <p><処理概要></p> <ul style="list-style-type: none"> ・引数の数のチェックを行ないエラーメッセージ出力関数を呼び出す。 ・HTMLテキストの作成関数を呼び出す。 	

モジュール設計			項目選択画面HTMLテキスト作成処理	
	ID	RSJS	ID	SJS02

ソースファイル名	sjs02/database.c
モジュール名	項目テーブル検索処理
モジュールID	database
処理概要	項目テーブルから項目一覧を検索して、HTML変換する。
1. インクルードファイルの定義	
<pre>#include <sqlhdr.h> #include <stdio.h> #include <stdlib.h> #include <string.h> #include "fd.h"</pre>	
2. 関数の一覧	
<p>(1) 関数名称：大項目一覧表示処理関数</p> <pre>int sjs_item_list_1(lang, key, type, plant) char *lang; /* 言語 (E:英語, J:日本語) */ char *key; /* 項目コード(大・中・小・データ項目) */ char *type; /* プラント種別(A, B, P) */ char *plant; /* (ALLP, BWRP, PWRP) + プラントコード */</pre> <p><処理概要></p> <ul style="list-style-type: none"> • ORACLEデータベースのオープンを行なう。 • SQLエラー処理の定義を行なう。 • 検索条件の設定する。 • SQLカーソルの準備を行なう。 • SQLカーソルをオープンする。 • カレント行移動行を得る • SQL FETCH文を発行する。 • 中項目一覧表示処理を呼び出す。 • カレント行移動アンカーを出力する。 • SQLカーソルをクローズする。 	

(2) 関数名称 : 中項目一覧表示処理関数

```
int sjs_item_list_2(lang, key, type, plant)
    char *lang; /* 言語 (E:英語, J:日本語) */
    char *key; /* 項目コード(大・中・小・データ項目) */
    char *type; /* プラント種別(A, B, P) */
    char *plant; /* (ALLP, BWRP, PWRP) + プラントコード */
```

〈処理概要〉

- ・SQLエラー処理の定義を行なう。
- ・検索条件の設定する。
- ・SQLカーソルの準備を行なう。
- ・SQLカーソルをオープンする。
- ・カレント行移動行を得る
- ・SQL FETCH文を発行する。
- ・小項目一覧表示処理を呼び出す。
- ・カレント行移動アンカーを出力する。
- ・SQLカーソルをクローズする。

(3) 関数名称 : 小項目一覧表示処理関数

```
int sjs_item_list_3(lang, key, type, plant)
    char *lang; /* 言語 (E:英語, J:日本語) */
    char *key; /* 項目コード(大・中・小・データ項目) */
    char *type; /* プラント種別(A, B, P) */
    char *plant; /* (ALLP, BWRP, PWRP) + プラントコード */
```

〈処理概要〉

- ・SQLエラー処理の定義を行なう。
- ・検索条件の設定する。
- ・SQLカーソルの準備を行なう。
- ・SQLカーソルをオープンする。
- ・カレント行移動行を得る
- ・SQL FETCH文を発行する。
- ・データ項目一覧表示処理を呼び出す。
- ・カレント行移動アンカーを出力する。
- ・SQLカーソルをクローズする。

(4) 関数名称 : データ項目一覧表示処理関数

```
int sjs_item_list_4(lang, key, type, plant)
    char *lang; /* 言語 (E:英語, J:日本語) */
    char *key; /* 項目コード(大・中・小・データ項目) */
    char *type; /* プラント種別(A, B, P) */
    char *plant; /* (ALLP, BWRP, PWRP) + プラントコード */
```

〈処理概要〉

- ・SQLエラー処理の定義を行なう。
- ・検索条件の設定する。
- ・SQLカーソルの準備を行なう。
- ・SQLカーソルをオープンする。
- ・カレント行移動行を得る
- ・SQL FETCH文を発行する。
- ・中項目一覧表示処理を呼び出す。
- ・カレント行移動アンカーを出力する。
- ・SQLカーソルをクローズする。

(5) 関数名称 : プラント型式検索条件の編集処理関数

```
int sjs_make_sqlstmt(item, key, type, plant)
    char *lang; /* 言語 (E:英語, J:日本語) */
    char *key; /* 項目コード(大・中・小・データ項目) */
    char *type; /* プラント種別(A, B, P) */
    char *plant; /* (ALLP, BWRP, PWRP) + プラントコード */
```

<処理概要>

- ・引数で渡されたプラント型式毎の検索SQL文を編集する。

モジュール設計			項目選択画面HTMLテキスト作成処理	
	ID	RSJS	ID	SJS02

ソースファイル名	sjs03/makehtml.c
モジュール名	HTMLテキストの作成
モジュールID	makehtml
処理概要	項目選択画面用HTMLテキストを作成する。
1. インクルードファイルの定義	
<pre>#include <stdio.h> #include <stdlib.h> #include <string.h> #include "fd.h"</pre>	
2. 関数の一覧	
<p>(1) 関数名称 : HTMLテキストの編集関数</p> <pre>int sjs_make_html(lang, type) char *lang; /* 言語 (E:英語, J:日本語) */ char *type; /* プラント種別(A, B, P) */</pre> <p><処理概要></p> <ul style="list-style-type: none"> ・テンプレートHTMLファイル名の編集を行なう。 ・テンプレートHTMLファイルのオープンを行なう。 ・テンプレートファイルの読み込みを行なう。 ・HTMLテキストの編集処理を呼び出す。 ・項目一覧表示処理を呼び出す。 ・テンプレートファイルのクローズを行なう。 	

モジュール一覧			データ表示画面HTMLテキスト作成処理	
	ID	RSJS	ID	SJS03

プログラム	データ表示画面HTMLテキスト作成処理	
プログラムID	SJS03	
モジュール名	モジュール概要	ソースファイル名
main	メイン処理	main.c
database	データの検索処理	database.c
makehtml	HTMLテキストの作成	makehtml.c

モジュール一覧			データ表示画面HTMLテキスト作成処理	
	ID	RSJS	ID	SJS03

ソースファイル名	sjs03/main.c
モジュール名	メイン処理
モジュールID	main
処理概要	データ表示画面HTMLテキスト作成メイン処理。
1. インクルードファイルの定義	
<pre>#include <stdio.h> #include <stdlib.h> #include <string.h> #include "fd.h"</pre>	
2. 関数の一覧	
<p>(1) 関数名称：メイン処理関数</p> <pre>int main(argc, argv) int argc; char **argv;</pre> <p><処理概要></p> <ul style="list-style-type: none"> ・引数の数のチェックを行ないエラーメッセージ出力関数を呼び出す。 ・HTMLテキストの作成関数を呼び出す。 	

モジュール一覧			データ表示画面HTMLテキスト作成処理	
	ID	R S J S	ID	S J S 0 3

ソースファイル名	sjs03/database.c
モジュール名	データテーブル一覧検索処理
モジュールID	database
処理概要	データテーブルからデータ一覧を検索して、HTML変換する。
1. インクルードファイルの定義	
<pre>#include <stdio.h> #include <stdlib.h> #include <string.h> #include <sqlhdr.h> #include "fd.h"</pre>	
2. 関数の一覧	
<p>(1) 関数名称：データ一覧検索処理関数</p> <pre>int sjs_data_list(lang, key, type, plant) char *lang; /* 言語 (E:英語, J:日本語) */ char *key; /* 項目コード(大・中・小・データ項目) */ char *type; /* プラント種別(A, B, P) */ char *plant; /* (ALLP, BWRP, PWRP) + プラントコード */</pre> <p><処理概要></p> <ul style="list-style-type: none"> ・ORACLEデータベースのオープンを行なう。 ・大項目、中項目、小項目検索SQLステートメントの編集を行なう。 ・原子炉型式検索SQLステートメントの編集を行なう。 ・プラントコード検索SQLステートメントの編集を行なう。 ・SQL命令の編集を行なう。 ・SQLカーソルの割り当てを行なう。 ・SQLカーソルのオープンを行なう。 ・データの読み込みを行なう。 ・項目名称を出力する。 ・読み込んだデータをHTMLに編集して出力する。 ・コメント情報の出力処理の呼び出しを行なう。 ・SQLカーソルのクローズを行なう。 	

(2) 関数名称 : 項目名称検索処理関数

```

int sjs_item_name(lang, item, type, key1, key2, key3, key4)
    char *lang; /* 言語 (E:英語, J:日本語) */
    int item /* 項目種別(大・中・小・データ項目) */
    char *type; /* プラント種別(A,B,P) */
    char *key1; /* 項目コード(大) */
    char *key2; /* 項目コード(中) */
    char *key3; /* 項目コード(小) */
    char *key4; /* 項目コード(データ項目) */

```

〈処理概要〉

- ・SQLエラー処理の定義を行なう。
- ・検索条件の設定を行なう。
- ・SQLカーソルの割り当てを行なう。
- ・SQLカーソルのオープンを行なう。
- ・データの読み込みを行なう。
- ・項目名称を編集して出力する。
- ・SQLカーソルのクローズを行なう。

(3) 関数名称 : コメント情報の出力処理関数

```

int sjs_get_comment(lang, key)
    char *lang; /* 言語 (E:英語, J:日本語) */
    char *key; /* 項目コード(大・中・小・データ項目) */

```

〈処理概要〉

- ・SQLエラー処理の定義を行なう。
- ・検索条件の設定を行なう。
- ・データの読み込みを行なう。
- ・コメント情報を編集して出力する。

モジュール一覧			データ表示画面HTMLテキスト作成処理	
	ID	R S J S	ID	S J S 0 3

ソースファイル名	sjs03/makehtml.c
モジュール名	HTMLテキストの作成
モジュールID	makehtml
処理概要	小項目選択画面用HTMLテキストを作成する。
1. インクルードファイルの定義	
<pre>#include <stdio.h> #include <stdlib.h> #include <string.h> #include "fd.h"</pre>	
2. 関数の一覧	
<p>(1) 関数名称 : HTMLテキストの編集関数</p> <pre>int sjs_make_html(lang, type) char *lang; /* 言語 (E:英語, J:日本語) */ char *type; /* プラント種別(A, B, P) */</pre> <p><処理概要></p> <ul style="list-style-type: none"> ・ テンプレートHTMLファイル名の編集を行なう。 ・ テンプレートHTMLファイルのオープンを行なう。 ・ テンプレートファイルの読み込みを行なう。 ・ HTMLテキストの編集処理を呼び出す。 ・ データ項目一覧表示処理を呼び出す。 ・ テンプレートファイルのクローズを行なう。 	

モジュール一覧			データ項目選択画面HTMLテキスト作成処理	
	ID	RSJS	ID	SJS04

プログラム	データ項目選択画面HTMLテキスト作成処理	
プログラムID	SJS04	
モジュール名	モジュール概要	ソースファイル名
main	メイン処理	main.c
search	データの検索処理	search.c
makehtml	HTMLテキストの作成	makehtml.c

モジュール設計			データ項目選択画面HTMLテキスト作成処理	
	ID	RSJS	ID	SJS04

ソースファイル名	sjs04/main.c
モジュール名	メイン処理
モジュールID	main
処理概要	データ項目選択画面HTMLテキスト作成メイン処理。
1. インクルードファイルの定義	
<pre>#include <stdio.h> #include <stdlib.h> #include <string.h> #include "fd.h" #include "rsjs004.h"</pre>	
2. 関数の一覧	
<p>(1) 関数名称 : メイン処理関数</p> <pre>int main(argc, argv) int argc; char **argv;</pre> <p><処理概要></p> <ul style="list-style-type: none"> ・引数の数のチェックを行ないエラーメッセージ出力関数を呼び出す。 ・プラント種別の設定を行なう。 ・検索文字列のデコード処理を行なう。 ・画面作成処理の呼び出しを行なう。 	

モジュール設計			データ項目選択画面HTMLテキスト作成処理	
	ID	R S J S	ID	S J S 0 4

ソースファイル名	sjs04/search.c
モジュール名	データの検索処理
モジュールID	search
処理概要	データ項目一覧を検索する。
1. インクルードファイルの定義	
<pre>#include <stdio.h> #include <stdlib.h> #include <string.h> #include <sqlhdr.h> #include "fd.h"</pre>	
2. 関数の一覧	
<p>(1) 関数名称 : 全文検索処理関数</p> <pre>int sjs_search(lang, type, plant, str) char *lang; /* 言語 (E:英語, J:日本語) */ char *type; /* プラント種別 (B,P) */ char *plant; /* プラントコード or ALLP */ char *str; /* 検索文字列 */</pre> <p><処理概要></p> <ul style="list-style-type: none"> • ORACLEデータベースのオープンを行なう。 • SQLエラー処理の定義を行なう。 • 項目テーブル検索条件の設定を行なう。 • SQLカーソルの準備を行なう。 • SQLカーソルのオープンを行なう。 • SQL FETCH文の発行を行なう。 • データを読み込み項目構造体に格納する。 • SQLカーソルのクローズを行なう。 • データテーブル検索条件の設定を行なう。 • SQLカーソルの準備を行なう。 • SQLカーソルのオープンを行なう。 • SQL FETCH文の発行を行なう。 • データを読み込みデータ構造体に格納する。 • SQLカーソルのクローズを行なう。 • 検索件数が0の時、「データ中に文字列が見つかりません。」のメッセージを出力する。 • 検索データをソートする。 • 項目一覧作成処理を呼び出す。 • 項目一覧を出力する。 	

(2) 関数名称 : 項目検索条件編集処理関数

```
int sjs_makeitem_sql(lang, type, plant, str)
    char *lang; /* 言語 (E:英語, J:日本語) */
    char *type; /* プラント種別(B,P) */
    char *plant; /* プラントコード or ALLP */
    char *str; /* 検索文字列 */
```

<処理概要>

- ・項目検索用SELECT文を作成する。

(3) 関数名称 : データ検索条件編集処理関数

```
int sjs_makeitem_sql(lang, type, plant, str)
    char *lang; /* 言語 (E:英語, J:日本語) */
    char *type; /* プラント種別(B,P) */
    char *plant; /* プラントコード or ALLP */
    char *str; /* 検索文字列 */
```

<処理概要>

- ・データ検索用SELECT文を作成する。

(4) 関数名称 : 全文検索条件編集処理関数

```
char *sjs_make_findstr(lang, str)
    char *lang; /* 言語 (E:英語, J:日本語) */
    char *str; /* 検索文字列 */
```

<処理概要>

- ・データ検索用SELECT文を作成する。

(5) 関数名称 : 全文 (項目) 検索条件編集処理関数

```
char *sjs_make_finditemstr(lang, str)
    char *lang; /* 言語 (E:英語, J:日本語) */
    char *str; /* 検索文字列 */
```

<処理概要>

- ・全文 (項目) 検索用SELECT文を作成する。

(6) 関数名称 : 全文 (データ) 検索条件編集処理関数

```
char *sjs_make_finddatastr(lang, str)
    char *lang; /* 言語 (E:英語, J:日本語) */
    char *str; /* 検索文字列 */
```

<処理概要>

- ・全文 (データ) 検索用SELECT文を作成する。

(7) 関数名称 : 全文字列検索条件編集処理関数

```
char *sjs_set_likes(lang, itemnam, str)
    char *lang; /* 言語 (E:英語, J:日本語) */
    char *itemnam; /* 項目名称 */
    char *str; /* 検索文字列 */
```

<処理概要>

- ・全文字列検索用SELECT文を作成する。

(8) 関数名称 : 文字列検索条件編集処理関数

```
char *sjs_set_like(lang, itemnam, str)
    char *lang;      /* 言語 (E:英語, J:日本語) */
    char *itemnam;   /* 項目名称 */
    char *str;       /* 検索文字列 */
```

<処理概要>

- ・文字列検索用SELECT文を作成する。

(10) 関数名称 : 項目構造体ソート処理関数

```
int item_sort(item1, item2)
    ItemList *item1; /* 全項目一覧ポインタ */
    ItemList *item2; /* 全項目一覧ポインタ + 1 */
```

<処理概要>

- ・引数で渡された文字列を比較し結果を返す。

(11) 関数名称 : 項目名称取得処理関数

```
int sjs_search_item(itemlist)
    ItemList *itemlist; /* 全項目一覧ポインタ */
```

<処理概要>

- ・SQLエラー処理の定義を行なう。
- ・項目テーブル検索条件の設定を行なう。
- ・SQLカーソルの準備を行なう。
- ・SQLカーソルのオープンを行なう。
- ・項目名称を読み込み項目一覧構造体にセットする。
- ・SQLカーソルのクローズを行なう。

(12) 関数名称 : 項目名称取得検索条件編集処理関数

```
int sjs_makeitemnm_sql(itemlist)
    ItemList *itemlist; /* 全項目一覧ポインタ */
```

<処理概要>

- ・項目名称取得検索用SELECT文を作成する。

(13) 関数名称 : 項目名称取得検索条件編集処理関数

```
int sjs_item_list(lang, itemlist)
    char *lang;      /* 言語 (E:英語, J:日本語) */
    ItemList *itemlist; /* 全項目一覧ポインタ */
```

<処理概要>

- ・大項目用HTMLを出力する。
- ・中項目用HTMLを出力する。
- ・小項目用HTMLを出力する。
- ・データ項目用HTMLを出力する。

モジュール設計			データ項目選択画面HTMLテキスト作成処理	
	ID	RSJS	ID	SJS04

ソースファイル名	sjs04/makehtml.c
モジュール名	HTMLテキストの作成
モジュールID	makehtml
処理概要	データ項目選択画面用HTMLテキストを作成する。
1. インクルードファイルの定義	
<pre>#include <stdio.h> #include <stdlib.h> #include <string.h> #include "fd.h"</pre>	
2. 関数の一覧	
<p>(1) 関数名称 : HTMLテキストの編集関数</p> <pre>int sjs_make_html(lang, type, plant, str) char *lang; /* 言語 (E:英語, J:日本語) */ char *type; /* プラント種別 (A, B, P) */ char *plant; /* (ALLP, BWRP, PWRP) + プラントコード */ char *str; /* 検索文字列 */</pre> <p><処理概要></p> <ul style="list-style-type: none"> ・ テンプレートHTMLファイル名の編集を行なう。 ・ テンプレートHTMLファイルのオープンを行なう。 ・ テンプレートファイルの読み込みを行なう。 ・ 全文検索項目一覧表示処理を呼び出す。 ・ テンプレートファイルのクローズを行なう。 	

モジュール一覧			属性フラグセット処理	
	ID	RSJS	ID	SETFLG

プログラム	属性フラグセット処理	
プログラムID	SETFLG	
モジュール名	モジュール概要	ソースファイル名
main	メイン処理	main.c
setone	単一フラグセット処理	setone.c
setitem	項目フラグセット処理	setitem.c
setdata	データフラグセット処理	setdata.c

モジュール設計			属性フラグセット処理	
	ID	RSJS	ID	SETFLG

ソースファイル名	setflg/main.c
モジュール名	メイン処理
モジュールID	main
処理概要	引数をチェックし各フラグセット処理を呼び出す。
1. インクルードファイルの定義	
<pre>#include <stdio.h> #include <stdlib.h> #include <string.h> #include "fd.h"</pre>	
2. 関数の一覧	
<p>(1) 関数名称 : メイン処理関数</p> <pre>int main(argc, argv) int argc; char **argv;</pre> <p><処理概要></p> <ul style="list-style-type: none"> 引数の数により以下の処理を行なう。 (引数が無い時) <ul style="list-style-type: none"> 項目テーブルの属性フラグをセットする。 データテーブルの属性フラグをセットする。 (引数が3~6の時) <ul style="list-style-type: none"> 引数チェック処理を呼び出す。 単一フラグセット処理を呼び出す。 (上記以外の時) <ul style="list-style-type: none"> エラーメッセージを出力する。 <p>(2) 関数名称 : 引数チェック処理</p> <pre>check_argv(argc, argv) int argc; char **argv;</pre> <p><処理概要></p> <ul style="list-style-type: none"> 処理フラグをチェックする。 プラントタイプをチェックする。 項目コードをチェックする。 上記エラーの場合にはエラーメッセージを出力しエラーリターンする。 	

モジュール設計			属性フラグセット処理	
	ID	RSJS	ID	SETFLG

ソースファイル名	setflg/setone.c
モジュール名	単一フラグセット処理
モジュールID	setone
処理概要	一項目の項目とデータにフラグをセットする。
<pre>#include <stdio.h> #include <stdlib.h> #include <string.h> #include "fd.h"</pre>	
2. 関数の一覧	
<p>(1) 関数名称 : 単一フラグセット処理関数</p> <pre>int setone(argc, argv) int argc; char **argv;</pre> <p><処理概要></p> <ul style="list-style-type: none"> ・ORACLEデータベースのオープンを行なう。 ・SQLエラー処理の定義を行なう。 ・項目更新SQLステートメントの編集を行なう。 ・動的SQLステートメントの編集を行なう。 ・項目更新SQLステートメントを実行する。 ・データ更新SQLステートメントの編集を行なう。 ・動的SQLステートメントの編集を行なう。 ・データ更新SQLステートメントを実行する。 <p>(2) 関数名称 : SQLエラー処理関数</p> <pre>void put_sql_error()</pre>	

モジュール設計			属性フラグセット処理	
	ID	RSJS	ID	SETFLG

ソースファイル名	setflg/setitem.c
モジュール名	項目フラグセット処理
モジュールID	setitem
処理概要	数値データ以外の項目に対してフラグをセットする。
1. インクルードファイルの定義	
<pre>#include <stdio.h> #include <stdlib.h> #include <string.h> #include "fd.h"</pre>	
2. 関数の一覧	
<p>(1) 関数名称：項目フラグセット処理関数 int setitem()</p> <p><処理概要></p> <ul style="list-style-type: none"> ・ORACLEデータベースのオープンを行なう。 ・SQLエラー処理の定義を行なう。 ・項目更新SQLステートメントの編集を行なう。 ・動的SQLステートメントの編集を行なう。 ・項目更新SQLステートメントを実行する。 	

モジュール設計			属性フラグセット処理	
	ID	RSJS	ID	SETFLG

ソースファイル名	setflg/setdata.c
モジュール名	データフラグセット処理
モジュールID	setdata
処理概要	数値データ以外のデータに対してフラグをセットする。
1. インクルードファイルの定義	
<pre>#include <stdio.h> #include <stdlib.h> #include <string.h> #include "fd.h"</pre>	
2. 関数の一覧	
<p>(1) 関数名称 : データフラグセット処理関数 int setdata()</p> <p><処理概要></p> <ul style="list-style-type: none"> • ORACLEデータベースのオープンを行なう。 • SQLエラー処理の定義を行なう。 • データ更新SQLステートメントの編集を行なう。 • 動的SQLステートメントの編集を行なう。 • データ更新SQLステートメントを実行する。 	

モジュール一覧			共通処理	
	ID	RSJS	ID	SJSLIB

プログラム	共通処理	
プログラムID	SJSLIB	
モジュール名	モジュール概要	ソースファイル名
filtlib	文字変換フィルター処理	filtlib.c
fdlib	共通ライブラリ処理	fdlib.c
Kcatoi	数値文字列変換処理	Kcatoi.c
kceuctoa	文字列変換処理	kceuctoaup.c
kcsblnksup	文字列空白圧縮処理	kcsblnksup.c
kcsstrreplace	文字列置換処理	kcsstrreplace.c
orasql	データベースオープン処理	orasql.c

モジュール設計			共通処理	
	ID	RSJS	ID	SJSLIB

ソースファイル名	lib/filtlib.c
モジュール名	文字変換フィルター処理
モジュールID	filtlib
処理概要	文字変換用フィルター関数
1. インクルードファイルの定義	
<pre>#include <stdio.h> #include <stdlib.h> #include <string.h></pre>	
2. 関数の一覧	
<p>(1) 関数名称：NK F (→EUCコード) フィルター関数</p> <pre>int code2euc(src, dst) char *src; /* 変換元文字列 */ char *dst; /* EUC文字列 */</pre> <p><処理概要></p> <ul style="list-style-type: none"> ・デコード処理を呼び出す。 ・NK Fコマンドを編集する。 ・双方向パイプをオープンし変換したEUC文字列を取得する。 	
<p>(2) 関数名称：NK F (→EUCコード) フィルター関数</p> <pre>int codetoeuc(src, dst) char *src; /* 変換元文字列 */ char *dst; /* EUC文字列 */</pre> <p><処理概要></p> <ul style="list-style-type: none"> ・デコード処理を呼び出す。 ・NK Fコマンドを編集する。 ・双方向パイプをオープンし変換したEUC文字列を取得する。 	
<p>(3) 関数名称：ACK (→EUCコード) フィルター関数</p> <pre>int code2euc2(src, dst) char *src; /* 変換元文字列 */ char *dst; /* EUC文字列 */</pre> <p><処理概要></p> <ul style="list-style-type: none"> ・デコード処理を呼び出す。 ・ACKコマンドを編集する。 ・双方向パイプをオープンし変換したEUC文字列を取得する。 	

(5) 関数名称 : エンコード文字列デコード関数

```
int strdecode(src, dst)
    char *src; /* エンコード文字列 */
    char *dst; /* デコード文字列 */
```

(6) 関数名称 : 双方向パイプ処理関数

```
int biwaypopen(command, fd_in, fd_out)
    char *command; /* コマンド文字列 */
    int *fd_in; /* 入力ファイルポインタ */
    int *fd_out; /* 出力ファイルポインタ */
```

<処理概要>

- 入力用パイプをオープンする。
- 出力用パイプをオープンする。
- コマンド文字列を入力用パイプに書き出す。
- 入力用パイプからコマンドの返答を取得する。

モジュール設計			共通処理	
	ID	RSJS	ID	SJSLIB

ソースファイル名	lib/fdlib.c
モジュール名	共通ライブラリ処理
モジュールID	fdlib
処理概要	共通ライブラリ。
1. インクルードファイルの定義	
<pre>#include <stdio.h> #include <stdlib.h> #include <string.h> #include <varargs.h> #include "fd.h"</pre>	
2. 関数の一覧	
<p>(1) 関数名称 : 転送方式の検査関数 int Fd_check_method()</p> <p><処理概要> ・転送方式の検査を行なう。</p> <p>(2) 関数名称 : 転送データの読み込み関数 char * Fd_get_data(method) int method;</p> <p><処理概要> ・データ長の読み込みを行なう。</p> <p>(3) 関数名称 : 転送データの項目読み込み関数 int Fd_get_entry(data, key, value) char *data; char **key; char **value;</p> <p><処理概要> ・入力データの複写を行なう。 ・データの編集を行なう。</p> <p>(4) 関数名称 : 入力キーの検関数 int Fd_key_check(text) char *text;</p>	

(5) 関数名称 : htmlのCGI格納ディレクトリとhtmlファイル格納ディレクトリの編集関数
 char * Fd_html_modify(text)
 char *text;

(6) 関数名称 : エラーhtmlテキストの作成関数
 int Fd_html_error(va_alist)

(7) 関数名称 :
 Fd_html_error(va_alist) va_dcl

<処理概要>

- ・メッセージの編集を行なう。
- ・テンプレートHTMLファイル名の編集を行なう。
- ・テンプレートHTMLファイルのオープンを行なう。
- ・テンプレートファイルの読み込みを行なう。
- ・メッセージの出力を行なう。
- ・HTMLテキストを出力する。
- ・テンプレートファイルのクローズを行なう。

(8) 関数名称 : デバッグメッセージ出力関数
 int DEBUG_MSG(va_alist)

(9) 関数名称 :
 DEBUG_MSG(va_alist) va_dcl

(10) 関数名称 : デバッグメッセージ出力関数
 int MSG(va_alist)

(11) 関数名称 :
 MSG(va_alist) va_dcl

(12) 関数名称 : デバッグメッセージ出力関数
 int ERR_MSG(va_alist)

(13) 関数名称 :
 ERR_MSG(va_alist) va_dcl

(14) 関数名称 : メニュー画面htmlテキストの作成関数
 int Fd_make_menu()

<処理概要>

- ・テンプレートHTMLファイル名の編集を行なう。
- ・テンプレートHTMLファイルのオープンを行なう。
- ・テンプレートファイルの読み込みを行なう。
- ・HTMLテキストを出力する。
- ・テンプレートファイルのクローズを行なう。

(15) 関数名称 : ascii空白を漢字空白に変換する関数
 int Fd_edit_item(text)
 char *text;

モジュール設計			共通処理	
	ID	RSJS	ID	SJSLIB

ソースファイル名	lib/Kcatoi.c
モジュール名	数値文字列変換処理
モジュールID	Kcatoi
処理概要	数値文字列の変換を行なう。
1. インクルードファイルの定義	
<pre>#include <stdlib.h></pre>	
2. 関数の一覧	
<p>(1) 関数名称 : 数値文字列のint型変換関数 int Kcatoi(string, length) char *string; int length;</p> <p>(2) 関数名称 : int型文字列変換関数 int Kcitoa(string, value, length) char *string; int value; int length;</p>	

モジュール設計			共通処理	
	ID	RSJS	ID	SJSLIB

ソースファイル名	lib/kceuctoa.c
モジュール名	文字列変換処理
モジュールID	kceuctoa
処理概要	EUC漢字コードからASCIIに変換する。
1. インクルードファイルの定義	
#include <stdio.h>	
2. 関数の一覧	
<p>(1) 関数名称 :</p> <pre>char * kceuctoa(in) char *in;</pre> <p>(2) 関数名称 : EUC漢字コードからASCIIに変換する関数</p> <pre>char * kceuctoan(in, length) char *in; int length;</pre>	

モジュール設計			共通処理	
	ID	RSJS	ID	SJSLIB

ソースファイル名	lib/kcsblnksup.c
モジュール名	文字列ブランク圧縮処理
モジュールID	kcsblnksup
処理概要	文字列のブランクを圧縮する。
1. インクルードファイルの定義	
#include <stdio.h>	
2. 関数の一覧	
<p>(1) 関数名称 : 文字列右側のブランクを圧縮する関数 char * kcsblnksup(text, count) unsigned char *text; int count;</p> <p>(2) 関数名称 : 文字列右側のブランクを圧縮する関数 char * kcsblank(text, count) unsigned char *text; int count;</p> <p>(3) 関数名称 : 文字列右側の指定された文字を圧縮する関数 char * kcscharsup(text, chars, count) char *text; char *chars; int count;</p> <p>(4) 関数名称 : 文字列右側のブランクを圧縮する関数 char * kcsblankright(text, count) unsigned char *text; int count;</p> <p>(5) 関数名称 : 文字列左側のブランクを圧縮する関数 char * kcsblankleft(text, count) unsigned char *text; int count;</p> <p>(6) 関数名称 : 文字列のブランクを全て圧縮する関数 char * kcsblankall(text, count) unsigned char *text; int count;</p>	

(7) 関数名称 : 文字列から連続する空白を圧縮する関数

```
char * kcsblankedit(text, count)
    unsigned char *text;
    int count;
```

(8) 関数名称 : 文字列から指定された文字を削除する関数

```
int kcsstringdel(text, delete)
    char *text;
    char *delete;
```

(9) 関数名称 : 文字列から指定された文字を削除する関数

```
char * kcsnewstringdel(text, delete)
    char *text;
    char *delete;
```

モジュール設計			共通処理	
	ID	RSJS	ID	SJSLIB

ソースファイル名	lib/kcsstrreplace.c
モジュール名	文字列置換処理
モジュールID	kcsstrreplace
処理概要	文字列を指定文字列に置き換える。
1. インクルードファイルの定義	
<pre>#include <string.h></pre>	
2. 関数の一覧	
<p>(1) 関数名称 :</p> <pre>char * kcstrrep(text, target, replace) char *text; char *target; char *replace;</pre>	

モジュール設計			共通処理	
	ID	RSJS	ID	SJSLIB

ソースファイル名	lib/orasql.c
モジュール名	データベースオープン処理
モジュールID	orasql
処理概要	データベースをオープンする。
1. インクルードファイルの定義	
<pre>#include <sqlhdr.h> #include <stdio.h> #include <stdlib.h> #include <string.h> #include <fd.h></pre>	
2. 関数の一覧	
<p>(1) 関数名称 : データベースオープン処理関数 int db_open()</p> <p>(2) 関数名称 : SQLエラー処理関数 void sql_error()</p>	

This is a blank page.

国際単位系 (SI) と換算表

表1 SI基本単位および補助単位

量	名称	記号
長さ	メートル	m
質量	キログラム	kg
時間	秒	s
電流	アンペア	A
熱力学温度	ケルビン	K
物質質量	モル	mol
光度	カンデラ	cd
平面角	ラジアン	rad
立体角	ステラジアン	sr

表3 固有の名称をもつ SI組立単位

量	名称	記号	他のSI単位による表現
周波数	ヘルツ	Hz	s ⁻¹
力	ニュートン	N	m·kg/s ²
圧力, 応力	パスカル	Pa	N/m ²
エネルギー, 仕事, 熱量	ジュール	J	N·m
工率, 放射束	ワット	W	J/s
電気量, 電荷	クーロン	C	A·s
電位, 電圧, 起電力	ボルト	V	W/A
静電容量	ファラド	F	C/V
電気抵抗	オーム	Ω	V/A
コンダクタンス	ジーメンズ	S	A/V
磁束	ウェーバ	Wb	V·s
磁束密度	テスラ	T	Wb/m ²
インダクタンス	ヘンリー	H	Wb/A
セルシウス温度	セルシウス度	°C	
光束度	ルーメン	lm	cd·sr
照射度	ルクス	lx	lm/m ²
放射能	ベクレル	Bq	s ⁻¹
吸収線量	グレイ	Gy	J/kg
線量当量	シーベルト	Sv	J/kg

表2 SIと併用される単位

名称	記号
分, 時, 日	min, h, d
度, 分, 秒	°, ', "
リットル	l, L
トン	t
電子ボルト	eV
原子質量単位	u

1 eV = 1.60218 × 10⁻¹⁹ J

1 u = 1.66054 × 10⁻²⁷ kg

表4 SIと共に暫定的に維持される単位

名称	記号
オングストローム	Å
バ	b
バール	bar
ガロン	Gal
キュリー	Ci
レントゲン	R
ラド	rad
レム	rem

1 Å = 0.1 nm = 10⁻¹⁰ m

1 b = 100 fm² = 10⁻²⁸ m²

1 bar = 0.1 MPa = 10⁵ Pa

1 Gal = 1 cm/s² = 10⁻² m/s²

1 Ci = 3.7 × 10¹⁰ Bq

1 R = 2.58 × 10⁻⁴ C/kg

1 rad = 1 cGy = 10⁻² Gy

1 rem = 1 cSv = 10⁻² Sv

表5 SI接頭語

倍数	接頭語	記号
10 ¹⁸	エクサ	E
10 ¹⁵	ペタ	P
10 ¹²	テラ	T
10 ⁹	ギガ	G
10 ⁶	メガ	M
10 ³	キロ	k
10 ²	ヘクト	h
10 ¹	デカ	da
10 ⁻¹	デシ	d
10 ⁻²	センチ	c
10 ⁻³	ミリ	m
10 ⁻⁶	マイクロ	μ
10 ⁻⁹	ナノ	n
10 ⁻¹²	ピコ	p
10 ⁻¹⁵	フェムト	f
10 ⁻¹⁸	アト	a

(注)

- 表1-5は「国際単位系」第5版, 国際度量衡局 1985年刊行による。ただし, 1 eV および 1 uの値は CODATAの1986年推奨値によった。
- 表4には海里, ノット, アール, ヘクトールも含まれているが日常の単位なのでここでは省略した。
- barは, JISでは流体の圧力を表わす場合に限り表2のカテゴリーに分類されている。
- EC閣僚理事会指令では bar, barn および「血圧の単位」mmHgを表2のカテゴリーに入れている。

換算表

力	N (=10 ⁵ dyn)	kgf	lbf
	1	0.101972	0.224809
	9.80665	1	2.20462
	4.44822	0.453592	1

粘度 1 Pa·s (=1 N·s/m²) = 10 P (ポアズ) (g/(cm·s))

動粘度 1 m²/s = 10⁴ St (ストークス) (cm²/s)

圧	MPa (=10 bar)	kgf/cm ²	atm	mmHg (Torr)	lbf/in ² (psi)
	1	10.1972	9.86923	7.50062 × 10 ³	145.038
力	0.0980665	1	0.967841	735.559	14.2233
	0.101325	1.03323	1	760	14.6959
	1.33322 × 10 ⁻⁴	1.35951 × 10 ⁻³	1.31579 × 10 ⁻³	1	1.93368 × 10 ⁻²
	6.89476 × 10 ⁻³	7.03070 × 10 ⁻²	6.80460 × 10 ⁻²	51.7149	1

エネルギー・仕事・熱量	J (=10 ⁷ erg)	kgf·m	kW·h	cal (計量法)	Btu	ft·lbf	eV
	1	0.101972	2.77778 × 10 ⁻⁷	0.238889	9.47813 × 10 ⁻⁴	0.737562	6.24150 × 10 ¹⁸
	9.80665	1	2.72407 × 10 ⁻⁶	2.34270	9.29487 × 10 ⁻³	7.23301	6.12082 × 10 ¹⁹
	3.6 × 10 ⁶	3.67098 × 10 ⁵	1	8.59999 × 10 ⁵	3412.13	2.65522 × 10 ⁶	2.24694 × 10 ²⁵
	4.18605	0.426858	1.16279 × 10 ⁻⁶	1	3.96759 × 10 ⁻³	3.08747	2.61272 × 10 ¹⁹
	1055.06	107.586	2.93072 × 10 ⁻⁴	252.042	1	778.172	6.58515 × 10 ²¹
	1.35582	0.138255	3.76616 × 10 ⁻⁷	0.323890	1.28506 × 10 ⁻³	1	8.46233 × 10 ¹⁸
	1.60218 × 10 ⁻¹⁹	1.63377 × 10 ⁻²⁰	4.45050 × 10 ⁻²⁶	3.82743 × 10 ⁻²⁰	1.51857 × 10 ⁻²²	1.18171 × 10 ⁻¹⁹	1

- 1 cal = 4.18605 J (計量法)
 = 4.184 J (熱化学)
 = 4.1855 J (15 °C)
 = 4.1868 J (国際蒸気表)
- 仕事率 1 PS (仏馬力)
 = 75 kgf·m/s
 = 735.499 W

放射能	Bq	Ci
	1	2.70270 × 10 ⁻¹¹
	3.7 × 10 ¹⁰	1

吸収線量	Gy	rad
	1	100
	0.01	1

照射線量	C/kg	R
	1	3876
	2.58 × 10 ⁻⁴	1

線量当量	Sv	rem
	1	100
	0.01	1

