学位論文

ユーザエクスペリエンスを考慮した
無線センサネットワークに関する研究

猿渡　俊介

"The best way to predict the future is to invent it."

Alan Kay, Xerox PARC, 1971.

目 次

図目次

表目次

第1章
序論

1.1 本論文の背景と目的

本論文は筆者が2001年に取り組んだときをスタートラインとしてその後6年間に渡り，次世代のコンピュータの形のあるべき姿を追い求めた結果である．10年後に本論文を読めば，本章で述べる筆者のアプローチが正しかったのか，間違っていたのかが分かるであろう．

筆者が2001年に研究者としての道をスタートしたとき，思ったことは「パーソナルコンピュータの次となる研究をしたい」ということであった．しかしながら，未来のことを予測するのが難しいように，パーソナルコンピュータの次は何かという問いに対する答えは簡単には見つからなかった．そこで筆者は，まず,「新しい技術の法則」を探すことにした．新しい技術の法則が存在するのであれば，それを知ればパーソナルコンピュータの次を予測することもできるのではないか．調査を進めるにつれて次第に「新しい技術の法則」の輪郭が見えてきた．

筆者が得た「新しい技術の法則」は2つある．

1つ目は「新しい技術は古い技術を取り込む」である．新しい技術は古い技術に比べて全く異なった特徴を持っているため，新しい技術が誕生した当初は単純なスペック値では古い技術よりも劣っている．しかし，新しい技術は単純さ，価格，信頼性，利便性の面で古い技術よりも優れているのである．そのため，新しい技術が成熟し，スペック値が改善されるにつれて徐々に古い技術の地位を取り込んでいく．この傾向はインターネットを中心に巻き起こった流れを見ると分かりやすい．

インターネットは誕生当初，自律分散性という機能を備えており，簡単に拡張できるネットワークであったものの質の低い通信しか行うことができなかった．そのため，電子メールのやりとりなどごく一部のアプリケーションのみにしか使われていなかった．しかしながら，インターネット技術が成熟していくにつれて古い技術の地位を次々と奪っていく．

インターネットを用いた音楽配信サービスの登場は既存の店舗型音楽販売の市場を脅かしつつある．CDを中心とした音楽市場の規模は1998年には6000億だったのに対して2005年には3600億円にまで落ち込んでいる．それに対して音楽配信サービスはナップスター190万曲，iTunesストア200万曲，モーラ50万曲と徐々に曲を揃え始め，市場規模も広がってきている．すでに音楽配信サービスの売り上げはシングルCDの売り上げ500億を超えようとしている．最近ではU2などの著名なアーティストですら先に音楽配信で新曲を公開するという動きもある．

音声通信の市場もインターネットに取りこまれつつある．音声通信の市場はインターネットの前に固定電話が携帯電話に取って代わられた．携帯電話は価格や音質では固定電話に劣っていたものの，徐々に価格や音質，バッテリの寿命が延長されていくにつれて固定電話に取って代わっていった．そして今ではインター

ネットが固定電話も携帯電話も両方とも吸収しようとしている．例えばSkypeを利用することでかなりの高音質で無料で国際電話をかけることができる．通信会社はNGNを立ち上げ，携帯電話網をIP化する動きも始まっている．

テレビの市場もインターネットが奪おうとしている．ブロードバンドが行き渡り，オンラインで動画を視聴するのが日常になりつつある．現在Yahoo!動画のユーザ数が505万人，GyaOのユーザ数が388万人とコンテンツを持つ企業による動画配信の市場は日々拡大している．さらに，ユーザが作成したコンテンツを簡単に共有できるYouTubeはユーザ数734万人とテレビでは実現できない動画サービスも生まれてきている．インターネットによる動画配信サービスの影響は確実にテレビを脅かしており，テレビスポットCM出稿料が2005年末の段階で前年比マイナス4〜5%となり，2006年も引き続きマイナスが続いている．

このように，新しい技術が古い技術の地位を奪っていく過程はマクルーハンのメディアに関する考察[1]や，クリステンセンのイノベーションに関する考察[2, 3, 4]が参考になる．

マクルーハンは著書「メディア論」[1]において新しいメディアと古いメディアの関係について次のように述べている．

> 新しいメディアは古いメディアに何かを付け加えるというものではない．また，古いメディアを平穏に放っておきもしない．それが古いメディアに変わって新しい形態と地位を見いだすまで，古いメディアを圧迫することを止めない．写本文化は教育において口誦の方式を維持していた．それが高水準の「スコラ哲学」と呼ばれるものであった．けれども，印刷が同一のテクストを任意の数の学生や読者の前に置くことによって，口頭の討論によるスコラ体制はあっという間に終わってしまった．

このように，メディアの観点から見ても新しい技術が古い技術の地位を奪っていることが分かる．

一方クリステンセンは著書「イノベーションのジレンマ」[2]において破壊的技術という言葉を用いて新しい技術が古い技術にとって変わる過程について述べている．その中でも，破壊的技術が生まれた時なぜ大企業が破壊的技術を用いて市場を開拓できないかに関する説明が興味深い．

> 通常，破壊的イノベーションは技術的には単純で，既製の部品を使い，アーキテクチャーも従来のものより単純な場合がある．確立された市場では，顧客の要望に応えるものではないため，当初はほとんど採用されない．主流からかけ離れた，とるに足らない新しい市場でしか評価されない特徴を備えた別のパッケージなのである．

図 1.1 にコンピュータの視点から新しい技術と古い技術の関係を示したグラフを示す．新しい技術は誕生当初は古い技術と全く異なった市場を持つ．そのため，新しい技術は古い技術にとって全く脅威にならないため，最初は無視される．これが新しい技術を見極めるのが難しい理由であると言える．

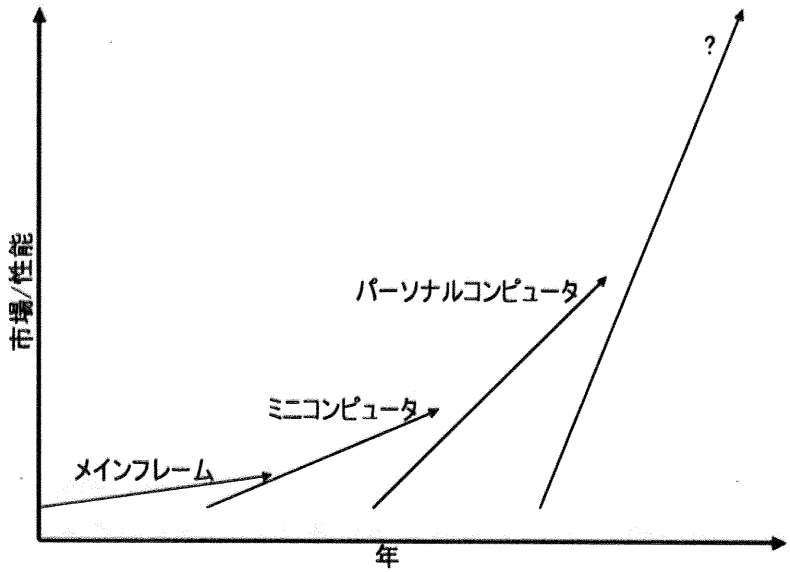


図 1.1: 技術と市場の関係

「イノベーションのジレンマ」を書いたクリステンセンはハーバード・ビジネス・スクールの教授であり,「イノベーションのジレンマ」もビジネス書である．一方の「メディア論」を書いたマクルーハンは英文学者であり,「メディア論」は社会学の本である．このように，全く異なる分野の人が技術に関して似たような視点で議論をしていることからも新しい技術が古い技術の地位を取り込む過程に規則性があることが伺い知れる．

2 つ目の法則は「新しい技術は個人を拡張する」である．新しい技術は，これまで個人ができなかったことを可能にする．これは個人が今まで存在しなかった全く新しいことをできるようになる，という意味では無い．今までやろうと思えばできたことでもコスト上の問題から個人レベルではできなかったことができるようになる，という意味である．つまり，昔も今も人類の用事 (やってきたこと) はほとんど変わっておらず，ある用事に対するアプローチの仕方を変えるのが新しい技術なのである．そして，技術は進歩すればするほど組織でしか使えなかったものが徐々に個人が使えるようになる．

例えば，コンピュータは生まれた当初，軍の弾道計算に用いられていた．このような背景を受けて IBM の創始者の Thomas Wilson は 1943 年に

"I think there is a world market for maybe five computers."

(世界のコンピュータの需要は5台ほどであろう)

と言った．しかしながら，企業の電子会計というアプリケーションが見つかるとメインフレームであるIBMのSystem/360が大ヒットし，1970年末までに約5,000台が販売された．1960年代の初頭にはミニコンピュータが誕生した．メインフレームは大企業の大規模システムを目的にしたものだったが，ミニコンピュータは当初研究室や産業制御，通信制御などを目的にして作られていた．しかしながら，徐々にミニコンピュータの性能が上がるにつれてメインフレームの市場も制していった．中でもDECのPDP-8は大ヒットし，30万台を超える売り上げ台数を記録した．当時のDECの創始者のKen Olsenは1977年に

"There is no reason for any individual to have a coputer in his home."

(個人が自宅でコンピュータを使う必要などない．)

と言い切った．しかしながらミニコンピュータの市場もパーソナルコンピュータに奪われる．パーソナルコンピュータは発売当初，個人向けを対象としたものであった．しかしながら1977年にAppleのApple IIが大ヒットすると売り上げ台数は200万台にも上った．その後マッキントッシュやWindows，BSD，Linuxなどが出現し，パーソナルコンピュータは世界を席巻した．

このような技術が持つ組織から個人へ向かっていく力は貨幣にすら及んでいる．2007年現在，貨幣は国しか生産することができない．しかしながら，その前提も企業通貨の登場により覆されようとしている．企業通貨とは，EdyやSuica，楽天ポイント，マイレージなど，企業が発行する商品が購入可能なポイントである．これらの企業通貨はTカードやマイレージのように単一の企業だけでなく，複数の企業が協力して構築しているものも存在するので貨幣のように交換性が高い．企業通貨の市場は年々膨れ上がっており3年後には数十兆円に上ると見られている．さらに，企業通貨は国の壁をも越えてしまう．例えばマイレージなどは航空会社が管理していることもあり，国境の概念は無い．将来的には個人が簡単に貨幣を発行できるようになり，地域通貨のようにそれぞれが異なった性質を持った膨大な種類の貨幣が誕生することも考えられる．地域通貨の可能性に関しては河邑の「エンデの遺言」[5]が詳しい．

ビジネスにおける「組織から個人へ向かう流れ」をフリードマンは著書「フラット化する世界」[6]においてGlobalizationという言葉で表している．フリードマンはグローバル化をGlobalization 1.0，Globalization 2.0，Globalization 3.0の3段階に定義している．Globalization 1.0はコロンブスがアメリカ大陸を発見した1492年から1800年までの国のグローバル化である．Globalization 1.0では自国をどの

ようにグローバルな競争やチャンスに適合するかが課題であった．Globalization 2.0は1800年から2000年までで産業革命によってもたらされた企業のグローバル化である．Globalization 2.0では自社をどのように世界経済に適合するのか，どのようにビジネスチャンスをものにするのかが課題であった．そしてGlobalization 3.0がインターネットやパーソナルコンピュータ，アウトソーシング，オフショアリング，Googleなどによってもたらされた個人のグローバル化である．Globalization 3.0では個人がグローバルに協力してチャンスをものにするかが課題である．

このように，技術が進歩すると共に個人の力は確実に拡張されている．では，「個人の能力を拡張する」技術を発見するにはどうすればいいのだろうか．クリステンセンの著書「イノベーションへの解」[3]によると破壊的技術を見出すにはターゲットとする顧客を属性ベースで分析するのではなく，状況ベースで分析すべきだと言っている．属性ベースの分析とは，顧客を年齢や性別，職業などで分類し，ある属性を満たす商品は何か，と考えるマーケティングの分析手法である．状況ベースに関する記述を以下に引用する．

> マーケティングで狙い通りの成果をあげるためには，顧客がものを購入したり利用したりする状況を理解することが欠かせない．具体的に言えば，顧客(個人や企業)の生活にはさまざまな「用事」がしょっちゅう発生し，彼らはとにかくそれを片づけなくてはならない．顧客は用事を片づけなければならないことに気付くと，その用事を片づけるために「雇える」製品やサービスがないものかと探し回る．顧客は実際，こんな風に暮らしているのだ．彼らの思考プロセスには，まず何かを片づけなくてはという認識が生じ，次に彼らはその用事をできるだけ効果的に，手軽に，そして安くこなせる物または人を雇おうとする．顧客が製品を購入する状況を構成するのは，顧客が片付け無くてはならない用事の機能的，感情的，社会的な側面である．わかりやすく言えば，顧客が片づけようとする「用事」や，その用事を通じて達成しようとする成果が，状況ベースの市場区分を構成するのである．製品のターゲットを顧客そのものではなく，顧客がおかれている状況に絞る企業が，狙い通り成功する製品を導入できる企業である．別の言い方をすれば，かぎとなる分析単位は，顧客ではなく状況なのだ．

つまり，新しい技術を開発するためにはユーザが何を実現したいかというユーザの体験(ユーザエクスペリエンス)に着目し，その体験をできるだけ簡単に実現する手法を考えれば良いということになる．

以上より得られた知見を基に抽出されたパーソナルコンピュータの次を暗示する指標を以下に示す．

- 現段階ではパーソナルコンピュータとは全く異なった市場を持つ
- パーソナルコンピュータより単純、低価格、高信頼性、便利
- 今まで個人では困難であったものを簡単にする

この3つの指標を満たすものとして筆者は無線センサネットワークに着目した．無線センサネットワークは，センサノード(無線通信機能を具備したセンサ)を空間に散布し，センサノード同士が自律的に通信を行いながらネットワークを構築することで実空間に存在する膨大な量の情報を取得するための技術である．センサノードはサイズが小さく，バッテリで動作するので設置の自由度がきわめて高いことが特徴である．

無線センサネットワークは現在のパーソナルコンピュータと全く異なる市場を持つ．現在のところ，ビルオートメーション，トレーサビリティ，構造ヘルスモニタリング，ヘルスケア支援，環境モニタリングなどさまざまな応用が考えられている．これらのどの応用もパーソナルコンピュータの市場とは一致しない．さらに，パーソナルコンピュータの対象が人であるのに対し，センサノードの対象は壁や椅子などのオブジェクトであるので市場規模はパーソナルコンピュータに比べて大きい．携帯電話をパーソナルコンピュータの次であると捉える向きもあるが，携帯電話の対象が人である以上パーソナルコンピュータの市場規模と同等である．つまり携帯電話はパーソナルコンピュータの次と捉えるよりもパーソナルコンピュータの一部であると捉える方が自然である．

また，センサノードはパーソナルコンピュータよりも単純，低価格，高信頼である．コンピュータは計算能力，保存能力，インプット/アウトプットする能力の3つの要素から構成される．コンピュータの歴史はこの3つの要素をどのような技術によって実現したかであると見ることができる．パーソナルコンピュータは32bit CPU，ハードディスク，インターネット，ディスプレイ，マウス，キーボードの組み合わせで発展してきた．センサノードは，1つの半導体デバイス上に8bit CPU，フラッシュメモリを具備している．また，インプット/アウトプット機能として無線通信機能とセンサ，アクチュエータを具備する．SoC(System-on-Chip)やMEMS(MicroElectroMechanical System)などの技術と組み合わせることでこれら全ての機能を1つの半導体デバイス上に統合することすら可能である．

センサノードが「今まで個人で困難であったものを簡単にする」かどうか，が本研究の最終的なテーマとなる．センサノードは小さいため，これまでコンピュータ化されてこなかったオブジェクトをもコンピュータ化し，ネットワークに接続することができる．このセンサノードの持つ特性を利用することで「今まで個人で困難であったものを簡単にする」ことはおそらくできるであろう．

以上のような背景から，筆者は無線センサネットワークを研究テーマとして選択した．そして，センサノードをわれわれの身の回りのオブジェクトに組み込むことが可能である点に着目し，オブジェクトが備えている機能同士を接続してユーザが欲しいサービスをユーザの手で作る事ができる環境の実現を目指した．例えばユーザが「目覚まし時計に合わせてカーテンを制御したい」と考えたときに，現在でも専門家に頼めば「目覚まし時計に合わせてカーテンを制御するサービス」は作ることができる．本研究の最終的なゴールは，このように今まで専門家に頼まなければ実現できなかったサービスをユーザがユーザ自身の手で必要に応じて簡単に作る事ができる環境を作ることである．

この目的を実現するために筆者はまず，無線センサノードの基盤技術に関する調査を行った．当時すでにカリフォルニア大学のバークレー校において開発されたMICA mote[7] 上で動作する nesC[8] で作成された TinyOS[9] が無線センサネットワークの分野の標準的なオペレーティングシステムとして扱われていた．筆者らも当初，TinyOS を使用することを検討した．しかしながら，TinyOS は非常に使い辛かったため，独自のオペレーティングシステムを作成することにした．このような背景で開発されたオペレーティングシステムが PAVENET OS である．つまり，PAVENET OS は開発当初はオペレーティングシステムの研究として開発されたものでは無く，必要に迫られて開発されていた．筆者は PAVENET OS を用いてデバイス連携技術 [10]，地震モニタリング [11]，コンテクストアウェアアプリケーション [12]，無線通信プロトコル [13] の研究などさまざまな研究で使用しながら，これらのアプリケーションの要求に応じて改変していった．その結果として研究成果として新規性を十分に持つオペレーティングシステムが完成した．PAVENET OS に関しては 3 章で詳細に述べる．

PAVENET OS を開発する傍ら，筆者の当初の目的であったユーザが身の回りのオブジェクト同士を連携させてさまざまなサービスを構築することが可能な ANTH と呼ばれるフレームワークの研究を行った．ANTH を用いることで目覚まし時計に合わせてテレビのスイッチを入れたり，携帯電話の着信に合わせて部屋の照明を点滅させたりといった空間を巻き込んだサービスがユーザの手で簡単に構築できるようになる．ANTH はきわめて少ない計算資源上でヒューマンインタフェースと柔軟なデバイス同士の接続を実現可能な機構を実現している．ANTH に関しては 4 章で詳細に述べる．

1.2 本論文の構成

本論文ではまず，2章において無線センサネットワークの技術動向について述べ，本研究の位置付けを明らかにする．具体的には，無線センサネットワークの基盤技術としてハードウェアとソフトウェアについて，無線センサネットワーク技術の利用形態として軍用，データ収集，ユビキタスコンピューティングのそれぞれの分類での応用を示す．3章では，無線センサネットワークの基盤技術であるPAVENET OSについて述べる．PAVENET OSは無線センサネットワークにおける通信プロトコルからアプリケーションまでを効率的に開発するための機能を備えたオペレーティングシステムである．特にTinyOSと同等の省資源性と性能を実現しつつも，TinyOSでは欠けているプログラムの書きやすさやハードリアルタイム処理のサポートを提供している点に特色がある．4章では無線センサネットワーク技術の応用研究であるANTHについて述べる．ANTHはコップや目覚まし時計や椅子などのユーザの身の回りのありとあらゆるオブジェクトにセンサやアクチュエータが組み込まれた環境においてこれらのオブジェクトを連携させることで新たなサービスを構築可能とするデバイス連携フレームワークである．そして最後に5章において本論文の結論を述べる．

第2章
無線センサネットワーク

2.1 はじめに

無線センサネットワークの研究は，アメリカを中心に発達した軍事研究を主体とした研究グループと，ヨーロッパ・日本を中心にユビキタスコンピューティングの実現を目指した研究グループの2つに分類される．

アメリカを中心に発達した研究グループは，カリフォルニア大学バークレー校の電子デバイス的視点を持つSmartDust ProjectのKris Pisterと，ネットワーク技術的視点を持つアドホックネットワークの分野のDeborah Estrinの2つのグループが1999年頃のほぼ同時期に研究を立ち上げている[14][15]．その後全米の大学を中心にDARPAのNEST Projectによって膨大な金額の研究予算が割り当てられ，軍事研究を中心に発達していた．

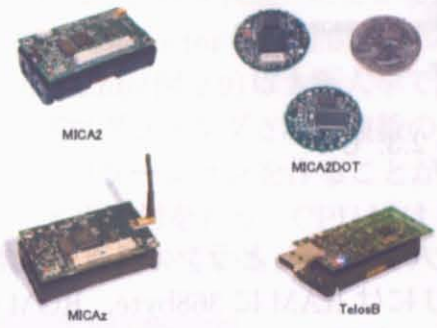
ヨーロッパ・日本を中心に発達した研究は当初，さまざまな地域で分散的に発生し，後にユビキタスコンピューティングの名の下に統一した流れとなっていった．2000年から2002年ぐらいまでに，ヨーロッパではイギリスのランカスター大学，ドイツのカールスルーエ大学，スイス連邦工科大学チューリッヒ校，スウェーデンのInteractive Institute，フィンランドのVTT，日本では東京大学のU^3などでそれぞれ研究が開始された．ヨーロッパ・日本の研究の特色は，人やわれわれの身の回りのオブジェクトに組み込んで使用する応用形態を中心に考えられている点である．

本章ではこれら2つのグループにおける研究動向を横断的に述べる．

2.2 無線センサノードの基盤技術

2.2.1 ハードウェア

無線センサノードの構成はCPU，無線通信モジュール，外部デバイスから構成される．外部デバイスとCPUはI2CやUART，ADCで接続され，用途に応じて取り外しが可能なように設計されるのが一般的となっている．ほとんどのセンサノードが似たような構成になっており，明確な違いは細かいスペック程度になっているのが現状である．



(出展: http://www.xbow.com)

図 2.1: moteシリーズ

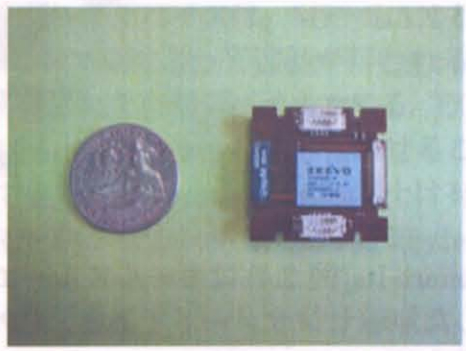

(出展: http://www.intel.com)

図 2.2: Intel Mote

Crossbow社から発売されているmoteシリーズは最も広く使われている無線センサノードである．moteシリーズを図2.1に示す．moteは，カリフォルニア大学バークレー校のSmart Dust Projectの大スケール版であるCots Dust Projectで開発されたセンサノードである．moteシリーズにはMICA2，MICA2DOT，MICAz，TelosBなどが存在する．Texas Instruments社のMSP430，ATMEL社のATmega128Lなどの CPUや，315MHz帯，433MHz帯，868MHz帯，916MHz帯，2.4GHz帯での無線通信とさまざまなラインナップが揃っているのが特徴である．Crossbow社からは発売されてないものの，Intel Mote(図2.2)とよばれるZeevoのBluetooth SOCであるTC2001を用いたセンサノードも存在する．TC2001はコアにARM7TDMIを用いており，無線通信には2.4GHz帯のBluetoothを用いている．RAMが64kbyte，ROMが512kbyteと通常のmoteシリーズに比べてやや豊富な計算資源や通信資源を具備しているのが特徴である．全てのmote上ではオペレーティングシステムとしてTinyOSが動作する．

U^3(図2.3)は東京大学で開発された無線センサノードである．U^3の最大の特徴は電源機能，無線通信機能，CPU機能，センシング機能の4つをハードウェアレ

ベルでモジュール化したことである．CPU には PIC18F452 を用い，無線通信には RFM 社の TR3001 を用いて 315MHz での通信を行うことができる．オペレーティングシステムは HI-TECH 社の PICC18 を用いて開発された PAVENET OS が用いられている．

図 2.3: U^3

Smart-Its(図 2.4) はカールスルーエ大学の Teco とランカスター大学が共同で開発した無線センサノードである．CPU には RAM に 368byte，ROM に 7kbyte を具備する Microchip 社の PIC16F87x を，無線には RFM 社の 868MHz 帯で動作する TR1001 を用いている．Smart-Its は後に Teco の Particle Computer へと発展した．Particle Computer は rfPIC12675H を用いた uPart(図 2.5) や，PIC18F6720 を用いた Particle(図 2.6) などから構成される．uPart は 315, 433, 868, 914 MHz 帯で動作する送信機のみを具備したセンサノードである．RAM を 64byte，ROM を 1792byte 具備している．Particle は Smart-Its と同様に TR1001 の無線を用いており，CPU に PIC18F6720 を用いており，4kbyte の RAM と 128kbyte の ROM を具備する．


(出展: http://www.smart-its.org)

図 2.4: Smart-Its


(出展: http://www.intel.com)

図 2.5: uPart


(出展: http://particle.teco.edu)

図 2.6: Particle

BTnodes(図 2.7) はスイス連邦工科大学チューリッヒ校で開発された無線セン

サノードである．BTnode の無線通信に Bluetooth を用いている点が特徴である．CPU には ATMEL 社の ATmega128L を用いている．

無線センサノードとはやや異なるものの，マイクロコントローラを用いてデバイス同士を有機的に連携させる技術としては，Pin&Play や Ubiquitous Chip が存在する．

Pin&Play(図 2.8) はランカスター大学で開発されているデバイスをボードにピン接続するだけでボード上のデバイスを連携させることができる技術である．notice board と呼ばれるネットワーク機能と電源機能を供給する面状のデバイスに Dallas Semiconductor 社の提供する 1-Wire Device に画鋲のようなピンを接続することで実現されている．notice board にピン接続されたデバイスは，MicroLAN と呼ばれる Dallas Semiconductor 社の提供するプロトコルによって通信が行われる．

Ubiquitous Chip(図 2.9) は大阪大学で開発された小型デバイスである．ECA ルールによってプログラミングされた複数の Ubiquitous Chip を組み合わせることでさまざまなアプリケーションを作ることができる．Ubiquitous Chip 同士は有線接続され，シリアル通信を行う．CPU には RAM に 192byte，ROM に 8kbyte 具備する PIC16F873 を用いている．

(出展: http://www.btnodc.ethz.ch)

図 2.7: BTnode

(出展: http://ubicomp.lancs.ac.uk)

図 2.8: Pin&Play

(出展: http://www-nishio.isc.cng.osaka-u.ac.jp)

図 2.9: Ubiquitous Chip

これらのハードウェアの構成要素を表 2.1 にまとめる．Intel Mote を除くと全てのセンサノードで用いている CPU は 8bit CPU である．CPU は複雑になればなるほど回路サイズが大きくなり，消費電力とコストが増加する．無線センサノードでは 8bit CPU を用いなければならないという明確な理由は無い．これまでパーソナルコンピュータでは CPU の動作速度の速さやハードディスクの容量の多さが実現できるアプリケーションの幅を決めていた．それに対して無線センサネットワークでは省資源性と省電力性がアプリケーションの幅を決める．そのため，無線センサネットワークではアプリケーションに必要最低限の CPU を選定することが重要となる．当然のことながら，16bit CPU を用いなければ実現できないようなア

プリケーションであれば16bit CPUを用いても良い．しかしながら，8bit CPUと16bit CPUでは複雑さに大きな開きがあるため，なるべくなら8bit CPUで済ませたいところである．もし可能であるならば4bit CPUの使用も視野にいれる必要がある．

無線通信ではCPUに比べてより多様なさまざまな規格が用いられる．これは国によって使用できる周波数帯が違うことに起因する．2.4 GHz帯はISM帯であるが，スペクトル拡散をかけなければならないので消費電力が大きくなる．現在ZigBeeが無線センサネットワークの標準的な無線通信プロトコルとされる向きがあるが，消費電力の観点からは減衰の激しい2.4GHzよりも数百MHz帯の方が適していると言える．また，無線通信の消費電力はMACプロトコルに強く依存し，さらに無線センサネットワークではアプリケーションに応じてMACプロトコルに求められる要件が全く異なることから無線モジュールは物理層だけ具備しているを選び，MAC層は自前で開発することも多い．これらの背景を踏まえてユーザはアプリケーションの要求と使用する国の法律に従って適切な無線モジュールを選定する必要がある．

CPU	ATmega128L	PIC16	PIC18	ARM7TDMI	MSP430
周波数	300 MHz	433 MHz	868 MHz	916 MHz	2.4 GHz
変調方式	ASK	FSK	PSK	FHSS	DSSS
プロトコル	独自	ZigBee	Bluetooth		

表 2.1: センサノードの構成要素

2.2.2 ソフトウェア

これまでのインターネットを前提としたミドルウェアではアプリケーションの作りやすさをどのように実現するかが最大のテーマであった．それに対して無線センサネットワークにおけるミドルウェアでは，アプリケーションの作りやすさと同時に低消費電力性や省資源性をも考慮しなければならない．特に無線通信ではMACプロトコルなどをアプリケーションに特化して設計することで低消費電力性が実現できるため，通信の下位層までをも考慮した仕組みが必須である．

オペレーティングシステム

無線センサネットワークにおけるオペレーティングシステムは，少ない計算資源で実現できること，無線センサネットワークの通信プロトコルからアプリケー

ションまでを構築できることの2つが求められる．現在存在する無線センサネットワーク用のオペレーティングシステムはevent modelで設計されたものとthread modelで設計されたものの2種類が存在する．

event modelを図2.10に示す．event modelは1つのevent loopと多数のevent handlerから構成される．event loopはeventの到着を待ち，eventが届くとeventに関連付けられているevent handlerを実行する．event modelではイベント駆動型プログラミング(event driven programing)によってアプリケーションが記述される．event handlerは寿命の短いrun-to-completionで記述され，pre-emptionされることが無い．つまり，event modelは実行ストリームが1つで実現されるため，省資源かつ低オーバヘッドで並列性を実現できる．しかしながら，ユーザが一連の処理を細かい処理に分割しなければならないという問題が発生する．さらに，event modelではタスクのpre-emptionを禁止しているのでハードリアルタイム処理のサポートができない．

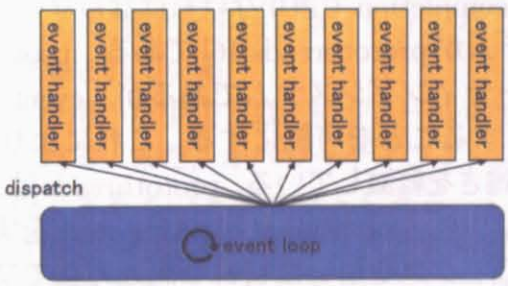


図2.10: event model

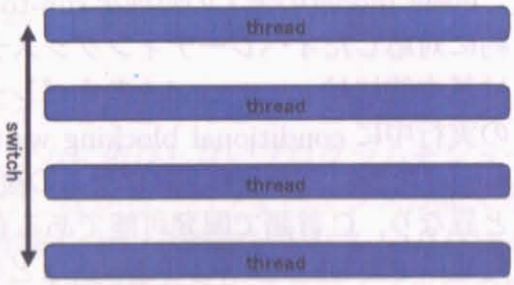


図2.11: thread model

thread modelを図2.11に示す．thread modelは複数のthreadから構成される．各threadはそれぞれ独立に実行ストリームを持っており，低い優先度のthreadは高い優先度のthreadにpre-emptionされるという特徴を持つ．thread modelではユーザは一連の処理を1つのスレッドとして記述することができる．また，pre-emptionを行うことも想定しているのでハードリアルタイム処理をサポートすることができる．しかしながら，pre-emption時のオーバヘッドの大きさやthread間の共有資源へのアクセス制御が困難であるという問題を持っている．

event modelで設計されたオペレーティングシステムの中で代表的なものがTinyOS[9]である．TinyOSはカリフォルニア大学バークレー校のCots Dust Projectで開発されたオペレーティングシステムである．現在無線センサネットワークの標準的なオペレーティングシステムとして扱われており，Crossbow社から発売されているMICA2やMICAz，Telos，iMote上で動作する．TinyOSはevent modelで構築されているのでタスクスイッチのオーバヘッドが51サイクルと小さい．そのた

め，ベストエフォートではあるものの無線通信における物理層などの記述も行うことができる．しかしながら，先ほども述べたようにevent modelでは一連の処理を複数のevent handlerに分解して記述しなければならないため，ユーザが処理の全体像を把握するのが困難であるという問題を抱えている．TinyOSではnesC[8]と呼ばれるevent model用の新しい言語で複数のevent handlerを1つのモジュールとして設計可能な機能を提供することでevent modelの持つプログラムの開発のし辛さを軽減している．一方でユーザはnesCを学ばなければならない点はTinyOSを使用するための敷居を上げている．また，TinyOSではシミュレータのTOSSIMやVMであるBombilla，ファイルシステムであるMatchbox，センサネットワークをデータベースとして扱うTinyDBなどさまざまなツールが揃っていることが最大の強みである．さらに，TinyOSはCPUの特別な機能を使用せずに実装可能であるため，移植性が高く，ATMEL社のATmega128LやTexsus Instruments社のMSP430，ARM7などさまざまなCPUに移植されている．

event modelの全ての処理をrun-to-completionで書かなければならないという制約に対応したオペレーティングシステムがprotothreads[16]である．protothreadsは基本的にはevent modelのオペレーティングシステムであるが，event handlerの実行中にconditional blocking waitすることを可能にする．これにより，event modelを用いた場合のプログラムの複雑さを軽減している．protothreadsはTinyOSと異なり，C言語で開発可能であるものの，conditional blocking waitを実現するためにPT_BEGINやPT_ENDなどのマクロを用いた特殊な記述方法でプログラムを開発しなければならない．そのため，ユーザはnesCを学習する手間ほどには負担がかからないものの特殊な記述方法を学ばなければならない．

MANTISはMICA2上で動作するオペレーティングシステムである．TinyOSやprotothreadsなどのevent modelで設計されたオペレーティングシステムに対してMANTISはthread modelで設計されている．MANTISのスレッドは10ms毎にOSがpre-emptionによってタスクスイッチを行うtime sliced multi-threadingで実現されているため，ユーザはタスクスイッチを意識せずにプログラムを開発することができる．しかしながら，タスクスイッチする際にCPUのレジスタの値などのコンテクストを保存しなければならないため，event modelで構築されたTinyOSのタスクスイッチのオーバヘッド51サイクルに比べて400サイクルと大きい．

このように，現状ではevent modelのTinyOSが省資源性とパフォーマンスの観点から最も優れていると言える．さらに，Crossbow社を介してMICA2やTelosが実際に購入できること，開発環境が整備されていること，移植性が高いこともTinyOSの長所である．しかしながら，event modelの抱えるプログラムの書き辛さやハードリアルタイム処理をサポートできない点はアプリケーションによっては問題になると考えられる．

プログラムモジュール

無線センサネットワークはネットワークを構成するノードの数が膨大であり，一度センサノードを配置した後は収集が困難であるという特徴を持つ．そのため，長期間の動作が求められることから既存の無線センサネットワークに対して新しい機能を追加したいという要求や既存のソフトウェアに含まれる不具合を修正したいといった要求が生まれる．このような背景から，センサノード上で動作するソフトウェアを動的に変更可能なプログラムモジュールの研究が多く行われている．

プログラムモジュールを実現する際には次の2点を考慮しなければならない．1つ目はプログラムモジュールのサイズである．無線センサネットワークでは無線の帯域が狭いため，効率的なモジュールの配布を実現するためにはモジュールサイズをできるだけ小さくする必要がある．2つ目はプログラムモジュールの安全な実行である．新しく配布されたモジュールがシステム全体が停止してしまうといった致命的な状況は避けなければならない．このような要件に対して，VM(Virtual Machine) を用いた手法とネイティブなコードを実行する手法の2通りのアプローチが提案されている．

VM を用いた手法では，モジュールのサイズを小さくできることと安全な実行が実現される．また，オペレーティングシステムに依存しないプログラムモジュールを実現できるというメリットもある．しかしながら，パフォーマンスが低くなるという問題や VM に必要とされる計算資源多いという問題が生じる．VM*はセンサノード用の Java VM である．VM 自体を必要に応じてカスタマイズ可能な機能を提供することで VM に必要とされる計算資源を必要最低限に抑えることを可能としている．ASVM(Application Specific Virtual Machine) や VAWZ(Virtualizing Architecture for Wireless Sensors) では，アプリケーションに特化した命令を具備した VM とそれに対応したコンパイラを自動生成することでプログラムのモジュールサイズの削減を実現している．例えば，VM に乱数を生成する命令「rnd」を組み込むことで，乱数を使用するプログラムモジュールを小さくすることができる．

ネイティブコードを実行する手法では，実行パフォーマンスが良いものの VM を用いた手法に比べてプログラムモジュールの安全な実行やプログラムモジュールのサイズが課題となる．SOS は event model を用いたオペレーティングシステムであり，event handler を動的にロード可能な仕組みを提供している．また，WDT(Watch Dog Timer) を用いてプログラムモジュールの暴走を検出することができる．t-kernel では，プログラムモジュールのロード時にプログラムモジュールを書き換えることでメモリ保護を実現している．

プロトコルスタック

無線センサネットワークではアプリケーションに応じて通信に対する要件が異なるため，さまざまなネットワークプロトコルやMACプロトコルが提案されている．例えばネットワークプロトコルの大まかな分類だけでもデータの収集 (Collection)，データの配布 (Dissemination)，データの集約 (Aggregation) と3種類存在し，それぞれが全く異なった動作をする．それぞれのプロトコルは異なるインタフェースで独立に開発されているため，相互運用性が無い．このような問題に対してネットワーク層とリンク層の間に各プロトコルを統一的に扱うSP(Sensornet Protocol) と呼ばれる中間層を用意するべきだという議論がなされている．SPはインターネットアーキテクチャで言うところのIPに相当し，ネットワーク層とリンク層のインタフェースをSPに統一することで相互運用性を高めることを目指している．

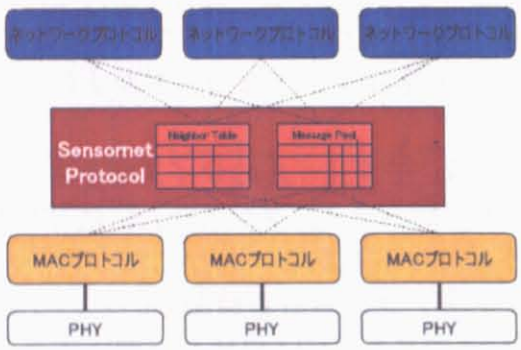


図 2.12: Sensornet Protocol

図 2.12 にSPの全体像を示す．SPは neighbor table と message pool の2つを管理している．neighbor table は隣接ノードを管理するテーブルである．隣接ノードのアドレスやスリープ状態とアクティブ状態のスケジュール，リンクの品質などを管理することでルーティングプロトコルをサポートする．message pool はネットワーク層から送信されるデータを一時的に保持し，MAC層へと渡すための機構である．SPでは全てのMACプロトコルや通信プロトコルで共通した message pool を用いることで複数のプロトコルを1つのネットワーク内に共存することを可能にしている．また，message pool ではパケットの許容される遅延や必要とされる信頼性，混雑状態などを管理しており，ネットワーク層やMAC層に対してフィードバック情報を提供する．

SPは既に TinyOS 上に実装されており，SPを用いてもオーバヘッドにならないことが確認されている [17].

2.3 無線センサネットワークの利用形態

無線センサネットワークは応用分野が極めて広いため，多岐に渡る分野に対するアプリケーションの研究が行われている．また，アプリケーションに応じて通信プロトコルや計算資源，ハードウェアに対する要件が大きく異なるのでアプリケーションからトップダウン的にアーキテクチャを検討する必要がある．本節ではこのような無線センサネットワークのさまざまな分野における利用形態を軍用，データ収集，ユビキタスコンピューティングの3つの視点で述べる．

2.3.1 軍用

EnviroTrack

EnviroTrack は無線センサネットワークを使って戦車や車などを追跡するためのオブジェクトベースのミドルウェアである．EnviroTrack ではユーザは sense と state の2つの関数を定義するだけでオブジェクトをトラッキングすることができる．sense はブール関数であり，センサからの値を取得して何かを検出したら true，検出しなかったら false を返す関数である．検出を判断する閾値などの値をユーザが定義する．state は追跡するオブジェクトの位置を算出するための関数である．位置を計算するために各センサノードから位置を算出するノードにどのような値を送るかや，送られてきた値をどのように集約して位置情報を算出するかをユーザが定義する．

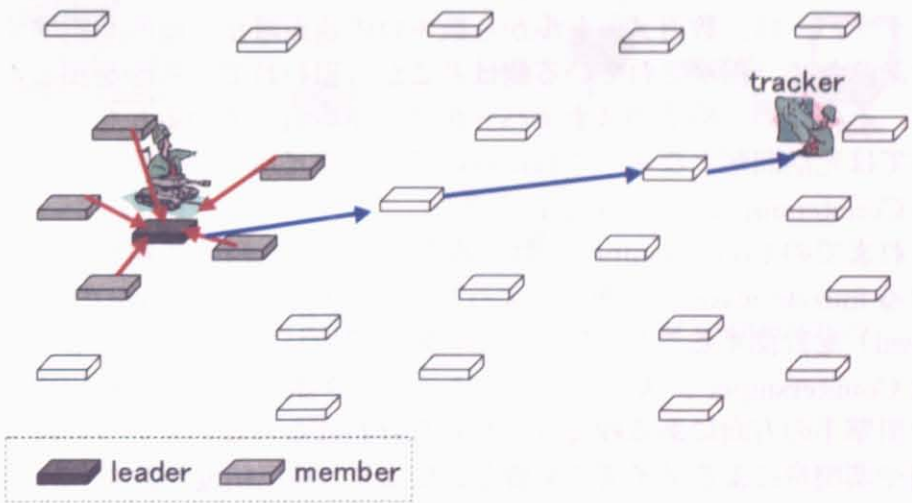


図 2.13: EnviroTrack

図 2.13 に EnviroTrack の動作例を示す．EnviroTrack は leader ノードと member ノードと tracker から構成される．EnviroTrack では各センサノードは固定的に配置されており，自分の位置を知っているという前提である．まず，それぞれのセンサノードは sense 関数を定期的に実行する．この動作例では sense 関数は磁気センサを用いている．センサノードの近くを戦車などのオブジェクトが通過すると磁気センサが反応し，sense 関数から true が返る．するとセンサノードは member ノードへと遷移し，member ノード同士でネゴシエーションを行って leader ノードの選出を行う．leader ノードは各 member ノードから位置など情報を収集する．そして state 関数を用いて検出したオブジェクトの位置を算出する．そして最後に算出した位置を位置情報に基いたルーティングプロトコルによって tracker まで配送する．

EnviroTrack では landmark routing[18] に似たルーティングプロトコルを用いている．EnviroTrack では LANMAR における landmark node の役割を leader が行っており，leader がデータを収集し，tracker へとデータを送信する．landmark routing では landmark node の役割は固定的であったが，EnviroTrack ではセンサノードは動かないので leader をターゲットに合わせてハンドオフしていく点が異なる．

EnviroTrack は TinyOS を用いて MICA2 上に実装されており，1000 分の 1 のモデルで実験した結果，時速 50km のオブジェクトを追跡することが確認されている．

Counter Sniper

スナイパーは，数百メートルから数キロの遠距離から標的を狙撃することが可能であるため，狙撃されている側はどこから狙われているかを知ることが困難である．そのため，相手のスナイパーがどこから狙っているかを知ることは軍隊にとっては死活問題となる．これに向けて，スナイパーの位置を検出するための技術，「Countersniper」が研究されてきた．

これまでの Countersniper では，複数のマイクを用いて狙撃時に銃口から発せられる muzzle wave と，弾丸によって生成される shock wave の TOA（Time Of Arrival）を計測することで狙撃手の位置を算出していた．しかしながら，このような Countersniper システムは muzzle wave や shock wave を検出するためのマイクを狙撃手の方向にある程度は合わせなければならないという問題や，波動の反射波や環境音によるノイズの影響などにより，検出精度が数%〜10%と低かった．

このような問題に対して，NEST プロジェクトの一環で，アメリカ Vanderbilt 大学の Simon らは無線センサネットワークを利用した Countersniper システムの研究を行っている [19]．図 2.14 に無線センサネットワークを用いた Countersniper

2.3.2 データ収集

TinyDB

TinyDB は無線センサネットワークにおいてデータ収集を行うアプリケーションのためのミドルウェアである．TinyOS を用いて MICA2 上に実装されている．TinyDB では，ユーザはデータを収集する時に SQL に似た宣言的問い合わせ言語を用いることで個別のセンサノードそのものよりもセンサノードの持つデータに注目して処理を記述することができる．また，ユーザの作成したクエリに応じてセンサデータのアグリゲーションやパケットの結合をシステムが自動的に行うことでパケットのルーティング時のオーバヘッドを減らし，低消費電力性を実現する．

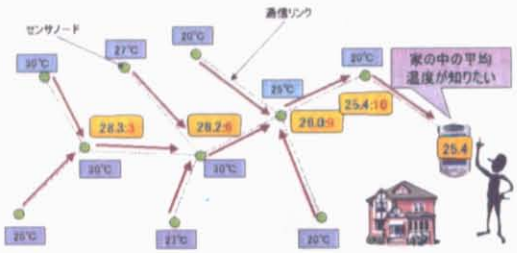


図 2.15: TinyDB

動作例として図 2.15 に 5 秒毎に部屋の中の平均温度を取得する場合を示す．まず，各センサノードはベースステーションを root とする routing tree を作成する．クエリやセンサデータは routing tree を用いた tree-based routing によって配送される．ユーザはまず，PDA や PC などのベースステーションで

```
SELECT AVG(temp)
FROM sensors
SAMPLE PERIOD 5s
```

を実行する．するとベースステーション内で無線センサネットワーク用のクエリが作成され，個々のノードへと配送される．クエリを受け取ったセンサノードは，5 秒毎にセンサデータを親ノードへと配送する．それぞれのセンサノードは，子ノードからセンサデータが送られてくるまで親ノードに対してデータを送信しない．子ノードからセンサデータを受け取った親ノードは，受け取ったセンサデータの平均と平均を算出したノードの数をさらに自分の親ノードへと送信する．これを繰り返すことにより，ユーザは部屋の平均温度を取得することができる．また，routing tree 内でセンサデータの集約が行われるため，単純にデータを収集し

2.3.2 データ収集

TinyDB

TinyDB は無線センサネットワークにおいてデータ収集を行うアプリケーションのためのミドルウェアである．TinyOS を用いて MICA2 上に実装されている．TinyDB では，ユーザはデータを収集する時に SQL に似た宣言的問い合わせ言語を用いることで個別のセンサノードそのものよりもセンサノードの持つデータに注目して処理を記述することができる．また，ユーザの作成したクエリに応じてセンサデータのアグリゲーションやパケットの結合をシステムが自動的に行うことでパケットのルーティング時のオーバヘッドを減らし，低消費電力性を実現する．

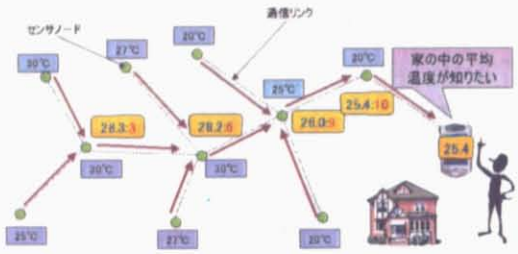


図 2.15: TinyDB

動作例として図 2.15 に 5 秒毎に部屋の中の平均温度を取得する場合を示す．まず，各センサノードはベースステーションを root とする routing tree を作成する．クエリやセンサデータは routing tree を用いた tree-based routing によって配送される．ユーザはまず，PDA や PC などのベースステーションで

```
SELECT AVG(temp)
FROM sensors
SAMPLE PERIOD 5s
```

を実行する．するとベースステーション内で無線センサネットワーク用のクエリが作成され，個々のノードへと配送される．クエリを受け取ったセンサノードは，5 秒毎にセンサデータを親ノードへと配送する．それぞれのセンサノードは，子ノードからセンサデータが送られてくるまで親ノードに対してデータを送信しない．子ノードからセンサデータを受け取った親ノードは，受け取ったセンサデータの平均と平均を算出したノードの数をさらに自分の親ノードへと送信する．これを繰り返すことにより，ユーザは部屋の平均温度を取得することができる．また，routing tree 内でセンサデータの集約が行われるため，単純にデータを収集し

てベースステーションで平均を計算した時に比べて少ない消費電力で平均温度を取得することができる.

生態観測

生態観測は生物学者が研究目的などで植物や動物の生態を観測する作業を意味する. これまで生態観測は人手によるデータの収集が主であったが, 長い期間に渡って観測する場合のコストが非常にかかることや, 人が動物や植物の生態に影響を与えてしまうことなどが問題であった. 無線センサネットワークを用いることでこのような問題を解決することができる.

Mainwaring らは, GDI(Great Duck Island) と呼ばれる島で海鳥の観測を行っている [21]. この研究では, 海鳥の巣の使用状況や季節による島の環境の変化, 海鳥の群れが島の環境に与える影響などを調べている.

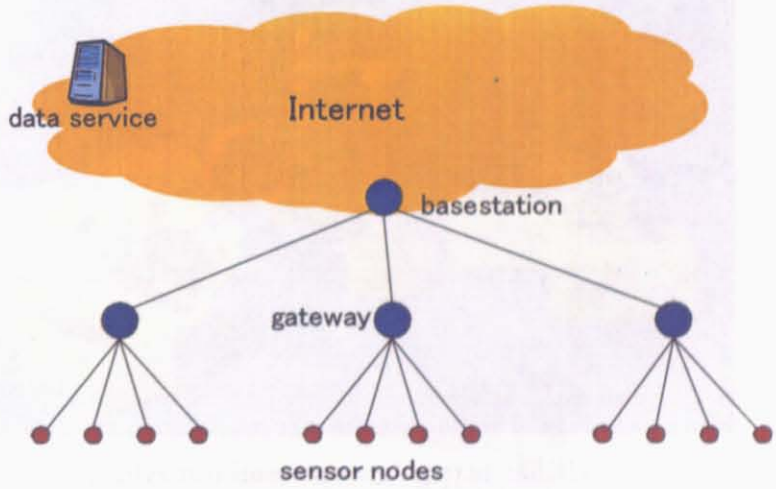


図 2.16: 生態観測

図 2.16 に [21] のシステムの全体像を示す. [21] は階層構造を持ったネットワーク形態を持つ. 末端のセンサノードは MICA2 を用いており, アクリルのケースに入れられている. 海鳥が巣にいるかどうかは巣の温度を測ることで得られるため, センサノードを巣の下に埋め込んでいる. また, センサノードは普段はスリープ状態にあり, 間欠的にゲートウェイに対してシングルホップでデータを送信している. ゲートウェイノードにも MICA2 を用いている. 末端のセンサノードとゲートウェイノードの違いは末端のセンサノードがバッテリで動作し, 間欠的な通信しか行わないのに対してゲートウェイノードはソーラーパネルで動作し, 常に無線をアクティブにしている点である. ゲートウェイノードからベースステーショ

ンにもシングルホップでデータが送信され，インターネット上にあるデータベースへと蓄えられる．

海中モニタリング

地表の70%を覆う海は未だ人類にとって謎の多い領域である．ダイバーであっても，1回に海中に潜っていられるのは1時間程度であり，深さも数十メートルまでしか潜る事はできない．このような海中をロボットとセンサノードを用いて観測することで海中の汚染状況や生態系の解明などを実現することができる [22]

(出展: http://groups.csail.mit.edu/)

図 2.17: 海中モニタリング

[22] におけるセンサノードは固定ノードと移動ノードから構成される．固定ノードは Aquafleck と呼ばれる．Aquafleck は，CPU に ATmega128，OS には TinyOS，通信に 320kbps の光通信モジュールと 50bps の超音波通信モジュールを具備している．また，センサには 255 × 143 × 8bit のカメラ，圧力センサ，温度センサを持つ．Aquafleck は 512kbyte のフラッシュメモリを備えており，データを蓄積している．また，Aquafleck 同士は音波を用いて通信を行うことができる．移動ノードは Amour AUV と呼ばれる．Amour AUV は 4 つの制御エンジンを具備しており，圧力センサと磁気コンパスを用いて海中での位置を把握しながら Aquafleck からデータを収集する．Amour AUV の底には円柱状の穴が開いており，Aquafleck と

結合し，光で通信を行ったり Aquafleck の場所を移動させたりすることができる．

構造ヘルスモニタリング

構造ヘルスモニタリングは建物の健全性を評価するための技術である．建造物は見た目は正常でも柱に亀裂が入っていたりする場合には専門家でなければその建物が安全かどうかを判断することはできない．このような問題に対して加速度センサを具備したセンサノードを高密度に配置し，データを収集し，加速度の変化を分析することで建物の健全性の評価の自動化を実現することができる [23].

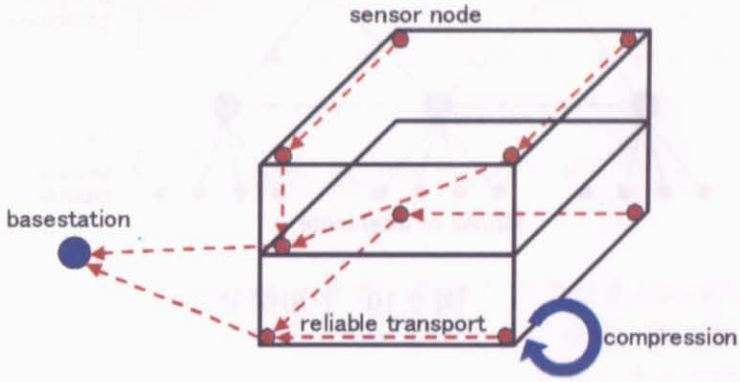


図 2.18: 構造ヘルスモニタリング

図 2.18 に [23] における無線センサネットワークである Wisden を示す．Wisden はベースステーションとセンサノードから構成される．センサノードには MICA2 を用いており，オペレーティングシステムには TinyOS を利用している．MICA2 には加速度センサが接続されており，100Hz で 3 軸 16bit の計測を行っている．各センサノードはアドホックネットワークを構築しており，BLAST[24] と呼ばれる信頼性の高いリンクを選択してルーティングツリーを構築する仕組みを採用することでデータ通信の信頼性を高めている．また，各センサノードでは Wavelet 変換などを用いた不可逆的な圧縮をすることで少ない帯域でも効率的にデータを収集することを可能としている．

予知保全

予知保全 (PdM) は半導体プラントやオイルタンカーなどに備え付けられた機械類の故障を予め検知し，メンテナンスをすることを目的としている．これまでの予知保全は数週間から数ヶ月毎にユーザが手作業で機械類を点検するために非常

にコストのかかる作業であった．それに対して予知保全に対して無線センサネットワーク技術を適用することで予知保全のコストを大幅に削減することが期待されている [25].

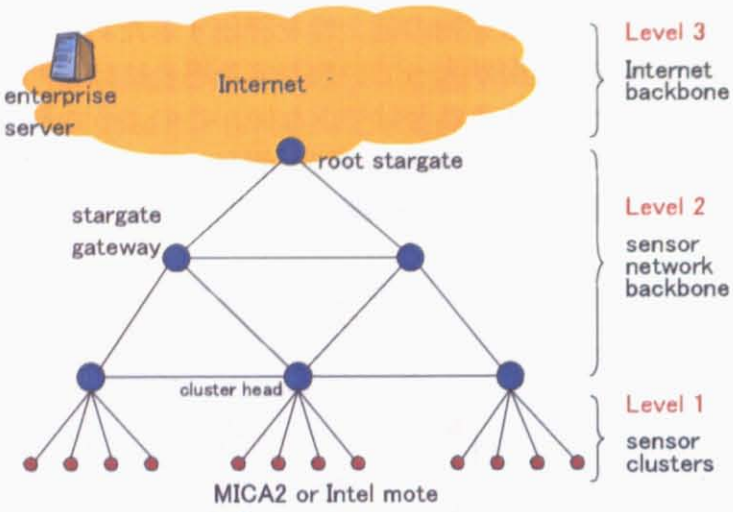


図 2.19: 予知保全

[25] では，MICA2[7] と Intel mote の 2 種類のプラットフォームを用いて実際に半導体プラントとオイルタンカー上に展開している．[25] では，各センサノードが加速度を定期的に集計することで機械類の故障を早めに検知することを試みている．図 2.19 に [25] で用いられているネットワークの構成を示す．[25] では無線センサネットワークを 3 つのレベルを持つ階層によって構築している．第 1 レベルの階層は，MICA2 や Intel Mote などのセンサノードが所属し，センサクラスタと呼ばれる．各センサノードはクラスタヘッドとサンプリング周期に合わせて Wake と Sleep を繰り返しながら通信を行うことで低消費電力性を実現する．第 2 レベルの階層は Stargate Gateway から構成され，センサネットワークバックボーンと呼ばれる．Stargate Gateway は第 1 レベルに属するセンサノードのクラスタヘッドの役割も担う．第 3 レベルの階層はインターネットバックボーンである．企業向けサーバや，第 2 レベルのネットワークとのブリッジを担うルートゲートウェイなどが属する．

森林モニタリング

森林では落雷や木々が極度に乾燥することによる火災が発生する．森林火災は一度火が広がると人手による鎮火は難しく，数週間に渡って続くこともある．その

ため，森林火災の事前の予知や即座に発見することが非常に重要である．しかしながら，森林は面積が巨大な上に森林火災は頻繁に起こるものでは無いため，人手による警備に非常にコストがかかる．このような問題に対して無線センサネットワークを用いることで低コストで森林火災の予知や発見をすることができる [26].

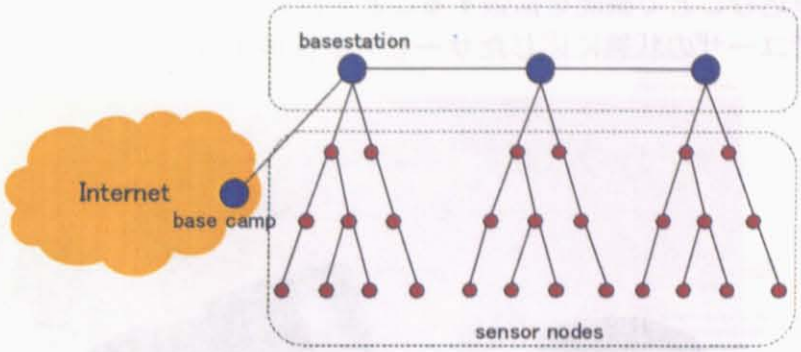


図 2.20: 森林モニタリング

図 2.20 に [26] における無線センサネットワークである FireWxNet の構成を示す．FireWxNet はベースキャンプ，ベースステーション，センサノードの 3 つから構成される．ベースキャンプは FireWxNet とインターネットのゲートウェイの役割を担う．衛星通信を用いて上り 128kbps，下り 512kbps でインターネットと接続している．ベースステーションはベースステーション同士で無線ネットワークを構築し，Backhaul Network を提供する．ベースステーションはバッテリとソーラーパネルを具備しているため，省電力性を考慮する必要は無く，オペレーティングシステムには Linux を用いている．ベースステーション同士は見晴らしのいい場所に設置され，900MHz 帯の無線帯域において指向性アンテナを用いて 10Mbps の通信を行う．各ベースステーションは可動性のウェブカムを具備しており，インターネット経由で森林の状況を観測する事ができる．センサノードはセンサノード同士でアドホックネットワークを構築し，ベースステーションへとセンサデータを配送する．センサノードには MICA2，ソフトウェアには MANTIS OS[27] を用いている．各センサノードはセンサとして温度センサ，湿度センサ，風速計，バッテリとして単 3 電池で動作している．各センサノード間は同期を取っており，15 分毎に 1 分間全てセンサノードが起床し，データをベースステーションまで配送することで低消費電力性を実現している．データの配送には信頼性が必要なため，フラッディングベースのルーティングプロトコルを用いることで信頼性を高めている．

2.3.3 ユビキタスコンピューティング

MediaCup

MediaCup プロジェクトは計算機能，センシング機能，通信機能をコーヒーカップに組み込むことで機能を拡張することを目的としている [28]．MediaCup を用いることでユーザの状態に応じたサービスを提供することが可能となる．

(出典: http://mediacup.teco.edu/)

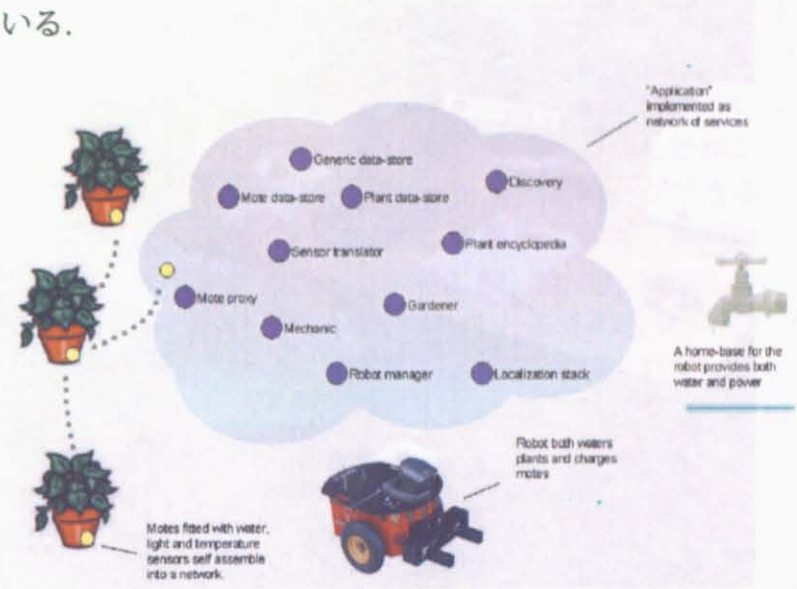
図 2.21: MediaCup

MediaCup を図 2.21 に示す．MediaCup にはセンサとして 3 つのボールスイッチと温度センサが組み込まれている．これらのセンサからの情報を CPU である PIC16F84 で抽出し，を用いてカップの「静止している」「動いている」「飲んでいる」「注がれている」の 4 つの状態を算出している．算出した情報は赤外線を介して天井に付けられた受信機に送信され，CAN(Car Area Network) を用いてパソコンなどに配送される．また，電源機能として 1F のキャパシタを具備しており，無線を介して電力を充電することもできる．

PlantCare

PlantCare は室内用観葉植物の手入れをロボットと無線センサネットワークを用いて自動化するための技術である [29]．ユーザに負担をできるだけかけずに設置から運用までを実現することを最終目的としている．

図 2.22 に PlantCare の全体像を示す．PlantCare はセンサノードとロボットから構成される．センサノードには MICA2 を用いている．MICA2 上では TinyOS が動作しており，光センサ，温度センサ，湿度センサを具備している．また，電源としてキャパシタを用いており，ロボットからの誘導電化によって電力を定期的に充電している．



(出典: http://seattleweb.intel-research.net/)

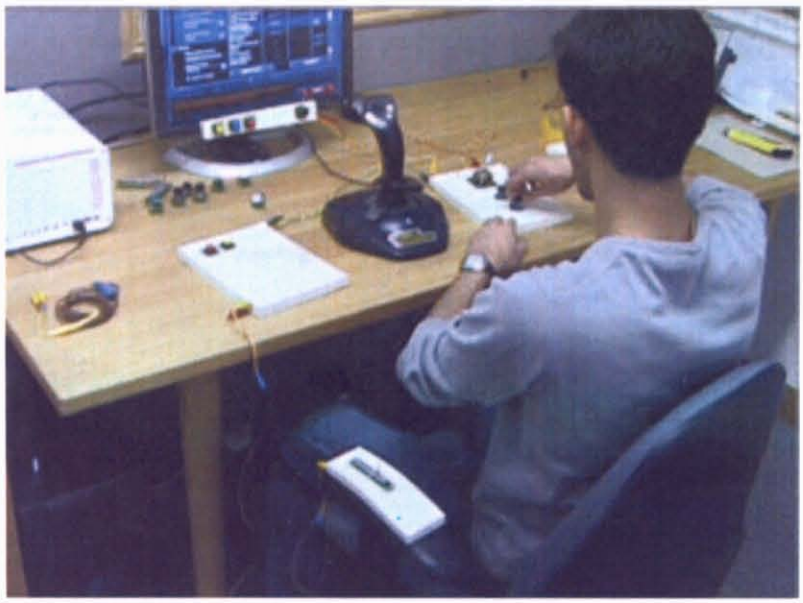
図 2.22: PlantCare

ロボットは ActivMedia Robotics の 2-DX に貯水タンクとポンプ，蛇口を取り付けたものを用いている．ロボットはセンサノードが具備しているセンサよりも高価で高精度なものを備えており，定期的に各センサノードの近くに訪れながらキャリブレーションを行う．また，各センサノードの値から日照状況を算出し，植物を日当たりのいい場所に移動したりといった作業も行う．

VoodooIO

これまでパソコン用のヒューマンインタフェースは企業があらかじめ用意したコントローラなどをパソコンに接続し，利用するというものであった．それに対して VoodooIO はユーザがボタンの位置や数，使いたい機能などを自由自在に組み合わせることでパソコンのヒューマンインタフェースを構築することができる [91]．VoodooIO は Pin&Play[92] から派生したプロジェクトであり，notice board と notice board に接続する各種デバイス，notice board とパソコンを接続する接続ケーブル，パソコンのゲームコントローラなどをエミュレーションするソフト

ウェアから構成される．VoodooIO では，実際に Microsoft の MechWarrior 4™ や Blizzard Entertainment の World of Warcraft™，彼らが自作したキャノンゲームなどのコントローラを作成している (図 2.23)．



(出典: http://eis.comp.lancs.ac.uk/)

図 2.23: VoodooIO

2.4 むすび

本章では 2.2 において無線センサネットワークの基盤技術を，2.3 において無線センサネットワークの利用形態について述べた．無線センサネットワークは出現当初アドホックネットワークの研究者が多く研究していたこともあり，ルーティングプロトコルや MAC プロトコルなどの通信プロトコルが注目されがちである．しかしながら本章ではあえて通信プロトコルについては述べていない．その理由は現在行われている通信プロトコルに関する研究のほとんどがシミュレーションによるものであり，MICA2 を用いた実証実験を行うことができる現在においては現実と乖離しすぎて使い物にならないからである．事実，無線センサネットワークを対象としたルーティングプロトコル [30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40] や MAC プロトコル [41, 42, 43, 44, 45, 46] など多くの提案がされているものの，2.3 で述べたアプリケーションではどれ 1 つとして使用されていない．シミュレーションと現実の乖離を指摘している論文 [24] によれば，現実の無線ネットワークでは一方向のリンクが発生することも多く，環境の変化によって電波の状態も大きく変動する．また，低消費電力性を考えるとマルチホップよりもシングルホップの方が低消費電力で実現できる状況も少なくない [47][48]．これらのことを鑑みると，無線センサネットワークで一番大事なのはアプリケーションであり，トップダウン的な視点で通信プロトコルを設計すべきであると言える．この時，複雑なルーティングプロトコルや MAC プロトコルは必要なく，現実の環境で運用に絶えうるかどうかを確認することが重要である．

一方で 2.3 で見て分かるように多くのアプリケーションで MICA2 と TinyOS が使用されている．これは MICA2 が市販されていることも大きな要因であるが，TinyOS 程度の省資源性と低オーバヘッドが実現できれば無線センサネットワークのさまざまなアプリケーションに対応可能であることを暗に示していると捉えることもできる．