

(財)日本建設情報総合センター研究助成事業

事故・災害時の初動体制を想定した
情報共有プラットフォームの提案に関する報告書

平成 15 年 9 月

目次

1	序	3
2	社会基盤構造物マネジメントシステム	4
2.1	社会基盤の維持管理	4
2.2	システムデザイン	5
2.3	システムの統合	6
3	情報共有プラットフォーム	8
3.1	はじめに	8
3.2	橋梁維持管理	8
3.3	橋梁マネジメントシステム	9
3.4	検査員用システムのプロトタイプ	9
3.5	まとめ	10
4	データマイニングによる知識獲得	12
4.1	はじめに	12
4.2	決定木によるデータマイニング	12
5	BridgeMLに基づくマルチモーダルアクセス可能な橋梁マネジメントアーキテクチャ	18
5.1	はじめに	18
5.2	情報共有型 BMS アーキテクチャ	18
5.3	Bridge Markup Language	29
5.4	まとめ	42
6	分散型情報技術による社会基盤施設防災・保全現場のインテリジェント化	44
6.1	はじめに	44
6.2	簡易加速度計測システム(RIMS)	46
6.3	RIMS の前提技術	48
6.4	通信技術概論	53
6.5	RIMS 開発の実際	59
6.6	評価試験	64
6.7	まとめ	67

1 序

構造物管理者にとって地震等の大規模災害時には、管理する構造物の状況を把握し、迅速に対応することが求められる。道路・鉄道などの交通ネットワークについては、規制による安全の確保だけでなく、人命救助や災害復旧に必要な緊急車両のためのルートの確保が重要となる。

現状では初動体制時の大きな問題として、現場からの情報が膨大でかつ整理されておらず、意思決定に必要な情報を抽出するために多くの時間とコストを要する。また、情報が分散しており、それらを統合的に扱うことができない。結果としてこれらの問題が意思決定の遅れにつながり、災害復旧プロジェクト全体に大きな影響を及ぼす。

本研究では、災害現場からの情報を集約的に管理する手法を提案する。本手法では、現場からの情報を収集するとともに情報を自動的にスクリーニングし、構造物管理者にとって有用な情報に変換する。また、各連携機関との情報共有を可能にし迅速な意思決定を実現する。

本報告書では、第 2 章で平常時の保全や緊急時の危機管理に関しての情報を取り扱う社会基盤構造物マネジメントシステムのアーキテクチャについて述べる。第 3 章では、災害現場から情報を効率的に収集し、また、現場での意志決定プロセスを支援する情報プラットフォームについて述べる。第 4 章では膨大なデータから有用な知識を抽出するデータマイニング技術により、橋梁の維持管理記録から有効な知識が抽出できることを確認した。第 5 章では具体的に橋梁の維持管理を取り扱う情報基盤 BrideML についてそのアーキテクチャについて説明し、プロトタイプ構築を行った。第 6 章では道路構造物の附属構造物を対象にモニタリングシステムを構築し、災害時の情報収集や異常の検知に有効であることを示した。

社会基盤構造物マネジメントシステムは、データベース、アプリケーションおよび Web インターフェースから構成される。データベースに関しては、構造物諸元・履歴等のテキスト情報に加えて、CAD、GIS、モニタリングシステムと連携させてこれらをデータソースとして利用することを検討している。

次に社会基盤施設の維持管理業務において複数の参画者またはグループが効率的に情報交換するためのプラットフォームについて検討を行った。複数のクライアント環境を想定したプロトタイプとして、PDA や PC からアクセス可能なシステムを構築した。

社会基盤のメンテナンスに関する情報の再利用・有効活用を行うため、データベースから知識を抽出するデータマイニングを行った。その結果、いくつかのルールが生成され、その有効性が確認された。

橋梁の維持管理に関する共通の言語を策定し、複数の事業者・組織間での情報交換を促進するためのデータフォーマットとして BridgeML を提案し、そのドラフトを作成した。またドラフトによるサンプルアプリケーションを作成し、その有効性を検証した。

簡易型加速度計測システム RIMS (Remote Intelligent Monitoring System)を開発し、超軽量アプライアンスによるモニタリング情報取得に成功した。このシステムは従来のモニタリング手法よりも施工面・通信コスト面で有利であり、今後の社会基盤維持活動のインフラストラクチャとしての将来性を期待できるものであると結論づけられる。

2 社会基盤構造物マネジメントシステム

2.1 社会基盤の維持管理

維持管理業務の一般的なプロセスを図 2.1 に示す。まず、既設構造物において定期的な検査またはモニタリングが行われ現状を把握するための基礎データが取得される。これらの情報をもとに健全度や耐久性などの現況把握を行い、また必要に応じて劣化等の将来予測を行う。その後、当該構造物に関して補修・補強の要否や監視項目といった要求項目をまとめる。対象とする構造物が交通ネットワークに属するように他の構造物と併せて管理・運用する必要がある場合、ボトムアップ方式で構造物群として要求項目をまとめることになる。これら要求項目は、構造物管理者側の制約によりトップダウン方式に最適化が行われる。構造物群、個別構造物の順に要求項目の最適化が行われ最終的に構造物の維持管理計画が策定される。

維持管理業務においてはそれぞれのプロセスに置いて現場検査員、橋梁技術者、構造物管理者の意志決定・判断が行われている。構造物マネジメントシステムはそれら意志決定・判断を支援するアプリケーションと維持管理に必要なとされる情報を保存・管理するためのデータベースにより構成され、ネットワークを通じてさまざまなユーザからのアクセスを可能にする。

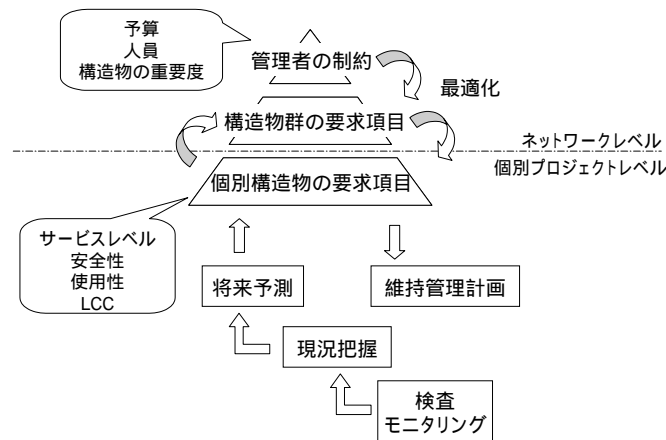


図 2.1 社会基盤の維持管理プロセス

本研究では図 2.2 に示すような社会基盤構造物マネジメントシステム Civil Asset Management Portal (CAMP) を提案する。CAMP の構成要素は、データベース、アプリケーションおよび Web インターフェースである。アプリケーションは維持管理業務を支援するためのツールを提供し、またユーザがデータの追加・削除・更新等の操作を行うデータ管理機能を含む。Web サーバはシステムをネットワークを通じて利用するためのインターフェースを提供する。データベースに関しては、構造物諸元・履歴等のテキスト情報に加えて、CAD、GIS、モニタリングシステムと連携させてこれらをデータソースとして利用することを検討している。

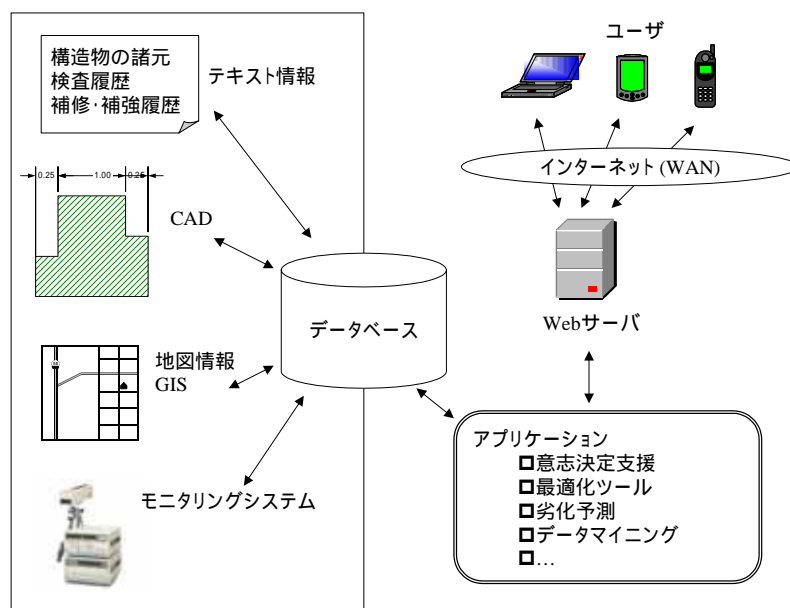


図 2.2 Civil Asset Management Portal

2.2 システムデザイン

CAMP では、Web アプリケーションベースのシステムを構築する。Web アプリケーションは図 2.3 のような多層型分散アーキテクチャを採用する。このようなアーキテクチャでは、システムを複数のコンポーネントに分割し、それぞれのインターフェースを定義することにより、個々のコンポーネントを独立して開発することができる。そのため、システムの使用性、拡張性、信頼性、メンテナンス性の向上を図ることができる。

クライアント層は、クライアント機器、Web ブラウザおよびプラグインソフトウェアで構成される。ユーザはクライアント層を通じてサーバにアクセスする。クライアント層は、ブラウザを主体とするシン・クライアントとプラグインやスクリプト言語を用いるリッチ・クライアントに分類できる。シン・クライアントはリッチ・クライアントに比べて、機能は HTML ベースに制限されるが、機器やブラウザへの要求項目は少なく、さまざまな機器・ブラウザで動作する汎用的なアプリケーションを構築するのに適している。それに対してリッチ・クライアントでは、プラグインを使用するため、ある程度クライアント機器やブラウザの種類が制限されるが、表現力の高いアプリケーションを構築できる。また、PC だけでなく、PDA (Personal Digital Assistant) や Web ブラウズ機能を持つ携帯電話をクライアント層に用いることが可能である。

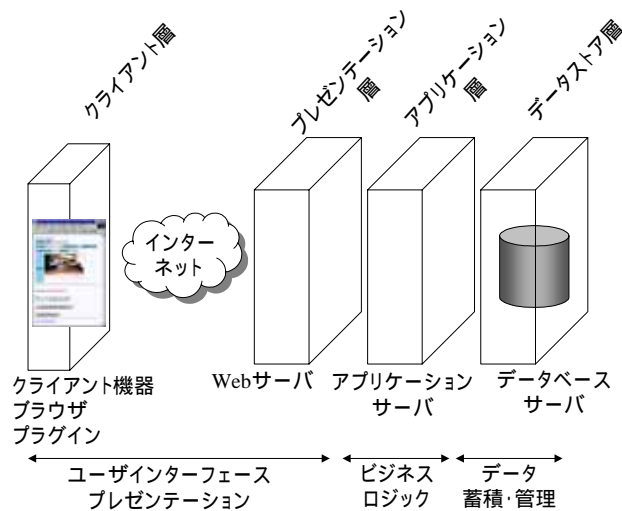


図 2.3 多層モデルによる Web アプリケーションの構成

サーバ側のアプリケーションは、プレゼンテーション層、アプリケーション層、データストア層から構成される。プレゼンテーション層は、サーバ側のクライアント層に対するインターフェースであり、クライアントからのリクエストを受け取り、次に示すアプリケーション層にリクエストを渡し、アプリケーション層からのレスポンスをクライアント層に対して、一般的には HTML のプレゼンテーションコードを動的に生成する。Web アプリケーションではプレゼンテーション層が認証、セッション管理、パーソナライズ機能を担当する。認証は、ユーザを特定するプロセスであり、通常はユーザ ID とパスワードを使用する。セッション管理は、ユーザがログインしてからログオフするまで、情報を継続的に保持するための機能であり、情報をクライアント側に持たせる方法とサーバ側で管理する方法がある。また、パーソナライズ機能では、ユーザのプロファイルや過去の履歴に応じてカスタマイズされたユーザインターフェースを提供する。

アプリケーション層では、アプリケーションで用いるロジックを実装し、ユーザのリクエストに応じて処理を実行する。また、データストアに対してデータの保存、操作、検索、分析を行う。

データストア層では、データを格納し、データを管理・操作するための機能を提供する。通常データストア層には、リレーショナルデータベースを用いる。

2.3 システムの統合

社会基盤構造物マネジメントシステムは、維持管理に関する情報を統合的に管理するためのプラットフォームを提供し、様々なシステムとの連携を考慮する必要がある。ここでは社会基盤構造物マネジメントシステムに統合すべき既存のシステムについて、その特徴と構造物の維持管理分野との関連について述べる。

2.3.1 GIS

地理情報システム(Geographic Information System: GIS)は、地理的位置と位置に付随する情報を管理するもので、それらの情報に関する加工・分析・可視化ツールを提供し、意志決定を支援するシステムである。GIS は多くの公共機関や企業に導入されており、様々な用途に利用されている。地方自治体では GIS を用いて、都市計画のための土地利用図や、

ハザードマップなど災害時の計画図を作成・管理している。また、電力・ガス会社では、送電線や配管などのネットワークを管理するために GIS を利用している。

社会基盤構造物の維持管理においても GIS による位置情報と属性情報の管理が重要である。橋梁やトンネルなどの交通ネットワークを構成する構造物は、個々の構造物の維持管理とともにネットワーク機能に着目した維持管理が必要である。また 盛土など面的な広がりを持つ構造物は、位置情報とその位置における属性情報をあわせて管理する必要がある。

2.3.2 CAD

CAD(Computer Aided Design)システムは、構造物の設計図を電子情報として作成・管理する。維持管理においては、変状・損傷箇所の特定や、補修・補強箇所の選定、大規模改修において構造物の設計図を参照する必要がある。このように構造物の維持管理において、部材・部位・部品等の位置情報を効率的に管理する必要があり、CAD システムとの連携が重要となる。

2.3.3 モニタリングシステム

モニタリングシステムは、継続的に構造物を監視し、現況を把握するために有用である。しかし、通常モニタリングシステムは同種のセンサ群などで完結している場合が多く、構造物全体や複数の構造物を統合的に管理することができない。また、モニタリングシステムはマネジメントシステムと独立しており、構造物管理者の意志決定が迅速に行われない。モニタリングシステムとマネジメントシステムの統合により、構造物の現状把握が迅速になり効率的な意志決定に繋がる。特に自然災害などの緊急時にはモニタリングシステムからの情報が意志決定に重要な役割を果たす。

3 情報共有プラットフォーム

3.1 はじめに

社会基盤施設の建設および維持管理業務は、施主、設計、施工、検査、財務管理など多種多様な専門技術・知識を持った職種により構成されている。プロジェクトの複雑化により参画する人員の数や職種の数は増大する傾向にある。特に、大規模プロジェクトでは、建設や維持管理を行う現場と意志決定を行う場は通常異なり、参画者も地理的に散在し、何らかの通信手段を用いて意志決定に必要な情報のやりとりをする場合がほとんどである。さらにこのような情報のやりとりがプロジェクトに与える影響はきわめて大きく、効率的な情報交換はプロジェクトの工期短縮、コスト削減につながる。

橋梁の検査を例に挙げると、現場においてはあらかじめ策定された検査手順により作業を進める。しかし、現場において何らかの重大な損傷が発見された場合、橋梁の使用停止を含めて緊急的な対応が必要な場合がある。このような場合、現場と橋梁技術者、交通ネットワーク管理者、意志決定者が密に連絡を取り合っ問題を開く必要がある。この中でも特に重要となるのが現場からの情報である。

3.2 橋梁維持管理

本研究では橋梁維持管理の参画者を表 3.1 のように分類した。この中でも橋梁技術者が中心的役割を果たし維持管理計画を作成する。検査者は現場において情報を収集するため、技術者の作成した検査計画に従い検査を実施し、検査報告書を作成することで情報を他に伝達する。技術者は、検査報告書を元に補修・補強計画および新たな検査計画を含む維持管理計画を策定する。財務担当者は策定された維持管理計画の予算を算出し、管理者が承認を行う。ただし、維持管理計画の策定は技術者、財務担当者、橋梁管理者間で修正が行うことができる。小規模な組織であれば、各参画者間の情報伝達に要する時間やコストは小さいが、大規模な組織や多数の橋梁を管理する組織においては情報共有・伝達のコストは大きい。そこで情報共有・伝達を促進するシステムが必要となる。

表 3.1 想定する維持管理業務参画者

役割 / 職種	業務内容
橋梁技術者	検査計画書の策定
	検査報告書の承認
	維持管理計画の策定
	施工計画書の作成
	施工作业完了の承認
	施工完了報告書の作成
検査員	現場検査
	検査報告書の作成
構造物管理者 / 所有者	維持管理計画書の承認
	施工完了報告書の承認
財務担当者	維持管理計画予算の算出
補修・補強施工者	補修・補強施工の実施
	作業報告書の作成

3.3 橋梁マネジメントシステム

橋梁マネジメントシステム（以下 BMS）は、諸元、検査記録、修復・改造履歴等の橋梁に関するデータをデータベース上で管理し、橋梁の劣化予測などの解析手法を提供し、また維持管理計画の最適化を行い、橋梁所有者または管理者の意思決定を支援するための包括的なシステムである。本研究では、表 3.1 に示す参画者間の情報の共有・伝達に着目し、協同作業を支援するシステムの設計を行った。図 3.1 は橋梁維持管理に関する協同作業環境の概念を示す。ここでは参画者のデータ・アプリケーション共有を行うために、インターネットを通じた Web システムを提案する。ユーザが使用するアプリケーションやデータはサーバ上で一元管理し、ユーザは Web ブラウザを通してシステムを利用する。ただし、ユーザによって利用形態は異なるため、複数のインターフェースを構築する。橋梁検査員は、主として現場において作業を行うため、現場作業に適したクライアント端末としてノート PC および PDA(Personal Digital Assistant)を想定する。PDA は、携帯情報端末と呼ばれ、主にスケジュールや住所録などの個人情報を管理する PIM(Personal Information Manager)機能を搭載している。サイズはノート PC に比べて非常に小型で軽量であるが、CPU パワー、メモリ等リソースの面で PC に劣る。そのため、共同作業環境を構築する場合、PC や PDA のリソースに応じてインターフェースを設計する必要がある。



図 3.1 BMS における協同作業環境

3.4 検査員用システムのプロトタイプ

前述のように橋梁マネジメントシステムには複数のインターフェースが存在するが、その中でも検査員に着目してインターフェースのプロトタイプを構築した。プロトタイプは、図 3.2 のようにクライアント端末である PDA、サーバとして機能するノート PC、および両者を接続する無線 LAN デバイスで構成される。今回は無線 LAN を用いて PDA-ノート PC 間で閉じたネットワークを構築したが、両者をインターネットを通じて接続することも可能である。

サーバ側は OS に Microsoft®Windows®2000Pro.を用い、Web サーバとして IIS5.0 をインストールした。データベースには Access2000 を採用した。クライアント側の PDA には OS として Windows®Powered Pocket PC(Windows®CE3.0), Web ブラウザとして Pocket IE を用いた。また、Java™実行環境として Jeode™Runtime をインストールした。

検査者は PDA を携帯し、検査記録を入力する。検査記録はサーバ上に保存され、検査者は検査に際しサーバから支援情報を受け取ることが可能である。図 3.3 は PDA 画面上に表示された検査メニューの一例である。

このプロトタイプにおいてサーバとクライアントの接続を無線 LAN で行い、PDA 用インターフェースの実用性の確認を行った。それに対し、PDA に PHS 端末を接続し両者をインターネットに接続したところ、PHS の通信速度がボトルネックとなり、実用的な速度での運用ができなかった。そのため、ナローバンド環境下でも利用できるコンテンツを構築する必要がある。

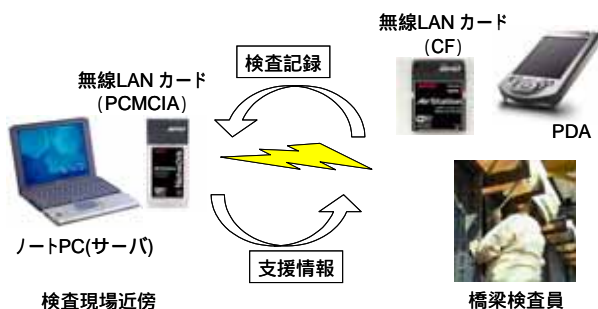


図 3.2 検査員用システムのプロトタイプ

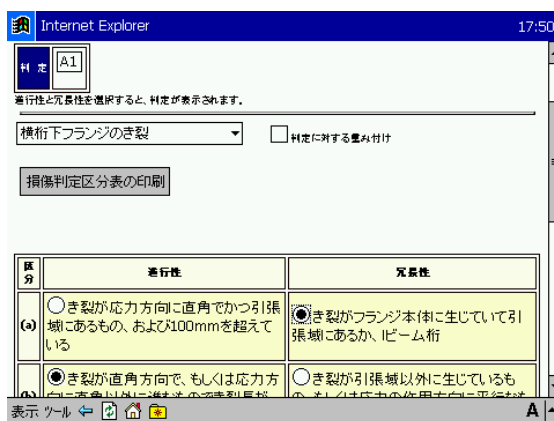


図 3.3 PDA 用検査画面

3.5 まとめ

本研究では、社会基盤施設の維持管理業務において複数の参画者またはグループが効率的に情報交換するためのプラットフォームについて検討を行った。まず、参画者の役割・業務形態を考慮し、それぞれに応じたクライアント端末を選定した。ユーザ間のアプリケーション・データ共有の手法として Web システムを提案し、クライアント端末に応じたインターフェースの設計を行った。複数のクライアント環境を想定したプロトタイプとして、PDA や PC からアクセス可能なシステムを構築した。

参考文献

- 水野, 阿部, 藤野, 阿部: 情報技術(IT)援用による橋梁の目視検査支援システムの構築, 土木情報システムシンポジウム論文集 Vol.9, pp.11-18, 2000年11月
- Y.Mizuno, M.Abe, Y.Fujino, M.Abe, Y.Abe: Development of Web-based Support System for Bridge Visual Inspection, IABSE Symposium, Melbourne, Sept. 2002.

4 データマイニングによる知識獲得

4.1 はじめに

構造物の運用・維持管理においては、取り扱う情報が多岐にわたり、何らかの意思決定を行う際には、構造諸元や履歴等の構造物自身が持つ情報に加えて、地盤状況や周辺環境、特に交通インフラの場合は交通ネットワーク全体を併せて検討する必要がある。また、モニタリングシステムを構造物に設置する場合は、取得した大量のデータを効率的に管理する必要がある。

ブロードバンドをはじめとするネットワークの大容量化や、携帯電話などの情報機器の高機能化・小型化代表される IT 技術の進展は、社会基盤の情報管理に大きなインパクトを与える。国土交通省では、公共事業改革の 1 つとして、コスト縮減の観点から IT を活用することを重要な柱としている。建設 CALS/EC（公共事業支援統合情報システム）では、3次元 CAD データ使用コード類等の情報共有・交換のための情報の標準化やライフサイクル全般にわたる情報を統合したデータベースの構築を推し進めている。

橋梁に関しては、アメリカにおいて各州が包括的な橋梁マネジメントシステム（BMS）の運用を行っており、多くの州が Pontis を採用している。Pontis は、橋梁に関する情報を保存するデータベースと劣化予測や補修・補強に関する予算配分を最適化するためのアプリケーションから構築される。特に劣化予測においてはデータベース中の過去の履歴をもとに劣化モデルを修正する機能を持つ。日本においても、さまざまな研究機関・大学で BMS の開発が行われており、例えば RC 橋の性能評価や劣化予測をサブシステムに持つ BMS の開発が行われている。

社会基盤構造物における情報は、現在においても紙ベースで管理されている場合がほとんどで、データの管理が十分に行われていない。BMS をはじめとする電子化された情報においても、解析等に用いられる情報はわずかで、いったん蓄積されたデータの運用・再利用が十分に行われていない。

そこで本研究では蓄積された情報の再利用・有効活用に着目したシステムの構築手法の検討を行う。

4.2 決定木によるデータマイニング

4.2.1 データマイニング

膨大なデータの中から隠れた有用な知識を獲得するデータマイニングが注目されている。データマイニングは、データからの知識発見（KDD: Knowledge Discovery in Data）とも呼ばれ、人工知能、機械学習、データベース、統計学や経営学にまたがる横断的な学問領域を形成している。

データマイニングは、主にデータの前処理、マイニングアルゴリズムの適用、後処理の 3 つの部分から構成される。前処理では、対象とするデータの品質を確保するために行われ、欠損値の処理やノイズデータの除去、離散化や数値化などのデータ変換が行われる。マイニングアルゴリズムには、統計解析、決定木、ニューラルネットワーク、遺伝的アルゴリズム等が用いられる。後処理では、データの可視化や分析・検証が含まれる。

4.2.2 CART による決定木の構築

決定木は、分類や予測を行う際に用いられている手法であり、ある注目する属性(従属変数)に関する重要な知識を、木構造によるルールの組み合わせで表現したものである。決定木を用いる利点として、ニューラルネットワークとは異なり、人が容易に理解できるルールが抽出できることが挙げられる。決定木によって得られたルールは IF-THEN 形式のプロダクションルールによって簡単に表現することができる。

CART(Classification and Regression Tree)は、決定木を構築する最も一般的なアルゴリズムの一つで、各ノードにおける分岐が 2 となる 2 進決定木である。以下にアルゴリズムについて分岐を行う分割テスト、決定木の構築、枝刈りについて述べる。

(1) 分割テスト

CART は、各ノードで 1 つの独立変数で分岐させていくことにより、2 進木を作成する。分岐の評価を行う尺度として、多様性を表す評価関数を用いる。評価関数は従属変数が均等に存在していれば大きな値をとり、従属変数が 1 つの値しかない状態に近づけば小さな値をとる。各ノードではこの評価関数をもっとも減少させるような最良分岐を行う。

CART では、評価関数として Gini 係数(Gini index)を用いる。従属変数が 2 値で表現される場合 Gini 係数は次式のようになる。 p_1 はノード内で従属変数がある一方の値である割合を示す。

$$2p_1(1-p_1)$$

ここでは、例として橋梁におけるき裂発生の有無について分割テストと評価関数について説明する。従属変数はき裂発生に関して有または無のいずれかの値をとる。対象とする橋梁は全 386 橋、そのうちき裂が発生しているものは 63 橋、き裂が発生していないものは 323 橋である。この状態を親ノードとして分割テストを行うと、子ノード 1 ではき裂あり 21 橋、き裂なし 26 橋、計 47 橋、子ノード 2 ではき裂あり 42 橋、き裂なし 300 橋、計 342 橋となる。親ノードでの Gini 係数は $2*(63/389)*(326/389)=0.271$ であり、子ノード 1, 2 はそれぞれ $2*(21/47)*(26/47)=0.494$, $2*(42/342)*(300/42)=0.215$ となる。分割テストによる評価関数の減少は親ノードの評価関数と子ノードの評価関数の重み付き平均との差で表され、 $0.271-(47/389)*0.494-(342/389)*0.215=0.022$ となる。

(2) 決定木の構築

根ノードにおいて上記のように分割テストを行い、下位の各ノードを分岐していく。木の上部において分岐変数となったカテゴリ値を持つ変数は、すぐに単一な値になる傾向がある。2 つ以上の値を持つ変数によって最良な分類を行い、多様性を大きく減少させる分岐が見つからなくなったとき、分割テストを終了し、ノードを葉とする。

分岐すべきところがなくなり、葉ノードだけになれば、すべての分岐処理は終了する。この時点で決定木はもっとも大きな木に成長している。後に述べるように、学習用データによく当てはまるように構築された最大決定木は、新しいデータレコードを最良に分類するものではない。そこで次に述べる枝刈りを行う。

(3) 枝刈り

CART アルゴリズムは、まず、はじめに根ノードにおいてデータ全体に対する最良分岐を行う。このときには、分岐を探す時に用いるデータレコード数は多い。分岐を進めて木が成長するとデータレコード数が徐々に少なくなっていく。そして木が終わりに向かうに従い、より少量のデータレコードによって及ぼされる傾向が強くなっていく。

ここでは、予測精度が高い部分木を選んでいく手順を述べる。どれだけ深く枝刈りするかを決めるために、はじめに繰り返し枝刈りを行うことによって候補となる部分木を決め

る．もっとも予測に貢献していない葉から枝刈りしていくことを目標とする．1 葉あたりの予測力がない枝葉を決めるために，調整済み誤分類率の概念を取り入れる．この指標は，葉数によるペナルティを与え，そのペナルティにうち勝つほどには誤分類率が下がらない弱い枝葉を見つけるためのものである．見つけられた枝葉は，枝刈りを行う候補になる．

4.2.3 橋梁の検査記録へのデータマイニングの適用

本研究では，鋼鉄道橋の検査データをもとに決定木によるルール抽出を行った．決定木はデータベース中の対象とする属性に関する知識を，木構造によるルールの組み合わせで表現したものである．ルールを頂点ノードから終端に至るまでの，分割テストの IF-THEN ルールとして簡単に表現することができる．

(1) 対象とするデータ

対象とする構造物を鋼鉄道橋とし，橋梁形式を上路プレートガーダー，組み立て構造をリベット形式とし，変状としてき裂の発生の有無を従属変数とし，それ以外の項目を独立変数としてデータマイニングを行った．橋梁連目数は 389 であり，そのうちき裂が発生している連目数は 63 であった．用いたデータ構造は表 4.1 に示す．対象データの項目は，橋梁情報を管理するデータベースから得られたもので，データの欠損や誤りの少ないものを選択した．

表 4.1 データ構造

項目	データ型	備考
設計荷重	テキスト型	
図面番号	テキスト型	
構造形式	テキスト型	
単複	テキスト型	単線 / 複線の区別
スパン	浮動小数点型	単位[m]
使用開始年	整数型	西暦
年間通トン	浮動小数点型	交通量，単位[ton]
斜角の有無	セット型	「あり」または「なし」
曲線	セット型	「あり」または「なし」
溶接補強	セット型	「あり」または「なし」
き裂の発生	セット型	「あり」または「なし」

(2) 決定木の導出

導出された決定木を図 4.1 に示す．根ノードでは「スパン」による分岐が行われ，次に子ノードでは「図面番号」，「年間通トン」による分岐が行われている．最終的に 7 つの葉ノードが生成されている．

この決定木を導出に用いた橋梁データに適用すると表 4.2 のようになる．決定木による予測と対象データを比較すると，予測の精度は正:361 件 (92.80%)，誤:28 件 (7.20%) となった．対象データのき裂発生連目は 63 (16.20%) であり，決定木による予測は有意なものと考えることができる．

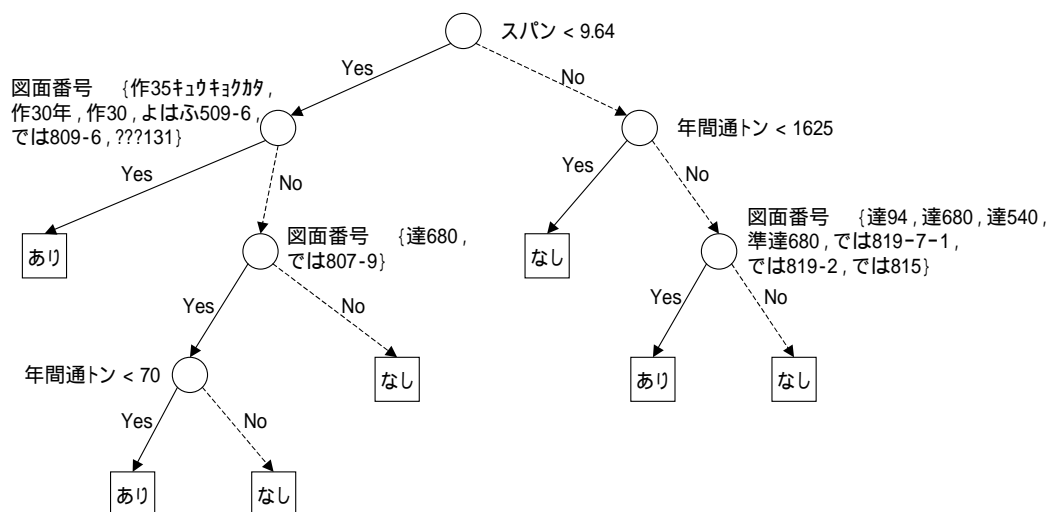


図 4.1 導出された決定木

表 4.2 決定木による予測

		決定木による予測	
		き裂あり	き裂なし
対象データ	き裂あり	39	24
	き裂なし	4	322

(3) プロダクションルールへの変換

前節では決定木全体の評価を行ったが，ここでは，決定木の根ノードから葉ノードへのルートを IF-THEN 形式のプロダクションルールに変換する事により，決定木を複数のルールに分割し，個々のルールについて評価する．プロダクションルールを以下に示す．なお，ルールに示された番号は，図 4.1 の葉ノードに対応する．

ルール 1： IF (スパン < 9.64) AND (図面番号 {作 35 キウキョカカ, 作 30 年, 作 30, よはふ 509-6, よはふ 506-13, では 809-6, ???131})
THEN き裂あり

ルール 2： IF (スパン < 9.64) AND (図面番号 {達 680, では 807-9}) AND (年間通トン < 70))
THEN き裂あり

ルール 3： IF (スパン < 9.64) AND (図面番号 {達 680, では 807-9}) AND (年間通トン < 70))
THEN き裂なし

ルール 4： IF (スパン < 9.64) AND (図面番号 {トケツ, 達 970, 達 540, 達 1084, では 508, 達 1084, 達 4.5', では 816, では 808, 806-7, Value})
THEN き裂なし

- ルール 5： IF (スパン 9.64) AND (年間通トン < 1625))
THEN き裂なし
- ルール 6： IF (スパン 9.64) AND (年間通トン 1625) AND (図面番号 { 達 94, 達 680, 達 540, 準達 680, では 819-7-1, では 819-2, では 815 }))
THEN き裂あり
- ルール 7： IF (スパン 9.64) AND (年間通トン 1625) AND (図面番号 { 曲では 864(8)-1, では 828-2, では 819-6, では 816-50, では 816, DG831-1, DG825-2, Value }))
THEN き裂なし

上記のルールの評価指標として、確信度とサポートを表 4.3 に示す。表中のルール該当データは、対象データ中でそれぞれのルールの前提部(IF 節)に該当するデータ数を表す。整合データはルール該当データのうち、ルールによる予測と実際のき裂発生が整合したデータ数で、矛盾データは予測と実際のき裂発生が一致しなかったデータ数である。確信度は、整合データをルール該当データ中の割合で示され、個々のルールの精度を表す。サポートは、整合データを対象とする全データ中の割合で示され、対象データ中でのルールの有効性を表す。

表 4.3 ルールの評価

ルール	ルール該当データ	整合データ	矛盾データ	確信度(%)	サポート(%)
1	14	14	0	100.00	3.60
2	4	4	0	100.00	1.03
3	8	6	2	75.00	1.54
4	21	20	1	95.24	5.14
5	294	273	21	92.86	70.18
6	25	21	4	84.00	5.40
7	23	23	0	100.00	5.91
計	389	361	28		

4.2.4 抽出されたルールの検証

抽出されたルールの中で、ルール 5 はサポートがもっとも大きく、確信度も高いため有効なルールと考えることができる。また、ルールの内容を検討すると、スパンがある程度以上で、交通荷重（年間通トン）がある値以下の橋梁にき裂が発生しないことは、橋梁維持管理に従事する技術者の経験と照らし合わせても妥当と判断できる。

ルール 2, 3 を見ると、ルール 5 とは異なり、交通荷重の少ない橋梁にき裂が発生する傾向が示されている。この傾向は、橋梁技術者の経験・知識に一見矛盾するように考えられるが、ルールの解釈方法により、この矛盾を解消できる。その解釈とは、交通荷重のきわめて少ない線区は閑散線区であり、適切な維持管理が欠落していることである。そのためき裂が発生しやすい傾向にあることは、抽出されたルールや技術者の経験・知識と整合する。

ルール 1 では、ある特定の図面番号を持つ橋梁にき裂が発生しやすい傾向が読みとれる。これらの橋梁は補強桁や明治期に製作されたものであり、経年劣化が進んでいるものと考えられる。

このように、抽出されたルールに関して橋梁技術者による検証によりその有効性を示すことができた。

4.2.5 まとめ

データベースに蓄えられた情報から知識を抽出するデータマイニングを行った。その結果、いくつかのルールが生成され、その有効性を検討した。鋼鉄道橋を対象とし、き裂発生の有無について決定木を構築した結果、いくつかの有効なプロダクションルールを生成した。これらのルールは橋梁技術者の検証が必要であるが、膨大なデータから効率的に知識を抽出する有効性を示した。

データベース活用という観点からデータマイニングを適用する場合、今後の課題として、対象データの獲得、ルールの後処理、ルールの再利用が考えられる。

本研究では、既存のデータベースからデータの欠損や誤りの少ないものを選別してデータマイニングを適用した。そのため橋梁形式として上路プレートガーダー、変状としてき裂発生の有無という比較的小さな領域を扱った。今後様々なデータソースに対して、データマイニングを行い、知識抽出の有効性を示す必要がある。

データマイニングにより抽出されたルールは、確信度やサポートなどの指標によってある程度有効性を検証することができるが、ルールの内容を検証するには専門技術者の協力が必要である。コンピュータによる自動化と専門家による検証を組み合わせた効率的なシステムの構築が、知識抽出の精度・効率を向上させる。

本研究では、データマイニングによりルールを抽出し、そのルールの検証を行ったが、抽出されたルールを社会基盤構造物マネジメントシステムにフィードバックする機構が必要である。マネジメントシステムにおけるルールの活用方法として、エキスパートシステムと組み合わせた意志決定支援や、抽出されたルールを技術者のための教育コンテンツとして利用する方法がある。

参考文献

- 福田，森本，徳山：データマイニング，共立出版，2001年9月
- M.J.A.Berry, G.S.Linoff: Mastering Data Mining, John Wiley & Sons, Inc., 2000.
- 水野，阿部，藤野，S.Meret，阿部：データベース技術を用いた社会基盤構造物に関する維持管理データ管理手法の提案，コンクリートの耐久性データベースフォーマットに関するシンポジウム論文集，土木学会，pp.77-82，2002年12月

5 BridgeML に基づくマルチモーダルアクセス可能な 橋梁マネジメントアーキテクチャ

5.1 はじめに

既存の橋梁マネジメントシステム(BMS)は、橋梁技術者や管理者に対し意思決定支援を行うためのアプリケーションやデータベースで構築されている。データベースには検査記録が含まれるが、検査員をサポートする機能が含まれていない。また、既存の BMS はスタンドアロンタイプのソフトウェアであり、複数のユーザによる情報の共有が考慮されていない。

橋梁の維持管理には構造物管理者や検査請負業者、コンサルタント、補修・補強施工業者など複数の組織が関与する。しかし、関与者間のコミュニケーションはそれぞれ独自の書式を用いる。また、維持管理基準や検査マニュアルは構造管理者ごとに書式が異なる。そのため、維持管理に関与する組織間のコミュニケーションは複雑であり、コンピュータによる情報の共有や再利用が困難になっている。

そこで、本研究ではまず、ネットワークに接続され、複数のユーザによるアクセスを可能にする BMS のアーキテクチャを提案する。特に、検査員に対してはコンピュータによる検査記録の入力だけでなく、音声による検査サポートについて検討し、プロトタイプを作成した。

橋梁の維持管理に関する情報の共有・再利用を実現するために、BridgeML を提案する。BridgeML は XML(eXtensible Markup Language)による橋梁の維持管理を記述するための言語である。また、BridgeML の優位性やアプリケーションについて検討した。

5.2 情報共有型 BMS アーキテクチャ

ここでは情報システムとしての BMS フレームワークを提案する。本システムの主に二つの特徴を持つ。一つは、インターネットを通じたシステムであり、複数のユーザによる情報共有を行う。もう一つは、ユーザに対し複数のアクセス方法を可能とすることである。これは、検査員を例にとると検査記録の入力を行う際に、インターネットに接続された PC を用いる方法と、電話などの音声によりアクセスする方法の両方を実現することである。

5.2.1 機能デザイン

構築するシステムは、検査記録を含むデータベースに対しインターネットを通じてリクエストを送信し、処理結果を返すシステムである。このシステムは、先に述べたように通常の PC 環境や音声入出力環境などの複数のアクセス方法を、橋梁検査員、技術者、管理者に提供する。

ただし、本件研究で開発したプロトタイプは制限事項として、ユーザを橋梁技術者に限定する。しかし、他の検査員や管理者に対するユーザインターフェースの実装は、技術者の場合と同様に構築できる。また、システムの操作方法として、橋梁や変状の選択のみをユーザからの入力とし、出力に関しては HTML による画面表示と音声による結果提示を行う。

システムの構成を図 5.1 に示す。図の中央にサーバがあり、多量のデータ管理やデータ処理を行う。ユーザである検査員、技術者、管理者はクライアント機器を用い、インターネットを通じてシステムを利用する。このようにクライアント/サーバモデルを採用することでクライアント機器の負担を軽減することができる。

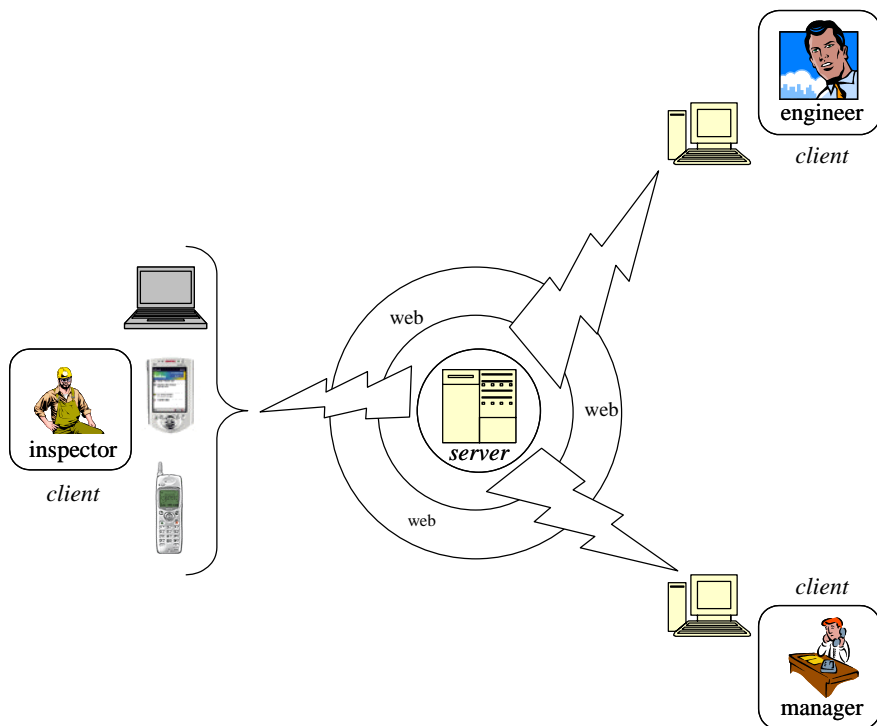


図 5.1 クライアント/サーバモデルによるシステム構成

5.2.2 システムの構成

(1) ソフトウェア

システムに採用したソフトウェアを表 5.1 に示す．以下にソフトウェアの特徴，選定理由を述べる．

表 5.1 ソフトウェア一覧

種類	ソフトウェア名	開発元
RDBMS(データベース)	PostgreSQL 7.2.1	PostgreSQL Global Development Group
Java Development Kit	J2SDK 1.3.1	Sun Microsystems
Servlet Engine	Tomcat	The Jakarta Project (Apache Foundation)
XML Parser	Xerces	Apache Foundation
XSL Processor	Xalan	Apache Foundation

(a) RDBMS (Relational Database Management System)

PostgreSQL はフリーのオープンソースソフトウェアであり，本研究で用いるデータ量に対して十分な機能を備えていることから採用した．データベースには鋼鉄道橋の橋梁諸元と検査記録が含まれている．データモデルを図 5.2 に示す．主キーは太字で，外部キーはイタリック体で示す．

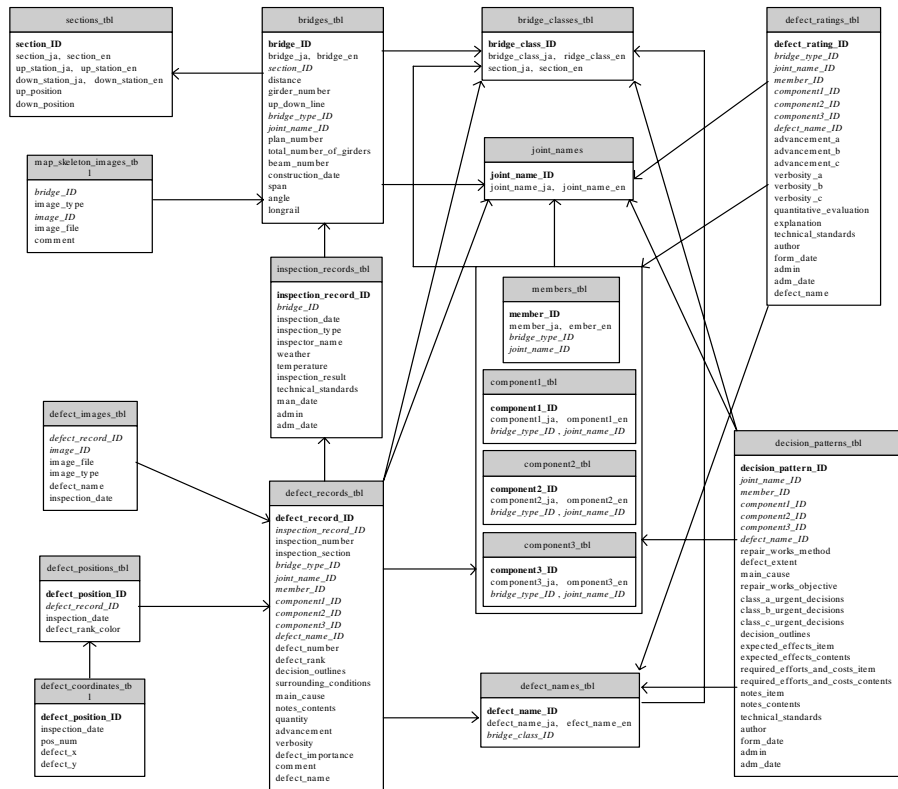


図 5.2 データモデル

(b) プログラミング言語 : Java

Java を選択した理由は以下ようになる。また、Java を採用理由はそのまま他のコンポーネントの採用理由となる。

- オブジェクト指向の言語
- サーバサイドで動的コンテンツを生成する Servlet が利用できる
- オペレーティングシステム(OS)に依存しない

(c) Java Servlet Engine : Tomcat

Servlet はサーバ側で動作するアプリケーションであり、ユーザからのリクエストを処理してクライアント側に応答を返す。Servlet は動的な Web サイトを構築するために最も一般的に用いられている技術の一つである。

Servlet を動作する環境を Servlet Engine または Servlet Container と呼び、Tomcat は利用に際し自由度が大きくかつ信頼性が高いため採用した。図 5.3 に Tomcat を用いた Web アプリケーションの構成を示す。

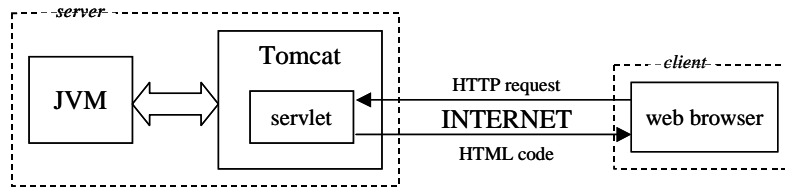


図 5.3 Tomcat による Web アプリケーション構成

(d) JDBC : データベースとの接続

Java アプリケーションからデータベースを操作するために JDBC(Java Database Connectivity)ドライバが必要である。図 5.4 のように JDBC ドライバは Java アプリケーションからのリクエストを SQL クエリに変換し、データベースからの SQL 結果をオブジェクトに変更する。

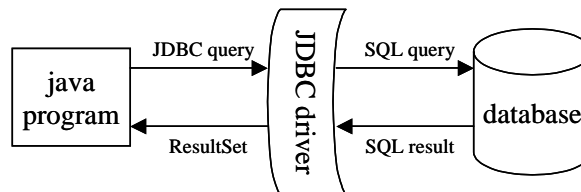


図 5.4 JDBC ドライバによるデータベース接続

(e) XML , DOM , Xerces

XML は木構造のデータを記述するための言語であり、タグを用いて情報をラベル付けする。本研究では、表構造のデータベースから取り出した情報を XML に変換して扱う。この利点については後で記述する。

DOM は XML にアクセスするための API(Application Programming Interface)仕様である。DOM オブジェクトを用いることでプログラムコードから容易に XML の木構造を処理することができる。

一般に XML から DOM を生成するプログラムをパーサと呼び、さまざまなプログラミング言語でパーサが実装されている。Xerces は Java で実装された XML-DOM間の変換を行うパーサであり、本研究では Xerces を採用した。図 5.5 にパーサの動作概念を示す。

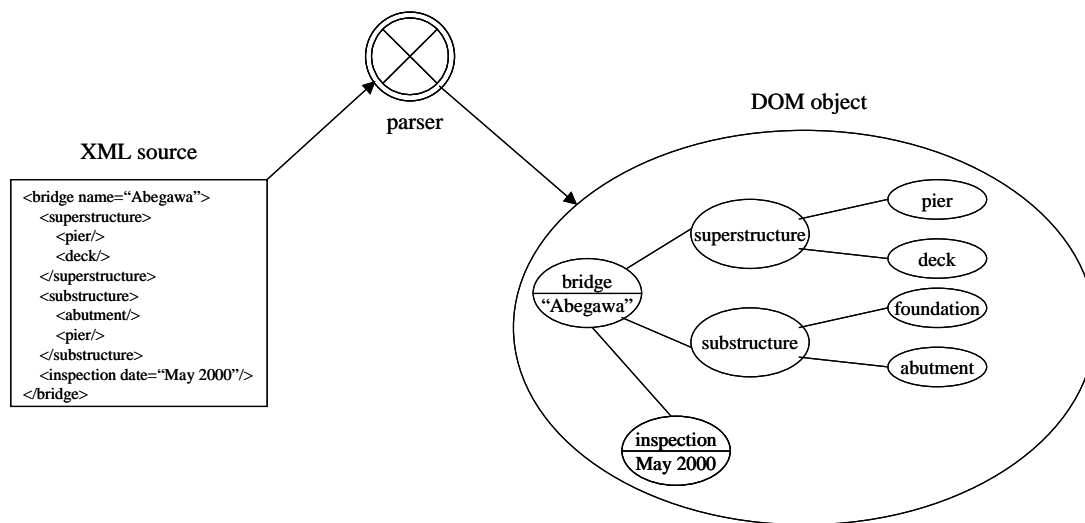


図 5.5 パーサによる XML-DOM 変換

(f) XML , XSL , Xalan

XSL は XML を HTML や VoiceXML 等の他の形式に変換するために用いられる。本研究では、XML に XSL を適用するプロセッサとして Xalan を採用した。図 5.6 に Xalan の動作概念を示す。

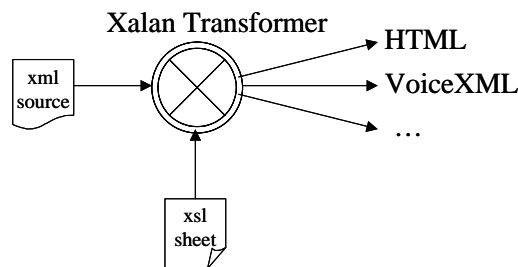


図 5.6 Xalan による XML 変換

(2) Web アプリケーションの多層モデル

図 5.7 に示すように構築するアプリケーションはデータ層、ビジネス層、プレゼンテーション層の多層構造を持つ。それぞれの層の役割を次に示す。なお、本研究ではこれらの層に用いるプログラムは Servlet または、Servlet から呼び出す Java Class であり、Tomcat に格納される。

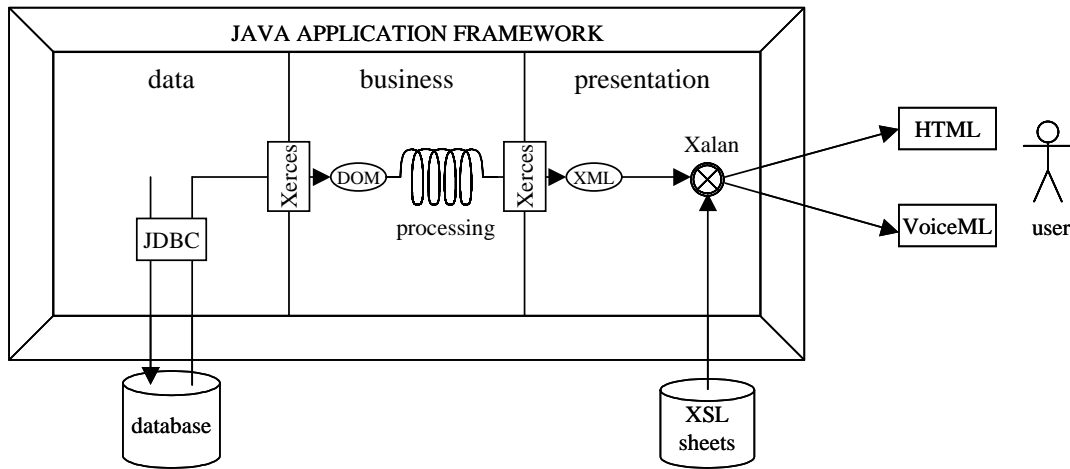


図 5.7 Web アプリケーションの多層構造

(a) データ層

データ層はデータベースとのインターフェースであり，データベースに対し SQL クエリを送信し，クエリ結果を取得する．Java アプリケーションにおいては JDBC ドライバがデータ層を構成する．

(b) ビジネス層

ビジネス層はアプリケーションの中心的な役割を果たす．ここではユーザからのリクエストに対してデータ処理を行うプログラムが格納される．この層では，データを DOM オブジェクトとして処理し，他層とのインターフェースに Xerces を用いて XML-DOM 間の変換を行う．橋梁マネジメントシステムにおける意思決定支援アプリケーションなどの工学的なアプリケーションはこの層に実装される．

(c) プレゼンテーション層

この層では，ビジネス層からの出力である XML を変換する．具体的には Xalan を用いて XML と XSL スタイルシートを組み合わせることで，ユーザの使用するクライアント機器に応じた出力形式に変換する．出力形式としては HTML や VoiceXML などがある．

(3) Web アプリケーションにおける情報の流れ

Web アプリケーションはこれまでに述べたコンポーネントにより構成される．図 5.8 に Web アプリケーションのコンポーネントとコンポーネント間の情報の流れを示す．

ここでは，ユーザがデータベース中から情報を検索することを想定して，情報の流れについて説明する．まず，ユーザはブラウザ上に示される HTML フォームにおいて，検索したい情報を提示する．提示方法はテキストエリアへの入力，チェックボックスの選択などがある．フォームの入力を行った後，Submit ボタンをクリックするとユーザの入力は HTTP リクエストに変換されてサーバに送信される．

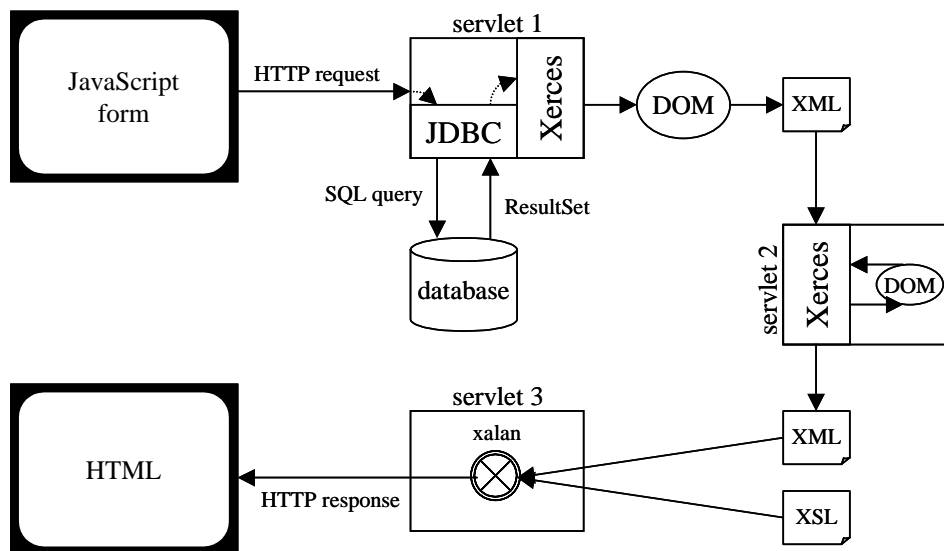


図 5.8 Web アプリケーションにおける情報の流れ

サーバ側では、データ層を担当する Servlet1 が HTTP リクエストを受け取り、データベースに検索を行う。検索はまず、HTTP リクエストを SQL クエリに変換し、JDBC によりデータベースに接続する。データベースからの応答は、JDBC を通じて Java オブジェクト ResultSet として受け取る。また、Servlet1 は Xerces を用いて ResultSet の中から必要な情報を DOM オブジェクトに変換する。他の Servlet に情報を渡す際には、いったん XML ファイルに変換する方法と DOM オブジェクトをそのまま用いる方法があるが、本研究では前者を採用し、Servlet1 は DOM オブジェクトから XML ファイルを生成し、他の Servlet がデータベースの検索結果を使用できるようにする。

Servlet2 はビジネス層を構成し、ユーザのリクエストに応じてデータベースの検索結果に対して処理を行う。例えば、ユーザが橋梁の変状記録を必要とする場合に、最近の検査記録のみを取り出したり、変状数をカウントしたりする処理をこの Servlet が担当する。Servlet2 も Xerces を用いて XML-DOM 間の変換を行い、Servlet 内の処理は DOM を使用し、他の Servlet に対しては XML ファイルを用いて情報を交換する。

Servlet3 はプレゼンテーション層であり、Xalan を用いて XML と XSL を組み合わせて、ユーザにプレゼンテーションコードを提示する。図では PC 環境のユーザを想定し、HTML でプレゼンテーションコードを生成している。Servlet3 では組み合わせる XSL を変更することで、XML から HTML、VoiceXML などさまざまなプレゼンテーションコードの生成を実現する。

5.2.3 プロトタイプの構築

(1) HTML ベースの BMS(HTML チャンネル)

ユーザとして橋梁技術者を対象に橋梁や変状を選択することで変状記録を参照できるプロトタイプを構築する。プロトタイプではまず、ユーザは橋梁を選択する。次に画面に示される変状一覧の中から参照したい変状を選択することで、詳細な変状記録を見ることができる。

このプロトタイプの機能は簡単ではあるが、用いるフレームワークは汎用的であり、他の機能を追加する場合にも容易に再利用可能である。このフレームワークは 5 つの

Servlet(DisplayDefectsList, SelectBridge, DefectsList, ShowDefect, XML2HTML)と 1 つの Java クラス(xmlFileMaker)で構成される。

Web アプリケーションの画面遷移は図 5.9 のようになる。まず、ユーザは index.html から DisplayDefectsList へ移り、橋梁を選択する。DisplayDefectsList は SelectBridge および DefectsList から構成されている。ユーザが初めて DisplayDefectsList を訪れたときは SelectBridge のみが機能し、橋梁選択を促す。橋梁が選択されると DisplayDefectsList の画面下方に変状一覧が表示されるが、この部分を DefectList が担当する。DefectList から変状を選択すると、ShowDefect が変状の詳細を表示する。一連の情報の流れを図 5.10 に示す。



図 5.9 Web アプリケーションにおける画面遷移

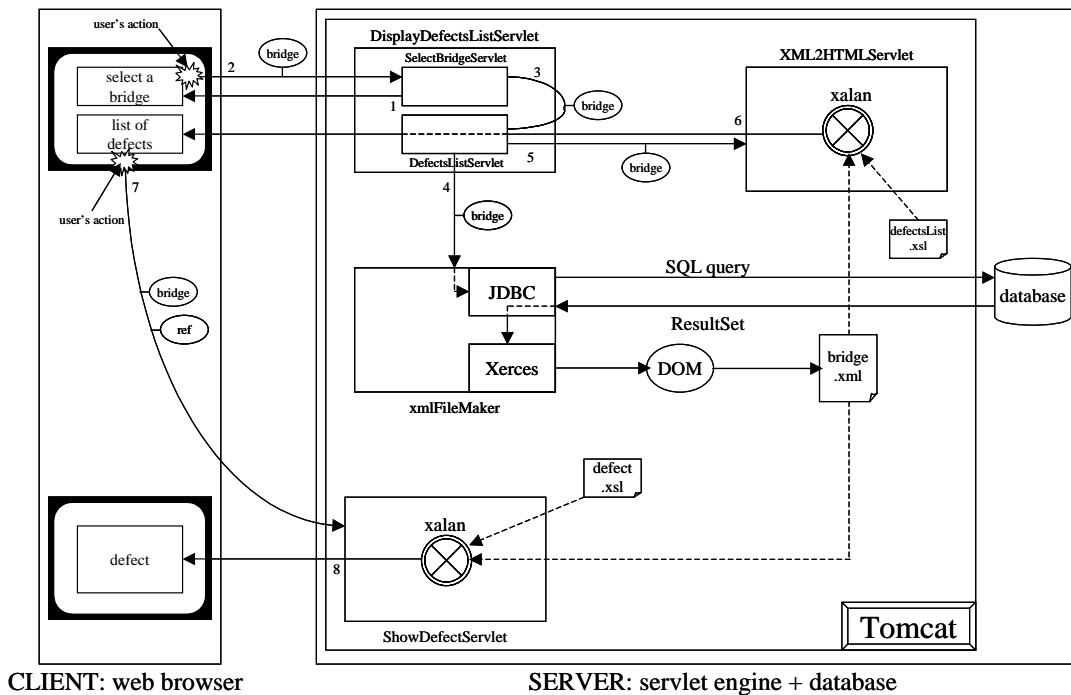


図 5.10 プロトタイプでの情報の流れ

このプロトタイプでは、橋梁を表すパラメータ bridge と橋梁に関する情報を保持する XML ファイル bridge.xml を用いて情報を管理する。まず、ユーザが初めて DisplayDefectsList を表示させたとき、パラメータ bridge は値を保持しておらず、SelectBridge のみがインクルードされる。SelectBridge は JDBC を用いてデータベースに接続し、登録されている橋梁名を呼び出し、HTML として橋梁一覧を提示し、ユーザが選択できるようにする。ユーザがある橋梁を選択し Submit ボタンをクリックすると、その橋梁名がパラメータ bridge として再び DisplayDefectList へリダイレクトされる。パラメータ bridge が値を保持している場合、SelectBridge と DefectsList がインクルードされる。

DefectsList は xmlFileMaker を呼び出しパラメータ bridge を渡す。xmlFileMaker はパラメータ bridge で渡される橋梁名に関して JDBC を通じてデータベースから変状一覧を取得する。変状一覧は JDBC により ResultSet オブジェクトとして得られるが、Xerces を用いて DOM オブジェクトに変換し、XML ファイルを生成する。

XML ファイルは橋梁に関するすべての変状情報を含んでおり、defectList.xls を用いて変状一覧を表示する HTML に変換される。DefectsList では XML ファイルから HTML に変換する処理を XML2HTML で行う。XML2HTML は XML ファイルと XLS を指定することで HTML を出力する。DefectsList に表示される変状一覧では個々の変状ごとに固有の ID が与えられており、ユーザは ID を選択することで詳細な変状記録を参照する。

詳細な変状記録は ShowDefect によって表示される。ShowDefect は DefectList からパラメータとして変状 ID を受け取り、XML ファイルの中から該当する変状の情報のみを取り出し、HTML として表示する。ここでは HTML の記述には defect.xml を用いる。

このように、ユーザが必要とする情報はデータベースから XML ファイルに変換する。ここでは、通常の PC 環境を想定し、Servlet を用いて XML ファイルから HTML への変換を行うプロトタイプを構築した。最終的な出力形式を VoceXML などの他の形式にする場合に関しても、XML ファイルからの変換プログラムのみを変更することで実現可能であり、汎用性の高いプラットフォームの構築が可能になった。

(2) 音声出力を備えた BMS(音声チャンネル)

音声出力をオンラインで行うシステムは VoceXML を出力するソフトウェアと、音声オンラインで出力する Voice サーバで構成される。本研究ではプロトタイプを作成し、音声出力を備えた BMS を試験的に実装するためより簡易なシステムを採用した。図 5.11 に示すように、システムはサーバ側では橋梁の変状に関する XML ファイルを作成する部分とクライアント側で XML ファイルを音声に変化するアプリケーションで構成される。

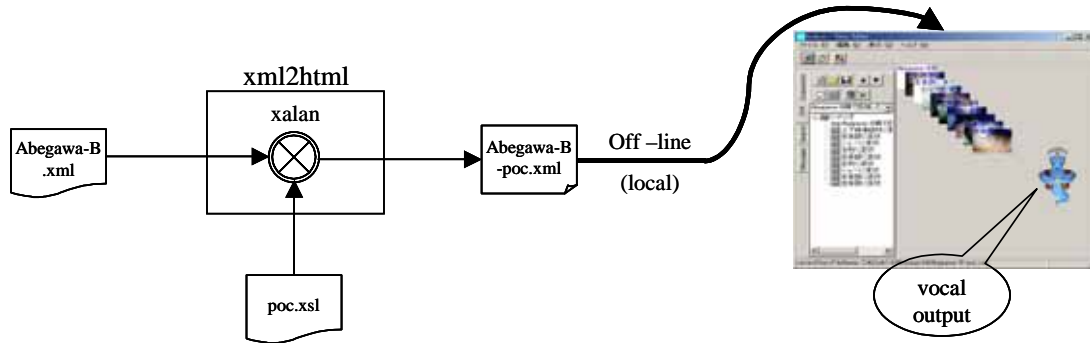


図 5.11 音声出力システム

クライアント側のアプリケーションとして POC Communicator を採用した。POC(Public Opinion Channel)は通信総合研究所情報通信ブレークスルー基礎研究 21 西田結集型特別グループによるプロジェクト(http://www.synsophy.go.jp/about_poc/index-j.html)で、コミュニティ支援を行うシステムである。POC ではコミュニティ・メンバが自由に情報を発信し、コミュニティで情報を共有する。共有される情報は XML 形式で記述され、音声として情報を出力することができる。本研究では、POC で採用された XML 形式(以下 POCXML)で橋梁情報を記述し、クライアントアプリケーションで音声出力を行う。

POCXML の生成はサーバ側で行う。先ほど HTML ベースの BMS 示したように、サーバ内で橋梁情報は xmlFilemaker クラスにより XML ファイルとして扱っている。音声出力を行うためには、橋梁情報の XML を POCXML に変換する必要がある。図 5.11 に示すように xml2html クラスを用いて、XML ファイルと poc.xsl を組み合わせることで、POCXML を生成する。

POCXML はクライアントアプリケーションにより音声出力を行う。図 5.12 および図 5.13 にクライアントアプリケーションの出力画面を示す。図 5.12 は橋梁の変状一覧を示し、図 5.13 は詳細な変状記録を示す。音声出力は Microsoft Agent が行う。

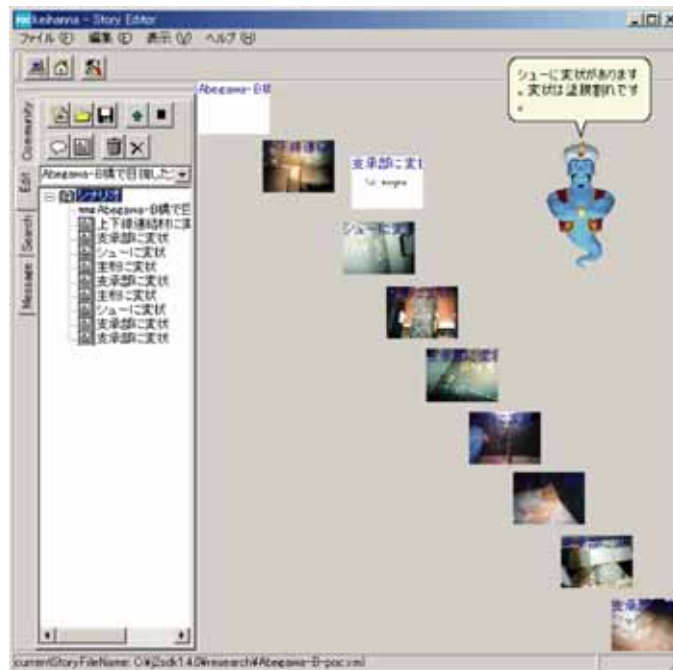


図 5.12 POC Communicator による変状一覧表示



図 5.13 POC Communicator による変状の詳細表示

5.2.4 多様なクライアント機器の採用の実現に向けて

先に述べたプロトタイプでは、HTML による出力と、PC 上のクライアントアプリケーションを用いた音声出力を実現した。実際に電話による音声ナビゲーションや i モード対応携帯電話(NTT DoCoMo)をクライアント機器に用いる場合のハードウェア・ソフトウェア構成について述べる。また、XML から多様な形式への変換について議論する。

(1) 電話による音声ナビゲーション(音声チャンネル)

音声出力を備えた BMS で述べた同様の手法を用いて橋梁情報の XML ファイルを VoiceXML に変換可能である。電話による音声ナビゲーションを実現するためには、本研究で構築した Web アプリケーションを Voice ゲートウェイまたは Voice サーバに接続する必要がある。Voice ゲートウェイ、Voice サーバは VoiceXML で記述された情報に従って、

ユーザに対して音声で情報をやりとりするためのハードウェアとソフトウェアで構成され、主要な機能は以下ようになる。

- VoiceXML ブラウザは VoiceXML を音声ナビゲーションに変換する
- 音声認識機能はユーザの音声による入力を解析・処理する
- 音声合成エンジンはユーザに対してテキスト情報を音声出力に変換する
- dialogic カードまたは CISCO ルータは公衆電話回線とのインターフェースとなる

Voice ゲートウェイの役割は、ユーザからの音声によるリクエストを HTTP リクエストとして Web サーバに送り、Web サーバが送信する VoiceXML を音声出力に変換することである。

(2) i モードによるクライアント機器の実現(i モードチャンネル)

i モードは NTT DoCoMo の携帯電話のためのインターネットサービスである。i モード対応携帯電話では Web ブラウズが可能であるが、通常の PC に比べて計算能力・記憶容量が乏しいため、HTML のサブセットである cHTML を採用している。クライアント機器として i モード対応携帯電話を用いる場合、XML-cHTML 変換を行う必要がある。

(3) 複数の出力チャンネルの実現

これまでシステムの出力形式として HTTP プロトコルによる HTML 出力を中心に述べてきたが、それ以外にもさまざまな出力形式が考えられる。例えば、橋梁維持管理における図面情報をデータベースで管理し、クライアント側では CAD 形式で出力することが考えられる。この場合 XML-CAD ファイル変換を行うコンバータが必要である。同様に橋梁維持管理の報告書を作成する場合、テキストデータ、ワードプロセッサファイル、表計算ファイルでの出力が考えられる。このようにさまざまな出力形式に変換する様子を図 5.14 に示す。これら複数の出力形式への変換は、データベース-XML 変換と XML-出力形式変換と 2 段階に分けることができる。

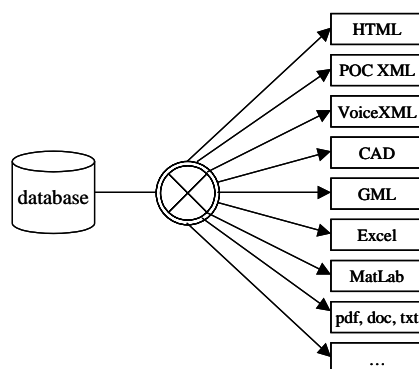


図 5.14 複数の出力チャンネル

5.3 Bridge Markup Language

先に示したプロトタイプでは、アプリケーション内部で橋梁情報を XML で表現し、ユーザの使用形態に応じて XML から HTML や音声形式に情報を変換した。本節ではアプリケーション内で使用する XML について述べる。

Web 上で情報を交換する手段として、XML ベースで情報を記述する仕様が策定されている。数学、音楽、ニュースに関する記述がそれぞれ MathML, MusicML, NewsML とし

て策定されている．本研究では橋梁の維持管理に関する記述言語 BridgeML(Bridge Markup Language)を提案する．

本研究では 2 つのドラフトについて述べる．ドラフト 1 は前節のプロトタイプシステムで用いたもので，非常に簡単な論理構造であるが，アプリケーション内において XML で情報を管理し，HTML などの他の出力形式に変換するフレームワークを実現した．ドラフト 2 は，論理構造を複雑にし表現力の高い仕様であるが，実際にアプリケーションを作成し検証が必要である．

5.3.1 BridgeML ドラフト 1

ドラフト 1 は前節のプロトタイプシステムで採用したもので，データベース上の橋梁情報をマッピングする．そのため，論理構造はデータベース構造に大きく依存し，またデータベース中の主要な部分のみを抽出している．ドラフト 1 の DTD(文書型定義)を図 5.15 に示し，またその論理構造を木構造に表現したものを図 5.16 に示す．

```
<!ELEMENT bridge (location|defect)>
<!ATTLIST bridge name CDATA #REQUIRED>

<!ELEMENT location (member, component1, component2, component3)>
<!ATTLIST location girderNb CDATA #REQUIRED>
<!ATTLIST location ref IDREF #REQUIRED>

<!ELEMENT member EMPTY>
<!ATTLIST member name.en CDATA #IMPLIED>
<!ATTLIST member name.ja CDATA #IMPLIED>

<!ELEMENT component1 EMPTY>
<!ATTLIST component1 name.en CDATA #IMPLIED>
<!ATTLIST component1 name.ja CDATA #IMPLIED>

<!ELEMENT component2 EMPTY>
<!ATTLIST component2 name.en CDATA #IMPLIED>
<!ATTLIST component2 name.ja CDATA #IMPLIED>

<!ELEMENT component3 EMPTY>
<!ATTLIST component3 name.en CDATA #IMPLIED>
<!ATTLIST component3 name.ja CDATA #IMPLIED>

<!ELEMENT defect (date, picture, rank, mainCause, quantity)>
<!ATTLIST defect location-ref ID #REQUIRED>
<!ATTLIST defect name.en CDATA #IMPLIED>
<!ATTLIST defect name.ja CDATA #IMPLIED>

<!ELEMENT date (#PCDATA)>
<!ELEMENT picture (#PCDATA)>
<!ELEMENT rank (#PCDATA)>
<!ELEMENT mainCause (#PCDATA)>
<!ELEMENT quantity (#PCDATA)>
```

図 5.15 BridgeML ドラフト 1 の DTD

ルート要素は橋梁を表す `bridge` であり，属性として橋梁名を表す `name` を持つ．また子要素として変状位置を表す `location` および変状の詳細を記述する `defect` を持つ．

要素 `location` は変状位置を表し，属性として桁番号を表す `girderNb` と対応する `defect` 要素を参照するポインタを持つ．子要素として橋梁の部材や位置を示す `member`，`component1`，`component2`，`component3` を持つ．要素 `member`，`component1`，`component2`，`component3` は橋梁の部材や部品名に相当しそれぞれ属性として日本語名，英語名を持つ．

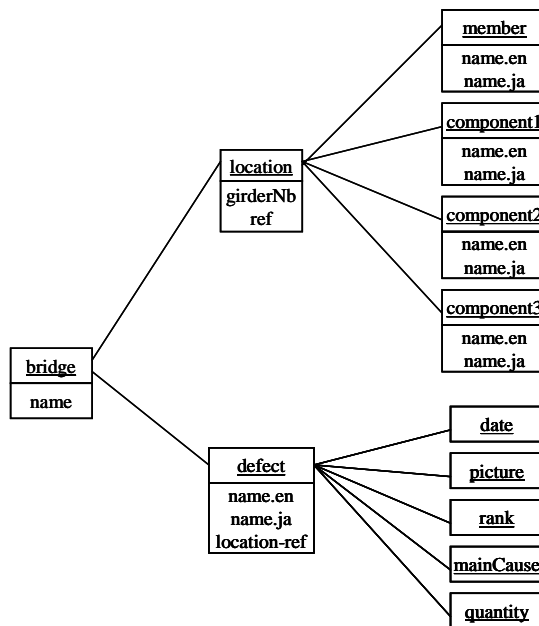


図 5.16 ドラフト 1 の論理構造

要素 defect は変状の詳細を表し，属性として要素 location から参照するための ID と日本語名，英語名を持つ．また子要素として，検査日を表す date，変状写真を表す picture，変状ランクを表す rank，変状原因を表す mainCause，そして変状量を表す quantity を持つ．

5.3.2 アプリケーションの構築

BridgeML を用いたアプリケーションとして，前節で述べたプロトタイプシステムでは，BridgeML から HTML および POCXML への変換を行った．

(1) BridgeML の記述例

BridgeML の記述例を図 5.17 に示す．ここでは橋梁 Abegawa-B を取り上げているが，記述する情報は変状の位置を表す location 要素と変状の詳細を表す defect 要素に分かれる．

```

<?xml version="1.0" encoding="EUC-JP"?>
<!DOCTYPE bridge SYSTEM "file:///usr/local/tomcat-4.0.3/webapps/engineer/WEB-INF/bmc.dtd">
<bridge name="Abegawa-B">
  <location girderNb="5" ref="JTS01697270501439141001">
    <member name.en="connection between lines" name.ja="上下線連結材"/>
    <component1 name.en="" name.ja=""/>
    <component2 name.en="" name.ja=""/>
    <component3 name.en="" name.ja=""/>
  </location>
  <location girderNb="2" ref="JTS01697270201437351005">
    <member name.en="bearing area" name.ja="支承部"/>
    <component1 name.en="bearing plate" name.ja="ベアリングプレート"/>
    <component2 name.en="" name.ja=""/>
    <component3 name.en="" name.ja=""/>
  </location>
  <defect location-ref="JTS01697270201437351005" name.en="damage" name.ja="損傷">
    <date>1994-07-14</date>
    <picture>blank.gif</picture>
    <rank>B</rank>
    <mainCause>建設時の据付不良と思われる。</mainCause>
    <quantity>すき間約0.5mm</quantity>
  </defect>
  <defect location-ref="JTS01697270501439141001" name.en="damage" name.ja="損傷">
    <date>1995-01-09</date>
    <picture>abegawB0501.jpg</picture>
    <rank>A2</rank>
    <mainCause>支点沈下によるものと思われる。</mainCause>
    <quantity>l=40.25mm, l=5.34mm</quantity>
  </defect>
  <defect location-ref="JTS01697270501439141001" name.en="damage" name.ja="損傷">
    <date>1995-01-09</date>
    <picture>abegawB0502.jpg</picture>
    <rank>A2</rank>
    <mainCause>支点沈下によるものと思われる。</mainCause>
    <quantity>l=40.25mm, l=5.34mm</quantity>
  </defect>
  <defect location-ref="JTS01697270501439141001" name.en="damage" name.ja="損傷">
    <date>1995-01-09</date>
    <picture>abegawB0503.jpg</picture>
    <rank>A2</rank>
    <mainCause>支点沈下によるものと思われる。</mainCause>
    <quantity>l=40.25mm, l=5.34mm</quantity>
  </defect>
</bridge>

```

図 5.17 BridgeML による橋梁情報の記述

(2) BridgeML から HTML への変換

BridgeML は XSL を介して HTML に変換することができる。本研究のプロトタイプでは、2 種類の変換を行った。一つは変状一覧を表示するためのもので、もう一つは詳細な変状記録を表示するためのものである。

変状一覧を表示するためには、図 5.18 に示す defectsList.xsl を用いた。なお変換された HTML 出力を図 5.19 に示す。ここでは BridgeML の location 要素を表形式で表し、一行が一つの location 要素に対応する。

変状記録を表示するためには、図 5.20 に示す defect.xsl を用いた。なお変換された HTML 出力を図 5.21 に示す。ここでは一つの変状に対し ID を参照することで location 要素と defect 要素を組み合わせ、変状位置と詳細を表示する。

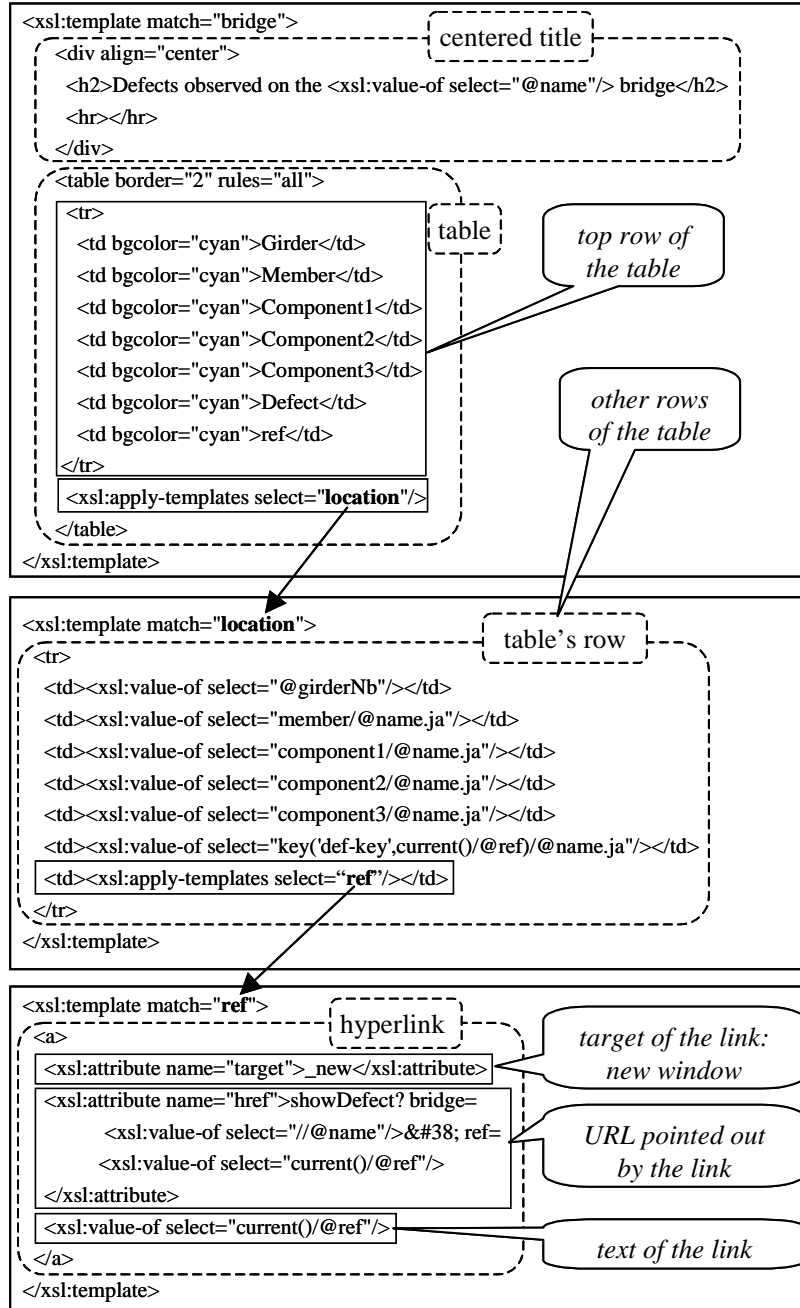
(3) BridgeML から POCXML への変換

BridgeML から POCXML への変換を行うために図 5.22 に示す pos.xsl を作成した。変換された POCXML を図 5.23 に示す。POCXML ではルートに scenario 要素があり、子要素として header および opinion がある。本研究では opinion 要素に変状の位置と詳細をマッピングした。


```

<?xml version="1.0" encoding="EUC-JP"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
<xsl:key name="def-key" match="/bridge/defect" use="@location-ref"/>

```



```

</xsl:stylesheet>

```

☒ 5.18 defectList.xsl

Defects observed on the Abegawa-B bridge

Order	Member	Component1	Component2	Component3	Defect	ref
5	上下線連絡材				損傷	JTS01697270201438141001
2	支承部	ヘアリングプレート			損傷	JTS01697270201437351005
1	シュー	取付部			塗膜剥離	JTS01697270101437351003
2	主桁	上フランジ	枕木支部		損傷	JTS01697270201437351006
1	支承部	ソールプレート	取付部		き裂	JTS01697270101437351001
2	主桁	ダイヤフラム			損傷	JTS01697270201437351004
4	シュー	取付部		リベット・ボルト	損傷	JTS01697270401438811001
1	支承部	シュー座		モルタル	損傷	JTS01697270101437351002
4	支承部	シュー座		モルタル	損傷	JTS01697270401438811002

図 5.19 変状の一覧表示


Defect observed on the bridge on 1994-07-14

On that day, the defect was judged as rank A2

The main cause is
ダイヤフラムの面外振動によるものと思われる。

Quantitative observation:
l=36mm

Member	Component1	Component2	Component3
主桁	ダイヤフラム		



Image

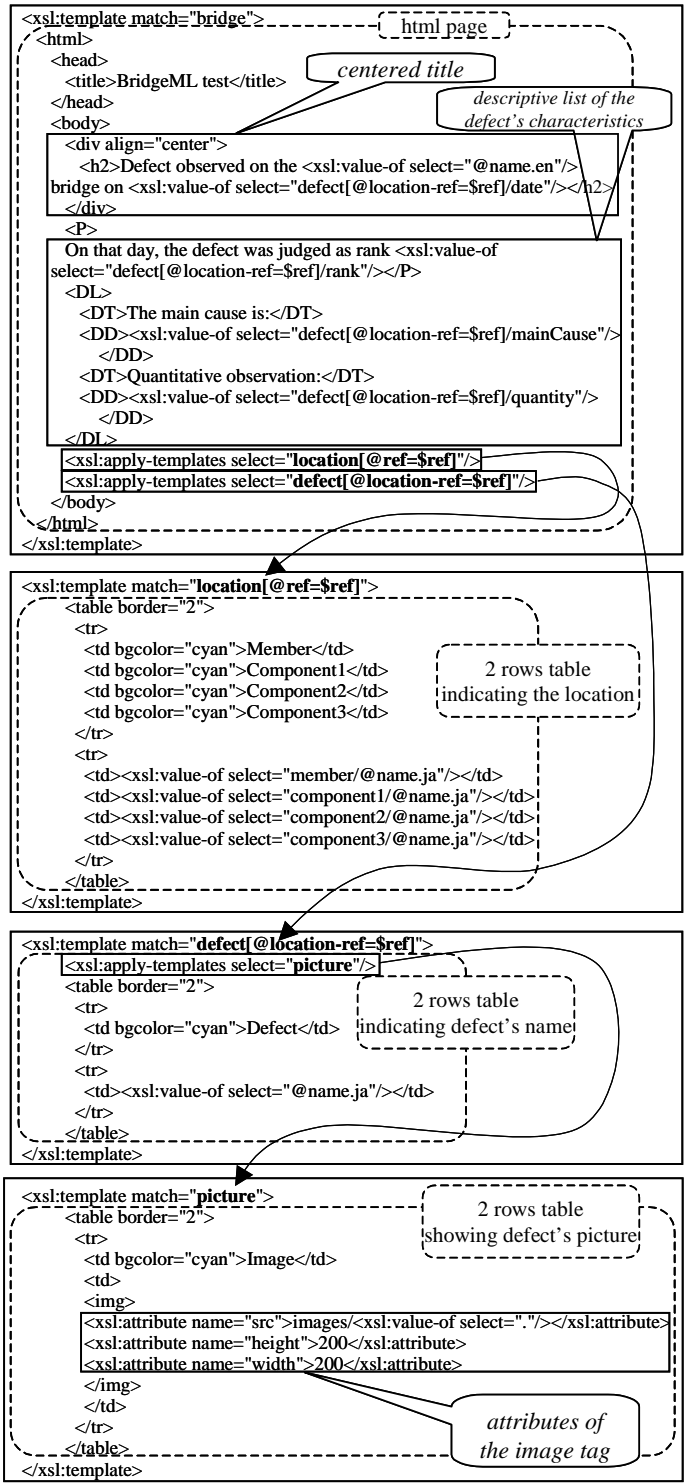
Defect
損傷

図 5.21 変状記録

```

<?xml version="1.0" encoding="EUC-JP"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
<xsl:param name="ref">JTS01697270201437351004</xsl:param>

```



```

</xsl:stylesheet>

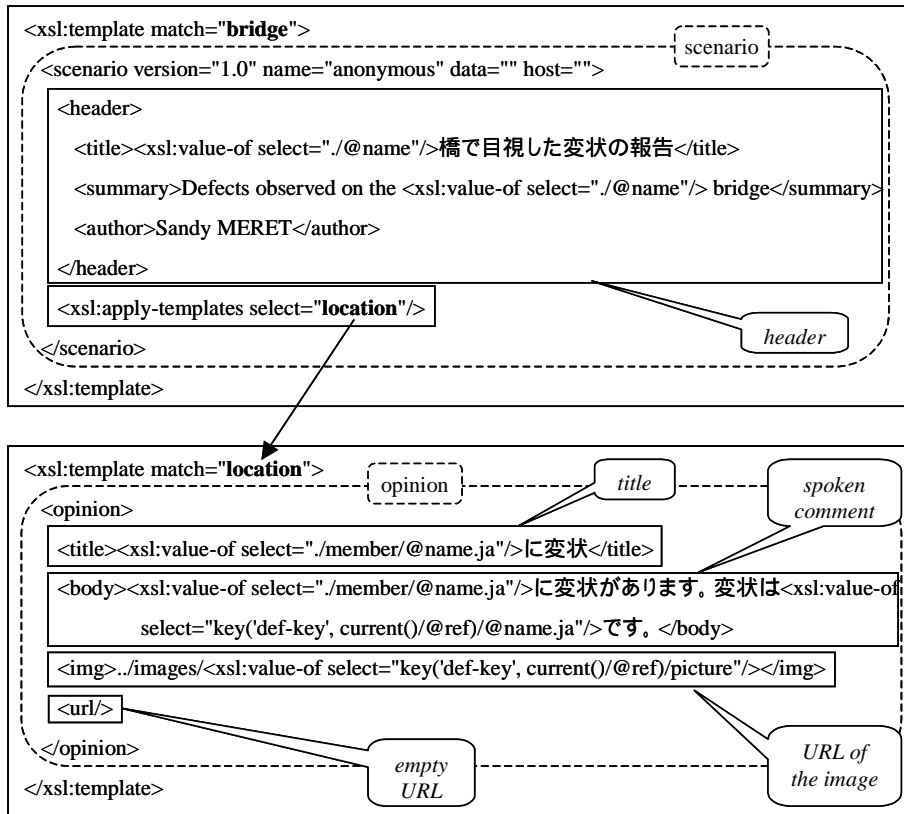
```

図 5.20 defect.xsl

```

<?xml version="1.0" encoding="Shift_JIS"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
<xsl:key name="def-key" match="/bridge/defect" use="@location-ref"/>

```



```

</xsl:stylesheet>

```

図 5.22 poc.xsl

```

<?xml version="1.0" encoding="UTF-8"?>
<scenario host="" data="" name="anonymous" version="1.0">
  <header>
    <title>Abegawa-B橋で目視した変状の報告</title>
    <summary>Defects observed on the Abegawa-B bridge</summary>
    <author>Sandy MERET</author>
  </header>
  <opinion>
    <title>上下線連結材に変状</title>
    <body>上下線連結材に変状があります。変状は損傷です。</body>
    <img>images/abegawB0501.jpg</img>
    <url/>
  </opinion>
  <opinion>
    <title>支承部に変状</title>
    <body>支承部に変状があります。変状は損傷です。</body>
    <img>images/blank.gif</img>
    <url/>
  </opinion>
  <opinion>
    <title>シューに変状</title>
    <body>シューに変状があります。変状は塗膜割れです。</body>
    <img>images/abegawB0106.jpg</img>
    <url/>
  </opinion>
  <opinion>
    <title>主桁に変状</title>
    <body>主桁に変状があります。変状は損傷です。</body>
    <img>images/abegawB0205.jpg</img>
    <url/>
  </opinion>
  <opinion>
    <title>支承部に変状</title>
    <body>支承部に変状があります。変状はき裂です。</body>
    <img>images/abegawB0108.jpg</img>
    <url/>
  </opinion>
  <opinion>
    <title>主桁に変状</title>
    <body>主桁に変状があります。変状は損傷です。</body>
    <img>images/abegawB0206.jpg</img>
    <url/>
  </opinion>
  <opinion>
    <title>シューに変状</title>
    <body>シューに変状があります。変状は損傷です。</body>
    <img>images/abegawB0404.jpg</img>
    <url/>
  </opinion>
  <opinion>
    <title>支承部に変状</title>
    <body>支承部に変状があります。変状は損傷です。</body>
    <img>images/abegawB0102.jpg</img>
    <url/>
  </opinion>
  <opinion>
    <title>支承部に変状</title>
    <body>支承部に変状があります。変状は損傷です。</body>
    <img>images/abegawB0401.jpg</img>
    <url/>
  </opinion>
</scenario>

```

図 5.23 POCXML による橋梁情報の記述

5.3.3 BridgeML ドラフト 2

BridgeML ドラフト 1 はプロトタイプアプリケーションの中で橋梁情報を記述するために用いたが、その構造は単純であり、実際に橋梁マネジメントシステムで用いるには不十分である。ドラフト 2 ではより詳細な情報を記述できるように仕様を策定した。

(1) 策定方法

ドラフト 1 と同様にルート要素は bridge であり、その下に当該橋梁に関する情報が含まれる。すなわち一つの XML ファイルが一つの橋梁を記述する。BridgeML は橋梁の時管理を対象としているため、重要な要素は変状であり、defect 要素に記述する。変状はその詳細以外にも、変状位置という空間的要素と検査日という時間的要素を持つ。そのため空間的要素を location 要素で記述し、時間的要素を inspection 要素で記述する。また inspection 要素には検査日以外にも検査員などの検査に関する情報を記述する。

一つの変状位置に複数の変状が存在し、また一つの検査情報は複数の変状情報を含む。そこで、情報の冗長性を省くために location 要素、inspection 要素、defect 要素を独立して記述し、location 要素と defect 要素および inspection 要素と defect 要素を 1 対多の関係で表現する。XML の木構造ではこれら 3 つの要素はすべて bridge 要素の子要素として同じレベルに位置する。これらの要素を関連づけるために、location 要素と inspection 要素に ID を付与し、defect 要素からこれらの ID を参照する。

変状を記述する defect 要素は、子要素として 3 つの要素を持つ。それらは検査員による調書である observation 要素、橋梁技術者による診断を記述する diagnosis 要素、橋梁管理者による対処などの意志決定を記述する decision 要素である。図 5.24 に文書型定義(DTD)、図 5.25 に論理構造を木構造に表したものを示す。

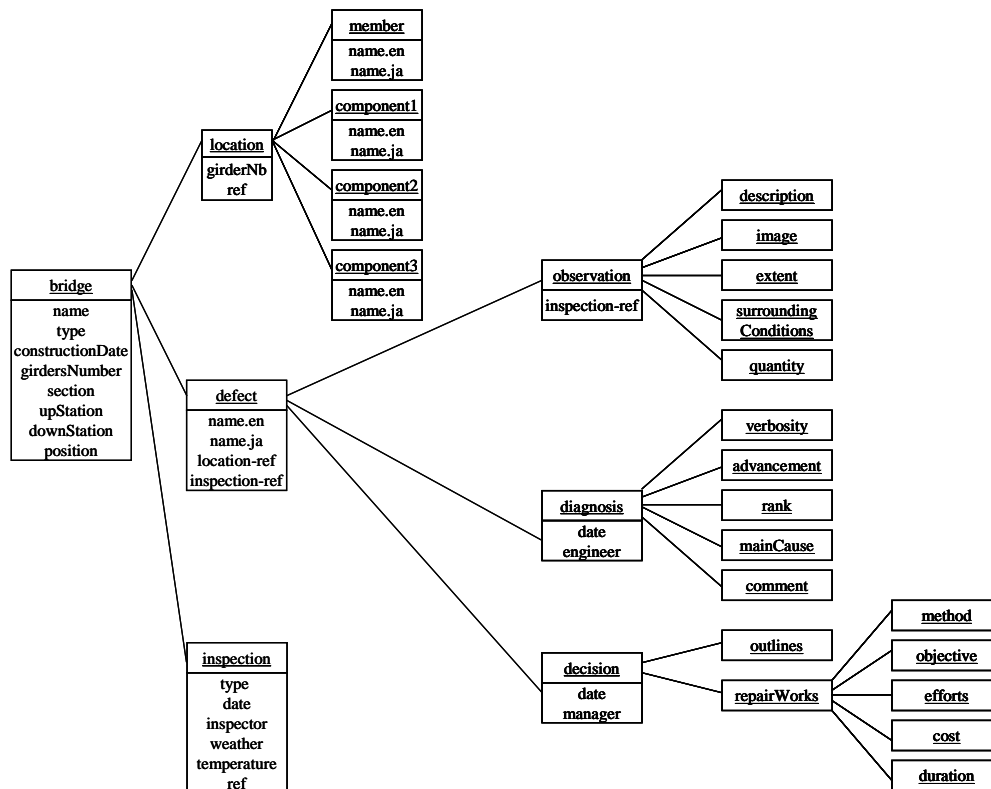


図 5.25 BridgeML ドラフト 2 の論理構造

```

<!ELEMENT bridge (defect|location|inspection)>
<!ATTLIST bridge name CDATA #REQUIRED>
<!ATTLIST bridge type CDATA #IMPLIED>
<!ATTLIST bridge constructionDate CDATA #IMPLIED>
<!ATTLIST bridge girdersNumber CDATA #IMPLIED>
<!ATTLIST bridge section CDATA #IMPLIED>
<!ATTLIST bridge upStation CDATA #IMPLIED>
<!ATTLIST bridge downStation CDATA #IMPLIED>
<!ATTLIST bridge position CDATA #IMPLIED>

<!ELEMENT location (member, component1, component2, component3)>
<!ATTLIST location girderNb CDATA #REQUIRED>
<!ATTLIST location ref IDREF #REQUIRED>

<!ELEMENT member EMPTY>
<!ATTLIST member name.en CDATA #IMPLIED>
<!ATTLIST member name.ja CDATA #IMPLIED>

<!ELEMENT component1 EMPTY>
<!ATTLIST component1 name.en CDATA #IMPLIED>
<!ATTLIST component1 name.ja CDATA #IMPLIED>

<!ELEMENT component2 EMPTY>
<!ATTLIST component2 name.en CDATA #IMPLIED>
<!ATTLIST component2 name.ja CDATA #IMPLIED>

<!ELEMENT component3 EMPTY>
<!ATTLIST component3 name.en CDATA #IMPLIED>
<!ATTLIST component3 name.ja CDATA #IMPLIED>

<!ELEMENT defect (observation|diagnosis|decision)>
<!ATTLIST defect location-ref ID #REQUIRED>
<!ATTLIST defect inspection-ref ID #REQUIRED>
<!ATTLIST defect name.en CDATA #IMPLIED>
<!ATTLIST defect name.ja CDATA #IMPLIED>

<!ELEMENT observation (description|image|extent|surroundingConditions|quantity)>
<!ATTLIST observation inspection-ref ID #REQUIRED>

<!ELEMENT description (#PCDATA)>
<!ELEMENT image (#PCDATA)>
<!ELEMENT extent (#PCDATA)>
<!ELEMENT surroundingConditions (#PCDATA)>
<!ELEMENT quantity (#PCDATA)>

<!ELEMENT diagnosis (verbosity|advancement|rank|mainCause|comment)>
<!ATTLIST diagnosis date CDATA #IMPLIED>
<!ATTLIST diagnosis engineer CDATA #IMPLIED>

<!ELEMENT decision (outlines|repairWorks)>
<!ELEMENT outlines (#PCDATA)>
<!ATTLIST decision date CDATA #IMPLIED>
<!ATTLIST decision manager CDATA #IMPLIED>

<!ELEMENT repairWorks (method|objective|efforts|cost|duration)>
<!ELEMENT method (#PCDATA)>
<!ELEMENT objective (#PCDATA)>
<!ELEMENT efforts (#PCDATA)>
<!ELEMENT cost (#PCDATA)>
<!ELEMENT duration (#PCDATA)>

<!ELEMENT inspection EMPTY>
<!ATTLIST inspection type CDATA #IMPLIED>
<!ATTLIST inspection date CDATA #IMPLIED>
<!ATTLIST inspection inspector CDATA #IMPLIED>
<!ATTLIST inspection weather CDATA #IMPLIED>
<!ATTLIST inspection temperature CDATA #IMPLIED>
<!ATTLIST inspection engineer CDATA #IMPLIED>
<!ATTLIST inspection engineerDate CDATA #IMPLIED>
<!ATTLIST inspection ref IDREF #IMPLIED>

```

図 5.24 BRIDGEML ドラフト 2 の DTD

(2) 仕様の詳細

(a) bridge 要素

ルート要素は bridge であり，橋梁の情報を保持，属性として橋梁の諸元を持つ．属性は次のようになる．

- name：橋梁名
- type：橋梁形式
- constructionDate：架設年月日
- girdersNumber：桁番号
- section：橋梁が位置する路線名
- upStation, downStation：橋梁の前後にある駅名
- position：起点からの距離

子要素として，defect，location，inspection を持つ．location 要素はドラフト 1 と同じである．defect 要素は項目を追加・修正し，inspection 要素は新たに加えた．

(b) defect 要素

defect 要素は次の 4 つの属性を持つ．

- name.en, name.ja：それぞれ変状名の英語表記，日本語表記
- location-ref：location 要素を参照するための ID
- inspection-ref：inspection 要素を参照するための ID

子要素として，observation，diagnosis，decision を持つ．

(c) observation 要素

observation 要素は，検査員が変状を発見した際の調書に相当する．属性として検査情報である inspection 要素を参照するための ID を持つ．また，次のテキストデータで記述された子要素を持つ．

- description：検査員による変状の概要説明
- image：変状写真のファイル名(パス名)
- extent：変状の定性的評価
- surroundingCondition：変状の周辺環境に関する説明
- quantity：変状量(き裂長，脱落ボルト数など定量的評価)

(d) diagnosis 要素

diagnosis 要素は，橋梁技術者の変状に対する評価・診断を記述する．属性として評価・診断を行った日付および担当技術者名を持つ．子要素は次のようになる．

- verbosity：変状の冗長性
- advancement：変状の進行性
- rank：変状の評価
- mainCause：変状原因
- comment：技術者のコメント

(e) decision 要素

decision 要素は，橋梁管理者の変状に対する判断・意志決定を記述する．属性として属性として判断・意志決定を行った日付および管理者名を持つ．子要素は次のようになる．

- outline：実施する対策に関する記述
- repairWorks：補修・補強工法に関する記述

(f) repairWorks 要素

repairWorks 要素は変状に対する補修・補強工法を記述し，次の子要素を持つ．

- method：対策の種類(補修，補強，大規模改修)
- objective：対策工法のねらい
- efforts：対策内容および要する労力
- cost：対策費用
- duration：対策効果

(g) inspection 要素

inspection 要素は検査に関する情報を記述し，属性は次のようになる．

- type：検査種別(定期検査，詳細検査)
- date：検査日
- inspector：検査員
- weather：天候
- temperature：気温
- ref：defect 要素から参照するための ID

(3) ドラフト 2 のまとめ

BridgeML ドラフト 2 は仕様の策定では，2 つの点においてドラフト 1 と大きく異なる．一つは変状，変状位置，検査の分離であり，もう一つは検査員，技術者，管理者といった維持管理に携わるユーザによる情報の分離である．

ドラフト 2 では，変状 defect，変状位置 location，検査 inspection に関する要素をすべてルート要素下の同じレベルに配置し，それぞれを独立して記述した．これにより，まず橋梁の維持管理において，橋梁の物理的情報である location，構造物の状態を表す defect を分離することができる．検査情報 inspection の分離は，橋梁の状態に時間の概念を与えるために必要である．

defect 要素は子要素として，observation 要素，diagnosis 要素，decision 要素を持つが，それぞれ検査員，技術者，管理者が記述する情報に相当する．このように情報を分離することでユーザ間の協調作業を支援する．

5.3.4 BridgeML による情報交換

BridgeML を使用する利点として情報交換が容易になることが挙げられる．橋梁のライフサイクルは非常に長く，計画，建設，維持管理，撤去・架替えという一連のサイクルには検査専門会社，建設請負業者，コンサルタント，管理者等複数の組織が関与する．そのため，これら複数の組織間における情報の共有または伝達が必要不可欠である．現状では，複数の組織に共通した橋梁データを管理するためのフォーマットは存在せず，必要に応じてあるフォーマットから別のフォーマットへ変換するためのコンバータを用意する必要がある．これは必ずしも電子情報のコンバータではなく，手作業により紙媒体を扱うことも含まれる．一般に情報を変換するコンバータの作成は非常に労力とコストを要する．

複数の組織が橋梁の維持管理に関してそれぞれ独自のフォーマットで定義されたデータベースを持つ場合，図 5.26 の左図のようにそれぞれのデータベース間におけるコンバータの作成が必要である．このような多くのコンバータを作成するためには多くのコストや労

力を必要とし、またデータ交換時の変換ミスなどのリスクが大きくなる。それに対し、橋梁の維持管理に関するデータフォーマットを BridgeML で統一した場合、図 5.26 の右図のようにデータ共有・交換が可能になる。図に示されているように 6 つのデータベースを考えた場合、それぞれの間でデータ交換を行うためには 15 のコンバータが必要となる。それに対し、BridgeML を用いた場合、コンバータは 6 となる。

BridgeML は複数の組織間でのデータ共有・交換を容易にする。これは橋梁の維持管理において複数の組織が関わる場合、情報共有・交換に必要なコストを削減する。また、BMS やモニタリングシステムなど橋梁の維持管理に関わるアプリケーションを構築する場合に、データフォーマットを BridgeML に準拠させることで開発コスト工期を削減することができる。これはアプリケーション開発者だけでなく、利用者側にとってもアプリケーション導入コストが小さくなる利点がある。

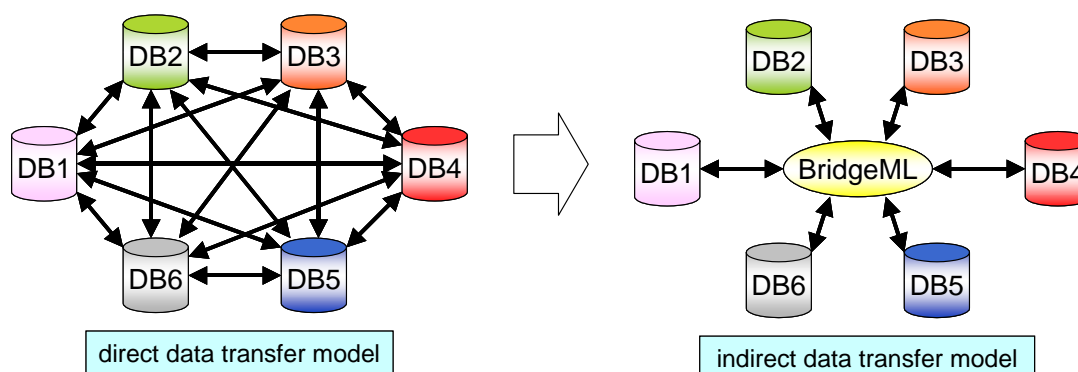


図 5.26 データ交換

5.4 まとめ

本研究では、今後増大すると予想される社会基盤構造物の維持管理を対象として、IT(情報技術)による維持管理支援の手法を提案した。

米国においては、すでに橋梁の維持管理の重要性を認識し、維持管理情報を蓄積・管理し、維持管理計画の最適化を支援するツールを提供する橋梁マネジメントシステム(BMS)の開発・運用を行っている。それに対し日本においては実用的な BMS の運用は行われていない。本研究では、BMS の実現に向けてマルチモーダルアクセスと BridgeML という 2 つの重要な機能を提案した。

BMS は橋梁維持管理における情報を管理するシステムであり、様々なユーザおよび利用形態が考えられる。本研究では、PC だけでなく、PDA や携帯電話による利用を想定したシステムアーキテクチャを提案した。また、プロトタイプを作成し、PC を利用した橋梁変状の参照や、音声による情報参照を可能にした。

橋梁の維持管理に関する情報は様々な事業者・組織が管理する現状では、組織間の情報共有・交換には非常に大きなコスト・労力を要する。本研究では、橋梁の維持管理に関するデータフォーマットとして BridgeML を提案し、そのドラフトを作成した。またドラフトによるサンプルアプリケーションを作成し、その有効性を検証した。このような橋梁の維持管理に関する共通の言語を策定することは、複数の事業者・組織間での情報交換を促進する。

参考文献

- A.Miyamoto: Development of a Bridge Management System (J-BMS) in Japan, Life-cycle Cost Analysis and Design of Civil Infrastructure Systems, 2000.

- P.D.Thompson: Pontis Bridge Management System, Pacific Rim TransTech Conference Proceeding
- Kushida, Miyamoto, and Kinoshita: Development of Concrete Bridge Rating Prototype Expert System (BREX) in Japan, Journal of Computing in Civil Engineering, 1997.
- Kawamura, Nakamura, and Miyamoto: Concrete Bridge Rating Expert System with Neuro-Fuzzy Hybrid Systems and its Application, The Sixth International Conference on the Application of AI to Civil and Structural Engineering, 2001.
- J.H.Garrett, and A. Smailagic: Wearable Computers for Field Inspectors – Delivering data and Knowledge-based support in the Field.
- T.Harada, Y.Tomita, and K. Yokoyama: Modeling of Maintenance Strategies for Highway Bridges by Using Multi-Layer Genetic Algorithms.
- S.Chikuyama, H.Furuta, and M.Dogaki: Application of Virus Evolutionary-genetic Algorithms to a Decision Support System for the Maintenance Program of Road Bridges.
- N.Yabuki and M.Furukawa: An Integrated Design and Project Management System for Prestressed Concrete Hollow Slab Bridges, The Ninth International Conference on Computing in Civil and Building Engineering.
- N.Yabuki and T.Shitani: An Integrated Design System for Connections of Steel Structures, ASCE 2001 FIAPP Specialty Conference.
- N.Fujiwara, T.Nishida, S. Azechi, K.Sumii, T.Hirata, and H.Yano: Public Opinion Channel (POC) Facilitates Communications in the Networked Community, The Third International Cognitive Technology Conference, pp. 141-148, 1999.
- A.Michard: XML language and applications, Eyrolles, 2000.

6 分散型情報技術による社会基盤施設防災・保全現場のインテリジェント化

6.1 はじめに

6.1.1 小規模な管理対象構造物の存在

維持・管理活動の対象となるのは、橋梁やトンネルといった構造物本体ばかりではない。例えば、道路には照明柱・標識柱・フェンス・監視カメラなどといった付属物が多数存在している。例えば照明柱は、十分な照明効果を得るために数十 m～100m おきに設置されており、ひとつの橋梁に数基～数十基が設置されている。よって、付属物の数は先に挙げた橋梁などの大規模構造物の数とは桁が異なる。単価が低いため、建設費を越えるような維持・管理手法を用いることは合理的とは考えられず(破損した際に交換した方が低コストとなる)、また数が膨大であるためにひとつひとつの小規模構造物に高コストな検査手法を用いることは経済的に不可能である。

しかし、これらの小規模構造物も、事故を起こせば大規模構造物と同様に少なからぬ損害を発生させることになる。照明柱が落下すれば車輛や人には致命的な損害を与えることになり、また事故のために通行が不可能になるなどの社会的影響は決して小さいものではない。照明柱や標識柱の破損はいくつも事例があり、その原因が交通荷重によるものか、あるいは暴風や地震などの災害によるものなのかといった切りわけさえできていないのが現状である。

道路付属物は道路管理者に責任の所在がある。この仕組みの下で低コストに安全性を確保するには、検査・モニタリングを効率よく行うシステムを新たに構築する必要がある。

6.1.2 従来の保全活動の問題点

(1) 従来の検査技術の問題点

従来の保全活動は、人手による検査(目視検査/打音検査など)や非破壊検査(超音波探照試験/磁粉探照試験など)を定期的、あるいは危険と判断される箇所に重点的に行うことにより損傷を検出/損傷程度を判断し、それらの資料を元に補修を計画・施工してきた。

目視・打音検査といった人手による検査手法は、経験と勘を頼りにするために客観性に乏しく、検査の精度は検査員のスキルに大きく依存する。このため、安定したデータを得ることは難しい。また、熟練した検査員の育成には長い時間がかかること、現場での作業は苛酷な環境にさらされることから、人的コストが莫大になることも問題である。

これらの問題点を解決するために、客観的な検査手法やモニタリング技術の開発が行われている。筆者の所属研究室で開発しているレーザードップラー振動計によるモード解析手法や、圧電セラミック素子を用いたモニタリング手法、光ファイバーセンサによる歪み計測などはその例である。

また、近年のパーソナルコンピュータの低価格化により、変位計・振動計といったセンサとコンピュータによるモニタリングシステムも比較的安価に構築が可能となっている。

しかし、これらの手法も上に挙げた「膨大な数の構造物のモニタリング」に適應するにはまだまだ高コストであり、より安価なモニタリングシステムの構築が望まれる。

今後も「高コストだが高精度」な検査技術の研究開発は進むであろう。それは重要度の高い構造物における精密検査に欠かせない基礎技術である。同時に、「実用的な精度でか

つ低コスト」な検査技術を開発し、定期的な検査には後者を、重点検査には前者を用いる「ハイ&ローミックス」を実現することにより、高度な社会基盤維持・管理戦略を立てることが可能となる。

このような安価な検査手法はどのようなものになるだろうか。シーズ側で注目されるのは、近年発達の著しい MEMS センサである。マイクロマシニング技術によって作られる小さく安価なセンサを MEMS センサと呼ぶ。MEMS は量産効果により価格が下げられるため、大量のセンサをあちこちに配置することにより空間密度的にリッチな情報を比較的安価に得ることが可能になることが予測される。「膨大な数の構造物のモニタリング」に適した手法である。

しかし、膨大な管理対象構造物に対して、時間的密度の高い情報を刻一刻と送る能力のあるセンサ群を配置し、それらの情報を集中管理することは不可能である。データ転送のために大きな帯域を必要とし、またデータの洪水の中から意味のある情報を拾い出すシステムを構築しなければならない。この問題を解決するためには、センサとコンピュータとをパッケージ化し必要な情報のみをデータとして報告するインテリジェントなセンサユニットを構築する必要がある。

(2) ユビキタス時代の検査活動

ユビキタス(Ubiquitous)とは「遍在」を意味するラテン語である。この言葉は、あらゆる場所・あらゆる物にコンピュータが存在すること、それらがネットワークで結ばれていることを意味しており、コンピュータ技術が実現する未来像として広く使われるようになった。Internet というグローバルなコンピュータネットワークは後者の前提となる技術であり、そのインフラ整備は IT 国家を標榜するわが国の国家目標となっている。同様の概念をパーヴェイシヴ(Pervasive)と呼ぶ向きもある。

ユビキタスの実現する世界は、あらゆる情報が場所を選ばず引き出せる世界である。一般的には、個人の欲している情報(その人の仕事に関わる情報や、趣味の情報)を、自宅であれオフィスであれ外出先であれ自由に引き出せる社会や、もっと公共性の高い情報についてだれでもどこからでもアクセスできる状態がイメージされているが、社会基盤施設の情報は公共性の高いものでありこれについてもだれもがどこからでもアクセスできる状態が望ましいことは言うまでもない。日本政府の推進する e-Japan 戦略では、高速ネットワーク網の整備とユビキタスを実現するコンテンツ面・技術面の枠組み作りがうたわれている。社会基盤施設の情報流通をこのような戦略面から捉えた場合、高精度な地理データの構築を行う「電子国土」構想と併せて、現実の情報を取得できる軽量情報端末の開発が必要不可欠である。

6.1.3 研究の目的

本研究の目的は、以上のような背景の下、MEMS センサによるモニタリングと、情報通信技術を基盤とした分散協調型の情報技術を用いて社会基盤設備の維持・管理を現場から合理化するシステムを構築することである。

ここで言う分散協調型情報技術とは、近年発達の著しいパーソナルコンピュータやマイクロコントローラといった安価で小規模なデバイスを分散配置し、これらが協調動作することによりシステムを構成するものである。構成要素となるデバイスは単なるテレメータとして動作するのではなく、それぞれがインテリジェントな情報処理を行うことによって協調動作を行う。

情報処理の社会基盤設備の維持管理への適用は、欧米を中心に積極的に進められている。しかし、それらのシステムは現場で直接使うものではなくオフィスでの作業を前提としている。

本研究では現場での使用を想定し、意思決定支援やヘルスマニタリング情報をリアルタイムに得ることを目指す。自然災害の多い我が国においてはこのような特色を持ったシステムは非常に有用であると考えられる。現場での使用が前提となるため、一般的な情報システムの構成要素であるパーソナルコンピュータやワークステーションをそのまま使うことはできない。このため、屋外環境での使用に耐えうる情報機器の開発が必要となる。

本研究では橋梁の防災・保全現場を例として、社会基盤設備の現場におけるさまざまな意思決定、および現場から得られる情報を用いた戦略的決定支援を可能とするシステムの構築を目的とする。

老朽化しつつある社会資本ストックの維持のためには、長期的な戦略が必要となる。様々なシナリオの中から経済性・利便性に優れるものを選択するために、現場からの情報を統合し判断することは必須である。

情報システムを導入することにより、現場での作業の合理化や高いサービス水準を実現する。建設・防災・保全現場において、熟練技術者の経験を活かすようなシステム作りは重要である。

振動・変位といったデータを計測することは、橋梁の損傷度判定・余寿命診断の精度向上に役立つ。しかし、従来の手法ではモニタリングにはデータ回収・整理のためのコストが高かった。情報通信技術の応用により、これらの問題を解決することができる。

本研究で開発を行うシステムは、マイコン(マイクロコンピュータ/コントローラ)と近年発達の著しい MEMS センサーデバイスを用い、安価に実用的精度で計測を行うことを目標とする。

MEMS 機器は、安価かつ低消費電力であることを特徴とし、従来のものと比べて精度も遜色はない。安全性も高く、モニタリングの核となる技術である。

21 世紀のマイコン技術は、ネットワーク接続機器さえ高価な PC を用いずに構築できるレベルに達している。PC を屋外環境で稼動することは現実的でないため、現場に対応したインテリジェントなシステムを構築するのに必要不可欠な技術である。

本研究では、道路付属物のモニタリングを対象に、インテリジェント加速度計測システムの開発を行い、分散型モニタリングシステムの有用性を検証する。

6.2 簡易加速度計測システム(RIMS)

6.2.1 構築の目的

劣化の進む膨大な量の社会資本ストックの維持のためには、安価で効率的なモニタリング手法が必要不可欠である。本研究では、適用可能な精度を持ち、かつ安価な計測システムの提案を行う。

本研究では、分散型の計測システムを提案する。小規模な自律型計測モジュールを通信技術により組み合わせ、整理された計測データを低コストに得られるようにすることを目的とする。

例として、簡易加速度計測システムの構築を行う。加速度計を選んだのは、次のような理由による。

- 加速度計は計測器としては比較的複雑な機構を持ち、事例として適切である。
- 近年、半導体を用いた加速度計が製品化されており、小規模計測機の例として研究が行いやすい。

本研究で提案するシステムは、次のような点で簡易的なものである。従来、センサーシステムはある物理量を記録し(データロガー)、あるいは遠隔地にその物理量を伝送する(テレメトリ)ものであった。しかし、実際にわれわれが必要としている情報は直接的な物理量ではなく、検査対象構造物がどのくらい健全であるか、どれだけの耐荷力・余寿命を持っているかといった情報である。そのため、これらの情報に結びつくインデックスが重要と

なる．単にテレメトリ装置を構築しても膨大なデータが垂れ流されるだけであり，データを受信する情報センターでインデックス作成を行うといった作業をしない限りモニタリング情報が活用されるとは考えにくい．

そこで，本研究では計測部とインデックスを計算するロジックとを小さなパッケージに納めた装置を作成し，これを対象構造物に取り付けるだけでインデックス情報を取り出すことのできるシステムを提案する．以後，このシステムを RIMS(Remote Intelligent Monitoring System)と呼ぶことにする．

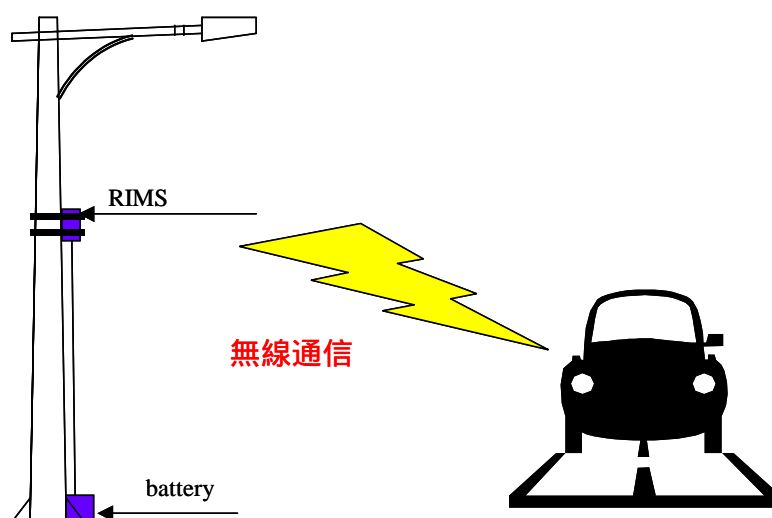


図 5.1 RIMS 概念図

6.2.2 RIMS の特徴

RIMS の特徴として，取り付け工事および情報取得が低コストであることが挙げられる．センサ・情報処理部・通信部が一体となったパッケージを取り付けるだけで工事完了である．従来のデータロガー方式では，センサを対象構造物に取り付けてからデータロガーのある観測小屋までケーブルを引き込む工事が必要であった．RIMS では大掛かりな工事は不要である(後述するレインボーブリッジにおける実証実験では，1 時間弱で取り付け工事は終了した)．

インデックス情報のみを通信することにより通信量を減らし，低コストかつ容易に対象構造物の健全度を取得できることを目標とする．具体的には，インデックス情報として閾値を超えるピーク加速度を記録した度数を用いることにし，また通信部に無線通信を採用して近接通過車両から情報を取得できるようにする．

6.2.3 開発目標

本システムの目的は，従来計測不可能だった構造物のモニタリングデータ取得を可能にすることと，最小限のコストでデータを回収できるようにすることである．

これらの目的を達成するためには，すべての計測器がネットワークで接続されていることが理想的である．実際，IPv6 によるネットワークが普及すれば，このようなインテリジェント機器はすべてインターネットに接続されることが予想される．

しかし、現状ではネットワークの存在を前提にすることは現実的ではない。例えば、我が国の橋梁の過半は山間部にあり、携帯電話での通信も不可能である場合が多い。このような状況下ではバックボーンとしての通信設備は期待できない。

そこで、本システムでは、データを蓄積し検査員が定期的にデータ回収を行うバッチ型システムとして設計を行い、同時に将来のネットワークインフラの整備を見越して、オンラインでの使用も可能なように仕様を決めることにする。

6.2.4 RIMS の構成

加速度計測システムは、加速度計測モジュール、コントローラ、通信システムからなる。これらの要素は、独立したモジュールとして設計するため、他のシステムにも容易に転用可能である。

- 加速度計測部：MEMS 加速度計を用い、加速度を計測するモジュールである。
- コントローラ：制御用マイクロコンピュータを用い、計測データの管理を行う。本システムはバッチ型であるため、計測したデータを時系列として保存しておくことは現実的に不可能である。そこで、なんらかの演算を行い、計測データを整理し情報量を圧縮しておく必要がある。本システムでは、加速度を閾値による度数分布として整理することにする。つまり、加速度の絶対値を計測し、これらのピークが閾値を越えた回数をカウントし、記憶する。閾値は、例えば 100gal, 200gal, 500gal, 1000gal というようにいくつかのカテゴリに分類・記憶できるようにする。また、データ回収の際に管理しているデータを外部に出力するのもコントローラの役目である。
- 通信システム：データ回収コストを最小限にとどめるため、無線通信によりコントローラのデータを近づいただけで回収できることを目標とする。

6.3 RIMS の前提技術

6.3.1 MEMS 加速度計概論

加速度計には MEMS 加速度計と呼ばれるカテゴリのものを使用する。MEMS とは、Micro Electric Mechanical System の略で、マイクロマシニング技術（印刷や電子ビーム加工などの超微細加工技術）により製作される超小型機械の総称である。応用例としてはセンサ(温度, pH, 圧力, etc.)やアクチュエータ(静電アクチュエータ, 圧電アクチュエータ etc.)がある。

加速度センサは MEMS センサアプリケーションの一種である。ニュートンの法則 ($F=ma$) に基づき慣性力による変形を検出するもので、検出軸の数から 1 軸型・2 軸型・3 軸型の種類が、また検出原理により piezo 抵抗型・静電容量型・圧電型の種類がある。通常、検出軸と検出原理とで種別が行われる。圧電型では静的加速度を検出することができないが piezo 抵抗型・静電容量型では可能であることから、検出原理でこの違いを表している。

piezo 抵抗型加速度センサは、piezo 抵抗素子に力が加わると抵抗が変化する現象から変形を検出する。1 軸加速度センサでは錘のついたカンチレバーの変形を piezo 抵抗素子で検出する。3 軸型では錘をダイヤフラム上に配置し、歪みを piezo 抵抗素子で検出する。静電容量型加速度センサでは、錘により変位する部分に変位電極を設け、固定電極との距離の変化により静電容量が変化するから変形を検出する。

MEMS 加速度センサは、従来のサーボ型・フォースバランス型などの非 MEMS 加速度計と比較して、小型・低消費電力・安価といった特徴を持つ。マイクロマシニング技術により製作されることから小型であることはもちろんであるが、微小な変形を検出する原理より消費電力が小さいという特徴にも繋がる。電流が小さいので感電などの事故も起こら

ない．単結晶シリコンなどから削り出して製作されるため破壊強度を大きくすることが可能で，レンジの広い計測が可能である．MEMS であることがすなわち安価であることには繋がらないのであるが，印刷技術の応用により同じ構造のものを大量に生産できる MEMS の特長からスケールメリットが生じ，大量生産するほど 1 個当たり製作コストを削減することができる．サーボ型加速度ピックアップが 10 万円前後であるのに対し，秋葉原で目にする静電容量型 2 軸加速度センサ ADXL202 はわずか 3000 円である．

MEMS 加速度センサの応用例としては，ロボットや車輛の加速度計測・制御，エアバッグのトリガー用途などが挙げられる．社会基盤施設用としては，東京ガスの超高密度リアルタイム地震防災システム「SUPREME」という，地震時防災用機器への応用例がある．入手が容易な加速度センサとしては，前述の ADXL202(アナログデバイス社)がある．静電容量 2 軸型のこのセンサは，エアバッグ用として開発され，ゲームなどにも利用されている． $\pm 2g$ までの加速度を検出でき，PWM 信号で出力する．精度があまりよくないこと，PWM 信号の扱いが難しいことが難点である．

RIMS には，マイクロストーン社の 3 軸加速度センサ，MA-3-04 を使用している．このデバイスは piezo 抵抗型 3 軸に分類される MEMS 加速度計で，人体振動検知やロボット制御用やゲームなどに広く使われている．サンプル価格で 1 個 25000 円であった．広いレンジ・低消費電力・小型といった特徴を持つ．MA3 シリーズは三軸加速度を電圧で計測できる事に加え，オペアンプ内蔵により出力範囲を $+0 \sim 5V$ としており，一般的な A/D コンバータに直結できる優れた設計となっている．この特徴により，RIMS では加速度計をコントローラに直結しただけの単純な回路構成で目標を達成することができた．

今回用いるものは，14 ピンの DIP パッケージ IC である．パッケージ内には錘とそれを支えるダイヤフラム構造が入っており，錘の慣性力による変形を電圧に変換しオペアンプで増幅し出力する．アプリケーション側では電圧を計測し，線形に加速度に変換するだけでよい．

MA-3 シリーズには，レンジによって $\pm 4G$ の 04， $\pm 1G$ の 10， $\pm 2G$ の 20 の三種類が存在する．今回はこれらのうち，小さな加速度を検出したかった事からもっともレンジの狭い 04 を採用した．

6.3.2 マイコン技術概論

(1) 組込みマイコンの概要

コントロールにはマイクロコンピュータ(以下，マイコンと略す)を用いる．マイコンは自動車や家電・オートメーション機器の制御に広く利用されており，インテリジェントな機器を開発するにあたってマイコン技術は欠かせないものであると言える．

マイコンの歴史は 1971 年発表の i4004 に始まる．当初は電卓を安価に製作するためにその構成要素であった ALU，メモリ，I/O などの部品を統合しストアドプログラム方式で柔軟に構成を変えられるようにしたものであったが，4bit 並列処理からスタートして現在では 64bit 並列処理となるなど高性能化が進み，計算機環境を一変させるに至った．多くの PC の心臓部に使われているのは i386 とその後継機に分類されるマイコンであり，PC はマイコンのアプリケーションとしてもっとも有名なものであるといえる．

しかし，PC よりも多くの数のマイコンが，前述したような機器の中で稼動しているのである．機器制御に用いられるマイコンは組込み用マイコンと呼ばれるカテゴリーのもので，PC の心臓部である i386 系列に代表される汎用マイコンとは異なり，より小さな能力のマイコンコアに次のような機構を追加したものである．

(a) 入出力(I/O)

組込みマイコンの主な用途は機器制御であるため，入出力機能が不可欠である．

もちろん、汎用マイコンにも入出力はあるのだが、専用のコントローラ IC を CPU の外側につけなければならない。組込みマイコンは、I/O コントローラを内蔵し、なかには A/D 変換機構を持つ物も存在する。

(b) メモリ (ROM/RAM)

機器に組込んで使用するという目的のため、部品点数を減らすことは電源の面からも実装スペースの面からも、さらに設計/製作コストの上でも重要なファクターである。

計算機の構成要素の中で大きな実装スペースを取るのがメモリであるが、組込みマイコンの多くはメモリを内蔵し、上記の目的を達成している。

ROM の種類には、製造時に内容を書き込んでしまう MaskROM、一度だけ書き込みが可能な PROM、書き込み/消去のくり返しが可能な(E)EPROM がある。

多くの組込みマイコンは、最低限の RAM しか搭載していない。このため、RAM はワークエリアとして固定的に使うなどのプログラミング上の工夫が必要となる。

(c) タイマ

機器制御では時間の計測が必要な局面がよくある(今回の加速度計における、計測間隔の生成はいい例である)。時間の計測のために、一定間隔でカウントアップ/ダウンする機構が用意されており、これをタイマと呼ぶ。精密な時間計測には欠かせない機能である。

(d) ウォッチドッグタイマ (Watch Dog Timer, WDT)

組込みマイコンの動作環境はノイズが多く、プログラムの暴走が避けられない。そこで、このような環境でも確実に動作するような工夫が必要となる。

WDT はこの矛盾する要請のもとに設計された特別なタイマである。自動的にカウントアップし、一定値になると WDT はマイコンをリセットする。リセットを防ぐために、プログラムは WDT をクリアしつづけなければならない。意図通りに動作しているプログラムであれば、WDT がカウントオーバーする前に WDT をクリアすることができるはずである。

この機構により、暴走時を検出して安全にシステムをリセットすることができる。

(e) 低消費電力機構

組込みマイコンは電池駆動を求められるなどシビアな電源環境で利用されるため、消費電力が小さいことが求められる。多くの組込みマイコンは処理速度よりも低消費電力であることを優先し、スリープモードなど特別な省電力機能を持つものもある。

(2) 組込みマイコンの具体例

本節では組み込みマイコンの例を挙げ、RIMS に用いるマイコン選択について述べる。

(a) Z80

Z80 は 1980 年代には PC 用として広く使われていた 8bit CPU である。現在でも高速化した Z80 をコアに WDT やタイマなどを内蔵した Z80 互換 CPU は組込み用として広く使われている。価格も CPU ボードだけなら 5,6 千円と非常に安価であり、組込み用途に向く。古くからのソフトウェア資産を多くかかえており、開発への不安も少ない。

しかし、現代的な OS サポートが存在しないこと、アーキテクチャが古く開発サイクルも ICE などの支援がなければ長期化すること、ROM が外付けとなるため開発にあたってはマイコンボードの他に ROM ライタが必要であることが欠点として挙げられる。

(b) PIC (Peripheral Interface Controller)

PIC は、MicroChip 社の組み込み用ワンチップマイコンの総称である。1990 年代初頭に登場し、PIC には、大別して 1. 12bit 命令セットのローエンドシリーズ、2. 14bit のミッドレンジシリーズ、3. 16bit のハイエンドシリーズの 3 種がある。これらのうち、ミッドレンジシリーズに位置するフラッシュメモリ搭載型マイコンは現在もっとも普及しているものである。このシリーズには、電子工作等によく使われる 16F84 やその後継にあたる 16F628、もっとも高機能な 16F877 などがある。それぞれのシリーズの中に内蔵するメモリ・I/O の違いによる様々な製品が存在する。16F84 はデジタルの汎用入出力ピンしか持たないのに対し、16F877 は A/D コンバータを内蔵している。

同じクラスのマイコンとしては、PIC とソフトウェア互換性があり高速動作の可能な UBICOM 社の SX、PIC よりも素直なインストラクションセットを持ち扱いやすい Atmel 社の AVR がある。PIC はこれらの中ではもっとも早く市場に登場したことから普及しているが、その特異なアーキテクチャゆえプログラム開発は容易ではなく、後発の AVR のほうが優れている点が多い。AVR が自由に扱える RAM を搭載しているのに対し、PIC は基本的に特殊用途用のレジスタしか持たない。このため、AVR では C 言語による開発が可能であるが、PIC ではアセンブラによる開発が前提となる。従来多数のディスクリット部品で構成していたコンポーネント回路の置き換えに適しており、単体でシステムを構成するような場合には向かないマイコンであるといえる。

(c) H8

H8 は日立製作所製の組み込み用マイコン群の名称で、消費電力の小さな H8/300L シリーズ、内部 16bit で高速処理が可能な H8/300H シリーズなどからなる。H8/300H シリーズはクロック 10~25MHz の高速処理機能に加え、A/D および D/A コンバータ内蔵、数十本の入出力端子、大容量の Flash Memory および RAM 搭載と高機能を使いやすくまとめているのが特徴である。C コンパイラなどの開発環境も日立製作所から純正品が比較的安価に提供されているのに加えて GNU C Compiler も利用が可能である。PIC や Z80 ではサブセットの C による開発を強いられる事になるが、H8/300H であれば一般的な C による開発が可能となるのである。価格も安く、近年では Z80 にかわって組み込み用に広く使われている。マイコン単価で 1000~2000 円、ボードでも 3000 円~20000 円程度で入手できる。

H8/300H シリーズのラインナップとしては、従来 3048F という製品がよく使われていた。ロボットなどの製作事例にはこのマイコンを用いたものが多い。現在は内蔵 ROM/RAM を強化した 3052F や、DRAM コントロール回路を内蔵した 3069F が製造されている。これらのマイコンは ROM512KB、RAM16KB と大容量の記憶装置を内蔵し、最新のプロセスで製造されているため動作速度も高速である。3069F は最大 25MHz のクロックで動作する。大容量の外付けメモリが必要な場合には 3069F、一時メモリとして内蔵 RAM だけで済むのであれば 3052F という使い分けとなる。

(d) L-CARD+

最後に、PC により近いマイコンシステムを紹介する。これまでに紹介したマイコンは、PC で使われているような汎用 OS を用いずに専用プログラムが単体で動作するようなソフトウェア設計が一般的であるのに対し、このボードは組み込み用 PC に分類され、PC とほぼ同じソフトウェアが利用できる点が優れている。

L-CARD+は、タンバック電子が製作し、Linux 技術で有名な Laser5 が発売している組み込み用 Linux コンピュータである。名刺サイズの小さな基盤に Linux が動作する必要最低限のハードウェアを実装している。CPU にはワークステーションなどで実績のある MIPS 互換 CPU、VR4181 を採用し、Ethernet1 チャンネル、シリアルポート 1 チャンネルを備える。ROM2MB と RAM16MB を実装しており、CF に Linux 環境を構築するようになっている。

欠点としては、基板だけで約 40,000 円と高価であることがあげられる。また、Linux はリアルタイム OS としての性能は高くないという不安材料もある。しかし、今後の計測用コンピュータは組み込み OS を搭載したものが主流となることが予想されることや、将来の計測器の最重要テクノロジーは通信となる事が予想される事から、PC 上で広く普及しており開発サイクルが小さくできる Linux を用いたシステムは今後大きな地位を占めることが予測される。

(3) マイコンの比較

代表的なマイコンとして、Z80,PIC,H8 について比較してみる。

(a) アーキテクチャ

Z80 と H8 は CISC、PIC は RISC に属する。特に PIC はハーバードアーキテクチャを用いたタイミング管理が容易であるなど先進的な機能を持つ。Z80 や H8 はそれと比べるとやや古い設計であるが、H8 のほうが現代の汎用 CPU に近い設計であり通常の用途には使いやすいといえる。シビアなタイミング管理が必要になる場合には PIC が、高速処理が必要な場合には H8 がそれぞれ優れている。

(b) 速度

標準的なクロック速度を比べると、Z80 は 1MHz ~ 10MHz、PIC は数 kHz ~ 4MHz、H8 は 12MHz ~ 25MHz である。PIC では消費電力を抑えるために低い動作周波数にすることも可能だが、多量のデータ処理には役者不足である。Z80 は高速化バージョンこそあるものの一世代前の 8bit CPU であるため処理速度は期待できない。H8 は 16bit の本格的なプロセッサであり、信号計測や制御用としては十分なパフォーマンスを発揮できる。

(c) ソフトウェア開発環境

Z80 は歴史も古く多くの開発用ソフトウェアが存在するが、8bit 処理でメモリアドレス空間も 16bit しかないため本格的な処理系を実装することは不可能である。C 言語のコンパイラはサブセットとなる。商用のコンパイラは何種類も存在する。ROM は内蔵されていないので、外付けの ROM にプログラムを書き込むことになる。従って、開発には ROM ライタが必要である。

PIC は Z80 以上に条件が厳しく、わずか数十 ~ 数百 bytes の RAM しか利用できないため、コンパイラ言語での開発は現実的でない。商用コンパイラを用いて開発を行ってみたところ、メモリ再利用に問題があり動作するプログラムを作ることができなかった。アセンブラでの開発が前提となる。なお、MicroChip 社はアセンブラを無料で公開しており、高価な開発環境を購入する必要がないことは魅力である。内蔵 ROM にプログラムを書き込むために PIC ライタと呼ばれる装置が必要となる。

H8 は本格的なプロセッサであるためフルセットの C/C++ コンパイラを実装することができる。日立製作所純正の処理系のほか、GCC(GNU Compiler Collection)が無料で利用できる。プログラムメモリも 3052F や 3069F では 512KB と多量に用意されており、開発のしやすいマイコンであるといえる。なお、内蔵 ROM へのプログラム書き込みには内蔵シリアルポートを利用する。書き込みに必要な回路は非常に単純で済み、オンボード書き込みが一般的である。

(d) 内蔵周辺機能

Z80 はもともと周辺機能を内蔵した CPU ではないが、組み込み用 Z80 互換 CPU には汎用 I/O・タイマ・シリアルポートなどがチップ上に組み込まれている。過去との互換性のため、これらの機能は I/O を通して行うことになり、使い勝手はあまりよくない。

PICには汎用 I/O・タイマ・A/D 変換機・コンパレータなどが装備されており、用途により内蔵機能で使い分けることになる。16F84 には汎用 I/O と 8bit タイマしかないが、16F877 にはそれらに加え 16bit タイマ・A/D 変換機・シリアルポートが用意されている。これらの機能はメモリマップド I/O 方式で、メモリアクセスを行うのと同じ要領で簡単に利用できる。

H8 は数多くの周辺機能を内蔵している。汎用 I/O・8bit/16bit タイマ・A/D 変換機・D/A 変換機・シリアルポート・パルスジェネレータなどが簡単に利用できるようになっている。PIC が少量の機器のコントロールに向くのに対し、H8 では多くの機器を 1 個の CPU で制御できるようになっている。

(e) まとめ

Z80 は古い歴史による豊富なソフトウェア資産に支えられており現在でも多くの組み込み機器で利用されているが、設計の古さは否めず新しくシステム開発を行う場合には適していない。作ることでできるシステムも小規模なものに限られる。Z80 と組み合わせる周辺 IC も入手困難になっており、もはや歴史的価値しかないものと判断される。

PIC は電気信号的に非常に扱いやすく、またタイミング管理も容易であるため、単機能なコンポーネントを作るのに非常に適している。例えば、プロトコル変換機や、簡単なシーケンサなどには最適なマイコンであるといえる。アーキテクチャが単純であるためアセンブラでも簡単なシステムの記述は容易であるが、コンパイラが事実上利用できないため大規模なシステム開発には適さない。

H8 は現代的な CPU であり、本格的なコンパイラを用いた開発が可能である。内蔵機能も豊富かつ使いやすく、いくつもの機器を同時に制御するといった用途にも使用可能である。价格的にも PIC に肉薄しており、幅広く手軽に利用できるマイコンであるといえる。

なお、前節で挙げた L-CARD+ はこれらの組み込みマイコンでは対処できないような複雑な機器向けの組み込みボードである。タッチパネルを利用した GUI や、本格的なネットワーク機能を組み込みたいときにはこのカテゴリの製品が用いられる。後述するネットワークアプライアンスにはこのクラスの組み込みボードが用いられている。

6.4 通信技術概論

(1) 有線通信

有線通信手段として標準的なものを以下に挙げる。

(a) シリアル通信

1 本の信号線上に直列に信号を送る方式をシリアル通信という。RS-232C などと呼ばれる信号線がこの規格によるものである。通信の同期を取る方式として、同期方式と調歩同期方式の二種類がある。同期方式は同期信号を通信線とは別に用意する方式、調歩同期方式は 1 バイトを表すビット列にスタート・ストップビットと呼ばれる同期信号を混在させることで同期を取る方式である。通常、実装の容易な調歩同期方式が用いられる。

調歩同期式シリアル通信は容易に実装できるが、転送速度が 150bps ~ 9600bps の何段階か、フロー制御にソフトウェア方式とハードウェア方式の別が、スタート・ストップビットの実装にも数種類があり、これらの組み合わせだけの種類が存在している。このため、通信方式としては扱いにくい部類に属する。近年の PC にはシリアル通信ポートを持たないものも出現しているが(後述する USB にその地位を代わられている)、計測器のインターフェースとしてはもっとも一般的な方法である。1 対 1 の通信にしか使えない、速度が遅い(規格上は 9600bps まで、実用的にも 100Kbps 程度止まり)といったことも欠点である。

(b) USB

シリアル通信の後継として登場したのが USB である。1本の線路上にシリアル信号を流すという点でシリアル通信の一種に属するのだが、信号パケットを標準化することにより記憶デバイス・入出力デバイスなどのデバイスの仮想化を行う高機能な規格となっている。従来のシリアル通信より高速なのだが、実装には専用のコントローラが必要であり、ソフトウェア上も OS の支援を必要とするなどハードルが高い。やはり1対1の通信が前提となる。

(c) Ethernet

Ethernet は CSMA/CD(Carrier Sense Multiple Access/Collision Detect)方式による通信手段である。上記の2つとは異なり、ひとつのネットワークを共有するすべてのノードで信号線を共有し、協調動作することでデータ通信を行うという特徴を持つ。方式名は、信号線上の搬送波を調べ(Carrier Sense)、複数のノードからのアクセスを許容し(Multiple Access)、同時に信号を送ってしまった場合にはそれを検出して再度データを送る(Collision Detect)ことを表している。現在もっとも普及しているコンピュータネットワーク通信手段である。速度により、10BASE(最高速度 10Mbps)、100BASE(同 100Mbps)、1000BASE(同 1Gbps)の種別がある。通信線としてはかつては同軸ケーブルが用いられていたが現在ではより対線を用いることが多い。

RIMS では、加速度計測システムとの通信インターフェースに Ethernet を用いる。当初は、ブラウザとなるコンピュータと計測器との間をシリアル通信、もしくはハンドシェイクによるパラレル通信で結ぶ設計を考えていた。しかし、Ethernet を用いた通信には次のような利点がある。

- ひとつの回線に様々なデータを流せる
- 多チャンネル化が容易である
- アプリケーションの開発が容易である
- 遠距離通信にも安定したソリューションが提供できる

Ethernet 上に流されるデータ形式としては IPX, NetBEUI, IP などが一般的である。これらのうちもっとも普及しかつ重要なものが IP(Internet Protocol)であろう。IP によるネットワークがインターネットであり、世界中に一般的な方法で情報を届けるには IP が最適である。

現在でも、北海道電力のモニタリングシステムに IP を用いたものがある。画像データを IP により監視するシステムである。

これらの点から、社会基盤施設モニタリング技術の将来像として、また本研究のテーマである分散型情報技術の基盤技術として、センサモジュールの LAN 接続を試みることにした。

簡単な計算を試みる。本システムの加速度計測モジュールは、通信に $60 \times 50 = 3000$ bps 程度の帯域を必要とする(50Hz 時)。10Base の Ethernet を使用した場合、3Mbps 程度を越えると輻輳が発生するので、有効な帯域のなかで 1000 チャンネルの計測モジュールが使用できる。実際には IP のオーバーヘッドがあるためここまでのチャンネル数は接続不能と判断されるが、シリアルポート接続では 4 チャンネル程度が限界であるため、多チャンネルを行う場合には LAN 接続機能を持った計測機器が有利になることがわかる。

(2) 無線通信

無線通信を実現するテクノロジーには、次のようなものがある。

(a) 微弱電波通信

電波法により無免許無申請で運用が認められている微弱な電波を用いた通信方式である。I/O データなどから、シリアル通信を無線化する装置が発売されている。信頼性などの点でやや問題がある。

(b) 携帯電話

近年普及の著しい携帯電話とモデムとの組み合わせにより、データ通信を実現することができる。携帯電話とひとくりに呼ぶが、その実体はデジタル/アナログの別、PHS など様々な技術の集合体である。

モニタリング情報を携帯電話により通信する技術は需要も多く、最近では報告も多いが、キャリアによりまったく別の機器が必要になる点や、通信インフラストラクチャに大きく依存するなどの難点もある。

(c) Bluetooth

家庭用機器などの無線通信手段の本命として期待されている方式である。拡散スペクトラム通信を用い、高速で手軽な無線通信環境を実現するといわれている。しかし、商品化の準備段階においていくつかの問題点が発見されたとも報道されており、現時点で手軽に利用できる技術ではない。本研究中、何度か Bluetooth 機器の開発を行っている企業に取材したのだが、残念ながら計測器の通信に利用できるモジュールを入手することはできなかった。

(d) 無線 LAN

IEEE802.11 に規定されている、11Mbps で Ethernet と同様の通信を行う機構である。無線 LAN のシステムは、Ethernet のハブに直結するアクセスポートと、PC に直結する無線 LAN インターフェースとで構成される。アクセスポートを経由することで Ethernet との混在が可能である。

今回のシステムでは、計測システムをネットワーク接続で構成していることから、無線 LAN のアクセスポートをハブに直結することにより、無線通信を行うことを実現している。

6.4.2 TCP/IP をベースとしたネットワーク技術

本節では、TCP/IP に基づく通信技術のスタンダードである Web システムに至るまでの歴史を概観し、その利点・欠点について述べる。RIMS は通信技術のアプリケーションでもあるので、その前提条件として実装が容易な通信システムが必要だった。Web システムは現在もっとも普及した通信システムであり、また低コストに利用できる標準化技術であることを示す。

(1) レガシーシステム

コンピュータが発明されたのは 1946 年で、弾道計算を目的としたものであった。以後、1950 年代より急速に開発が進み、1960 年代にはビジネス分野にも使用されるようになった。この時期のコンピュータは大型でコストも高く、またヒューマンインターフェースも未熟であったため、大量のデータ処理を順次実行する「バッチ処理」が行われていた。

1970 年代になると、支社の端末機と本社の大型コンピュータとを回線で結び、集中的に処理を行うオンライン処理が発明され、1980 年代に広く普及していった。バッチではなくリアルタイムな処理へのニーズがあったことを背景とし、ディスプレイとキーボードの組み合わせからなる端末機の登場や、コンピュータの処理能力の増大がそのニーズに答えた結果である。

これらのコンピュータシステムを、レガシーシステムと呼ぶ。

レガシーシステムを構成するコンピュータは、各メーカーが独自仕様に基づきハード・ソフトを設計したものであった。このため、競争原理が働かずに高コストとなることや、他のメーカーとの互換性がなく相互接続性が低かったこと、メーカー主導の開発となるためユーザのニーズに即応する商品開発ができなかったこと、などの問題点を抱えることとなった。

(2) オープンシステムとダウンサイジング

1970年代のはじめに電卓の構成部品としてマイクロコンピュータが発明される。

1970年代の中盤にはこれを用いたパーソナルコンピュータが出現し、以後急速に発展を遂げていく。

1980年代には各社から様々なアーキテクチャのパーソナルコンピュータが発売されたが、IBM社の仕様がデファクト・スタンダードとして80年代の終わりには収束していく。

当初は限られた処理能力しか持たない玩具のような存在であったパーソナルコンピュータだが、32bit並列処理能力を持つマイクロコンピュータである80386の登場により、大型コンピュータとかわらない性能を発揮するようになった。

一方、初心者にも使いやすいGUIシステムであるWindowsが普及し、パーソナルコンピュータ上のOSとしてのデファクト・スタンダードになった。

通信システムの基盤であるTCP/IP環境も整備され、デファクト・スタンダードによるオープンな仕様に基づく計算機環境がパーソナルコンピュータの世界に登場することになった。

このような計算機環境を、レガシーシステムに対してオープンシステムと呼ぶ。オープンシステムでは開発がレガシーシステムに比べ非常に容易であることから、ユーザ寄りの商品開発が盛んに行われ、性能差もあまりないことから大型機を駆逐していくことになる。いわゆるダウンサイジングである。

(3) クライアント/サーバシステム

オープンシステムにおいて、処理をサーバとクライアントとに分散するようにしたシステムを、クライアント/サーバシステムと呼ぶ。これに対し、ネットワークを前提とせず一台のパーソナルコンピュータで処理を行うシステムを、スタンドアロンシステムと呼ぶ。

クライアント/サーバシステムでは、ユーザインターフェースとロジックの部分をクライアント側で、データ処理部分をサーバ側で行う。サーバにデータを集中させることにより、一貫したデータ管理を保証するのである。

この形のシステムは、うまく設計すればニーズに合ったものを作れるという利点があるが、クライアント側の処理に負荷がかかり、しかもクライアントの仕様変更があった場合にすべてのクライアントのプログラムを入れ替えなければいけないため、運用が難しいという欠点があった。

(4) Web型クライアント/サーバシステム

1990年代初頭、論文を発表するためのシステムとしてWWWがCERNで開発された。WWWは二つのアイデアからなるシステムである。

- Internet上の計算機にあるファイルに、ユニークな名前(URI)をつける
- ファイルを表示する際に、一定の形式(HTML)で解釈を行う

WWWはクライアント/サーバシステムの一つだが、その処理内容は前節で述べた典型的なシステムに比べて非常に単純である。クライアントはサーバにURLを送り、サーバは

その URL に結びつけられたファイルの内容を送り返す。クライアントは送られてきたファイルを HTML 形式として解釈し、表示するだけである。

後に NCSA のマーク・アンドリーセンの手により HTML の拡張に画像表示機能などが拡張されたが、基本的なシステムは今でも変わっていない。

このシステムにおいて、URL に結びつけられた内容を単なるファイルではなくなんらかのデータ処理結果として返すというアイデアが、Web 型クライアント/サーバシステムである。Web 型クライアント/サーバシステムでは、ロジックはクライアント側ではなくサーバ側にある。クライアント側に必要なのは HTML 文書を解釈するブラウザだけである。

この意味で、Web 型クライアント/サーバシステムは、レガシーシステムのオンライン処理とよく似ている。大型機が Web サーバ(アプリケーションサーバ)に、端末機が Web ブラウザに置き換わったものとも言えよう。端末機の代わりに Web ブラウザを用いることにより、より使いやすいヒューマンインターフェースが実現できる点が大きな違いである。

Web 型クライアント/サーバシステムでは、ロジックの入れ替えが必要な時でもサーバ側のプログラム修正だけで済み、クライアント管理のコストが抑えられるのが特徴である。

Web システムは、サーバよりクライアント側に処理コストがかかる設計になっている。ブラウザを作ることは難しいが、サーバを作るのは容易である。Web サーバは基本的にはテキスト処理を扱うだけであり、また他のプロトコル(メールを扱う SMTP/POP3、ファイル転送を行う FTP など)と比べて状態管理・認証といったプロセスがなく、実装が容易である。このことは Web アプリケーションが状態・認証を必要とするときには障害となるのであるが、単純な動作を行う通信システムとしては非常に開発効率がいいことを示している。

Web クライアントはまた、簡単に実装できる GUI のプラットフォームでもある。従来 GUI を持つシステムは多くのコーディング量を必要としたのだが、Web システムではサーバ側に簡単な HTML での記述を行うことにより GUI が実現できる。PC 用のほか、携帯電話・PDA・ゲーム機・車載用機器などさまざまな Web クライアントが存在し、現在もっとも標準的な通信テクノロジーであろう。

6.4.3 超軽量ネットワークアプライアンス概論

(1) ネットワークアプライアンス

PC やワークステーションではない、単機能に特化したネットワーク接続機器をネットワークアプライアンスと呼ぶ。もともとは Web ブラウザと電子メール機能を組み込んだ簡易端末(STB などと呼ばれる)を指す用語であったが、この種の簡易端末が PC と携帯電話にニッチを奪われる中でクライアント的なものよりもサーバ的な製品を指すように変化してきた。

現在実用化されているネットワークアプライアンスには次のようなものがある。

(a) ルータ

ネットワークインターフェースを二つ装備し、二つのネットワークを結ぶための装置である。通常、WAN(=Internet)と LAN とを結ぶために用いられる。UNIX ワークステーションからルーティング機能だけを抜き出した製品であるが、近年の Internet 接続環境の普及と Linux をはじめとする無償 UNIX 互換 OS の登場により安価な製品が登場している。

(b) NAS

ネットワークアプライアンスストレージの略で、記憶装置として特化したネットワークアプライアンスである。これも UNIX ワークステーションにあった記憶領域共有機能を取

り出して製品化したものであるが、容易に LAN 内で共有する情報スペースを作ることができるものとして普及を始めている。

(c) Web カメラ

ネットワークに接続するだけで Web サーバとして機能するカメラである。静止画や動画を配信することができる。監視用途に普及を始めており、手軽に映像情報を遠隔地に送ることができるのが特徴である。上記の 2 つが UNIX ワークステーションの機能の一部を抜き出したものであるのに対し、Web カメラは PC やワークステーションだけでは実現できない機能を一つのパッケージに収めた製品である点が興味深い。

(d) LAN アダプタ

従来の計測器はシリアル通信により PC と接続できるものがほとんどであるが、シリアル通信では多くの機器を接続できない・複数の PC から共有することができないことから、これらを直接 LAN に接続したいというニーズが存在する。LAN アダプタは、シリアル・パラレルなどの信号を LAN 接続できるようにプロトコル変換を行う装置で、通常通信ポートを仮想化するドライバソフトウェアとセットで販売されている。このカテゴリの製品の中の特異なものとしてプリントサーバがある。プリンタの共有を行うための製品であるが、プリンタの共有は LAN ではよくあるシチュエーションであるため、それに特化した製品が存在する。

これらのネットワークアプライアンスは、組み込み用コンピュータを用いて実装されている。CPU には 3 種のマイコン(Z80,PIC,H8)よりも PC に近いクラスのコンピュータである x86 や SH,ARM などが用いられている。従来ネットワークに接続するにはそれだけの CPU パワーが必要だと考えられてきた。そのため、PC との差別化は難しく、以上に挙げた 4 種の製品以外のネットワークアプライアンスは普及していない。

(2) 超軽量ネットワークアプライアンスの登場

PICNIC は、落合正弘氏の設計による PIC アプリケーションである。秋月電子でキットまたは完成品として入手できる。Realtek 社の Ethernet コントローラ、RTL8019 とミッドレンジ PIC、16F877 の組み合わせにより、16F877 の持つ I/O のうちアナログ入力 4 チャンネル、デジタル入出力合わせて 8bit、シリアルポート 1 チャンネルを IP でアクセスできるようにするハードウェアである。シリアルポートと汎用入出力ピンを持つので、そのまま LAN アダプタとして利用することができる。

8019 は、かつてネットワークコントローラとして事実上標準の地位を占めていた Novel 社の NE2000 と互換のある Ethernet コントローラで、8bit のバスでの運用が可能である。性能的には PCI などの高性能な 32bit バスを前提とした現代のコントローラに遥かにおよばないものの、10Mbps 転送用のコントローラとしては十分な性能を有している。

PICNIC 以前の LAN アダプタと比較してみると、従来製品の 1/3 ~ 1/10 の低価格を実現している。単純な構造であるため次の点が省略されている。

- エラーリカバリを行わない
- セキュリティを考慮していない
- バッファサイズが小さい
- http 以外のプロトコルは UDP しかサポートしない

これらの点が批判されているが、よく考えると必ずしも欠点ではないことがわかる。

まず、リアルタイムで計測したデータはエラーがどこかで起こった時点で失われてしまうので、エラーリカバリすることに意味はない。

セキュリティ面はネットワークをローカルに構築し、ファイアウォールにより保護することにより解決可能である。PICNIC 自体が低機能過ぎて、PC で構築されたネットワークのように外部からワームを注入、内側からセキュリティを脅かす、といったことが不可能であり、この意味では従来製品より堅牢であるとも言える。

バッファサイズの小ささはなんともしがたいが、そもそも I/O のリカバリタイムがそれほど短くないので取りこぼしのないようにシステム設計を行うことで回避可能である。その意味ではセッションを張るのに多くのやり取りを必要とする TCP をサポートせずにシンプルな UDP を採用しているというのは、バランスの取れた設計と言えよう。ネットワーク関連のライブラリは現代の言語であれば充実しているので、他の言語からの制御も容易である。

PICNIC は Ethernet で低コストに監視・制御システムを構築できる素材として、簡易気象台やホームコントロールに利用されている報告がある。研究用途としても、東京大学物性研究所にて He ガスマーターの遠隔制御に用いた例がある。

PICNIC の意義は、安価なマイコンでネットワークアプライアンスを実現していることにある。PICNIC の出現以降、AVR や H8 を用いたネットワークアプライアンスの製作事例が次々と発表された。これらの、制御用マイコンによるネットワーク機器を超軽量ネットワークアプライアンスと呼ぶことにする。



図 6.2 PICNIC

6.5 RIMS 開発の実際

6.5.1 初期の RIMS デザイン

開発当初の RIMS は、加速度計測モジュールと PICNIC および L-CARD+ の 3 つの部分からなるものとしてデザインしていた。最初の加速度計測モジュールは、PIC16F84 とアナログデバイス社の 2 軸 MEMS 加速度計 ADXL202 を一枚の基板上に配置したもので、加速度計の値をシリアル通信で出力する。PICNIC はシリアル通信を Ethernet に変換し、L-CARD+ に接続する LAN アダプタとして用いる。L-CARD+ では Linux 上に加速度計からのデータを処理するプログラムを動作させ、加速度計の情報から指標値を計算しネットワークに流す。この段階では指標値の通信に HTTP を採用することは決めていなかった。

この設計ではうまく動作させることができなかった。まず、加速度計の値を PIC で読み出すソフトウェアを開発したのだが、加速度計から出力される PWM 波形を安定して読み出すことが困難であった。また、PICNIC によるシリアル UDP 変換及び L-CARD+ での読

み込みも動作するものの、途中から読み込みデータがおかしくなるなど連続動作させることができなかった。

RIMS では長時間にわたるデータの取得が求められる。ADXL202 と PICNIC の組み合わせは要求に見合うだけの信頼性がなく、システムとして動作させることは困難であった。また、モジュールが 3 つに分けたのも、それぞれで不具合が生じたときに問題の切り分けを行うことを難しくしており、システム設計上の失敗であった。

以上の失敗の経験から、RIMS のデザインはモノリシックにするべきであること、各コンポーネントを小さく分けるのはよくないことを学び、現在の RIMS のデザインに改めた。ちょうどこの時期に秋月電子から H8 ネットワーク基板が発売され、それまで開発経験がないことから敬遠していた H8 を採用し、デザインをやり直すことにした。

6.5.2 システムの構成

現行の RIMS のシステムの構成は、図 6.3 のようになる。

加速度計はコントローラ(H8/3069F)の A/D コンバータに直結する。コントローラは随時加速度計の値をモニタし、RAM 上に確保されたリングバッファに記録を行うようプログラミングする。コントローラは TCP/IP により外部と接続され、HTTP によるコマンドを受け付けて設定・データ表示を行うようにした。HTTP コマンドとしては、指標値である加速度が閾値を越えた回数を表示するコマンド、リングバッファから取り出した波形データをそのまま表示するコマンド、各種設定を行うコマンドなどを設けた。

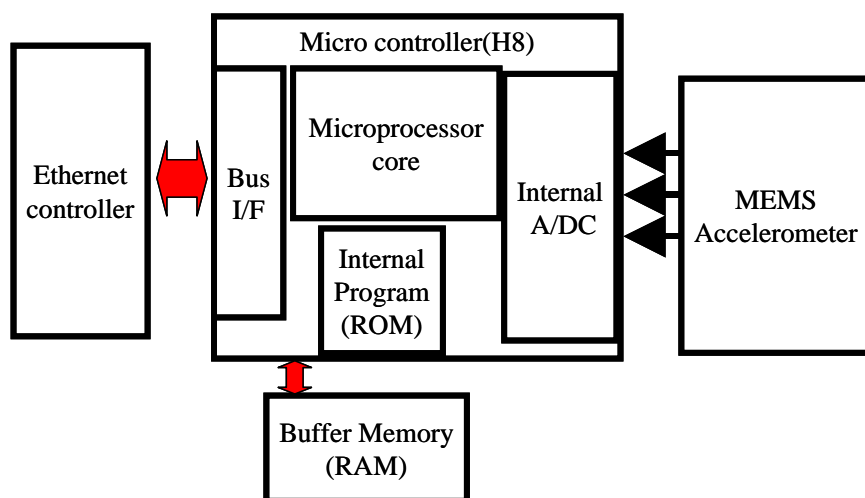


図 6.3 RIMS のハードウェア構成

RIMS 計測器をネットワーク接続により増設すれば、多チャンネルの観測も可能となる。1 個の RIMS は同時に 6 つのアクセスしか許容しないが、1 つのネットワーク上に RIMS はいくつあってもよい。実際には輻輳が起こるため無限に接続できるわけではないが、RIMS が通信を行うのは指標値の情報だけであり計測加速度全てがネットワーク上を流れるわけではないので、使用する帯域はわずかである。この点は、加速度情報を全てネットワークに流していた初期の RIMS デザインよりも優れている点である。

すべてのプログラムは H8/3069F の内蔵 ROM に書き込まれており、電源 ON により即計測が始まるようになっている。図 6.4 に示す今回作成した加速度計測プログラムは次の 2 つの部分からなる。なお、デーモンの用語は UNIX における常時駆動プログラムをこの名で呼ぶことから、サブレット関数は Web アプリケーションサーバにおけるサブレット

に相当する部分を C 言語の関数として実装していることから、それぞれこの用語を充てている。

- 加速度計測デーモン：MEMS 加速度計から送られてくる加速度データを監視し、リングバッファに記録する。リングバッファから信号を遅れて読み出し、波形のピークを閾値により分類、記録する。ピークの検出は±両方について絶対値を用いている。
- HTTP 処理部：簡易 HTTP サーバである。HTTP によりコマンドを受け付け、サーブレット関数を呼び出す。各サーブレット関数では、デーモンが記録したデータの取得・消去、及び加速度計のオフセット調整を行う。

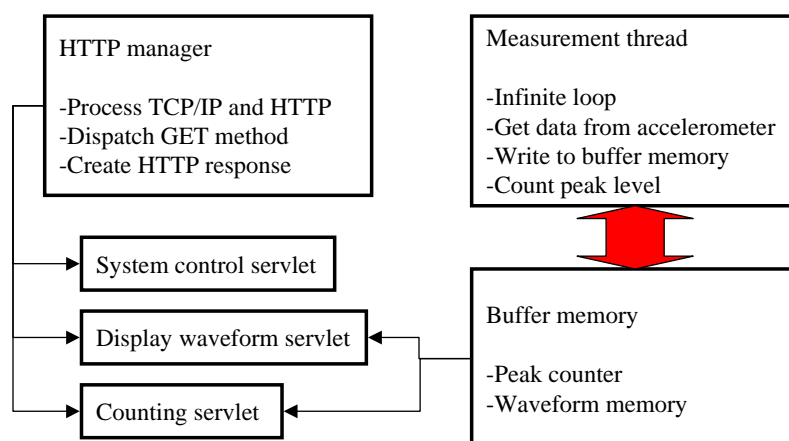


図 6.4 RIMS のソフトウェア構成

6.5.3 計測部の実装

計測部では、1.加速度計からの電圧を A/D 変換し、2.値をリングバッファに格納し、3.リングバッファに格納された値から指標値を計算するという 3 つの処理を行っている。

加速度値から直接指標を計算せずに一旦リングバッファを経由しているのは、時刻歴データを保持するためと、ノイズ対策のため過去のデータを記録しておく必要があるためである。当初はリングバッファなしで実装していたが、時刻歴データを 100Hz で通信することが困難だったこと、ノイズ対策処理が不可能であったことからこの構造とした。このことは、RIMS の実装には大容量 RAM が必要不可欠であることを示唆しており、基本的にレジスタしかもたない PIC などでは実装が不可能であることを意味する。

H8/3069F には A/D コンバータが 8 チャンネル内蔵されている。逐次比較型 10bit、レンジを +0 ~ 5V とすると 5mV の分解能となる。MA-3-04 の感度は 50mV/m/s² であるので、分解能はおおよそ 10gal となる。プログラム中では 1 カウント=10gal として計算を行うようにする。

実際には、逐次比較型 A/D コンバータの精度は下位 2bit はあてにならないとされている。実際、静かに設置した場合でもプラスマイナス 5 カウント程度の変動を記録した。対策として、精度確保のためにオーバーサンプリングを行う。具体的には、サンプリングレートを 500Hz とし、5 つの平均値から 100Hz サンプリング値として採用することとした。

先に 8 チャンネルの A/D コンバータ内蔵と述べたが、実は内部的には A/D コンバータは 1 チャンネルのみであり、これを内蔵マルチプレクサにより切り換えて使用する事になる。A/D 変換後の値は、チャンネル毎に特定のアドレスに割り付けられた ADDR_A ~ ADDR_D と呼ばれるレジスタに格納される。リングバッファへは単純にメモリコピーにより転送を行うことができた。現行のファームウェアではリングバッファは 40 秒分用意してあり、後述

するサブレット函数によりそのうち最新 15 秒間分について波形データとして外部に出力可能にしてある。

H8 の A/D コンバータの使い方には単一モードとスキャンモードとがある。単一モードでは明示的にマルチプレクサを切り換えて指定されたチャンネルから読込を行い、結果は ADDRA ~ ADDR D レジスタのうち対応するものに格納される。このモードでは A/D 変換終了をポーリングまたは割り込みで待たなければならない。スキャンモードでは 1 ~ 4ch / 5 ~ 8ch の 2 群にアナログ入力端子をグループ分けし、どちらか一つのグループについて自動的に順次変換・読込を行い、ADDRA ~ ADDR D レジスタに最新の値が格納される。ユーザプログラム側で A/D 変換終了のタイミングを待つ必要はない。RIMS では 3 軸の加速度を別々のチャンネルで取り込むため、1 ~ 3ch についてスキャンモードでサンプリングを行うことにした。内部的にはチャンネル切り替え サンプルホールド A/D 変換のプロセスが行われており、チャンネルごとに数十 μ s の遅延が生じていることになる。サンプリングレート 100Hz に対し無視できる数値であると判断している。

ハードウェア的には、アナログ入力ピン(ポート 7)に加速度計の出力ピンを直結している。MA3 は OP アンプを内蔵し、加速度値を 0 ~ 5V の電圧に変換しているため、0 ~ 5V の計測レンジを持つ内蔵 A/D 変換機と直結することができる。通常、マイコンには安定化された電源を入力するため、A/D コンバータへの電源・リファレンス電圧はマイコン電源 (Vcc) を短絡して用いるのだが、今回のシステムでは通信時にネットワークコントローラが大電流を要求するためアナログ入力値が乱れる現象が観測された。この問題は、アナログ部電源およびリファレンス電圧をデジタル電源から分離し、電源ラインに大容量コンデンサを配置する事により解決している。

指標値のカウント方法としては、ラインクロス法を採用した。これは、あらかじめ設けた閾値をまたいだ回数をカウントする手法である。本来ならピークホールドしたあとその値で分類を行うべきであるが、A/D 変換された値は細かく変動するため「ピーク」の定義が難しく、このような手法で代用することにした。実際には、この手法でも細かい変動によりラインクロスしてしまうため期待される答えよりも多くカウントしてしまう結果となってしまう。閾値ラインは中心値より \pm 対称にとるものとし、+ 側のときは下から上に、- 側のときは上から下に、それぞれまたいだ回数をカウントすることにした。

6.5.4 通信部の実装

RIMS は HTTP サーバとして動作し、計測データの出力と初期設定をブラウザから行う。通信部は TCP/IP を受け付けてコマンド処理を行う、簡易 HTTP サーバとしての役割を持つ。RIMS では OS として TRON などの高機能 OS ではなく、三岩氏による簡易 OS、H8/OS を利用している。この OS は TCP/IP の機能を最低限実装しており、ネットワークサーバとしての利用には十分な API を持っている。しかし、クライアントとしての機能は弱く、クライアントプルとしての使い方は可能だがサーバプッシュとしての使い方をすることはできない。このため、計測器のコマンドデザインは HTTP によるリクエストを能動的に処理するものとなった。

当初、計測器からデータをリアルタイムに取得するために、TCP/IP よりも軽い通信方法である UDP/IP によりクライアントとなる PC から直接 100Hz でデータ要求を送りつける方法をテストしていた。この方法でも計測器側は 100Hz=10ms の反応時間を確保できたのだが、問題はクライアント側で Linux では 10ms のリアルタイム性能でもギリギリ、Windows ではまったくリアルタイム性能を確保できず数 Hz の計測データを取得するのがやっとであった。このため、UDP による通信はあきらめてリングバッファを新設し、HTTP によりリングバッファの内容を整形・転送するコマンド体系に改めた。よって、現状の RIMS は現在リアルタイムでの計測機能を持たない。H8 側のリアルタイム性能には問題がないため、将来的には UDP による多チャンネル同期機能を実装する事も可能であると考えている。こ

の際、UDP であっても 10ms 程度の遅延が発生するため、同期制度には限界があると割り切って使う必要がある。

H8/OS では、TCP/IP の状態遷移も自前で管理しなければならない。いちいちこの部分を管理するのは大変なので、HTTP ハンドラと各コマンド処理部(これをサブレット函数と呼ぶことにする)とに分離することにした。HTTP ハンドラはコマンドを受け付け、登録されているサブレット函数を呼び出して本文を得る(本文はメモリ上に保存しておく)。あとは HTTP ハンドラがレスポンスパケットを本文を元に組み立て、送信可能なサイズに区切って出力する。

サブレット函数には、コマンドに対応するレスポンスを HTML として整形する処理を記述する。デフォルトでは指標値情報を表形式で返すサブレットが呼び出され、他に初期化(過去 10 秒間の平均値を原点に設定する)、時刻歴(過去 15 秒間の時刻歴を CSV 形式で出力する)を用意した。

RIMS プロトタイプでは秋月電子より発売されている H8/3069F ネットワークボードを使用したのだが、このボードではオシレータとしてセラロックを用いているために精度が悪く、長いパケットを送ることができない欠点があった。安定性向上のために(有)追坂電子機器との共同開発でセラロックの代わりに水晶を用いた CPU ボードを作成し、IP として最大長となる長さのパケットでも問題なく通信できる事を確認している。

H8/OS にはいくつかの不具合があり、長時間連続稼働させるとネットワークに反応しなくなる。これは RTL8019 のレジスタの使用法にいくつか誤りがあるためであることが後日わかったが、RIMS では安定して長時間稼働させる必要があるため、1 分毎にネットワーク部を再起動する処理をタイマで実装している。加速度計測部の処理に追加する形で実装したのだが、この処理によりそれまでは 1 時間程度しか連続稼働できなかったのが、3 日間放置しても問題なく動作していることを確認できた。

RIMS の開発目標として、無線通信がある。今回は、ELECOM 社の無線 LAN アクセスポイント LD-WL11/AP2 をパッケージ内に納め、H8-NIC とクロスケーブルで接続することで無線 LAN カードによるアクセスを可能とした。クライアント側からはアクセスポイントとして見えるため、インフラストラクチャモードでの運用が可能である。

アクセスポイントを使用したことで、RIMS 全体の消費電力が大きくなってしまった。RIMS 自体の消費電力は 5V100mA であるが、Ethernet 信号を無線化するアクセスポイントがその 3~4 倍もの電力を消費している。無線 LAN コントローラの消費電力削減は携帯情報端末のニーズもあり、開発中に TI 社より TNETW1100B というコントローラ発売のニュースがあった。このコントローラを用いることにより、未使用時の使用電力を 2mW まで下げることができるという。RIMS では通信が行われるのは指標値取得とセットアップ時にわずかな時間だけであるため、省電力化のこうかは大きいものと予想される。無線化のためのソリューションとして非常に魅力的であり、引き続き調査中である。

マイコン本体をより消費電力の小さなものにするのも課題である。電池駆動を考えた場合、500mW の消費電力はまだまだ大きい。前述したとおり RIMS の処理にはある程度の大きさのリングバッファが必要であること、ネットワークコントローラを利用するための入出力ピンが必要であることから候補機種は限られるが、H8/3069F よりも消費電力が小さくこれらの条件を充たすマイコンは存在するため、検討の余地がある。

現状ではアクセスポイントを通じて近接時のデータ取得が利用スタイルとなっているが、もともと Internet 接続を前提としているため光ケーブル・無線 LAN などの情報インフラを整えば、RIMS はそのまま Internet に接続される情報収集端末となる。これらのインフラ整備がまだ不十分な現状でも、携帯電話のキャリアを利用することにより常時接続のモニタシステムを構築することは可能である。データをどこでも取得できるのと現地まで足を運ばなければならないのでは大違いであり、Internet に常時接続された情報収集端末の潜在ニーズは非常に大きいものと思われる。RIMS コピキタス時代の社会基盤施設保全のキーテクノロジーになることは間違いない。

なお、電源の改良も大きな課題である。RIMS プロトタイプでは安定化電源またはシール鉛蓄電池を使用した。無線 LAN アダプタが電力を消費することもあって、大容量の鉛蓄電池を使用する必要があった。RIMS 自体は太陽電池でも駆動できるため、太陽電池によるトリクル充電と、停電時でも取得した情報を失わないようにするソリューションの組み合わせにより電池交換なしでも半永久的に使用できる電源が構築できる。現状の RIMS ではデータを DRAM 上に保存しているため、電源断により記憶は失われてしまう。この問題を解決するため、フラッシュメモリによるデータ記憶回路と、太陽電池を用いたトリクル充電回路を設計中である。

6.6 評価試験

6.6.1 校正評価

校正作業は、IMV(株)の工場にて行った。IMV 社は振動計と振動試験装置のメーカーで、RIMS の試験には振動台を貸していただいた。

校正評価は、6Hz で加速度振幅を変化させながら振動台を作動させることで行った。図 6.5 に校正評価の様子を示す。図 6.6 は片側振幅 100gal の時のものである。1 カウントが 10gal に相当することより、振幅は約 200gal となる。

校正評価により、RIMS が精度よく加速度を検出できることが実証された。



図 6.5 校正評価風景

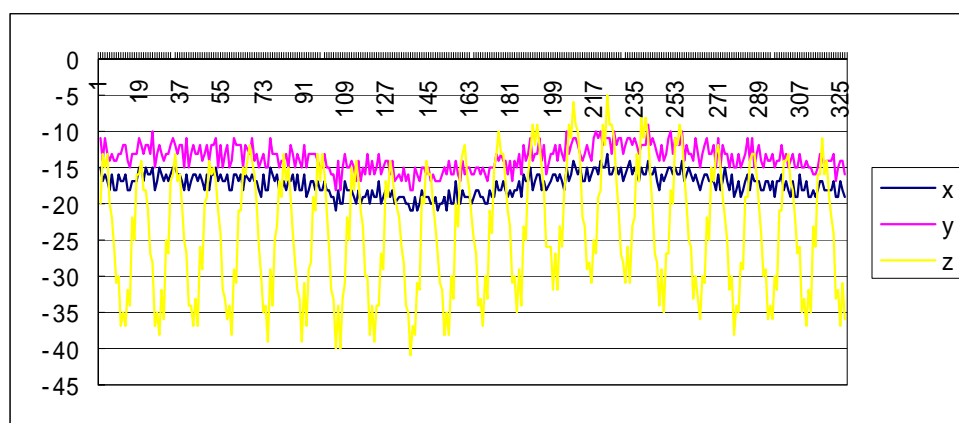


図 6.6 観測された時刻歴 (縦軸は 1 カウント=10gal 単位)

6.6.2 現場実験

現場実験は、図 6.7 に示すレインボーブリッジ下路にある遊歩道ノースルートにある照明柱にて行った。取り付けには(株)大谷工業の協力を得て、現実の取り付け作業の検証を行う意味を含めて行った。

図 6.8 のように RIMS をこの照明柱に取り付け、実際にどれだけの振動が記録されるかを調べた。取り付けには金属製バンドと RIMS のケースを取り付ける治具を使用した。電源については、AC 電源を確保することができなかつたため、シール鉛電池を用いて照明柱下部に配置した。取り付け作業は作業員 2 人で 1 時間弱で終了した。計測器取り付けとしては非常に作業効率が高いことが実証された。



図 6.7 遊歩道照明柱



図 6.8 RIMS 取付

この後無線 LAN によるデータ取得の実験を行った。遊歩道上で照明柱が見通せる範囲内(およそ 50m)ではデータ通信が行えたが、歩道連絡フロアの自動ドアに入るとリンクが安定しなくなることがわかった。

この状態で指標値を表示させたところ、数秒間で 100 を超える値がカウントされてしまっていた。時刻歴は図 6.9 のようになっている。

このグラフからわかるように、細かいノイズが混入していたのが原因であった。フーリエ変換してスペクトルを調べたところ、50Hz のノイズが入っていることがわかった。おそらく、照明柱のための交流電源によるノイズであると思われる。サンプリングレートが

100Hz であることから，2 ステップ前の値との平均を取ることによってノイズを除去できると考え，簡単なフィルタとして実装した．この処理により 50Hz のノイズを除去することができた．

以上の改良を RIMS の内部プログラムに対して行い，再度取り付けを行った．改良 RIMS で取得できた波形の例を図 6.10 に示す．

この状態で 1 時間毎の X 軸指標値をデータ取得時より過去 18 時間分表示させたものを図 6.11 に示す．このように，時刻による交通量の変化で指標値が変化していることを示している．

図 6.10 から明らかかなようにこの照明柱の固有振動周期は 3Hz 程度であり，100gal 以下の指標値 7 万前後というのは振動のピークの数としては多すぎる値である．指標値カウンタアルゴリズムではヒステリシスを考慮せず閾値をまたいだ回数を単純に数えてしまっているため，数えすぎが発生しているものと考えられる．

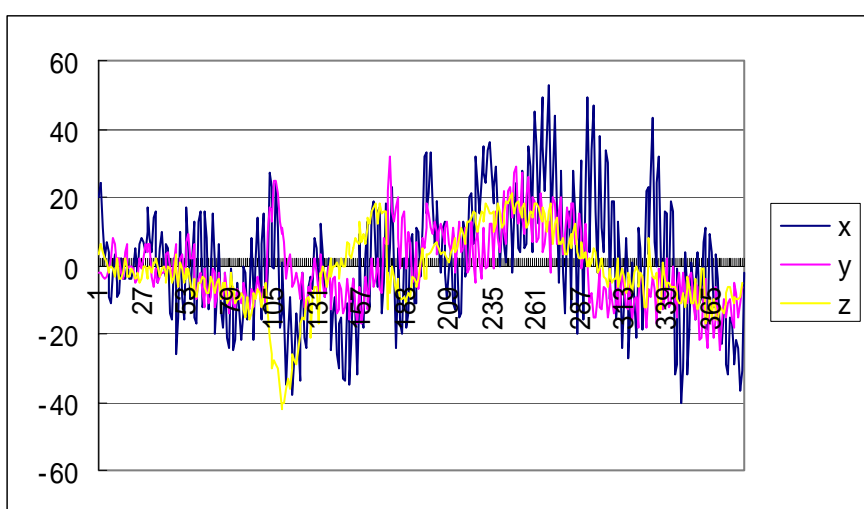


図 6.9 観測された時刻歴

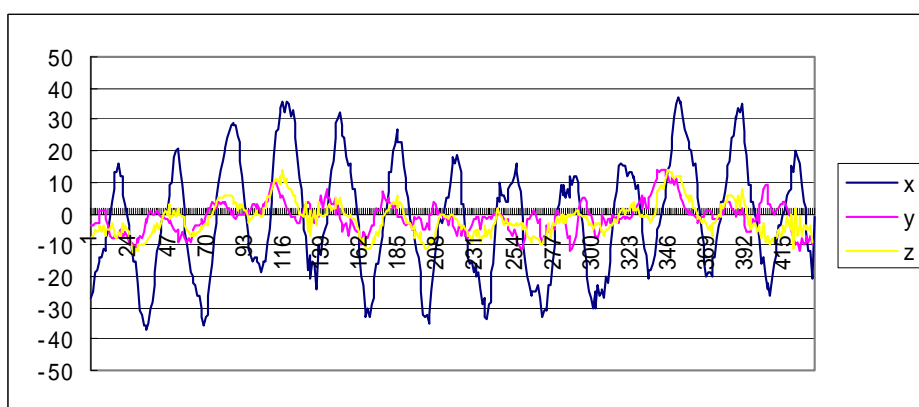


図 6.10 改良後の時刻歴表示

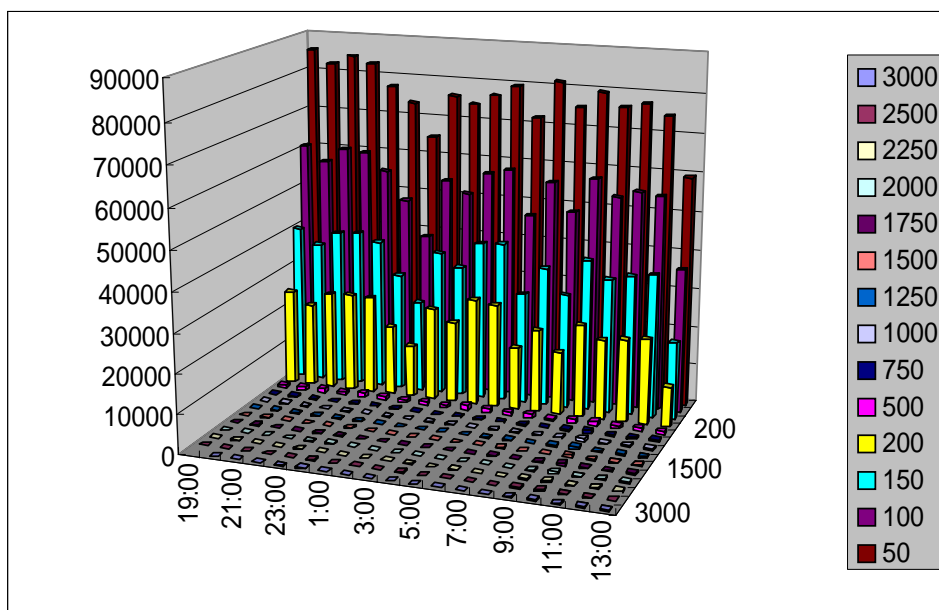


図 6.11 19 時間分の指標値の変化

6.7 まとめ

6.7.1 システム開発の成果

本研究では、簡易型加速度計測システム RIMS を開発し、超軽量アプライアンスによるモニタリング情報取得に成功した。このシステムは従来のモニタリング手法よりも施工面・通信コスト面で有利であり、今後の社会基盤維持活動のインフラストラクチャとしての将来性を期待できるものであると結論づけられる。

RIMS では一般的な通信プロトコルである HTTP を TCP/IP 上に実装することにより、PC のみならず PDA から利用することに成功した。また、HTTP の採用により、Web ブラウザというインフラを用いることで、従来では開発コストのかかる部分であった表示装置・スイッチ類を一切用いずに GUI による操作を実現した。

MEMS センサとマイコンの組み合わせにより小型のパッケージに納めることができた。このためモニタリングに必要な取り付け施工のコストは最小限で済み、従来のように観測小屋などを設置することなく安価にモニタリングを行うことを可能とした。

RIMS では、センサをネットワーク接続することによる低コストな多チャンネル化・マルチメディア化を実現できる。今回使用したセンサは加速度計だけであるが、アナログ入力やデジタル入力ピンにはその他各種のセンサを接続することができ、また近年発達の著しい画像センサと組み合わせることによってさらに複雑なモニタリングが可能となる。

6.7.2 今後の課題

今回作成したシステムはプロトタイプであるため、残された課題も多い。

まず加速度計測システムについては、コスト・消費電力などの点で改良の余地がある。マイコン H8/3069F は単価 2,000 円と安価であるが、加速度計 MA3 は高性能であるが高価である。より安価な加速度計への置き換えが望ましい。無線通信部でも触れたが、消費電力の点はさらなる改良が必要である。

情報の形式としては HTML と CSV を用いた。指標値の表示と各種設定は HTML で、加速度時刻歴は CSV で表示される。特に時刻歴に関してはグラフ化するためには一旦 CSV 形式で保存した後 Excel などのアプリケーションで開きなおす必要があり、不便であった。クライアント側でデータ処理を行うアプリケーションを開発するか、あるいは Java アプレットによるグラフ表示を実現するなどの改良を行いたい。

アプリケーションとしてぜひとも実現する必要があるのが、GIS との連動である。データの取得位置は GPS により特定可能である。この位置情報をレコードに埋めこみ、GIS で処理することにより、地図の上にデータを書き込むことが可能となる。

参考文献

- 土木学会 情報利用技術委員会 情報共有技術小委員会 『何でもどこでもネットワークの未来』 「土木施工」2003 年 2 月号
- 岡田和廣 『慣性センサへの応用』 「計測と制御」2003 年 1 月号
- 東京ガス 『超高密度地震防災システム SUPREME』 <http://www.tokyo-gas.co.jp/supreme/>
- ANALOG DEVICES “ADXL202/210 Data Sheet”
- マイクロストーン 『MA3 データシート』
- トランジスタ技術編集部編 『PIC マイコン活用ハンドブック』 CQ 出版,2001
- BUG 『多機能遠隔モニタリングシステム「IP-CAM」』,2000
<http://www.bug.co.jp/topics/ipcam.html>
- 鷲山玲子,土屋光 『PICNIC を利用した He ガスマーターの遠隔自動計測』,2002
- 落合正弘 『PICNIC の製作』 トランジスタ技術 2001 年 1 月号
- 落合正弘 『PICNIC Ver.2 の概要と付属ライブラリの使い方』 トランジスタ技術 2001 年 9 月号
- 三岩幸夫 『H8/3067F マイコンボードの設計&製作』 トランジスタ技術 2001 年 9 月号
- インターネットノード株式会社ホームページ <http://www.i-node.co.jp/>
- 日立製作所 『H8/3069 F-ZTAT ハードウェアマニュアル』 2001

助成研究者紹介

阿部^{あへ}雅人^{まさと}

現職：東京大学大学院工学系研究科助教授（Ph.D.）

主な著書：社会基盤メンテナンス工学（東京大学出版会 平成 16 年刊行予定）