オープンソース・データベースを活用した
医療機器管理用データベースに関する研究

2014 年 3 月

北海道工業大学大学院

渡邉 翔太郎

目　次

第 1 章

序　論

我が国は，すべての国民が良質な医療を享受できることを前提とし，世界でも上位を誇る医療提供体制を整えている．その中で，高度かつ効果的な医療を遂行するためにも，医療機器は必要不可欠な存在となっており，良質な医療提供に多大な貢献をしている．しかし，高度・複雑化が進む医療機器に対して，医療従事者の知識不足による不適切な使用や，医療機器本体の整備不良による故障など，医療機器に起因する事故が問題となっている．そこで，政府は医療安全確保の観点からも，平成 19 年 4 月に第 5 次改正医療法を施行し，すべての医療機関に医療機器の安全管理に係る体制整備を義務付け，医療機器の適切な保守管理と安全使用に関する対策を講じること課せた[1-1]．特に，医療機器の使用前や定期的な保守点検の実施とその実施内容の記録を含めた包括的な医療機器管理を求めるようになった．

このような中で，医療機器の保守管理を適切に実施するためには，医療機器個々の情報を管理する医療機器管理台帳を作成し整備することが重要である[1-2]．この台帳には，保有機器個々を一意に識別する管理番号や機器名称，製造年月，使用期限などを記載し，これを基に医療機器の保守点検実施状況の評価など，購入から廃棄までのライフサイクル管理を行う．従来は，記録用紙を基にして台帳を作成していたが，近年の医療機関が保有する医療機器台数とその種類，使用頻度の増加により，それらの情報量も膨大となることから，施設内に散見するすべての医療機器の包括管理は非常に困難な状況となっていた．そこで，今日では各医療機関においてデータベースシステムを活用した市販の医療機器管理システムの導入が進められている．これにより，膨大な医療機器情報の一元管理を可能とし，さらに施設内のネットワークや専用の携帯情報端末を利用することで，煩雑な医療機器管理業務の効率化を図ることができる．しかし，その導入費用は医療機関の規模にもよるが，数百万円と高額であることに加えて，自施設の医療機器管理スタイルに合わせたカスタマ

イズにも費用が掛かる他，診療報酬に応じた業務収入も乏しいことから，積極的な導入に至らない現状にある[1-3][1-4]．また，このような中で比較的安価な汎用のデータベースソフトウェアである Microsoft Access（Microsoft Access，Inc.）や File Maker（File Maker，Inc.）を活用して，独自の医療機器管理システムを開発している医療機関もある[1-5][1-6]．これにより，自施設の特徴を活かした医療機器管理が可能となる反面，これらの開発を初めから行うには膨大な時間を必要とし，さらにデータベースの概念を理解した専従の開発者の存在が不可欠となる．また，ソフトウェアに掛かるライセンスの問題から他施設での導入とカスタマイズは十分に行えず，医療機器管理に必要な項目や機能も医療機関で様々であるため，その普及には至っていない．

一方で，我が国の医療をめぐる様々な問題に対し，効果的かつ効率的な医療を提供していくための施策として，医事会計システムや電子カルテシステムといった保健医療分野の情報化が推進されている[1-7]．近年，それらの情報化に OSS（Open Source Software）が活用され，その普及促進と標準規格化に期待が寄せられている[1-8]．OSS の利点はソフトウェアの設計図となるソースコードを共有できることであり，このことは，便利で多様な機能を持ったシステムを特定の開発者に依存せず，多数の施設で導入できることを意味する．しかし，現在まで医療機器管理に関する OSS は国内外で存在していない．

このような背景から本研究では，有用な医療機器管理システムの導入や開発が十分に行えない医療機関を対象として，その普及促進と標準規格化に繋がる OSS を活用した新たな医療機器管理ソフトウェアの開発を行った．具体的には，改正医療法で求められる包括的な医療機器管理が行えるシステムを，多数の施設で簡易に導入できる Open-MEMS（Medical Equipment Management Software）と命名する新たな医療機器管理ソフトウェアの開発を行った．また，本研究に協力頂いた病院に導入を行い，Open-MEMS が医療機器管理システム導入のための新たなツールとして，有用であるか検討を行った．

Open-MEMS は，本研究の目的を実現するために，医療機器管理用に作成したアプリケーションサーバを構成する OSS を簡易な方法で導入できるようにしたパッケージソフトウェアとした．具体的には，医療機器情報を管理するデータベースサーバの PostgreSQL 8.4.13 と医療機器管理用 Web アプリケーションを提供する Web サーバの Tomcat 6.0.37 より構成し、RPM Package Manager を使用して，拡張子を「.rpm」としてパッケージ化を行ったソフトウェアとした．このソフトウェアを導入することで，クライアント・サーバ型の医療機器管理システムを簡易に構築できることが最大の特徴である．また，このように開発した Open-MEMS の導入を当該病院で行った結果，医療機器情報のデータベース化と Web アプリケーションによる情報アクセスの即時性を実現し，改正医療法に求められる医療機器の適切な保守点検の実施と記録を含めた包括的な医療機器管理を実施することができた．OSS を活用したソフトウェアは医事会計システムなどで開発が進んでいるが、

現在のところ「医療機器管理」に関するソフトウェアは本研究で開発を行った「Open-MEMS」以外存在していないことから、Open-MEMS が医療機器管理システム導入のための新たなツールとして期待できることが示唆された.

本論文は，本章の序論を第 1 章とする全 7 章から構成されており，次章以降は以下の内容を記述する．第 2 章は，我が国の医療分野の現状と医療機器の安全管理をめぐる動向について述べる．第 3 章は，Open-MEMS の開発に活用した OSS の定義と開発メリットを概説する. 第 4 章は, 本研究で開発を行った Open-MEMS のシステム構成およびソフトウェア構成，パッケージ作成方法について説明する．第 5 章は，研究協力病院を対象に行った Open-MEMS の導入実験について述べる．第 6 章は，Open-MEMS の有用性と臨床導入について考察する．最後に第 7 章にて，本研究で得られた知見を整理し結論を述べる．

第 2 章

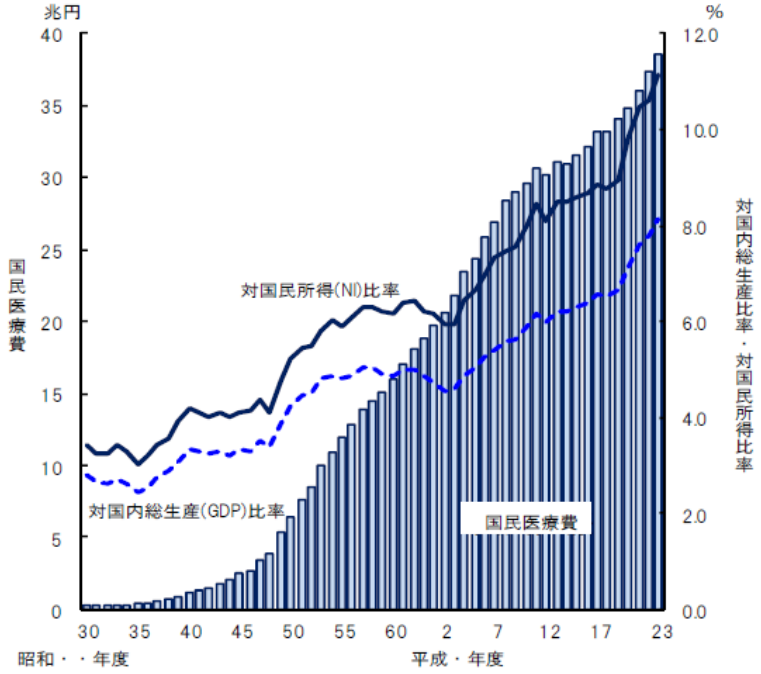
医療分野の現状と
医療機器の安全管理をめぐる動向

我が国は，すべての国民が良質かつ平等な医療を享受できることを前提とし，世界でも上位を誇る医療体制を整えている．その中で，高度に発達してきた医療技術に伴い，医療機器も日々進歩・発展を続け，現代医療を支える上で必要不可欠な存在となっている．しかし，その医療機器に起因する事故が問題をめぐり，いかに医療機器を安全に使用し，管理するかが課題となっている．厚生労働省は薬事法や医療法を改正し，医療機器の安全管理に関する対策を講じている．中でも改正医療法では，すべての医療機関に医療機器の安全管理に係る体制整備を義務付け，医療機器の適切な保守点検の実施と記録を含めた包括的な医療機器管理を求めるようになった．

本章では，我が国の医療提供の現状を踏まえ，医療機器の安全管理をめぐる動向について述べる．

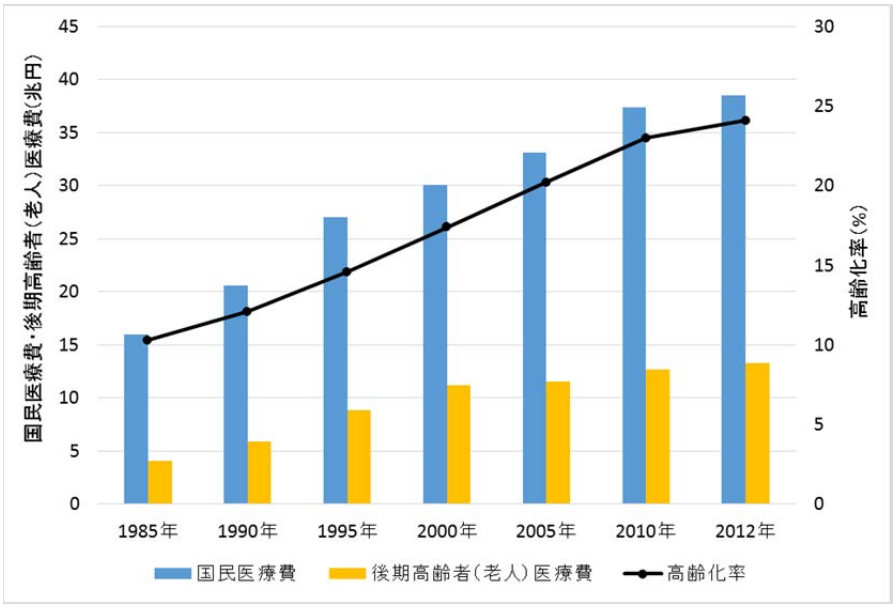
2-1　我が国における医療提供の現状

我が国では，すべての国民が良質かつ平等な医療を享受できることを目指し，医療水準の向上や医療保険制度の整備などに努めている．中でも，戦後における医学や医療技術の進歩・発展は我が国の平均寿命を急速に延ばし，いまや世界有数の長寿国となっている．しかし，平均寿命が延び高齢者人口が増える一方で，新生児の出生率は年々減少する傾向にあり，我が国はいわゆる少子高齢化と呼ばれる状態となっている[2-1]．この少子高齢化の特徴である高齢者人口の増加に伴い，現れた大きな課題として年々増加傾向にある国民医療費がある．これは，図 2-1 に示す「国民医療費・対国内総生産及び国民対所得比の年次推移」と図 2-2 に示す「国民医療費・後期高齢者（老人）医療費・高齢化率の年次推移」から，高齢者人口に比例して国民医療費が増加していることがわかる[2-2][2-3][2-4]．また，景気によって左右される国民所得に対して，国民医療費は無関係に伸び続けている現状も伺える．さらに，図 2-3 からは 65 歳以上の高齢者一人当たりの医療費が 64 歳以下と比べて高額であり，図 2-1 と図 2-2 と合わせることで将来的な医療費負担の増加を容易に予測することができる．



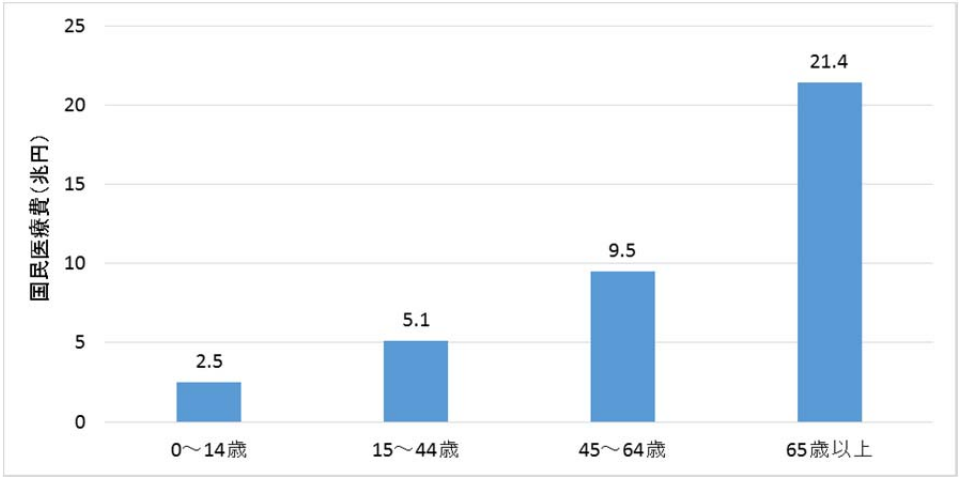
資料　厚生労働省　平成 23 年度国民医療費の概況

図 2-1　国民医療費・対国内総生産及び対国民所得比の年次推移



国民医療費，後期高齢者（老人）医療費は厚生労働省「医療費の動向」、高齢化率は総務省統計局「国勢調査」より作成

図 2-2　国民医療費・後期高齢者（老人）医療費・高齢化率の推移



厚生労働省「平成 23 年度国民医療費の概況」より作成

図 2-3　年齢階級別一人当たりの国民医療費

一方で，国民医療費を基に医療サービスを提供する側である医療機関においては，手厚いケアを必要とする高齢者の受療率が進んだことにより，医療従事者の数が患者数に対して決して十分とは言えない状態にある[2-5]．実際に，我が国での医療機関における 100 床当たりの従事者数は 2012 年時点で，医師が 12.9 人，看護師が 46.0 人となっており年々増加傾向にあるが，これは諸先進国に比べて 1 / 3 ～ 1 / 6 といずれもかなり低い数字であり，医療従事者一人が抱える仕事がいかに多いかを示している [2-6] [2-7]．このことは医療の質に大きく影を落とす一因にもなっており，現実に一人の看護師が二人の患者のベッドを病室から運んだ際に，患者を取り違えてしまったという事故も起きている．これは十分な看護師数を確保して対応していれば起きえなかった事故であり，現場の看護師不足の実態を示すものである．また，近年は地方の病院や救急・産科・小児科における勤務医の医師不足と過度の勤務が深刻な問題となっている．特に，地方における医師確保が困難な現状から残った医師や病院に掛かる業務負担は増大しており，地域の医療機関の診療科縮小や廃止，廃業といった事態が相次いでいる．このことは，とりわけ急性期医療や産科に大きな不安を抱かせるものとなっており，地域における医療体制を根幹から揺るがす事態に陥っている．産科や小児医療に至っては，訴訟リスクの高さや夜間・休日患者が多いことなどから，医療体制が整っている都市部でもこの問題が顕在化している．

これらのことから，我が国における医療の質と安全性の確保に繋がる課題として，国民にとって安心できる医療提供体制と実際の医療を担う医療従事者がある程度の余裕を持って働くことのできる体制の構築を推進していくことが求められている．

2-2　医療機器の安全管理をめぐる動向

2-2-1　現代医療における医療機器の存在

古来より人の目で診断し人の手で治療を行ってきた医療は，聴診器やメスといった「医療用の道具」の発明により「新しい目・手」を得ることができ，診断・治療技術のさらなる進歩・発展を遂げることになった．そして，19 世紀末から 20 世紀初頭にかけて，ドイツの物理学者レントゲンによりX線装置，ロシアの物理学者コロトコフにより血圧計，オランダの生理学者アイントーフェンにより心電計が開発され，医療の世界に「医療用の機械」，いわゆる医療機器が使用されるようになりその重要性が認識されるようになった．20 世紀に入ると機器開発の基盤となる機械・電気電子など工学分野の著しい発達に支えられて多種多様な医療機器が開発され，今や治療のみならず臨床検査・放射線検査および治療・生体計測など多岐に渡る分野にて日常的に使用されるようになった．そして，医療機器の導入はこれらの分野のさらなる進歩・発展をもたらし，そこから得られた知見が医療機器の更なる発展を促すことになり，これらの相互の発展が結果として医療そのものの発展と高度化に繋がることとなった[2-8]．

実際の臨床現場における医療の遂行にも医療機器は多く用いられており，その中でも機器を利用した検査による早期での病気発見や生命維持管理装置による救急患者や重症患者の治療のように，医療機器なくしては分からない・救えないといった医療場面も多くなっている．また，近年は医療分野の情報化に伴い，遠隔地における医療行為の手段としてネットワークを介した医療機器の遠隔操作による治療や医療情報のデータベース管理と複数の医療機関における情報共有など，ネットワークやほかの様々な情報技術と医療機器の結び付きが強くなっている．

このように，現代医療における医療技術の発展は医学と工学の領域を融合した医用工学の発達によるものであり，研究開発され高度化する多くの医療機器が安全かつ効果的な医療の遂行に多大な貢献をしている．特に，医療機器の果たす役割は検査，診断から治療，リハビリ，そして情報技術の活用など多様化し，医療機器は現代医療を支える上で必要不可欠な存在となっている．

2-2-2　医療機器に起因する事故の問題

高度な医療機器が医療現場に導入され，国民の医療に貢献していることは既述の通りである．しかしその反面，医療従事者の不適切な使用や整備不良による事故が問題となっている．

財団法人日本医療機能評価機構による医療事故の調査によると，報告義務対象医療機関（大学病院，ナショナルセンター，国立病院機構など）による医療事故事例の報告は，平成 24 年 1 月から 12 月までの間で 2,535 件に上る．また，その内で死亡に至った例は 180 件，死亡に至らなくとも重度ないし軽度の障害が残存した可能性のある例でも 994 件となり，ここまでで事例総数の 46.4%を死亡・障害残存例が占める状況となる．さらに，医療機器に起因する事故事例は全体の 2.9%に当たる 73 件で，その内死亡に至った例が 8 件，重度ないし軽度の障害が残存した可能性のある例が 15 件となっている. 事故の内容は多い順に「不適切使用」,「破損」,「使用中の点検・管理ミス」と挙げられている．また，ヒヤリ・ハットの発生件数は平成 24 年 1 月から 12 月までの間で 690,109 件に上り，その内 20,393 件（全体の 2.9%）が医療機器に起因している [2-9].

これらの医療機器に起因する事故としては，その使用頻度と必要性が増したことによる医療機関での保有台数・利用場面の増加がある．保有する医療機器の増加は，その管理や保守点検に係る労力と医療機器そのものの管理費用増大に繋がる．また，近年は医療機器そのものが非常に高度・複雑化しているため，看護師のように臨床現場にて医療機器を操作する医療従事者が正しい操作法や使用法を理解していない慢性的な知識不足にあることが問題視されている．いずれも臨床現場における医療機器の安全かつ効率的な使用に支障をきたす要因となっている.

この現状に対して政府や関連団体は，医療機器に係る事故防止と安全確保のための関連法の制定や立案･実施に当たっている. 次項に医療機器を取り巻く法規制について述べる.

2-2-3　医療機器を取り巻く法規制

2-2-3-1　薬事法

我が国における医療機器は，薬事法に規定されている．薬事法第 2 条 4 項によると「人若しくは動物の疾病の診断，治療若しくは予防に使用されること，または人若しくは動物の身体の構造若しくは機能に影響を及ぼすことが目的とされている機械器具など」とされており，平成 17 年までは医療で用いられる用具や機械はすべて「医療用具」とされていたが，同年 4 月 1 日の薬事法改正によって「医療機器」と総称されるようになった[2-10]．現在，薬事法によって医療機器として指定されている機器はその使用用途，使用部位，使用方法，構造，材質などから病院用機器・診断用機器・手術用機器・処置用機器・歯科用機器・簡易医療機器に分けられている．加えて，その機器が人の生命・健康におよぼすリスクの大きさに応じてクラス分類がされている．表 2-1 にその分類と不具合発生時のリスクの大きさ，対象医療機器を示す.

さらに，この中でもクラス分類に関わらず保守点検・修理・その他の管理に必要な専門的知識・技術を必要とする医療機器を「特定保守管理医療機器」として指定しており，コンタクトレンズや自動体外式除細動器（AED）などがこれにあたる．現在，薬事法で指定されている医療機器の総数は非常に多く，特定保守管理医療機器に指定されているだけでも約 1000 種にも上っている．

また，薬事法第 63 条 2 項では医療機器製造販売業者に医療機器の有効性・安全性の確保，適正使用・管理のために必要な情報が記載された添付文書を医療機器ごとに提供しなければならないと定めている．添付文書に記載される事項は，「使用方法，取扱上の注意事項（禁忌・禁止事項を含む)」，「保守点検に関する事項」，「省令に定める事項」と様式の統一化が図られている．また，独立行政法人医薬品医療機器総合機構（PMDA）が電子登録によるデータベース化を推進している．

さらに，薬事法第 77 条の 4 では医療機器による危害の防止および副作用などを厚生労働大臣に報告すること義務付けている．製造販売業者などは，医療機器の使用によって保健衛生上の危害が発生し，また拡大の恐れがあることを知ったときは，これを防止するために廃棄，回収，販売の停止，情報の提供などの措置を講じなければならないと定めている．また，医療機器の副作用（不具合）などに起因するものと疑われる疾病，傷害または死亡の発生または感染症の発生その他，医療機器の有効性および安全性に関する事項を知ったときは，その旨を報告しなければならないとしている．

表 2-1　医療機器のクラス分類

クラス分類	リスクの大きさ	対象医療機器
クラスⅠ （一般医療機器）	人体へのリスクが極めて低いと考えられるもの	メス，ピンセット，医療用不織布，手術用照明器，X線フィルムなど
クラスⅡ （管理医療機器）	人体へのリスクが比較的低いと考えられるもの	MRI，X 線撮影装置，超音波診断装置，汎用心電計など
クラスⅢ （高度管理医療機器）	人体へのリスクが比較的高いと考えられるもの	人工透析装置，人工呼吸器，バルーンカテーテルなど
クラスⅣ （高度管理医療機器）	人体への侵襲性が高いため，生命の危険に直結する恐れのあるもの	ペースメーカ，人工心臓弁，ステントなど

2-2-3-2　医療法

医療機関における医療機器の安全確保に係る対策については，医療法で規定されている．医療法第 6 条 10 項では「病院，診療所または助産所の管理者は，厚生労働省令で定めるところにより，医療の安全を確保するための指針の策定，従業者に対する研修の実施その他の当該病院，診療所または助産所における医療の安全を確保するための措置を講じなければならない」としている[2-11]．つまりここでは，病院などの医療機関に対し医療の安全確保のための指針を策定すること，従業員に対する研修を実施することなどの安全確保のための措置を講じることを義務付けている．さらに，この規定を受け，医療法施行規則では病院などの医療機関において医療の安全管理のための体制確保を求め，医療機器については「医療機器の安全管理のための体制確保に係る措置」を講じなければならないと規定している[2-12]．これには，「医療機器の安全対策のための責任者（医療機器安全管理責任者）の配置」，「従業者に対する医療機器の安全使用のための研修の実施」，「医療機器の保守点検に関する計画の策定および保守点検の実施」，「医療機器の安全使用のために必要となる情報の収集その他の医療機器の安全使用を目的とした改善のための方策の実施」の 4 項目が義務付けられている．これにより，医療機器の安全使用と管理体制の整備が法令に明記され，医療機関は安全な医療機器を提供することが責務となった．また，この詳細事項については，平成 19 年 3 月 30 日厚生労働省医政局指導課長・研究開発課長通知「医療機器に係る安全管理のための体制の確保に係る運営上の留意点について」により示されている．この中で，特に医療機器の使用に当たっては医療機器製造販売業者が作成した添付文書などの記載に従って実施するよう指導されており，そのため医療機器安全管理責任者に添付文書などの医療機器の安全使用・保守点検などに関する情報を整理し，その管理を行う事を求めている．

2-2-4　臨床工学技士の誕生

従来，我が国では医療機器を管理する専門の医療職種が存在せず，医師や看護師が本来の業務に加えて医療機器の保守管理などを行っていた．しかし，高度・複雑化する医療機器に対して医師や看護師による対応は困難となり，医用工学の知識を持ち医療機器を専門に取り扱う医療従事者として，昭和 62 年に臨床工学技士が誕生した[2-13]．臨床工学技士は，臨床工学技士法の定める業務指針により「生命維持管理装置（人の呼吸，循環又は代謝の機能の一部を代替し，補助することを目的とする装置）の操作および保守点検を業とする者」とされていることから医療機器全般の保守管理業務を担当するようになった．しかし，当初の臨床工学技士による業務は生命維持管理装置の操作，いわゆる臨床業務に重点を置

かれることやその絶対数が少ないこともあり，医療機器管理業務を行う臨床工学技士の定員化が進展せず，医療機器管理の対策や規模は十分に至らなかった．そこで，厚生労働省は医療機器管理の推進力を強化するために,平成 19 年 4 月に第 5 次改正医療法を施行し，病院や診療所または助産所などの医療機関に「医療機器の安全管理のための制確保に係る措置」として，医療機器安全管理責任者の配置を義務付けた．この医療機器安全管理責任者の資格は,「医療機器の適切な使用方法や保守点検の方法など医療機器に関する十分な経験および知識を有する常勤職員であり，医師，歯科医師，薬剤師，助産師（助産所の場合に限る)，看護師，歯科衛生師（主として歯科偉業を行う診療所に限る)，診療放射線技師，臨床検査技師または，臨床工学技士のいずれかの資格を有していることと．なお，医療機器の適切な保守管理を含めた包括的な管理に係わる実務を行う事ができる者であること.」とされている．この資格に最もふさわしいのは実際に医療機器管理業務を担っている臨床工学技士であり，臨床工学技士による医療機器安全管理体制の強化が強く求められるようになった．このよう中，医療機器安全管理責任者としての臨床工学技士の役割は「医療機器を安全に使用できる環境整備」,「医療機器の安全性と性能維持」,「他の医療従事者との調整・連携（状況の把握)」である．また，病院管理者の指示の下，医療機器管理に関する医療機関内での体制の構築や円滑な運営を図り，医療安全管理部門と連携して，医療機器に関わる職員への教育・研修なども行わなければならない．これは，医療機器の取り扱いには臨床工学技士だけではなく医師・看護師など，他の医療従事者も多数関わっているからであり，医療現場で安全に医療機器を使用するためにはすべての関係者への教育・指導が必要である．そのためには臨床工学技士が医療機器安全管理責任者として，その教育・指導の中心的役割を担うことにより安全な医療が確立できることとなる．

2-2-5　医療機器管理の現状

医療機器の保守管理を適切に実施するためには，保有機器個々の情報を管理する医療機器管理台帳を作成し整備することが重要である．この台帳には，医療機器を一意に識別する管理番号や機器名称，製造年月，使用期限などを記載し，これを基に医療機器の保守点検実施状況の評価など，購入から廃棄までのライフサイクル管理が行われる．この台帳の作成は，医療機関の規模や管理形態により異なるが，基本的な作成方法と管理方法については医療機器管理指針対策委員会が策定した「医療機器の保守点検に関する計画の策定及び保守点検の適切な実施に関する指針」に具体的な明記がされている[1-2]．表 2-2 に医療機器管理指針対策委員会が策定した医療機器管理台帳の作成例を示す．

従来，この医療機器管理台帳は記録用紙を基にして作成していたが，近年の医療機関が保有する医療機器台数とその種類，使用頻度の増加により，それらの情報量も膨大となることから，施設内に散見するすべての医療機器に対する情報管理は非常に困難な状況となっていた．そこで，今日では各医療機関においてデータベースシステムを活用した市販の医療機器管理システムの導入が進められている．これにより，膨大な医療機器情報の一元管理を可能とし，さらに施設内のネットワークや専用の携帯情報端末を利用することで，煩雑な医療機器管理業務の効率化を図ることができる．しかし，市販の医療機器管理システムの導入は各医療機関の規模や管理形態にもよるが，導入費用に数百万円，加えて維持・ライセンス費用，医療機関独自の医療機器管理方法に合わせたカスタマイズに掛かる費用が高額である．また，医療機器管理の診療報酬に応じた業務収入も乏しいことから，積極的な資本投入はされず，経済的に余裕のある医療機関でなければ積極的な導入に至らない現状にある．また，このような中で比較的安価な汎用のデータベースソフトウェアである Microsoft Access（Microsoft, Inc.）や File Maker（File Maker, Inc.）を活用して，独自の医療機器管理システムを開発している医療機関もある[1-5][1-6]．これにより，自施設の特徴を活かした医療機器管理が可能となる反面，これらの開発を初めから行うには膨大な時間を必要とし，本来の業務の片手間となってしまう．さらに，ソフトウェアに掛かるライセンスの問題から他施設での導入とカスタマイズは十分と言えず，その普及には至っていない．また，データベースの機能にも限界があり，スタンドアロンとしての使用が前提となるため，単純な医療機器個体の情報管理程度に留まり，分散している機器に対して情報の参照や共有に滞りが生じている．

このように，有用な医療機器管理援システムの導入や開発が十分に行えず，従来通りの管理しかできないという医療機関が非常に多くあり，医療機器管理の重要性は高まる一方で，その対策や規模は十分とは言えない現状にある．

表 2-2　医療機器管理指針対策委員会が策定した医療機器管理台帳の作成例

管理番号	設置場所	機器区分	機種名	製造番号	製造年月	購入年月	使用期限	破棄年月	備考
A0001	集中治療室	人工呼吸器	○△×○	○△×○	oo/o/o	oo/o/o	oo/o/o	oo/o/o	
A0002	集中治療室	IABP			oo/o/o	oo/o/o	oo/o/o	oo/o/o	
A0003	手術室	麻酔器			oo/o/o	oo/o/o	oo/o/o	oo/o/o	
A0004	手術室	電気メス			oo/o/o	oo/o/o	oo/o/o	oo/o/o	
A0005	3階南病棟	輸液ポンプ			oo/o/o	oo/o/o	oo/o/o	oo/o/o	
A0006	4階北病棟	シリンジポンプ			oo/o/o	oo/o/o	oo/o/o	oo/o/o	
A0007	4階北病棟	除細動器			oo/o/o	oo/o/o	oo/o/o	oo/o/o	
A0008	生理検査室	心電計			oo/o/o	oo/o/o	oo/o/o	oo/o/o	

第 3 章

オープンソース・ソフトウェア

オープンソースとは，コンピュータソフトウェアを提供する一つの方法であり，この方法で提供されたソフトウェアをオープンソース・ソフトウェアと呼ばれる．

本章では，オープンソースの定義からソフトウェアの知的所有権について概説する．さらに，オープンソースによる開発メリットと課題，医療分野でのオープンソース・ソフトウェア活用の現状を述べる．

3-1　オープンソースの定義

オープンソース・ソフトウェア（OSS：Open Source software）とは，ソフトウェアの設計図にあたるソースコードが公開され，誰でも自由にそのソースコードを利用でき，また改変と再配布を認めるソフトウェアをいう．この定義は，オープンソースイニシアティブ（OSI：Open Source Initiative）に明記されており，これに準拠していなければならない．以下にオープンソースの定義を示す[3-1]．

1. Free Redistribution（自由な再配布）
2. Source Code（ソースコードの開示）
3. Derived Works（派生著作物に関するライセンス条件）
4. Integrity of the Author's Source Code（著作者のソースコードの完全性）
5. No Discrimination Against Persons or Groups（個人やグループに対する差別の禁止）
6. No Discrimination Against Fields of Endeavor（利用分野に対する差別の禁止）
7. Distribution of License（ライセンスの継承）
8. License Must Not Be Specific to a Product（特定製品でのみ有効なライセンスの禁止）
9. License Must Not Restrict Other Software（他のソフトウェアを制限するライセンスの禁止）
10. License Must Be Technology-Neutral（技術的な中立を保っている）

この定義に書かれている項目は，OSS を維持することを目的とするだけではなく，OSS と商用ソフトウェアの共存やオープンソースコミュニティ間にある確執を調停するための項目となっている．以下に OSS の定義における各項目の具体的な意味を記述する．

1. 自由な再配布

誰に対しても，以下の条件での完全な自由な再配布を承諾する．

(1) 他のソフトウェアと一緒に販売したり，無償で提供したりすることを制限してはいけない．

(2) ソフトウェアの販売に関して，ライセンスの中で使用料，またはその他の報酬を要求してはいけない．

この条件の目的は，完全に自由な再配布を承諾することでオープンソースコミュニティとしての「オープンソースの普及」という長期的な目標を達成することを目指している．

2. ソースコードの開示

いかなる製品に関してもソースコードを適切な価格で提供する．

(1) OSS を配布する場合には，必ずソースコードを含まなければならない．

(2) 情報家電や携帯電話などソースコードと共に配布できない製品形態の場合には，別の方法でのソースコードの開示が求められる．その場合には，インターネットから無償でダウンロードできることが望ましい．そして，この形態でのソースコードの提供に対しては，ソースコードが公開されていることを十分に告知する必要がある．

加えて，ソースコードの提供形態に関してもガイドラインがあり，そこにはプログラムが変更しやすい形態でなければならず，ソースコードを意図的にわかりづらくすることは許されない．

3. 派生著作物に関するライセンス条件

以下に示す条件でソフトウェアの改変と派生ソフトウェアの再配布を承諾する．

(1) ソフトウェアの自由な改変を承諾しなければならない．

(2) 改変して作られた派生ソフトウェアに対しても元のソフトウェアと同じ条件での配布を承諾しなければならない．

多くのユーザが自由にソースコードを改変し，発展させ続けられることがオープンソースの持つ意味であり，「ソースコードの開示」だけでは，オープンソースとはいえない．従って，ソフトウェアの「改変権」が無条件に認められなければならない．また，改変結果の「再配布」も認められなければならない．

4. 著作者のソースコードの完全性

著作者はオリジナルのソースコードと修正をみわけられるようにするために，配布方法について派生著作物に元のソフトウェアとは異なる名前やバージョン番号を付けることが必要とされている．ソフトウェアが改良され，改変バージョンが数多く配布されることは

よいことであるが，改変バージョンが同じ名前で配布されてしまうとオリジナルとの区別がつかなくなってしまう．また，ユーザは自分が使っているソフトウェアがどのバージョンで誰が責任を持っているか知る権利を持っている．

5. 個人やグループに対する差別の禁止

いかなる個人やグループを差別してはいけないとされ，国によっては外国為替法によりソフトウェアに輸出制限がかけられている場合があるが，ライセンスではそのような制限を盛り込んではいけない．オープンソースが進歩していくためには，多種多様な人々やグループによって試験され，改変されていく必要がある．そのために，オープンソースライセンスが誰かを差別して締め出すことを禁止している．

6. 利用分野に対する差別の禁止

(1) オープンソースが商業的に利用されることを妨げてはいけない．

(2) 特定の学術・商業分野での使用を制限してはいけない．

オープンソースコミュニティは，商業的なユーザの参加も奨励しており，ライセンスでの商業利用の制限を禁止している．

7. ライセンスの継承

ソフトウェアに付随する権利は，そのプログラムが再配布されたすべての人に対し，そのまま適用されなければならず，オリジナルのライセンス形態を守ることが原則である．

8. 特定製品でのみ有効なライセンスの禁止

ライセンスは，特定のソフトウェア配布物との関係を前提としてはいけない．これは，その配布物から取り出されて利用されたり，配布した場合であってもそのソフトウェアが元のソフトウェア配布物に対して与えられたのと同じ権利がなければならない．

9. 他のソフトウェアの制限するライセンスの禁止

ライセンスは，ライセンスされるソフトウェアと共に配布される別のソフトウェアに制限を設けてはいけない．また，いかなる条件を課してはならない．

10. 技術的な中立を保っていること

ライセンスのいかなる条項もライセンスする技術に対して制限を設けてはならない．

この定義でのオープンソースとは，単にソースコードが開示されていたり，ソースコードへのアクセスが許されていることだけを意味しているわけではない．また，この 10 項目に準拠しているソフトウェアライセンスには図 3-1 に示す「OSI 認定マーク」が付与される．

図 3-1　OSI 認定マーク

3-2　ソフトウェアの知的所有権

OSS は，その定義からもソースコードが公開され，その利用や改変，再配布が自由なソフトウェアである．そのため，オープンソースを考える上で商用ソフトウェアの知的所有権を守ることが必要となる．

一般的に，ソフトウェアの開発は多くの時間と労力，知力を掛けた労働の結果として生まれる生産物であるが，プログラムには複製によってまったく同一の品質，機能を持つ類似品を作ることが技術的に容易だという特質がある．そこで，プログラムの複製行為を技術的ではなく，法的に阻止することを目的に著作権法改正（明治 32 年法律第 39 号）によりプログラムに対して著作権を中心とした保護体制が確立された[3-2]．これには，プログラムのことを「電子計算機を機能させて一つの結果を得ることができるようにこれに対する指令を組み合わせたものとして表現したものをいう」と定義し，プログラムを著作物と認定し著作権を確立させた．著作権とは,「著作者がその著作物を直接的かつ排他的に支配しうる権利」であり複製権，翻訳権などの複数の権利を総合したものである．つまり，プログラムも個人の私的利用を除いては，原則として著作者に無断で複製することは法的に禁止となる．このように，プログラムを保護するものとして著作権が整備されてきたが，一方では著作権だけでは守れないという考えもあった．例えば，ソフトウェアの開発で使われる様々なアイディアがある．例として，カーソルを点滅させたり，ウィンドウのタイトルバーにファイル名を表示したりといった「アイディア」は，著作権法での著作物の範囲には入らない．著作権法の保護対象物は，他人が知ることができるように外部に公表された著作物の表現であり，その中に含まれる「アイディア」は保護対象とはならなくなる．そこで，このようなアイディアを特許法で保護するとされた．

このように，プログラムの知的所有権は保護されるようになったが，OSS は前述したとおりソースコードが公開され，その利用や改変，再配布が自由なソフトウェアである．そのため，入手した人の判断で複製やプログラムの改変，機能の追加が可能となる．しかし，オープンソースの定義には特許権についての取り決めはなく，自分のアイディアを組み込んだプログラムを正式にリリースすることには問題が出る可能性がある[3-3]．

3-3　オープンソースによる開発メリットと課題

オープンソースがもたらすメリットは，その定義であるソースコードを公開し，自由に利用・改変・再配布できることにある．このメリットを利用するとシステム構築やカスタムソフトウェアの開発など容易に行える．以下に具体的なメリットについて記述する[3-3]．

① OSS から必要なソフトを作ることができる

一般的にソフトウェアの開発には，設計・プログラミング・デバック・試験の工程がある．たとえ小規模なソフトウェアであっても必ずこの工程をたどり，一から開発するには多大な労力が必要となる．しかし，既存の OSS を基に，それを改変して新しいソフトウェアを作成することができる．

② OSS は機能の追加が可能であり，それを自由に配布することができる

ソースコードが提供されていない商用ソフトウェアは，基本的に機能の追加は困難であり，追加を行うにしてもそのソフトウェアが提供している機能の範囲内で考える必要がある．しかし，OSS ではソースコードが入手できるため機能の追加はある程度の技術があればそれ程難しいものではない．また，OSS の改変に関する法的な障害はまったくないため，機能追加を行ったソフトウェアを自身で利用するだけでなく，配布することもできる．

③ OSS を使用することでコストが削減できる

OSS を使ってシステムを構築する場合には，そのソフトウェアがすでに使用できる状態で提供されている．これらのソフトウェアは商用ソフトウェアに匹敵する機能や性能，安定性を持っているソフトウェアであり，現実にビジネスの場でよく使用されている．つまり，ソフトウェア購入費用やライセンス費用，開発費用まで削減することができる．

以上が OSS による開発メリットとなるが，開発における注意すべきこともある．それは，OSS は，責任の所在がはっきりしないという点である．商用ソフトウェアでは，指定された環境であれば動作することが保障されており，そこで問題が生じてもサポートセンターなどで対応される．OSS を使用する場合には，この免責事項に注意を払う必要があり，常に危機意識を持って，問題が起きないかどうかを確認しながら開発していく必要がある．

オープンソースライセンスでは責任の所在について一般的に以下のように記している．

① 無保証の提供条件

OSS は無保証であり, 著作権者だけでなく配布だけ行っているような関係者も含め, 明示されていない場合 OSS は，一切の保証を付けずに提供されるものである.

② 商品性や特定目的への適合性に関する保障

OSS では，ソフトウェアに関する商品性や特定の目的への適合性については，一切の保証がされない.

③ 品質や性能に関する責任

OSS の品質や性能に関するすべてのリスクは, ユーザが追うものとする. ソフトウェアに欠陥があるとわかった場合には，ユーザ自身による対処や補修または訂正が求められる.

④ 使用結果の責任

ソフトウェアを使用したことやソフトウェアにトラブルがあって正しく動かなかったことに起因するいかなる損害も補償されない

⑤ 機能追加や機能変更における自己責任

OSS では，自由に機能を変更したり追加したりできるが，実際にこの作業を行うには自分自身でソースコードを書き換えなければならない．以上のようにソースコードが公開されている OSS では，ソフトウェアに問題があれば，基本的には自分で対応することになる．これは，開発者にとっては，自由に扱えることを意味するが責任はすべて使用者にあることも同時に意味する.

しかし，有名な OSS では，オープンソースプロジェクトがバクなどのソフトウェアの修正を行ってアップロードしてくれることが多く，一般ユーザが自らソフトウェアを修正するといったことは多くはない.

3-4　医療分野におけるオープンソース・ソフトウェア活用の現状

医療分野への情報技術の導入は，安全かつ効率的な医療提供を可能とし，放射線画像や薬剤管理，医事会計などの各部門システムや，電子カルテなどの診療支援システムなどに活用されている．しかし，このような情報システムを導入するには，膨大な費用が掛かることが問題となっている．電子カルテの導入コストは 1 床あたり 80～120 万円掛かるとされており，日本医師会の試算では全国の病院に導入した場合に掛かる費用は 10 年間で約 18 兆円としている[3-4]．この費用を我が国の医療経済から支出することは事実上不可能であるとされ，これらの普及のためには導入費用の低減が不可欠としている．このような中で，医療分野においても OSS が活用される事例が増えている．日本医師会では，OSS を活用した医事会計システムを中核とする ORCA Project（Online Receipt Computer Advantage）を発足させ，ネットワークを用い全国の医師，医療関係機関が誰でも無料で使用・改良できる公開ソフトウェア（オープンソース）方式で、毎月の診療報酬を請求するための専用コンピュータ（レセコン）の開発を行っている[3-5][3-6]．また，レセコンのプログラム部分だけでなく，医療情報の標準規格化を進めるため，医療情報データベースも公開している．一方で，OSS の開発を支援するサイト「Source Forge」では，2013 年現在で約 700 本近くの“medical” に関連したソフトウェアが登録されており，医療用画像処理ツールや医療従事者によって行われる検診訪問を管理するアプリケーションソフトウェアなどが公開されている[3-7]．今後も医療分野において OSS の導入が積極的に試みられるようになってきており，その普及と標準規格化に期待が寄せられている．

このように，保健医療分野などではそのシステムの導入に OSS が活用されているが，医療機器管理に関する OSS は国内外含めてこれまでに開発された事例はない．この理由として，医療機器管理に関する項目が標準規格化されていないことや情報工学と医療機器管理に精通した開発者の存在が少ないことが考えられる．

第 4 章

医療機器管理ソフトウェア
「Open-MEMS」

本研究で開発を行った Open-MEMS（Medical Equipment Management Software）は，医療機器管理用に作成したアプリケーションサーバを構成する OSS を簡易な方法で導入できるようにしたパッケージソフトウェアである．このソフトウェアを導入することで，クライアント・サーバ型の医療機器管理システムを簡易に構築できることが最大の特徴である．

本章では，Open-MEMS のシステム構成およびソフトウェア構成，パッケージ作成方法について説明し，さらに医療機器管理用に作成したデータベースサーバおよび Web アプリケーションの構成について述べる．

4-1　システム構成

Open-MEMS は，本研究の目的を実現するために医療機器管理用に作成した機器情報を管理するデータベースサーバと Web アプリケーションを提供する Web サーバを簡易な方法で導入できるようパッケージ化したソフトウェアである. これを 1 台の PC にインストールすることで医療機器管理用アプリケーションサーバを実装することができる.

システム構成は，図 4-1 に示すように Open-MEMS がインストールされた PC に，病院内 LAN を介して, 機器情報の入出力を行うタブレット PC などのクライアント端末で構成されるクライアント・サーバ型の医療機器管理システムである．クライアント側とサーバ側で役割を分散することにより，システム全体の処理を高速化でき安定したシステムを実現できる．また，ネットワーク上にサーバを置くことにより，複数のクライアントで情報の入出力が可能となる．これにより，システム使用者はクライアント端末上の Web ブラウザを使用してサーバにアクセスし，提供される医療機器管理アプリケーションから機器情報の閲覧や保守点検記録などを行う.

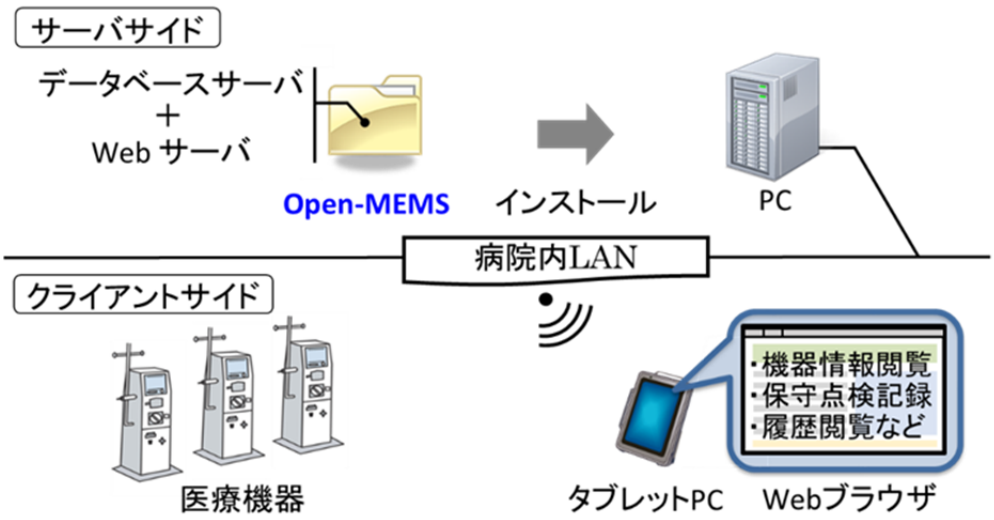


図 4-1　システム構成

4-2　ソフトウェア構成

Open-MEMS のソフトウェア構成を表 4-1 に示す．これは，主にデータベースサーバの PostgreSQL 8.4.13 と Web サーバの Tomcat 6.0.37 から成り，それぞれのスクリプトファイルと設定ファイル, 医療機器管理用に作成したデータベースファイルと Web アプリケーションファイルから構成される．さらに，開発環境の JDK 1.5（Java Development Kit）と API (Application Programming Interface) の JDBC 3 (Java Database Connectivity) が加えられる．Open-MEMS はこれらのソフトウェアを簡易に導入できるように一つのファイルにまとめ，パッケージ化を行っている．具体的なパッケージ方法は 4-4 節で示すが，これにより作成した Open-MEMS は，PC 上のダブルクリックによる簡易な操作でインストールすることができる．

なお, 本ソフトウェアはサーバに用いる PC の OS (Operating System) として， Linux 系 OS の中でもサーバ用途として開発され，その安定性に定評のある CentOS 上で動作することを前提としている．

上記の各ソフトウェアの詳細と特徴，選定理由については次項に記述する．

表 4-1　Open-MEMS のソフトウェア構成

Open-MEMS		
PostgreSQL 8.4.13	データベースサーバ	・スクリプトファイル ・設定ファイル ・医療機器管理用データベースファイル
Tomcat 6.0.37	Web サーバ	・スクリプトファイル ・設定ファイル ・医療機器管理アプリケーションファイル
JDK 1.5	開発環境	・構成ファイル一式
JDBC 3	API	・構成ファイル一式
CentOS	OS	・サーバに使用する OS

4-2-1　OS：CentOS

Open-MEMS の導入で使用した CentOS（Community ENTerprise Operating System）は，サーバ用 OS として現在最も大きなシェアを持つ Red Hat 社の Red Hat Enterprise Linux（RHEL）と完全互換を目指した Linux ディストリビューションである[4-1]．Linux ディストリビューションとは，必要な構成要素を組み合わせて OS 全体を構築したもので，一般的に以下に示す要素から図 4-2 に示す構成から成る．

① カーネル
OS の中核部分であり，コンピュータの作業を割り振ったり，メモリとファイルの管理を行う．

② ドライバモジュール
各ハードウェアを制御するためのプログラムを構成する．

③ API（Application Program Interface）
アプリケーションと OS を取り持つ部分で，アプリケーションから OS のサービスを呼び出すために使用する．

④ ユーザインターフェース
ユーザとのやり取りを制御するプログラムやキーボード・マウスの制御などに使用する．

⑤ ツール
基本的な作業を行うアプリケーションやユーティリティソフトウェアが構成される．

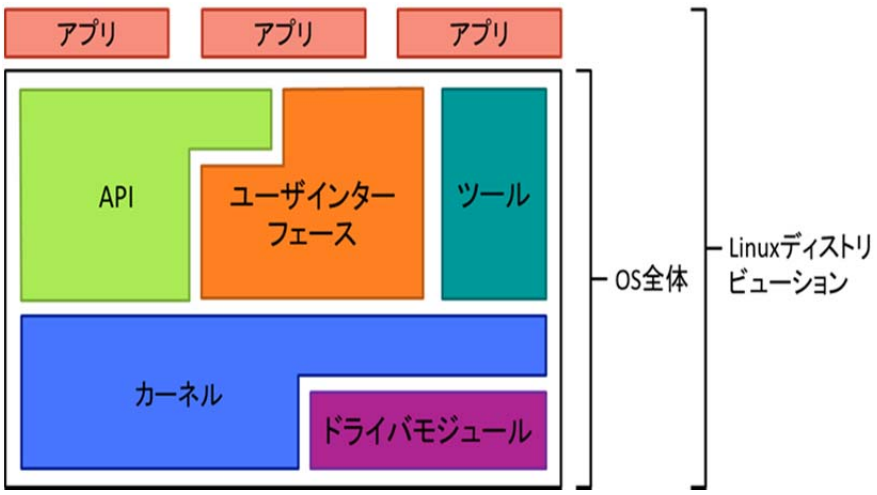


図 4-2　Linux ディストリビューションの構成

上記のようなLinuxディストリビューションの一つであるCentOSは，世界中の数万人規模によるコミュニティによってRed Hat社が無償公開したソースコードを同社の商標・商用パッケージなどを含まない形で再編集，再コンパイルして制作されたオープンソースのOSでもある．また，そのコミュニティによりバグの発見やセキュリティ対策が商用OSを上回る速度で行われている．さらに，サーバ用 OS として開発されていることから設計が非常にシンプルに作られており，連続稼働させても非常に高い安定性を備えている．サーバ使用用途やアーキテクチャによって必要なハードウェアのスペックは多少変わるが，比較的低スペックなハードウェアに対しても十分な速度で動作する[4-2]．表 4-2 に CentOS の導入に最小限必要なスペックを示す．また，各種のサーバソフトウェアやプログラム開発環境が充実し，これらを特別なプロセスを生成せずに導入することができる．これには，本研究で使用する PostgreSQL と Tomcat も含まれている．

以上のことから，本研究では Open-MEMS の導入で使用するサーバ PC の OS として，CentOS を選定し，中でも 2011 年 7 月にリリースされた CentOS 6.x の最終バージョンである CentOS 6.4 を使用した[4-3]．

表 4-2　CentOS 導入に最小限必要なスペック

アーキテクチャ / ハードウェア	32bit（i386）	64bit（x86_64）
CPU	400MHz（Pentium Ⅱクラスまたはそれ以上）	1GHz（Pentium4 クラスまたはそれ以上）
最小メモリ	512MB 以上	1GB 以上
最小ストレージ	1GB 以上	3GB 以上

4-2-2 データベース：PostgreSQL

医療機器情報の一元管理を行うデータベースサーバとして使用した PostgreSQL は，1973 年にカリフォルニア大学バークレイ校のデータベース研究プロジェクトにより開発されたリレーショナルデータベース管理システム（RDBMS：Relational DataBase Management System）の「Ingres」に端を発している．1986 年には同大学の Michael Stonebraker 教授を中心とした研究グループによって「Ingres」の後（post）に誕生したという意味で「Postgres」と名付けられ，以後，多くのコミュニティの開発により「PostgreSQL」に名称変更された．PostgreSQL の特徴を以下に示す[4-4]．

① データベースを操作するための問い合わせ言語である SQL（Structured Query Language）を完全サポートしている．

② ネットワーク対応型であるためクライアントとサーバが異なるプラットフォームでも問題なく動作する．つまり，データベースを利用するクライアントとデータベースエンジンを提供するサーバをそれぞれ独立させることができるため，クライアント側のライブラリが軽量になり，データベースエンジンのプログラムが変更されてもクライアント側は影響を受けにくくなる．

③ データベースの文字コードとクライアント側の文字コードが異なる場合は，サーバ側が自動的に文字コードを変換する．この機能はデータベースとデータベースアプリケーションの文字コードが異なる場合において有効であり，コード変換の際にはメモリ空間を不必要に圧迫しない．

④ Java や C 言語，SQL 言語，PL/pgSQL 等といったプログラミング言語を用いてサーバ側の処理関数を動的に定義できる．

⑤ データ型を動的に定義できる．通常，他のデータベース製品では数値型や文字列型等の決まったデータ型しか使用できない．

⑥ オブジェクト指向要素を含む RDMS（ORDMS）であるため，トランザクション（関連する複数の処理を 1 つの処理転移としてまとめたもの）や膨大なデータの高速処理，セキュリティといった点を維持しつつ，データモデルの柔軟性や拡張性を取り入れる

ことができる．ORDMS の機能にはオブジェクト識別子，配列，拡張可能な型システム，テーブル継承等が該当する．

⑦ ポイントタイムリカバリによってトランザクションに関する履歴の保持が可能となりこの履歴とバックアップファイルを用いて, 障害発生直前までのデータ復元ができるようになった．

⑧ データベースを停止させることなく格納データのバックアップを取ることができるため，24 時間 365 日の連続稼働が可能となる．

以上の特徴の中でも特に⑦, ⑧はデータベースの保存性と保守性を向上させることから，Open-MEMS のデータベースサーバとして PostgreSQL を選定し，PostgreSQL 8.x の最終バージョンである PostgreSQL 8.4.13 を使用した[4-5]．

4-2-3　Web サーバ：Tomcat

医療機器管理用 Web アプリケーションの提供を行う Tomcat は，Apache Software Foundation が運営している Jakarta プロジェクトが開発した Servlet/JSP（Java Server Pages）コンテナを実装した Web サーバである． Web サーバは，Web ブラウザを通してユーザからのリクエストに応えて Web ページを配信するプログラムのことである．ただし，この Web サーバは静的な HTML ページだけをユーザに配信するのではなく，ユーザからのリクエストに応じてプログラムを実行し，その実行結果をブラウザに返す動的な Web ページも配信する．その仕組みが Servlet/JSP コンテナにある．Servlet はサーバ上で動作する Java プログラムであり，JSP は HTML 内に Java プログラムを埋め込みクライアントに返す技術である．すなわち，サーバ上で Web ページを動的に生成しクライアントに結果を返すこととなる．これは，クライアントのプラットフォームに依存せず動作可能であることや予めサーバ上でプログラムをコンパイルさせて動作させるため処理速度が速いなどの利点がある[4-6]．

これらの点から Open-MEMS の Web サーバとして Tomcat を選定し, 中でも Tomcat 6.x の最終バージョンである Tomcat 6.0.37 を使用した[4-7]．

4-2-4　Java 開発環境

前項で既述したように, Tomcat は Servlet/JSP を実行するためのサーブレットコンテナを実装した Web サーバである．この Servlet/JSP を動作させるためには Java 開発環境が必要不可欠であり, その Java 開発環境として JDK（Java Development Kit）と JDBC（Java Database Connectivity）を使用した．JDK とは Java 言語を利用してプログラム開発用のコマンドラインツール一式であり，Oracle 社（旧 Sun Microsystem 社）が開発，配布している．コンパイラやデバッガ，クラスライブラリ，Java プログラム実行環境（Java 仮想マシン）等が含まれる．Open-MEMS では，JDK 1.5 を Servlet/JSP を実行するための開発環境として導入した[4-8].

4-2-5　API

API（Application Program Interface）とは，あるソフトェアの機能や管理するデータなどを、外部の他のプログラムから呼び出して利用するための手順やデータ形式などを定めた規約のことである，本システムでは，Java プログラムからデータベースへのアクセスと制御に JDBC（Java Database Connectivity）使用した．実際に稼働させるためには個々のデータベースに対応したドライバ（JDBC ドライバ）が必要となり，Open-MEMS の開発においては，PostgreSQL 8.4.13 に対応する JDBC 3 を導入した[4-8].

4-3　開発に使用したプログラミング言語

4-3-3　SQL

SQL（Structured Query Language：構造化問い合わせ言語）は，RDBMS（Relational DataBase Management System）においてデータの操作や定義を行うためのデータベース言語である．SQL の歴史は，当時 IBM 社の研究所に勤務していた Edgar F. Codd が 1970 年に発表した「A Relational Model of Data for Large Shared Data Banks（大規模共有データバンク用のデータリレーションモデル）」という論文が始まりとされている．これを基に RDBMS のプロトタイプとして System R が誕生し，そのデータベース言語には SEQUEL（Structured English Query Language）が実装された．これが後の SQL となり，SQL を実装した RDBMS は高い操作性が評価され，その使用に準拠した製品が様々な企業から発表された．その後も ANSI（American National Standards Institute：米国国家規格協会）や ISO（International Organization for Standardization：国際標準化機構），JIS（Japanese Industrial Standards：日本工業規格）等の規格団体により SQL の規格化がなされ，現在に至っている．

本来，データベースを管理しているのは RDBMS であるが，RDBMS 自体はデータベースユーザからの指示に従って格納データを管理する．この際，ユーザはデータベース言語である SQL を使用して RDBMS を制御し，データベースを管理することとなる．RDBMS における制御機能としては以下の 3 点が挙げられる．

1．データベース定義
　データを格納すべきテーブルの定義，複数のテーブルを関連付けるための制約，データベースに必要な機密保護の宣言．

2．データベース操作
　テーブルに対するデータ登録・修正・削除，複数のテーブルの結合，ビュー表作成等の集合操作，テーブル内のデータ検索．

3．トランザクション
　障害の回復処理や同時実行のための最小単位として保証される一連の処理操作．

また，RDBMS の操作方法は命令と演算子，関数からなる式で構成され，データベースに対するデータ取得・登録を目的としている．中でも SQL の役割は他のプログラミング言語からのデータ処理であり，テーブルに対する主な命令として次に示すものがある．

ｉ．SELECT（抽出）：テーブル表示や条件付きテーブル表示（検索）ができる．
　例）　SELECT “列名” FROM “テーブル名” ［WHERE “条件”］;

ii．INSERT（挿入）：テーブルに新しい行を挿入できる．

例） INSERT INTO “テーブル名（列名，列名，・・・）” VALUES “（データ，データ，・・・）”；

iii．UPDATE（更新）：テーブル内のデータを変更（更新）する．

例） UPDATE “テーブル名” SET “列名”＝“データ”，“列名”＝“データ”，・・・；

iv．DELETE（削除）：テーブルから行を削除する．

例） DELETE FROM “テーブル名” ［WHERE “条件”］；

ｉ～ivのような，データ操作用のコマンドはSQLの中でも特に重要であり，データ操作言語（DML：Data Manipulation Language）と呼ばれている．これと同類のものにテーブル操作用のコマンドであるデータ定義言語（DDL：Data Definition Language）と呼ばれるものも存在し，こちらには以下の命令が該当する．

v．CREATE TABLE：データを格納するテーブルを作成する．

例） CREATE TABLE “テーブル名”（“列名”，“データ型”，・・・）；

vi．ALTER TABLE：既存テーブルの定義を変更する（列の追加・削除等）．

例） ALTER TABLE “テーブル名” ADD “列名”“データ型”；

ALTER TABLE “テーブル名” DROP “列名”；

vii．DROP TABLE：既存テーブルを削除する．

例） DROP TABLE “テーブル名”；

この他にも，SQL はユーザ権限の管理等に関する様々な操作を実行できるが，Visual Basic や C 言語等で使用される条件分岐（if），ループ（for）等の構文が備わっておらず，「～ならば～する」といった命令を順番に記述し，全体で 1 つの目的を達成する（手続き言語）ことができない．しかし，データ演算のみを行うという点から 1 つの命令で完結し目的を達成できる（非手続き言語）ことが SQL の特徴でもある．また，SQL の使用に関してはいくつか注意点があり，命令は「半角」で記述する，命令と同じテーブル名・列名は付けられない，テーブル名・列名に全角文字を使用する際は「“”」で囲む等のルールを守る必要がある．これらのルールを守り，命令をデータベースに伝え，実行してもらうことで SQL による一連の処理（SQL の発行，SQL の問い合わせ，SQL の実行）が実行される．

4-3-2　HTML

HTML（Hyper Text Markup Language）は，Web ページを表示するためのマークアップ言語であり，文書の論理構造（見出し，段落等）や表示の仕方（フォントサイズ等）をテキストエディタ（Windows：メモ帳，CentOS：GNOME 等）に記述する．マークアップとは「目印をつける」という意味であり，HTML では『HTML タグ』と呼ばれるタグを用いて目印をつけることでテキストファイルに既述したソースコード（人間がプログラミング言語を用いて既述したコンピュータプログラム）の構造や構成をコンピュータに理解させる．この際，HTML タグを付けたテキストファイルを「HTML ファイル」とし，Internet Explorer（Microsoft, Inc.）やといった Web ブラウザを通して確認することで綺麗に整形された文書となる．大半の Web ブラウザはファイル中に記述された HTML タグを標準で解釈し，ファイル中で指定されている内容を Web ページとして画面上に再構成する．さらに，HTML ファイル中に挿入した画像や音声，ハイパーリンクを Web ページ上で表示することもできることから，HTML は予め決められた文書・画像（静的 Web ページ）を表示するための言語とも捉えられる．

HTML を用いて Web ページ上で表示させるソースコードを記述するためには，既述したようなテキストエディタと呼ばれるアプリケーションが必要となる．このテキストエディタに HTML を記述した上でファイルの拡張子を「.html」もしくは「.htm」として保存し，Web ブラウザを通して対象ファイル名を指定することで Web ページを表示させることができる．以前までは，HTML に関するある程度の知識が無ければ Web ページの作成は困難であったが，最近では高機能な HTML エディタを使用することで簡単に作成できるようになった．このエディタを使用する場合は HTML の知識を然程必要としないが，丁寧な Web ページの作成と細部までの修正を心がける際は HTML の基礎を学び，利用することが望ましい．

また，序盤で述べたように HTML の規則は基本的にタグを用いることである．タグには改行を示す
等の「単独タグ」と，<タグ>文字列</タグ>という書式で記述し挟まれた文字列等を修飾する「対タグ」の 2 種類があり，大文字もしくは小文字で記述する．また，タグにはオプションを持つものがあり，オプションは<タグオプション>を用いて<タグオプション>文字列</タグオプション>といった，通常のタグと同様の方法で記述する．加えて，「<」から「>」等の記号は特別な意味を持ち，表示する場合は特殊な形式で表示する必要がある．以下に簡単な HTML ソースコードを示す．

<HTML>（ルート要素）
<HEAD>（文書ヘッダ情報の明示）
<META>（文書メタ情報の明示）
<TITLE>（文書タイトルの明示）　</TITLE>
</HEAD>
<BODY>（文書内容の明示）
<DIV>（ブロックの明示）
<H1>（見出しの明示）</H1>
<P>（段落の明示）</P>
</BODY>
</HTML>

4-3-3　Java

Java は，1995 年に Sun Microsystems 社（現 Oracle 社）が開発したプログラミング言語であり，C 言語と類似した表記法を採用しているが，既存言語の欠点を踏まえて設計されたものである．また，最初からオブジェクト指向性を備えていることに加えて，強力なセキュリティ機構や豊富なネットワーク関連機能が標準で用意されており，ネットワーク環境で利用されることを強く意識した仕様になっている．Java で開発されたソフトウェアは特定の OS やマイクロプロッセサに依存することなく，基本的にはプラットフォームを選ばず動作するが，標準ではどのプラットフォームでも実現できる最大公約数的な機能しか利用できないため，プラットフォーム固有の機能を利用する用途に不向きである．

Java で記述されたソースコードは，コンパイル（ソースコードをコンピュータ上で実行可能な形式にすること）時にバイトコードと呼ばれる中間コードにいったん変換される．ソフトウェアはバイトコードの状態で配布され，実行時には JVM（Java Virtual Machine：Java 仮想マシン）によって各プラットフォームに対応した形式に変換され，実行される．そのため，開発時にはプラットフォームの違いを意識する必要はないが，バイトコードからネイティブコードへ変換する際にある程度の時間を必要とするため，動作が遅くなる．そこで，現在では Just In Time コンパイラという技術を用いて実行速度の高速化を図っている．

また，Servlet と呼ばれるサーバ上で動作する Java プログラムは，クライアントからの要求に応じてプログラムを処理し，動的な Web ページを提供できる．しかしながら，静的なテキストや画像を出力する際にプログラムが冗長になってしまう欠点があったため，静的な HTML ページに Java コードを記述する JSP という Servlet 拡張技術が開発された．

4-3-4　Java Script

Java Script は，1994 年に Netscape Communications 社の Brendan Eich 氏によって開発された「Live Script」と呼ばれていたスクリプト言語を起源とし，1995 年に Sun Microsystems 社と Netscape 社が共同開発したスクリプト言語である．この年は，3-3-3 で既述したように Java が開発され，当時大きな注目を浴びていたため，それに便乗する形で Live Script から Java Script という名称になったとされている．後にこの類似したネーミングの影響で Java Script が Java の別バージョンであるかのような印象を与え，開発やユーザに大きな混乱を招いたが，両言語は全くの別物である．Java がコンパイル時にコード変換されるプログラミング言語であるのに対し，Java Script はコンパイルせず簡易的に実行できる簡易プログラミング言語を指し，これをスクリプト言語という．

スクリプト言語は Java や C 言語等のプログラミング言語同様，英単語や記号・数字の組み合わせによってソースコードを記述するが，コンパイル作業が省略されるため手間をかけることなく実行することができる．小規模なプログラムを素早く作成することを目的に開発されたため，一般的なプログラミング言語と比較して機能面で多少劣るが，習得が容易で記述法も簡単であるという特徴を有している．また，主要な Web ブラウザの殆どにそのエンジンが搭載されているため，HTML 内にソースコードを埋め込むことで Web ページ上に時刻表示やアラート表示等の様々な機能を付加できることに加え，データベースとの連携も簡単に実行できるため，利便性と拡張性に優れている言語でもある．

このように十分な言語仕様を備えている Java Script であるが，然程プログラミングの知識がなく，開発環境が整っていなくとも取り組むことができる手軽さとリリース時の不十分なセキュリティ対策等の点から，正当な評価を得られてこなかった．しかしながら，近年では Google による Java Script を利用した積極的なサービス展開や Ajax（Asynchronous Java Script + XML）と呼ばれる技術が誕生し，再び注目を浴びるようになってきている．

4-4　パッケージファイルの作成

通常，何らかのソフトウェアをインストールする場合，システム上でのコンパイル作業が不可欠である．しかし，これにはある程度の情報処理に関する知識が必要であり，ソフトウェア使用者はこのような作業は望んではいない．パッケージ化を行うことで，面倒なコンパイル作業を省くことができ，誰でも簡単にそのソフトウェアのインストールが行える．そこで Open-MEMS も，4-2 節で既述した各種ソフトウェアを簡易な方法で導入できるようパッケージ化を行った．具体的なファイル構成を図 4-3 に示す．これには，主にデータベースサーバの PostgreSQL 8.4.13，Web サーバの Tomcat 6.0.37 に，それぞれの起動スクリプトファイル（init.d）と設定ファイル（profile）に加えて，医療機器管理用データベースファイル（me_db）に医療機器管理アプリケーションファイル（me_app）と Java 開発環境の JDK 1.5，Java と PostgreSQL の API である JDBC 3 である．これらは実際に CentOS 上にインストールされ，各ディレクトリ内に格納される．例えば，PostgreSQL 8.4.13 と Tomcat 6.0.37 の起動スクリプトファイル（init.d）は，CentOS 上の「/etc/rc.d/init.d」に格納され，それらの設定ファイル（profile）は「/etc/rc.d」に格納される．また，PostgreSQL および Tomcat に関するデータファイルは「/usr/local/」に格納され，中でも医療機器間用データベースファイル（me_db）は，「/usr/local/PostgreSQL 8.4.13/data/base」内に，医療機器管理アプリケーションファイル（me_app）は「/usr/local/Tomcat 6.0.37/webapps」内に保存されることとなる．

パッケージファイルの作成に当たって，本研究では Open-MEMS の導入条件として，サーバに用いる PC の OS を CentOS としたため，Open-MEMS のパッケージ化にあたり Linux ディストリビューション上で最も有名かつ高機能なパッケージ管理システムである RPM Package Manager を採用した．RPM Package Manager は，Red Hat 社が開発したパッケージ管理システムであり，多くの Linux ディストリビューションで採用されている．この RPM Package Manager には，プログラムのバイナリファイルや設定ファイル，ドキュメントなどがまとめられており，「rpm」というコマンドによりソフトウェアのインストールやアップデートが行える．また，一般的な GUI（Graphical User Interface）上でもダブルクリックによる簡易な操作でソフトウェアのインストールが行える．さらに，インストールを行うソフトウェアに関する情報を同一の手段で取得することができるため，不完全なインストールや誤ったインストールを防ぐことできる他，パッケージ内のファイルを簡単に追加・削除ができアップデートなどのソフトウェアの管理作業を容易にすることができる．なお，RPM Package Manager によりパッケージ化したソフトウェアは「.rpm」の拡張子となり実行形式のインストールファイルとなる．

以上より作成した「Open-MEMS.rpm」は実行形式のインストールファイルとなり，PC上のダブルクリックによる簡易な操作でインストールを可能とした．

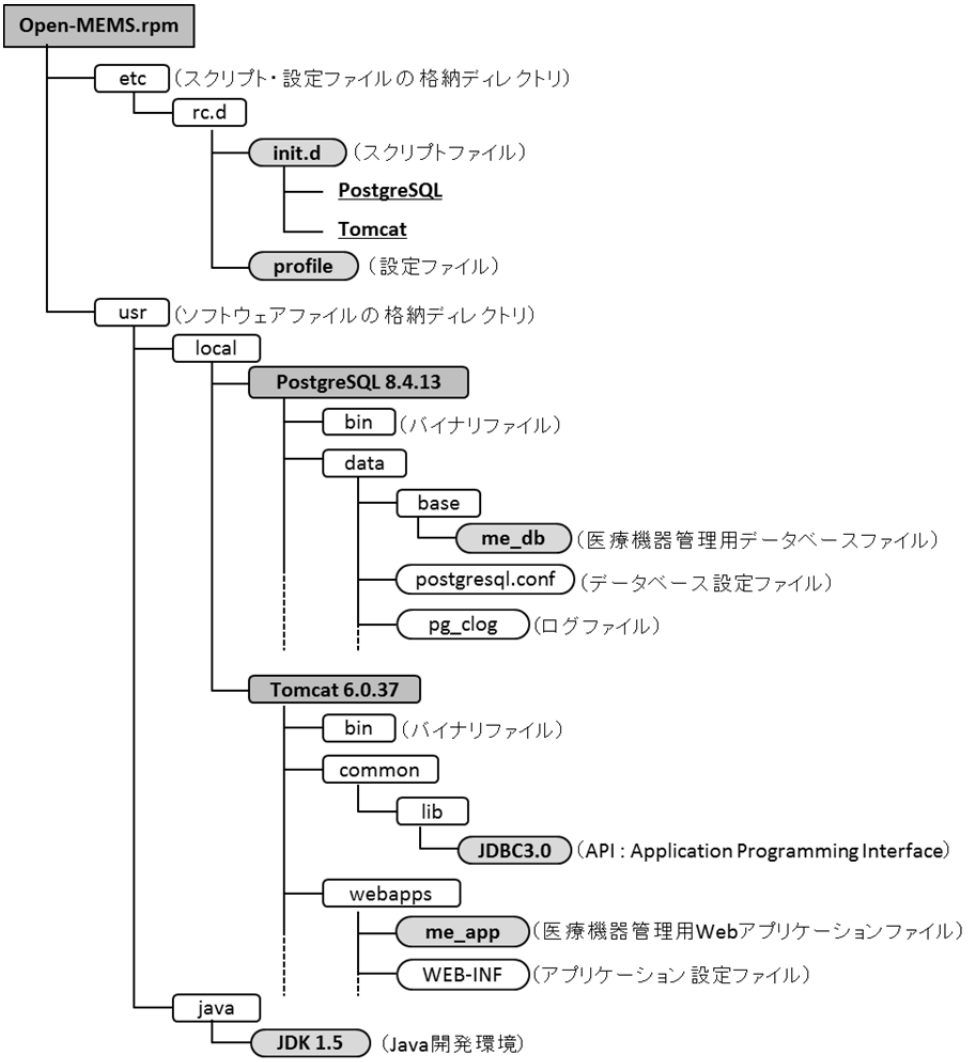


図 4-3　Open-MEMS のパッケージ構成

4-5　医療機器管理用データベース

4-5-1　データベースの設計

データベースを設計するためには，現場での実稼働を考慮した上でのデータモデリングと呼ばれる作業が必要となる[3-10]．データモデリングとは，システム構築の対象を調査・分析し, それらの集まりを抽象化した概念データモデルを作成することである. データベースとして表現すべき対象物をエンティティ（Entity：実体）と呼び，エンティティ間の相互関係をリレーションシップ（Relationship：関連)と呼ぶ．この関係を図示し，分析したものを E-R（Entity-Relationship）モデルという．また，データモデリングにおいて優れたデータベースを構築するためには「正規化」と呼ばれる概念も念頭におく必要がある．正規化とは，格納データをデータベース上で効率的に利用できるようにモデル化することであり, データ量を最小限に抑え, データ重複の防止や容易なデータ管理等を実現できる．RDBMS では，複雑なテーブルを特定の規則に従って単純なテーブルに分割することが正規化となるため，非常にユーザビリティの高いデータベースを設計することができる．また，正規化の 1 つとして，データ重複の防止によってテーブル間の整合性を保持することができる主キー（PRIMARY KEY）制約と外部キー（FOREIGN KEY）制約がある．これは複数存在するレコードから，その条件に対応するレコードを読み書きする際に用いられる．

以上の点を踏まえデータベースを作成する必要がある．次項に今回作成した医療機器管理用データベースの構成を述べる．

4-5-2　医療機器管理用データベースの構成

図 4-4 に今回設計した医療機器管理用データベースの E-R モデルを示す．これは，医療機器管理指針策定委員会が策定した「医療機器の保守点検に関する計画の策定及び保守点検の適切な実施に関する指針」を基にデータモデリングを行い，その結果，6 つの情報格納テーブルを構成した．具体的には，医療機器情報テーブル（me_ifno_tbl），医療機器台帳テーブル（me_registar_tbl），保守点検項目テーブル（check_item_tbl），保守点検情報テーブル（check_info_tbl），修理情報テーブル（repair_info_tbl），保守点検計画テーブル（check_plan_tbl）から成る．次項では各テーブルで管理される項目について述べる．

図 4-4　データベースの E-R モデル（一部抜粋）

4-5-2-1　医療機器情報テーブル

医療機器情報テーブル（me_info_tbl）は，医療機器のモデル（型式）毎の基本情報を管理する．具体的には，医療機器をモデル（型式）毎に割り振る医療機器 ID（me_info_id），モデル（型式）名（model_name），医療機器名称（me_name），メーカ名（maker_name），ディーラ名（dealer_name），JAN コード（jan_code），JMDN コード（jmdn_code），特定保守管理医療機器（specific_me），クラス分類（me_class），添付文書（me_document），操作マニュアル（operation_manual），保守点検マニュアル（maintenance_manual），外観（me_view），備考（me_note）である．なお，医療機器 ID には情報入力時の重複防止と医療機器台帳テーブル，保守点検項目テーブル間の整合性を保持するための PRIMARY KEY 制約，必ず文字入力を必須とする NOT NULL 制約をそれぞれ設けた．表 4-3 に機器情報テーブルの定義を示す．

表 4-3　医療機器情報テーブル（me_info_tbl）の定義

項目名（フィールド名）	データ型	属性
医療機器 ID（me_info_id）	SERIAL	PRIMARY KEY NOT NULL
モデル（型式）名（model_name）	TEXT	NOT NULL
医療機器名称（me_name）	TEXT	
メーカ名（maker_name）	TEXT	
ディーラ名（dealer_name）	TEXT	
JAN コード（jan_code）	TEXT	
JMDN コード（jmdn_code）	TEXT	
特定保守管理医療機器（specific_me）	TEXT	
クラス分類（me_class）	TEXT	
添付文書（me_document）	TEXT	
操作説明書（operation_manual）	TEXT	
保守点検説明書（maintenance_manual）	TEXT	
外観（me_view）	TEXT	
備考（me_note）	TEXT	

4-5-2-2　医療機器台帳テーブル

医療機器台帳テーブル（me_register_tbl）は，医療機器の個々の詳細情報を管理する．具体的には，医療機器台帳を管理する機器台帳 ID（me_register_tbl），医療機器情報テーブルとの連携を保つための医療機器 ID（me_ino_id），施設内で医療機器を一意に識別するために割り振る管理番号（me_no），登録年月日（reg_time），シリアル（製造）番号（serial_no），GS1-128 バーコード（gs1_128barcode），購入年月日（buy_date），廃棄年月日（scrap_date），管理状態（management_state），管理部署（management_dept），備考（register_note）である．また，機器台帳 ID には情報入力時の重複防止と保守点検情報テーブル，修理情報テーブル，保守点検計画テーブルとの整合性を保持するための PRIMARY KEY 制約，NOT NULL 制約をそれぞれ設けた．さらに，医療機器 ID には FOREIGN KEY 制約と NOT NULL 制約を設けた. 表 4-4 に機器台帳テーブルの定義を示す.

表 4-4　機器台帳テーブル（me_register_tbl）の定義

項目名（フィールド名）	データ型	属性
機器台帳 ID（me_register_id）	SERIAL	PRIMARY KEY, NOT NULL
医療機器 ID（me_info_id）	TEXT	FOREIGN KEY, NOT NULL
管理番号（me_no）	TEXT	NOT NULL
登録日時（reg_time）	TIME STAMP	
シリアル（製造）番号（serial_no）	TEXT	
GS1-128 バーコード（gs1_128barcode）	TEXT	
購入年月日（buy_date）	DATE	
廃棄年月日（scrap_date）	DATE	
管理状態（management_state）	TEXT	
管理部署（management_dept）	TEXT	
備考（register_note）	TEXT	

4-5-2-3　保守点検項目テーブル

保守点検項目テーブル（check_item_tbl）は，多種多様にある医療機器に対して様々な保守点検項目があるため，医療機器のモデル（型式）毎の保守点検項目を管理する．また，医療機器の保守点検には日常点検（始業点検・使用中点検・終業点検）および定期点検が実施されるため，それぞれを分類して管理する．さらに，その内で保守点検大項目 10 項目，保守点検小項目 30 項目の範囲で登録できるようにした．管理する項目の詳細は，保守点検項目 ID（check_id），医療機器 ID（me_info_id），保守点検種別（check_class），保守点検大項目（10 項目：check_major_item_1~10），保守点検小項目（30 項目：check_minor_head_1~30），備考（check_item_note）である．なお，医療機器 ID には，FOREIGN KEY 制約と NOT NULL 制約を設けている．表 4-5 に保守点検テーブルの定義を示す．

表 4-5　保守点検項目テーブル（check_item_tbl）の定義

項目名（フィールド名）	データ型	属性
保守点検項目 ID（check_id）	SERIAL	NOT NULL
医療機器 ID（me_info_id）	TEXT	FOREIGN KEY, NOT NULL
保守点検種別（check_class）	TEXT	
保守点検大項目（10 項目：check_major_item_1~10）	TEXT	
保守点検小項目（30 項目：check_minor_head_1~30）	TEXT	
備考（check_item_note）	TEXT	

4-5-2-4　保守点検情報テーブル

保守点検情報テーブル（check_info_tbl）は，保守点検項目情報テーブルと間接的に連携して医療機器個々の保守点検結果を管理する．管理する項目の詳細は，保守点検 ID（check_info_id），機器台帳 ID（me_register_id），保守点検実施日時（check_time），保守点検担当者（check_person），保守点検種別（check_class），保守点検大項目（10 項目：check_major_item_1~10），保守点検小項目（30 項目：cehck_minor_head_1~30），保守点検実施結果（check_text_result），保守点合否検果（check_result），保守点検総合結果（check_total_result），備考（check_note）である．なお，機器台帳 ID には医療機器台帳テーブルとの連携を保つように FOREIGN KEY 制約，NOT NULL 制約を設けた．表 4-6 に保守点検情報テーブルの定義を示す．

表 4-6　保守点検情報テーブル（check_info_tbl）の定義

項目名（フィールド名）	データ型	属性
保守点検 ID（check_info_id）	SERIAL	NOT NULL
機器台帳 ID（me_register_id）	TEXT	FOREIGN KEY, NOT NULL
保守点検実施日時（check_time）	TIME STAMP	
保守点検担当者（check_person）	TEXT	
保守点検種別（check_class）	TEXT	
保守点検大項目（10 項目：check_major_item_1~10）	TEXT	
保守点検小項目（30 項目：cehck_minor_head_1~30）	TEXT	
保守点検実施結果（check_text_result）	TEXT	
保守点合否検果（check_result）	TEXT	
保守点検総合結果（check_total_result）	TEXT	
備考（check_note）	TEXT	

4-5-2-5　修理情報テーブル

修理情報テーブル（repair_info_tbl）は，医療機器個々の修理内容を管理する．管理する項目は，修理情報 ID（repair_info_id），機器台帳 ID（me_register_id），故障発生日時（trouble_time），故障原因（trouble_factor），修理実施日時（repair_time），修理担当者（repair_person），修理内容（repair_text），修理費用（repair_cost），備考（repair_note）である．なお，機器台帳 ID には医療機器台帳テーブルとの連携保つように FOREIGN KEY 制約，NOT NULL 制約を設けた．表 4-7 に修理情報テーブルの定義を示す．

表 4-7　修理情報テーブル（repair_info_tbl）の定義

項目名（フィールド名）	データ型	属性
修理情報 ID（repair_info_id）	SERIAL	PRIMARY KEY, NOT NULL
機器台帳 ID（me_register_id）	TEXT	FOREIGN KEY, NOT NULL
故障発生日時（trouble_time）	TIME STAMP	
故障原因（trouble_factor）	TEXT	
修理実施日時（repair_time）	TEXT	
修理担当者（repair_person）	TEXT	
修理内容（repair_text）	TEXT	
修理費用（repair_cost）	TEXT	
備考（repair_note）	TEXT	

4-5-2-6　保守点検計画テーブル

保守点検計画テーブル（check_plan_tbl）は，定期点検実施時における保守点検周期を管理する. 具体的な項目は保守点検計画 ID（check_plan_id）, 機器台帳 ID（me_register_id）, 保守点検計画（check_plan），保守点検期間（check_plan_span），次回保守点検日（check_plan_date）, 保守点検予定者（check_plan_person）, 保守契約（check_contract）, 備考（check_plan_note）である．なお，医療機器台帳 ID には医療機器台帳テーブルとの連携保つように FOREIGN KEY 制約と NOT NULL 制約を設けた．表 4-8 に保守点検情報テーブルの定義を示す.

表 4-8　保守点検計画テーブル（check_plan_tbl）の定義

項目名（フィールド名）	データ型	属性
保守点検計画 ID（check_plan_id）	SERIAL	UNIQU NOT NULL
機器台帳 ID（me_register_id）	TEXT	FOREIGN KEY NOT NULL
保守点検計画（check_plan）	TIME STAMP	
保守点検期間（check_plan_span）	INT	
次回保守点検日（check_plan_date）	TEXT	
保守点検予定者（check_plan_person）	TEXT	
保守契約（check_contract）	TEXT	
備考（check_plan_note）	TEXT	

4-6　医療機器管理用 Web アプリケーション

医療機器管理用 Web アプリケーションは, Internet Explorer（Microsoft, Inc.）や Google Chrome（Google, Inc.）などの Web ブラウザを使用して，医療機器情報の閲覧や保守点検記録を行うアプリケーションであり，Open-MEMS のインストールされたサーバにアクセスすることで起動する．図 4-5 に Internet Explorer より出力した本アプリケーションのトップ画面，図 4-6 アプリケーションのフローチャートを示す．トップ画面は，管理者専用の情報管理メニューと一般者用の保守管理メニューから構成され，情報管理メニューでは主に医療機器情報の登録/編集，保守点検項目の登録/編集，保守点検計画の作成/編集などの情報入力を行う．保守管理メニューでは保有機器の一覧や詳細情報の閲覧，保守点検実施内容の記録・履歴管理，修理実施内容の記録・履歴管理を行う．なお，これらの機能は研究協力病院の医療機器管理方法を基に作成した．

以降では，情報管理メニューと保守管理メニューの各機能について述べる．

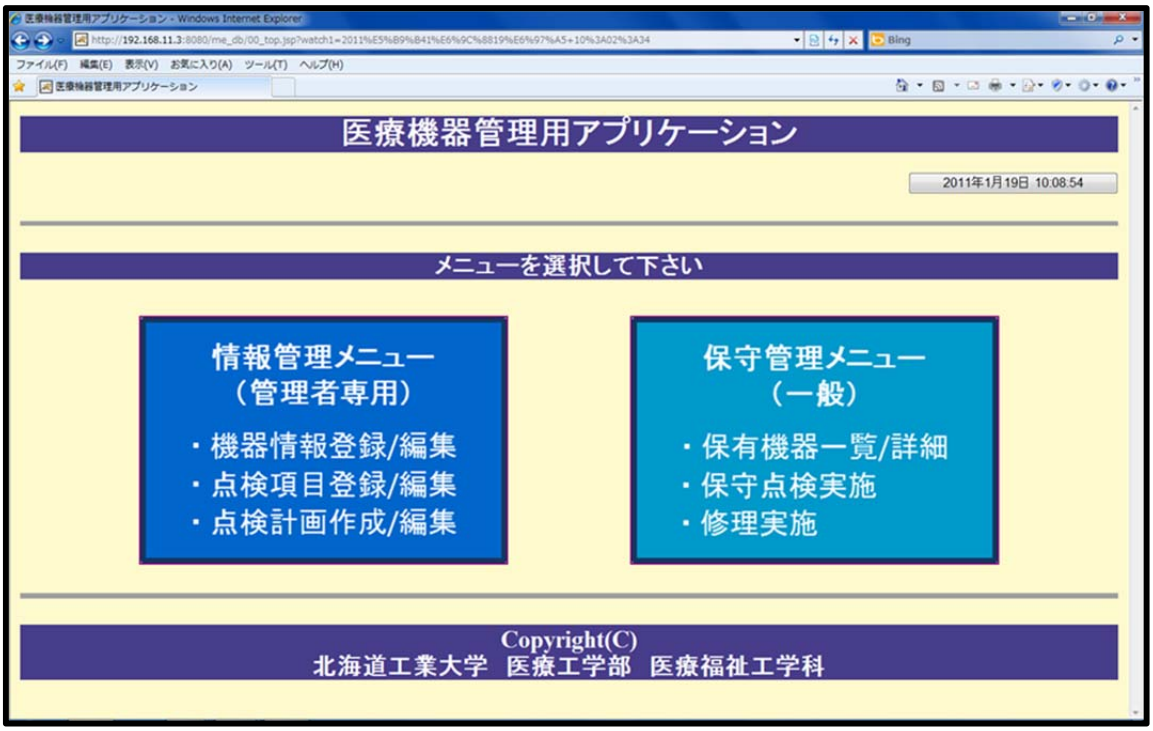


図 4-5　医療機器管理アプリケーションの TOP 画面

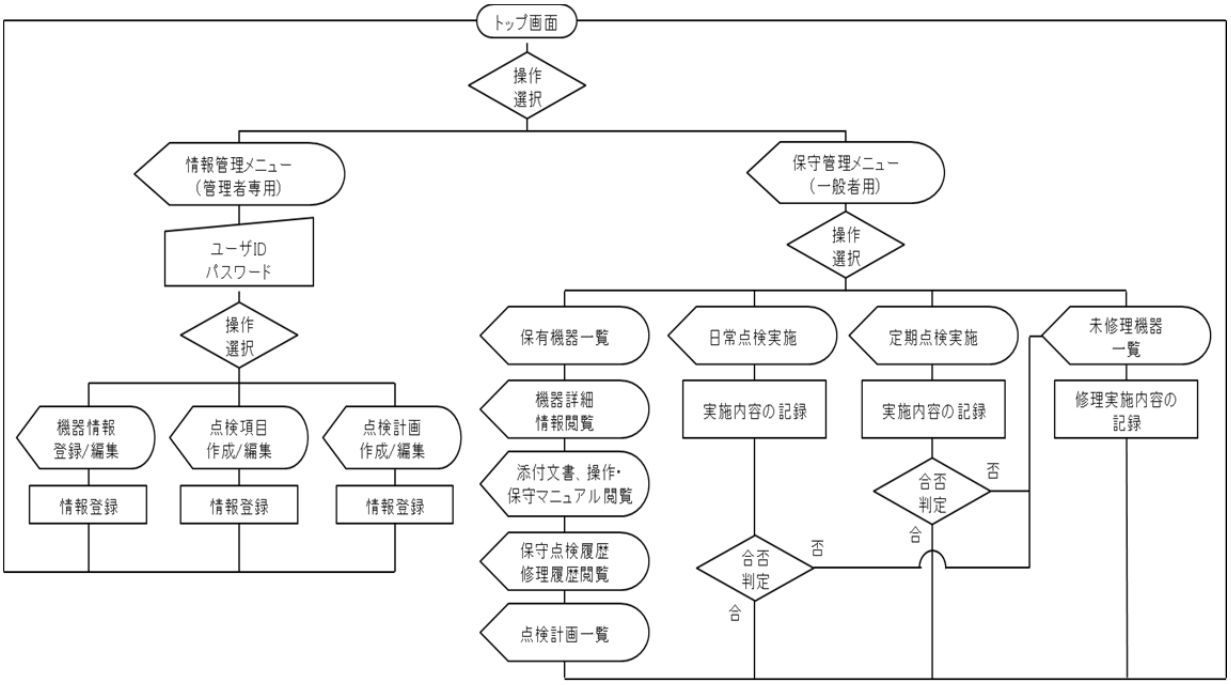


図 4-6　医療機器管理用 Web アプリケーションのフローチャート

4-6-1　情報管理メニュー

4-6-1-1　ログイン認証フォーム

情報管理メニューでは，医療機器情報の登録/編集，保守点検項目の作成/編集，保守点検計画の作成/編集などの情報入力を行う．そのため，真正性を確保するためにログイン認証フォームを設け，管理者専用の ID，パスワードでログインする．これにより，いつ，だれが，どのような情報を書き換えたのか把握することができる．図 4-7 にログイン認証フォームを示す．認証方式は，Digest 認証方式を採用し，ID とパスワードはサーバ側で管理される．管理場所は，/usr/local/tomcat/webapps/me_app/WEB-INF/web.xml となる．

なお．情報管理メニューでは，情報入力が主な機能となるため，ノート PC 上の作業を想定している．そのため，以降に示す図はノート PC 上の Internet Explorer での出力画面を示す．

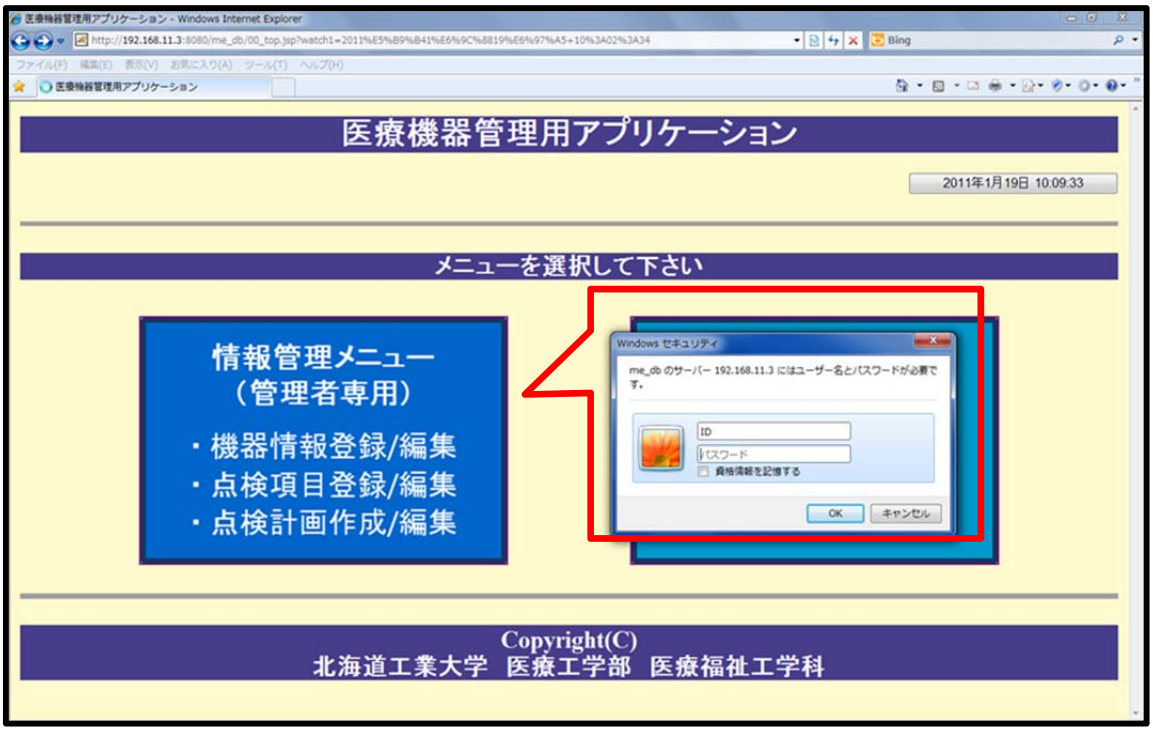


図 4-7　情報管理メニューの認証画面

4-6-1-2　医療機器情報登録/編集

医療機器情報の新規登録画面を図 4-8 に示す．ここでは，上段に医療機器の基本情報，下段に医療機器個々の詳細情報を登録できる．基本情報は医療機器のモデル（型式）ごとに機器名称や機器区分，メーカ，ディーラ，JAN コード，薬事法で定められている医療機器の一般名称である JMDN コードを登録する．また，特定保守管理医療機器に該当するか，またクラス分類の選択を行う．また，機器の外観は画像ファイル（jpg など）を参照し，添付文書，操作・保守マニュアルは PDF を参照する．なお，事前に登録されている内容については，リストダウンボタン「▼」をクリックすることで，入力操作を省くことできる．

詳細情報は，医療機器個々に割り振る管理番号，シリアル（製造）番号，購入年月日，廃棄年月日，使用可能であるか保守点検が必要であるかを示す管理状態，設置部署，定期点検間隔，担当予定者，次回定期点検日，保守契約を登録する．管理番号は登録方式を統一するために必ず半角入力となるように設定し，登録日時は自動入力される．

上記の必要事項を入力し登録ボタンをクリックすることでデータベースの医療機器情報テーブル（me_info_tbl）と医療機器台帳テーブル（me_register_tbl）に記録される．また，登録時，警告ダイヤログを表示し，登録の確認をするように設定し，入力漏れや管理番号の重複がある場合には，警告を返すようになっている．

登録が完了することで，図 4-9 に示すような形で画面が表示される．情報の編集に関しても，同様の形式となる．

図 4-8　医療機器新規登録画面

図 4-9　医療機器情報新規登録確認画面

4-6-1-3　保守点検項目作成/編集

図 4-10 に保守点検項目一覧画面を示す．ここでは，医療機器のモデル別に日常点検（始業点検・使用中点検・終業点検）項目および定期点検項目の作成と編集を行う．作成または編集を行うモデルの点検種別を選択することで，図 4-11 に示す保守点検項目の作成/編集画面となる．ここでは，保守点検の分類（例．外観点検，機能点検など）から保守点検項目を登録する．これらの情報は保守点検項目情報テーブル（check_item_tbl）に記録される．

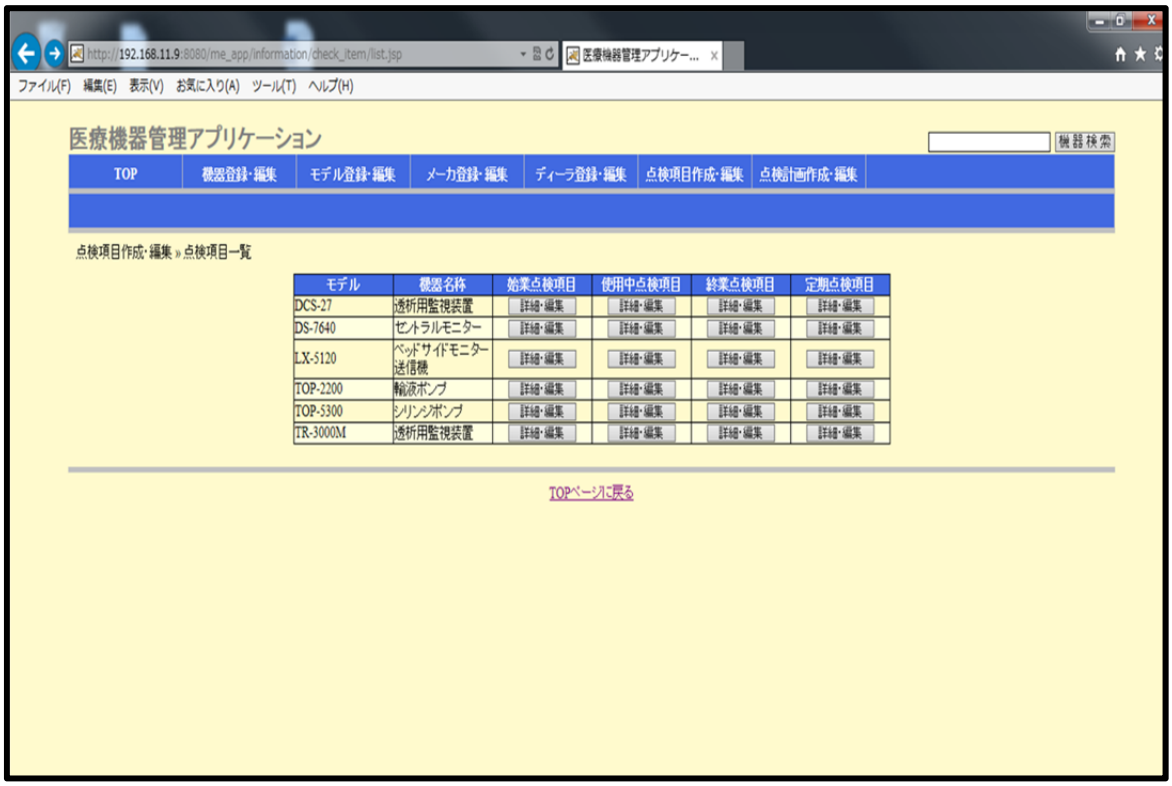


図 4-10　保守点検項目一覧画面

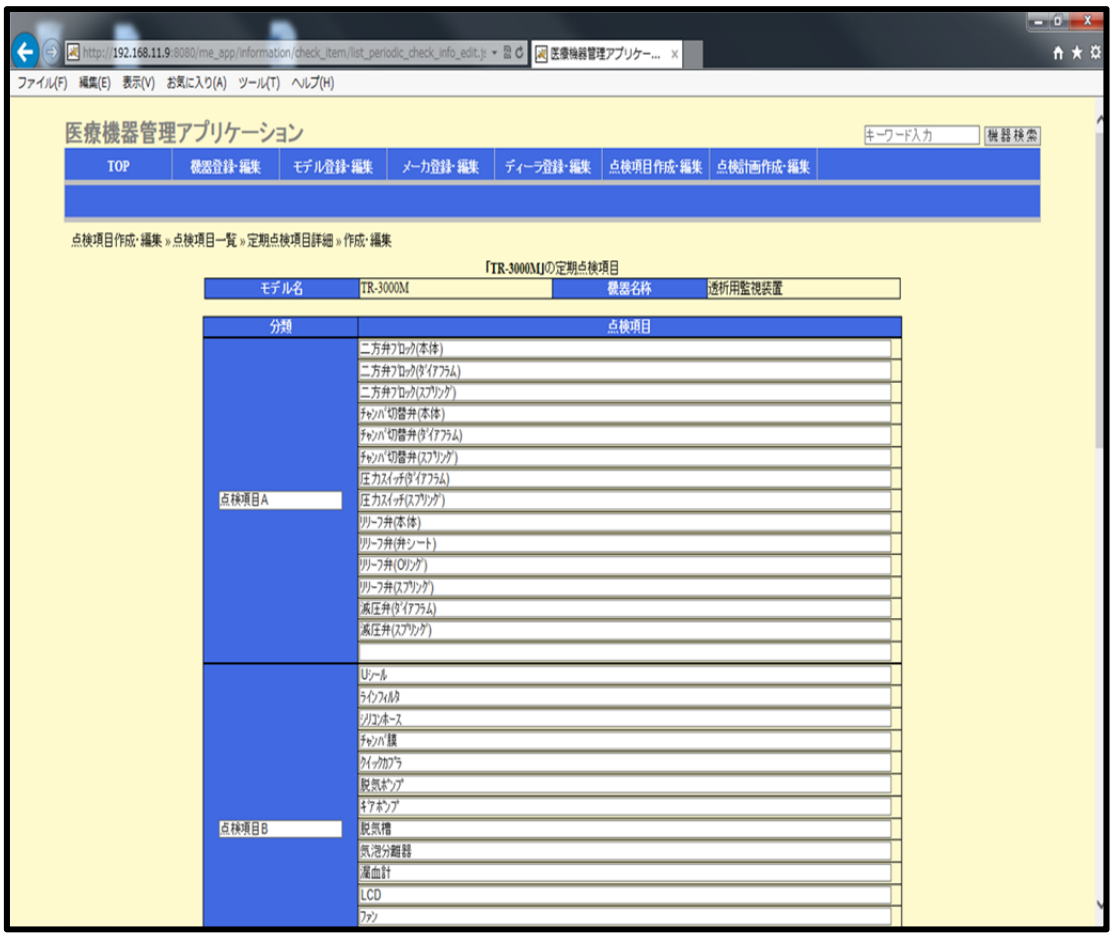


図 4-11　保守点検項目作成/編集画面

4-6-1-4　保守点検計画作成/編集

図 4-12 に保守点検計画の作成/編集画面を示す．ここでは，医療機器個々に対して定期点検を行う周期を指定し，点検計画の作成と編集を行う．保守点検間隔を数字入力することで，最終保守点検日から次回点検日が自動計算される．また，次回点検日が過ぎている場合に赤文字で警告表示する．これらの情報は，保守管理メニューの定期点検および点検計画一覧に反映され，保守点検計画テーブル（check_plan_tbl）に記録される．

図 4-12　保守点検計画の作成と編集

4-6-2　保守管理メニュー

4-6-2-1　医療機器詳細情報閲覧

図 4-13 に保有機器一覧画面を示す．ここでは，管理者メニューの医療機器情報登録画面で登録した医療機器の一覧を閲覧することができる．また，管理番号，機器名称，モデル，メーカ，管理部署でソートし保有機器を整列することや画面右上のテキストボックスより検索することができる．ここで，詳細ボタンを選択することで，機器詳細情報画面へと移行する．なお．保守管理メニューでは，情報閲覧が主な機能となるため，タブレット PC 上の作業を想定している．そのため，以降に示す図はタブレット PC 上の Google Chrome での出力画面を示す．

図 4-13　保有機器一覧画面

図 4-14 に医療機器詳細情報画面を示す．ここでは，医療機器の基本的な情報に加え，添付文書の閲覧，日常点検（始業点検，使用中点検，終業点検）および定期点検の実施回数や履歴を閲覧することができる．例として，図 4-15 に添付文書閲覧画面，図 4-16 に定期点検履歴詳細情報画面を示す．また，機器詳細情報や添付文書，日常点検（始業点検，使用中点検，中行点検）および定期点検の履歴はすべて印刷可能であり，書面に出力することができる．

図 4-14　医療機器詳細情報画面

図 4-15　添付文書閲覧画面

図 4-16　定期点検履歴詳細情報画面

4-6-2-2　保守点検実施

図 4-17 に保守点検実施画面（例：定期点検）を示す．ここでは，日常点検（始業点検・使用中点検・終業点検）および定期点検の記録を行う．医療機器のモデル別に情報管理メニューで登録した点検項目を引出し，その項目に対してラジオボタンで合否を付けていく形式となっている．また，必要に応じて測定値などを記録する備考欄と点検実施日時，点検担当者を記録する．最後に総合評価の判定を行い，点検を終了する．定期点検実施においては点検計画に基づいて行われる．これらの情報は保守点検情報テーブル（check_info_tbl）に記録される．

図 4-17　保守点検実施画面（例：定期点検）

4-6-2-3　修理実施

図 4-18 に修理実施画面を示す．ここでは，点検結果に否があった医療機器に対して修理実施内容を記録する．上段に，医療機器の基本情報を表示し，下段に修理担当者，修理費用，修理内容，備考を入力する．また，修理実施日時は自動入力され，修理情報テーブル（repair_info_tbl）に記録される．

図 4-18　修理実施画面

第 5 章

Open-MEMS の臨床導入

本研究にご協力頂いた泌尿器科を専門とする 70 床の病院に Open-MEMS の導入を行い，クライアント・サーバ型医療機器管理システムを構築した．これにより，医療機器情報のデータベース化と Web アプリケーションによる情報アクセスの即時性を実現し，医療機器管理業務の省力化と効率が向上したことが示唆された．

本章では，導入対象病院の医療機器管理の現状を述べ，Open-MEMS の導入とその結果を述べる．

5-1　導入対象病院における医療機器管理業務の解析

5-1-1　病院施設概要

本研究にご協力頂いた医療機関は，病床数 70 床（うち入院ベッド数 40 床，人工透析ベッド数 30 床）の泌尿器科を専門とする病院である．診療科目は，泌尿器科を始め循環器科があり，人工透析治療や体外衝撃波腎・尿管結石破砕術，ペースメーカ移植・交換術などが行われている．そのうち，手術症例数は年間 1,056 件（2012 年）である．施設基準評価は，日本泌尿器科学会専門医教育施設であり，2012 年に日本医療機能評価機構 Ver6.0 を認定取得している．また，2013 年には電子カルテシステム，透析監視システム，検査システムを導入し，経営や治療をサポートする情報システムを用いた体制整備が積極的に行われている．

5-1-2　医療機器管理の現状

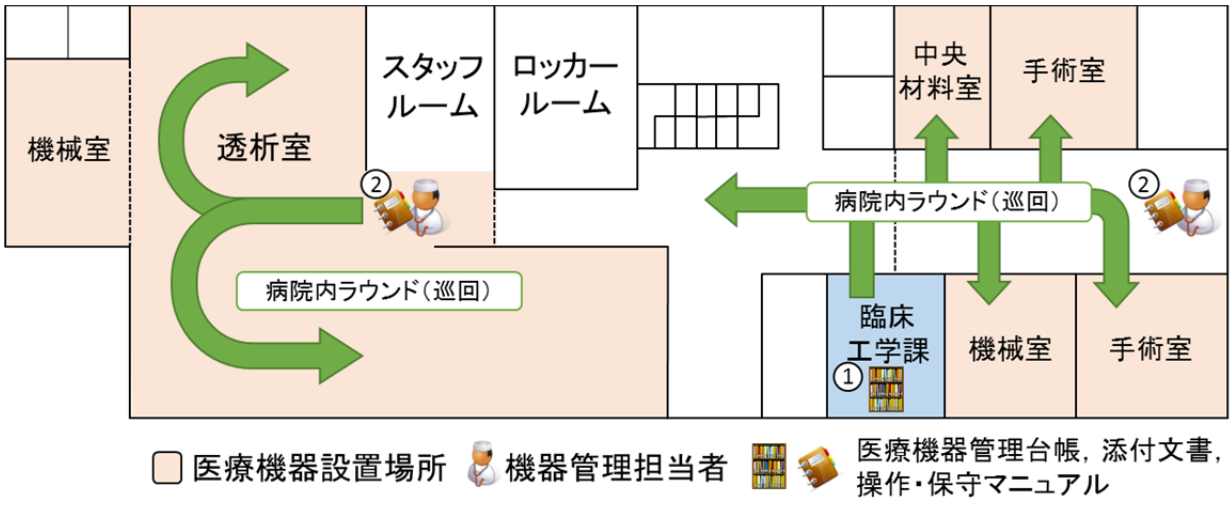
Open-MEMS の導入にあたり，導入対象病院における医療機器管理業務の解析を現場視察と聞取り調査より行った．当該病院では，臨床工学技士を医療機器安全管理責任者として配置し，医療機器の保守点検から安全使用に係る安全教育まで行っていた．その中で，臨床工学技士は 10 名配置され，人工透析業務と医療機器管理業務を各週のローテーションにより兼務して行われていた．医療機器は表 5-1 の保有医療機器一覧に示す 58 台を保有し，これらは手術室や透析室，外来，病棟の各部署で分散管理されていた．臨床工学課では医療機器管理台帳や添付文書，保守点検記録票，操作・保守マニュアルなどの医療機器情報を紙ベースで A4 ファイルに収めて管理していた．保守点検の実施に当たっては，臨床工学課から保守点検記録票と保守マニュアルを持ち出し，病院内をラウンド（巡回）することによって行われていた．図 5-1 に当該病院の見取り図と保守点検実施方法を示す．

これらのことから，臨床工学課において紙ベースの医療機器情報が一元管理されており，医療機器を管理している場所（手術室や透析室など）から必要とする情報へ即座にアクセスすることが困難な状況となっていた．

なお，保守点検記録票の作成においては，メーカの作成した医療機器個々の添付文書と医療機器管理指針対策委員会が策定しているガイドラインを参照し作成している．実際に使用されている保守点検記録票の一例を図 5-2 に示す．

表 5-1　保有医療機器一覧

機器名	台数［台］	機器名	台数［台］
透析装置	30	RO 装置	1
A 液溶解装置	1	B 液溶解装置	1
シリンジポンプ	3	輸液ポンプ	8
心電計	4	電気メス	2
Ho レーザ	1	ダイレーザ	1
ESWL	1	内視鏡装置	2
超音波診断装置	2	血液ガス分析装置	1
		計	58



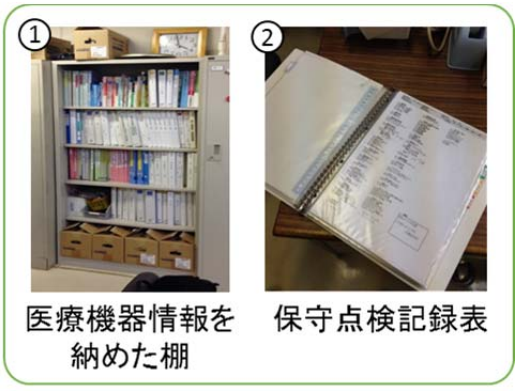
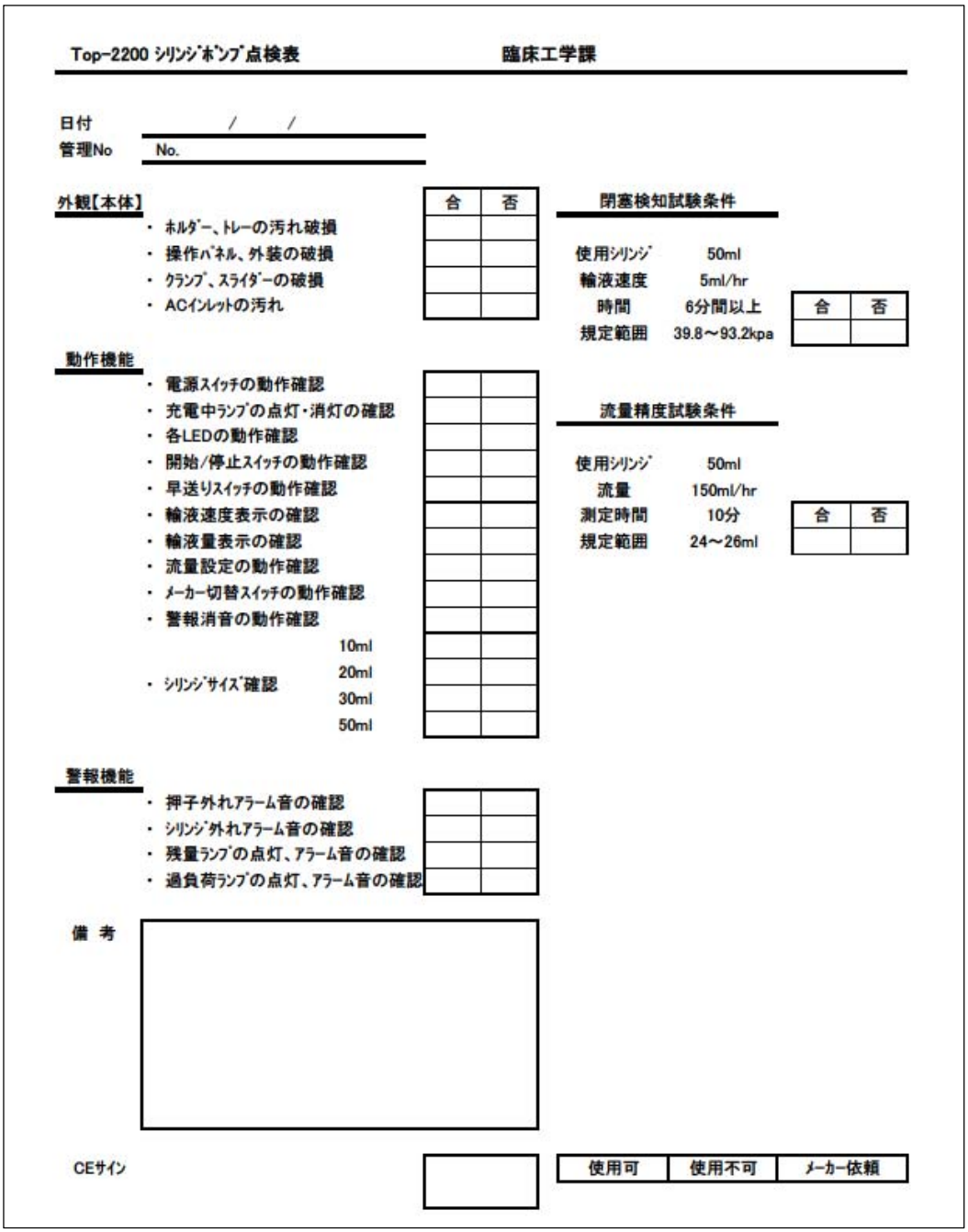


図 5-1　当該病院の見取り図と保守点検実施方法



Top-2200 ｼﾘﾝｼﾞﾎﾟﾝﾌﾟ点検表　　臨床工学課

日付　　/　　/

管理No　No.

外観【本体】

	合	否
・ ﾎﾙﾀﾞｰ、ﾄﾚｰの汚れ破損		
・ 操作ﾊﾟﾈﾙ、外装の破損		
・ ｸﾗﾝﾌﾟ、ｽﾗｲﾀﾞｰの破損		
・ ACｲﾝﾚｯﾄの汚れ		

閉塞検知試験条件

使用ｼﾘﾝｼﾞ	50ml
輸液速度	5ml/hr
時間	6分間以上
規定範囲	39.8～93.2kpa

合	否

動作機能

		合	否
・ 電源ｽｲｯﾁの動作確認			
・ 充電中ﾗﾝﾌﾟの点灯・消灯の確認			
・ 各LEDの動作確認			
・ 開始/停止ｽｲｯﾁの動作確認			
・ 早送りｽｲｯﾁの動作確認			
・ 輸液速度表示の確認			
・ 輸液量表示の確認			
・ 流量設定の動作確認			
・ ﾒｰｶｰ切替ｽｲｯﾁの動作確認			
・ 警報消音の動作確認			
・ ｼﾘﾝｼﾞｻｲｽﾞ確認	10ml		
	20ml		
	30ml		
	50ml		

流量精度試験条件

使用ｼﾘﾝｼﾞ	50ml
流量	150ml/hr
測定時間	10分
規定範囲	24～26ml

合	否

警報機能

	合	否
・ 押子外れｱﾗｰﾑ音の確認		
・ ｼﾘﾝｼﾞ外れｱﾗｰﾑ音の確認		
・ 残量ﾗﾝﾌﾟの点灯、ｱﾗｰﾑ音の確認		
・ 過負荷ﾗﾝﾌﾟの点灯、ｱﾗｰﾑ音の確認		

備 考

CEｻｲﾝ

使用可	使用不可	ﾒｰｶｰ依頼

図 5-2　保守点検記録票の一例（シリンジポンプ）

5-2　Open-MEMS の導入

医療機器情報へのアクセスの即時性を実現する環境へと整備することを目的に，当該病院に Open-MEMS を導入し，図 5-3 に示すようにクライアント・サーバ型医療機器管理システムを構築した. サーバには B5 版サイズのノート PC である Let's note（Panasonic, Inc., OS：CentOS 6.4）を使用し，クライアント端末にはアクセスの即時性を確保するため，7 インチサイズのタブレット PC である Nexus 7（Samsung, Inc.，OS：Android 4.2）を 2 台使用した．サーバは臨床工学課に設置し病院内 LAN に接続した．また，無線 LAN アクセスポイント（Aterm，NEC, Inc.，IEEE802.11n，2.4GHz 帯）を施設既存の電子カルテシステム，透析監視システムと分離する形で新たに 3 台設置した．設置場所は，各手術室の近くに 1 台ずつ，透析室に 1 台の計 3 台とし，厚生労働省の発表している医療情報システムの安全管理に関するガイドラインを参照した上で，WPA2 によるユーザ認証および SSL 通信（Secure Sockets Layer）による通信データの暗号化，ファイヤーフォールと MAC アドレスフィリルタリングによる第 3 者からの不正アクセス防止のセキュリティ対策を行った[5-1]．また，透析室および手術室で，信号受信強度（RSSI：Received Signal Strength Indicator）を計測した結果, 透析室で -80 ～ -40 dB, 手術室側で -61 ～ -35 dB と良好であった．なお，総務省より，埋め込み型医療機器を含むすべての医療機器に対して，無線 LAN の電波は特別注意を必要ないとしている[5-2].

なお，Open-MEMS の導入に当たっては，まず CentOS のインストールを行った．インストールには，CentOS の専門書を参考にし，操作時間とインストール時間を含めて 1 時間程度で完了した[5-3]．また，Open-MEMS のインストール方法は，図 5-4 に示す通りサーバとして使用する CentOS のデスクトップ画面に「Open-MEMS.rpm」を用意し，このソフトウェアの特徴であるダブルクリックによる簡易な方法でインストールを完了した．その後，サーバに用いる IP アドレスの設定と当該病院で保有している 58 台の医療機器情報を医療機器管理用 Web アプリケーションより入力した．医療機器情報の入力には，当該病院の医療機器管理台帳を参照し，2 時間程度で完了した．入力後，クライアント端末からサーバにアクセスし医療機器管理用 Web アプリケーションの動作検証を行った.

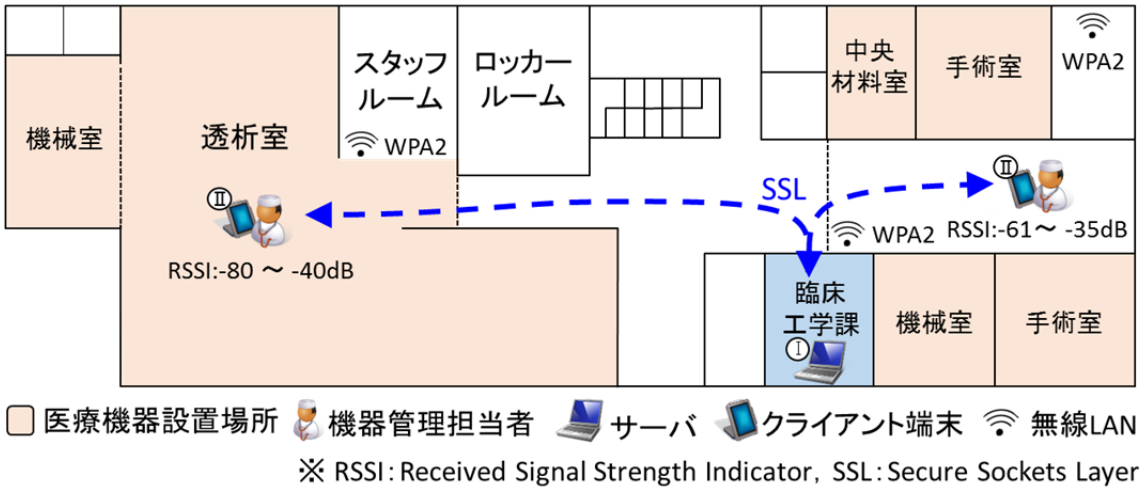


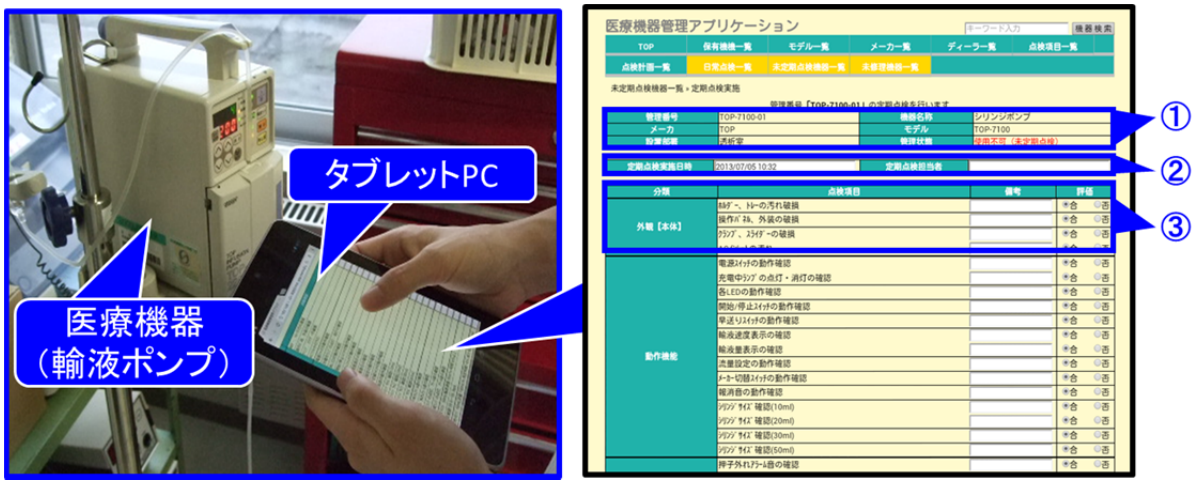
図 5-3　Open-MEMS の導入とシステム構成

図 5-4　Open-MEMS のインストール画面

5-3　Open-MEMS を導入した医療管理の実施

本システムを用いて，医療機器管理業務を行った実施例を図 5-5 に示す．これは，タブレット PC の Web ブラウザより当該病院で使用している定期点検記録票を出力し，保守点検実施状況を記録している場面である．実際に透析室で使用されている輸液ポンプを対象に行っている．ここでの Web アプリケーションは，4-6-2 項で説明した通り，上段に機器情報を表示し，中段に保守点検実施日時と点検担当者，下段は点検項目とそれらに対して合否判定を行うラジオボタン，必要に応じて測定値等を記録する備考欄を出力する．また，このとき図 5-6 に示す保守マニュアルを出力し，それを閲覧しながらの保守点検を可能とした．

以上の Web アプリケーションを使用し，保守点検および医療機器管理を実施した．



① 医療機器情報

管理番号	TOP-7100-01	機器名称	シリンジポンプ
メーカ	TOP	モデル	TOP-7100
設置部署	透析室	管理状態	使用不可（未定期点検）

② 保守点検実施日時・担当者

定期点検実施日時	2014/01/05 07:31	定期点検担当者	

③ 保守点検項目・備考欄・合否判定欄

分類	点検項目	備考	評価
外観【本体】	ﾎﾙﾀﾞｰ、ﾄﾚｰの汚れ破損		◉合 ○否
	操作ﾊﾟﾈﾙ、外装の破損		◉合 ○否
	ｸﾗﾝﾌﾟ、ｽﾗｲﾀﾞｰの破損		◉合 ○否
	ACｲﾝﾚｯﾄの汚れ		◉合 ○否

図 5-5　Open-MEM を使用した保守点検の実施例

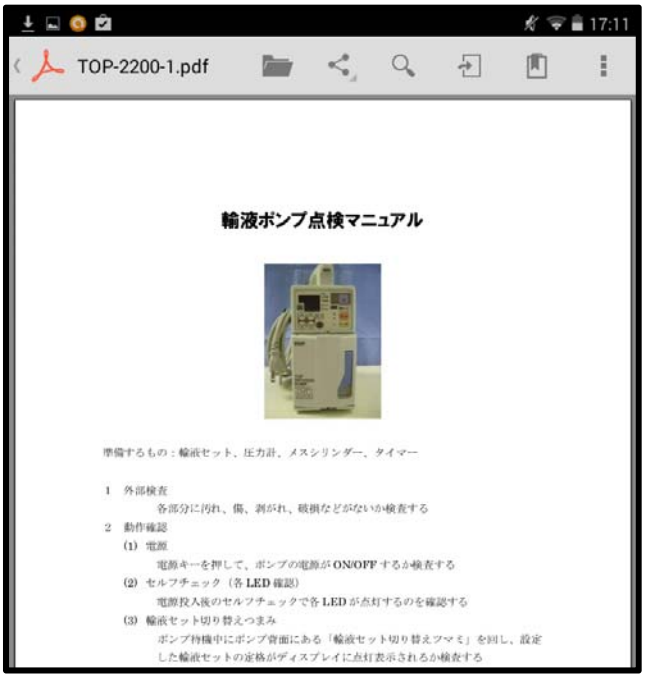


図 5-6　対象機器の保守マニュアルの閲覧

5-4　導入結果

Open-MEMS をサーバとして使用する PC のデスクトップ画面上に用意し，ダブルクリックによる簡易な操作でインストールを行った．これにより当該病院に医療機器管理用のアプリケーションサーバを設置することができた.今回はサーバとして使用したPCに，一般的なデスクワークに使用される B5 版サイズのノート PC を用いた．特に，運用中に操作ができないなどといったフリーズ状態に陥ることはなく，安定して稼働し，かつ省スペースな設置を可能としたことから，臨床工学課内に収めることが容易であった．また，CentOS のインストールも 1 冊の専門書で十分であり，1 時間程度でインストールが可能であった.

システム構築に際しては，既存システムと分離するため，新たに無線 LAN アクセスポイントを設置した．電波受信強度は良好であり，透析室や手術室内でもアクセスに混信などの不憫は生じることはなかった．なお，今回ハードウェアの導入に掛かった費用はサーバ用 PC 1 台とタブレット PC 2 台，無線 LAN アクセスポイント 3 台を合計して約 30 万円を要した.

また，当該病院では臨床工学課に医療機器情報を紙ベースで一元管理することによる情報へのアクセスに不便が生じていた．今回は，その環境を整備することを目的に Open-MEMS を導入し，病院内 LAN を利用したクライアント・サーバ型医療機器管理システムを構築した．これにより，医療機器管理台帳および添付文書，操作・保守マニュアル，保守点検記録票などの医療機器情報のデータベース化と，タブレット PC を活用した Web アプリケーションによる情報アクセスの即時性を実現することができた．保守点検実施時は，A4 サイズのファイルに収められた保守点検記録票と保守マニュアルを持ち出し，病院内をラウンドすることにより行われおり，非効率的であったが，小型軽量な 7 インチサイズのタブレット PC 1 台で，分散管理されている医療機器の保守点検記録を可能とした．特に，保守点検を実施しながらその実施内容の記録と保守マニュアルの閲覧を即座に行えたことは，医療機器管理業務の省力化と効率向上に繋がり，このことは現場の医療機器管理を担当している臨床工学技士より示唆された．しかし，Web アプリケーション実行端末として使用したタブレット PC は，従来までの記録用紙による方法から一新され，その操作に慣れるなれるために時間を必要とした．また，表示画面の文字サイズやボタンサイズなどにシステム使用者は適応する必要があった．しかし，アプリケーション画面は当該病院で使用されている医療機器管理台帳や点検記録票などのフォーマットに反映させているため，すぐに適応できたとの意見があった.

以上より，Open-MEMS は，当該病院の情報アクセスの不便さを改善し，かつ改正医療法の求める医療機器の適切な保守点検の実施と記録を含めた包括的な医療機器管理システムの導入を支援することが示唆された.

第 6 章

考　察

本研究は，有用な医療機器管理システムの導入や開発が十分に行えない医療機関を対象として，その簡易な導入支援と包括的な医療機器管理が行えるシステム構築を目的に OSS を活用した Open-MEMS と命名する新たな医療機器管理ソフトウェアの開発を行った. また，臨床導入を行いその有用性について検討を行った.

本章では，開発を行った Open-MEMS の有用性と臨床導入について考察する.

6-1　Open-MEMS の開発について

一般的に OSS を使用したサーバシステムの導入は，コマンドラインによる入力や多くの設定を必要とし複雑である．今回のようなアプリケーションサーバを作成するには，データベースサーバと Web サーバをそれぞれインストールし，それらを連動させる環境変数の設定などを行う必要がある．さらに，コマンドラインにより医療機器管理用のデータベースを設計・構築し，そのデータベースに連動した Web アプリケーションを作成する必要がある．これには，データベースの概念と医療機器管理に精通した技術者が不可欠となる．しかし，Open-MEMS は医療機器管理に必要なデータベースおよび Web アプリケーションと多様な設定を一つのパッケージソフトウェアとして作成したため，PC 上のダブルクリックによる簡易な操作で導入でき，システム構築の負担を著しく軽減したユーザフレンドリなソフトウェアとすることができた．また，サーバに使用する OS は Linux 系 OS の一つである CentOS が前提となるが，これはサーバ用途として開発されている商用の Red Hat Enterprise Linux（RHEL）との完全互換を目指したクローン OS の OSS であり，システムの安全性やセキュリティが確保され，ホームページから無償でダウンロードすることができる[4-3]．さらに，必要とするハードウェアのスペックは比較的低くても動作し，今回使用した B5 版サイズのノート PC（CPU：1.33GHz，RAM：1GHz，HDD：120GB）でも十分に安定して稼働することが示された．しかし，Linux 系 OS の使用経験のない者には導入時点で障壁になり，本ソフトウェアの普及における今後の課題であると考える．しかし，CentOS が導入できれば，その後のシステム導入は，既述のパッケージ化により簡便化しているため，従来よりも導入の障害を軽減されると考える．

今回の開発に当たり，データベースで管理する医療機器情報や医療機器管理 Web アプリケーションの機能について，医療機器管理指針対策委員会の策定したガイドラインを精査し，また現場の医療機器管理方法を調査することにより，医療機器管理業務に必要な項目や機能に対して過不足のない，かつある程度の雛形を完成させることができたと考える．特に，医療機器管理業務を行う上で必要となる添付文書，操作・保守マニュアルの閲覧や医療機器によって異なる保守点検項目を大項目 10 項目，小項目 30 項目の範囲内で作成できるように工夫して作成した．そのため，他施設の保守点検項目に合わせた保守点検記録票の作成が可能となっている．また，OSS を活用することで Open-MEMS も，そのソースコードが公開され，入手した人の判断でその改変，機能の追加が可能となる．このことは，Open-MEMS のデータベース及び Web アプリケーションを拡張することで，他施設の医療機器管理に合わせたカスタマイズが行える．特に近年では，RFID（Radio Frequency Identification）や PDA（Personal Digital Assistant），WiFi タグの位置情報を用いた医療機器の包括的管理の試みがなされており，本システムでもデータベースおよび Web アプ

リケーションを拡張することで対応できると考えている[6-1][6-2][6-3]．中には，市販の医療機器管理システムでは不足している機能を Microsoft Access を使用して拡張し，対応している医療機関もある[6-4]．これは，市販の医療機器管理システムにおけるカスタマイズが柔軟でないことを理由とし，市販システムと独立した形で開発・運用している．Open-MEMS は機能追加などのカスタマイズも可能とし，システムを独立させることはない．

また，Open-MEMS が提供する機能は，主に医療機器情報の登録・参照・管理，保守点検・修理情報の記録・履歴管理，保守点検項目および計画の作成などであり，これらは改正医療法の求める医療機器管理の中では必要最低限の機能と考える．しかし，医療機器を安全かつ効率的に運用していく上で有益な機能が様々に提案されている．例えば，医療機器管理の多くは機器管理室に集中管理する体制が構築されており，医療機器を使用する場合には，そこから貸借が行われる．この貸借情報を記録することは，医療機器使用時のトレーサビリティを確保する上で必要不可欠な情報となる[6-5]．また，医療機器の稼働時間の算出・記録や保守点検・修理情報より故障率の算出を行う機能など，今後は，このような機能を追加していくことが望まれ，発展させていくことが必要であると考える．

一方で，OSS はソフトウェアに掛かる責任の所在が明確でなく，Open-MEMS も該当する．しかし，OSS は多数の開発者がバクなどのソフトウェアの修正を行ってアップロードされることが多く，一般ユーザが自らソフトウェアを修正するといったことはほとんどない．また，Open-MEMS を構成するソフトウェアは最新かつ最終バージョンのものを使用し，データベースおよび Web サーバソフトウェア自体のバグやエラーに関しては十分に対処されている．このことから，特に致命的なトラブルは今後もないことが予測されるが，Open-MEMS の安定性を評価するためにも，今後も臨床導入を続けて行き，継続して検討する必要があると考える．

6-2　臨床導入について

導入対象病院の医療機器管理業務の解析を行った結果，問題点として，臨床工学課に医療機器情報を紙ベースで一元管理することによる情報へのアクセスに不便が生じていた．その理由として，医療機器は治療や手術の行われる場所（透析室や手術室）で分散管理されているが，機器管理台帳や添付文書，操作・保守マニュアルなどの医療機器情報は臨床工学課に一元管理されていることが原因として考えられた．情報が一元化されていることは望ましいが，当該病院のように情報の存在する臨床工学課以外で保守点検を行う必要がある場合，A4 ファイルに収められた保守点検記録票と保守マニュアルを持ちだす必要があり，非常に非効率であった．このことは，医療法に求められる包括的管理を実施する上で

の障害ともなり得た．このことから，Open-MEMS の導入による病院内 LAN を利用したクライアント・サーバ型医療機器管理システムの構築は，医療機器情報のデータベース化と Web アプリケーションによる情報アクセスの即時性を図ることができ，医療機器管理業務における利便性と安全性の向上に繋がると考える．また，今回の導入では情報アクセスの即時性を確保するためにタブレット PC を使用したが，このことは，簡易的にデータベースシステムを構築できるスタンドアロン向けの Microsoft Access などでは，機能不足であると考える．Microsoft Access は情報共有を行うことはできるが，クライアントの数に限りがあることに加えて，情報更新速度が十分ではない点は，データベースをより安全に運用することや，今後大規模なシステムに発展させていくためにも考慮すべき点であると考える．さらに，近年では情報通信技術が発展し，SaaS（Software as a Service）型システムが医療機器管理分野でも利用されている[6-6]．さらに，これにより多数の医療施設をカバーし，施設間での情報共有により組織的な医療機器管理戦略を立てている地域も存在する[6-7][6-8]．しかし，医療情報を病院外に晒すことやその通信速度が危惧されている現状があるなかでは，今回のクライアント・サーバ型システムの構築は有用であると考える．

一方で，Open-MEMS の導入により従来までの記録用紙を基にした医療機器管理方法からタブレット PC を用いた医療機器管理業務へと一新され，その操作に適応するための時間を必要とした．また，表示画面の文字サイズやボタンサイズなどにシステム使用者が慣れる必要があった．今回の Web アプリケーションの開発では「HTML」を使用したことからも，表示画面の設定などは FileMaker など比べて直観的ではないことが欠点として挙げられる．そのため，直観的かつ高いユーザビリティを備えるユーザインターフェースの検討と，限られた画面サイズで提供可能な情報項目の精査は，システム使用者の作業効率を向上させることからも，重要な検討課題となった．

また，当該病院の規模で今回のようなクライアント・サーバ型システムを市販の機器管理システムで導入する場合，最小構成（サーバ PC 1 台，クライアント PC 1 台）で約 500 万円の見積もりが提示された．さらにライセンス・カスタマイズ費用が加えられる．今回の臨床導入では，1 台のサーバと 2 台のクライアント端末，3 台の無線 LAN アクセスポイントなどのハードウェアに約 30 万円の費用を要したが，比較的安価に導入できたものと考える．施設既存のものや近年急速に発展し低価格化が進んでいるタブレット PC を利用できることは，医療機器管理システム導入におけるイニシャルコストを抑えることができると考える．

第 7 章

結　論

本研究は，有用な医療機器管理システムの導入や開発が十分に行えない医療機関を対象として，その簡易な導入支援と包括的な医療機器管理が行えるシステム構築を目的にOSSを活用した Open-MEMS と命名する新たな医療機器管理ソフトウェアの開発と臨床導入を行い，その有用性について検討した.

本章では，本研究によって得られた知見をまとめ，結論とする.

（1）本研究で開発を行ったOpen-MEMSの有用性を以下に総括する.

① システム構成

Open-MEMSは，本研究の目的を実現するために医療機器管理用に作成した医療機器情報を管理するデータベースサーバとWebアプリケーションを提供するWebサーバを簡易な方法で導入できるようパッケージ化したソフトウェアである．これを1台のPCにインストールすることで医療機器管理用アプリケーションサーバを実装できることが最大の特徴である．Open-MEMSをインストールしたPCに，病院内LANを利用することでクライアント・サーバ型の医療機器管理システムを構築することができる．これにより，システム使用者はクライアント端末上のWebブラウザを使用してサーバにアクセスし，提供される医療機器管理用 Web アプリケーションから機器情報の閲覧や保守点検実施状況の記録を行うことが可能である．このように，クライアント側とサーバ側で役割を分散することにより，システム全体の処理を高速化でき安定したシステムを実現できる．また，ネットワーク上にサーバを置くことにより，複数のクライアントで情報の入出力が可能となった．以上より，改正医療法に求められる包括的な医療機器管理を実施できるシステムの導入が行えた.

② Open-MEMSについて

Open-MEMS は，主にデータベースサーバの PostgreSQL 8.4.13 と Web サーバの Tomcat 6.0.37 から成り，それぞれのスクリプトファイルと設定ファイル，医療機器管理用に作成したデータベースファイルと Web アプリケーションファイルから構成される．Open-MEMSは，これらのソフトウェアを簡易に導入できるように一つのファイルにまとめ，パッケージ化を行っている．パケージ化の方法は，パッケージ管理システムであるRPM Package Manager を使用し，拡張子を「.rpm」としている．これにより，作成した「Open-MEMS.rpm」は実行形式のインストールファイルとなり，PC 上のダブルクリックによる簡易な操作でインストール可能となった.

③　Open Source Software による開発

Open Source Software（OSS）による開発は，その定義からもソースコードの自由な利用・改変・再配布が可能である．また，オープンソースコミュニティにより日々ソフトウェアの開発が進められ，商用のソフトウェアにも劣らない高度な機能を取り備えたソフトウェアが多くある．つまり，OSS を使用した医療機器管理システムを導入することができれば，安価でありながら効率的かつ高度な医療機器管理が実現可能となる．また，本研究で開発した Open-MEMS は多施設での導入を支援し，今後公開していくことができる．このことは，医療機器管理業務における医療機器管理システムの普及と向上に寄与できるものと考える．

以上のことから，本研究で開発した Open-MEMS を用いることで，医療機器のより安全かつ効率的な管理・運用が可能となる．

（2）本ソフトウェアを使用するにあたり以下の点に注意しなければならない．

①　情報処理に関する知識の必要性

本システムを有効利用するためには，データベースやプログラミングに関する知識がある程度必要であり，実際にシステムを用いて医療機器管理に当たる担当者が，これらの知識を持って運用にあたることが望ましい．そのため，OSS の使用経験のない者には導入時点で障壁となるが，書店には初心者に向けて多くの専門書が販売されており，初歩が理解できれば，十分に運用できると考える．また，システム導入は既述のパッケージ化により簡便化しているため，従来よりもはるかに導入の煩雑さを低減できると考える．

③ 病院内ネットワークの存在

クライアント・データベースシステムを有効利用するには，病院内 LAN のようなネットワーク環境が整備されていることが前提条件となる．ネットワーク環境が未整備の病院において使用する場合，新たにネットワーク環境の整備が必要となり，そのために初期投資などの諸問題を考慮しなければならない．しかし，本ソフトウェアは Open-MEMS がインストールされた PC のみでも，スタンドアロンでの医療機器情報閲覧，保守点検記録の管理，暦閲覧などが行える．

④ オープンソースによる開発の注意点

OSS は，その定義からもソースコードが公開され，入手した人の判断で複製やプログラムの改変，機能の追加が可能となる．そのため，オープンソースを考える上で商用ソフトウェアの知的所有権を守ることが必要となる．また，オープンソースの定義には特許権についての取り決めはなく，自分のアイディアを組み込んだプログラムを正式にリリースすることには問題が出る可能性がある．さらに，OSS は，責任の所在がはっきりしない．そのため OSS を使用する場合には，この免責事項に注意を払う必要があり，常に危機意識を持って，問題が起きないかどうかを確認しながら開発していく必要がある．

（3）本システムの構築時に浮上した課題について以下に示す．

① 機能の拡張

今回開発を行った Open-MEMS は，研究協力病院の医療機器管理方法を基に，主に医療機器情報の登録・参照・管理，また保守点検・修理情報の記録・履歴管理，さらに保守点検項目および計画の作成などの機能を有する．これらは改正医療法の求める医療機器管理の中では必要最低限の機能と考える．しかし，医療機器管理に必要な機能は多様である．例えば，医療機器のトレーサビリティを確保するための貸借管理や，医療機器の故障期間を把握するために必要な稼働時間の算出を行う機能など，医療機器を安全かつ効率的に運用していく上で必要な機能がある．今後は，このような機能を追加していく必要があり，発展させていくことが望まれる．

② ソフトウェアおよびシステムの保守管理

OSS の利用は，その品質や性能に関するすべてのリスクは，ユーザが追うものとされる．そのため，ソフトウェアに欠陥があるとわかった場合には，ユーザ自身による対処や補修または訂正が求められる．Open-MEMS についても，例外ではなく今回の臨床導入を今後も続けていき，その必要な内容を解析し，安全かつ安定したシステムを構築していく必要がある．

以上の点を考慮することで，医療機器管理システムの普及と包括的な医療機器管理が可能となり，安全かつ効率的な医療機器管理業務の向上に寄与できるものと考える．

最後に，医療機器の包括的管理が求められている中で，自施設の規模や管理スタイルに合わせた医療機器管理システムを構築することは，安全で適切な医療機器管理が可能になるだけではなく，業務の効率性や費用面からみても有用な点が多いと言える．実際に多く

の医療機関では，比較的安価なデータベースソフトウェアを活用し，独自の医療機器管理システムの開発を行っている．しかし，その開発時間やソフトウェアに掛かるライセンスなどが影響し，十分な開発や導入が行えない医療機関が存在している．

このことからも，本研究のようにOSSを活用して開発を行った新たな医療機器管理ソフトウェア「Open-MEMS」は，その簡易な導入支援と包括的な医療機器管理を可能とし，有用であることが臨床導入より示唆され，非常に貴重であると考える．

また，OSSを使用したソフトウェアは医事会計システムなどで開発が進んでいるが，現在のところ「医療機器管理」に関するソフトウェアは本研究の「Open-MEMS」以外存在していない．

以上のことから，Open-MEMSが医療機器管理システム構築のための新たなツールとして今後発展し，安全で高品位な医療提供に大きく寄与すること、また医療情報工学に大きく貢献するものと期待する．

謝　辞

本研究の遂行と本論文をまとめるにあたり，修士課程を含め直接の指導教員として 5 年間にわたり惜しみない御指導を賜ることとなった北海道工業大学大学院工学研究科応用電子工学専攻有澤準二教授に厚く御礼申し上げる．

そして，副査として本論文を御査読して頂き，多くの御指導・御助言を賜った木村主幸教授並びに北間正崇教授に厚く御礼申し上げる．

また，学会への投稿及び発表の際に連名者として多くの示唆並びに貴重な御助言を頂戴した山下政司教授，清水久恵教授，守田憲崇講師に厚く御礼申し上げる．

本研究は，これら各位の多大な御指導・御助言の下に遂行できたものであり，ここに改めて感謝申し上げる．

平成 26 年 3 月

渡邉　翔太郎

参考文献

第１章 序論

［1-1］ 厚生労働省：医療機器に係る安全管理のための体制確保に係る運用上の留意点について，医政指発第 0330001 号及び医政研発第 0330018 号，2007.

［1-2］ 医療機器管理指針対策委員会：医療機器の保守点検に関する計画の策定及び保守点検の適切な実施に関する指針 Ver.1.02，㈳日本臨床工学技士会，2007.

［1-3］ 厚生労働省：B011-4 医療機器安全管理料，平成 24 年医科診療報酬点数表，第 2 章特掲診察料第 1 部医学管理等，2012.

［1-4］ 高倉 照彦：保守管理部門におけるコストの現状，Clinical Engineering，Vol14，No.3，pp.273-282，2003.

［1-5］ 西謙一：医療機器管理ソフトの開発と無償提供，医工学治療，Vol.19，No.3，pp.189-196，2007.

［1-6］ 新 秀直，玉井 久義：市販の機器管理支援ソフトの現状と展望，医器学，Vol.76，No.11，pp.55-62，2006.

［1-7］ 厚生労働省：保健医療分他の情報化にむけてのグランドデザイン，2001.

［1-8］ 小林慎治，八幡勝也，宮司正道，岡田昌史，中原孝洋，石原謙：医療分野における Open Source Software 活用の現状と問題点，医療情報学，Vol.26，No.5，pp.341-350，2006.

第２章 医療分野の現状と医療機器の安全管理をめぐる動向

［2-1］ 国立社会保障・人口問題研究所：日本の将来推計人口，2012.

［2-2］ 厚生労働省：平成 23 年度国民医療費の概況，国民医療費の状況，2013.

［2-3］ 厚生労働省：医療費の動向，2012.

［2-4］ 総務省統計局：平成 22 年国勢調査，2012.

［2-5］ 厚生労働省：平成 22 年度我が国の保健統計，2012.

［2-6］ 厚生労働省：平成 24 年医療施設（動態）調査・病院報告の概況，2014.
［2-7］ 中澤勇一 : 医師不足の現状と対策, 信州医誌, Vol.58, No.6, pp.291-300, 2010.
［2-8］ 久保田 博南 : 医療機器の歴史（最先端技術のツールを探る），（株）医書出版部，2003.
［2-9］ （財）日本医療機能評価機構：平成 24 年度年俸，2012.
［2-10］ 薬事法（昭和 35 年法律第 145 号）第 2 条 4 項，1960.
［2-11］ 医療法（昭和 23 年法律第 205 号）第 6 条 10 項，1948.
［2-12］ 医療法施行規則（昭和 23 年厚生省令第 50 号）第 1 条 11 項，2007 改正.
［2-13］ 臨床工学技士法（昭和 62 年法律第 60 号），1987.

第 3 章 オープンソース・ソフトウェア

［3-1］ オープンソースイニシアティブ：オープンソースの定義，http://www.opensource.jp/osd/osd-japanese.html
［3-2］ 著作権法（明治 32 年法律第 39 号）
［3-3］ 秋本 芳伸, 岡田 泰子 : オープンソースの理解,（株）ディー・アート, pp49-80, 2004.
［3-4］ 内閣府総合規制改革会議：第 9 回総合規制改革会議資料，2004.
［3-5］ 日本医師会：https://www.med.or.jp/
［3-6］ ORCA Project：http://theorcaproject.wordpress.com/
［3-7］ SorceForge：http://sourceforge.net/

第 4 章 医療機器管理ソフトウェア「Open-MEMS」

［4-1］ 中島 能和，飛田 伸一郎：CentOS 徹底入門 第 3 版，（株）翔泳社，2012.
［4-2］ 斉藤 和邦 : はじめての CentOS6 Linux サーバ構築編,（株）秀和システム, 2011.
［4-3］ CentOS プロジェクトホームページ：http://www.centos.org/
［4-4］ 浅羽義之，石田朗雄：PostgreSQL 徹底入門 第 2 版，（株）翔泳社，2008.
［4-5］ NPO 法人日本 PostgreSQL ユーザ会：http://www.postgresql.jp/
［4-6］ Jason Brittain, Ian F. Darwin : Tomcat ハンドブック 第 2 版,（株）オライリー・ジャパン，2008.
［4-7］ Apache Software Foundation：http://jakarta.apache.org/
［4-8］ Oracle：http://www.oracle.com/index.html
［4-9］ 谷尻 かおり : これだけはおさえたいデータベースの基礎の基礎 改訂新版,（株）技術評論社，2009.

第 5 章 Open-MEMS の臨床導入

［5-1］ 厚生労働省 : 医療情報システムの安全管理に関するガイドライン第 4.1 版，2010.

［5-2］ 総務省電波利用ホームページ：http://www.tele.soumu.go.jp/index.htm.
［5-3］ 齊藤和邦：TECHNICAL MASTER はじめての CentOS6 Linux サーバ構築編，㈱秀和システム，2011.

第６章 考察

［6-1］ 赤羽智幸，保坂良資，室橋高男，大谷真：病院内 ME 機器包括管理への応用のための UHF 帯パッシブタグの認証機能に関する基礎的検討，生体医工学，Vol.50，No.1，pp.124-130, 2012.
［6-2］ 嶋﨑公司，鎌田元喜，加藤友子，天木啓孝：PDA を利用した医療機器の保守管理，医器学，Vol.80，No.1，pp.14-19，2010.
［6-3］ 新秀直，田中勝弥，玉井久義，大江和彦：医療機器の保守管理のための Wifi 位置情報検出システムの開発と評価，医器学，Vol.79，No.6，pp.9-17，2009.
［6-4］ 村上幸司，林啓介，高橋啓歩，上田貴美子，高尾晃輔，高橋雅人，岡田弘毅，角幸奈，佐藤芳美，田尾信幸，青木豪，佐々木新，内海美智子，片岡かおり，千田芝樹:病棟で使用するＭＥ機器運用システムの開発と運用, 医器学, Vol.83, No.3，pp.27-31，2013.
［6-5］ 社）日本臨床工学技士会：ME 室ハンドブックー医療機器中央管理のすべてー，じほう，2006.
［6-6］ 澤田務，赤枝卓則，神田千佳代，本田靖雅，遠藤明，武隈良治，三城正紘：SaaS 型 ME 機器管理システムの開発，医機学，Vol.79，No.7，pp.475-476,2009.
［6-7］ Lyad Mobarek，Waleed Al Tarawneh，Francois Langevin，Mohammad Ibbini，“Fully Automated Clinical Engineering Technical Management System”，Journal of Clinical Engineering，pp.46-52，2006.
［6-8］ Chia-Hung Chein，Yi-You Huang，Fok-Ching Chong，“A framework of medical equipment management system for in-house clinical engineering department”，IEEE Engineering in Medicine & Biology Society，pp.6054-6057，2010.

研究業績

原著論文

1. 渡邉 翔太郎，守田 憲崇，北間 正崇，清水 久恵，山下 政司，木村 主幸，有澤 準二，有澤 博明，坂 武敏，“オープンソース・ソフトウェアを使用した医療機器管理システムの開発”，医工学治療，第 26 巻，1 号，2014.

国際会議

1. S. Watanabe, N. Mamorita, M. Kitama, H. Shimizu, M. Yamashita, K. Kimura, J. Arisawa, H. Arisawa, T. Saka, “Medical equipment management support system developed from open source software”, International Conference on Biomedical Engineering 2013, Singapore, December, 2013.
2. S. Watanabe, N. Mamorita, M. Kitama, H. Shimizu, M. Yamashita, K. Kimura, J. Arisawa, H. Arisawa, T. Saka, “System for medical equipment management based on open source software”, IEEE Engineering in Medicine and Biology Society 2013, OSAKA, July, 2013.

シンポジウム

1. 渡邉 翔太郎，守田 憲崇，北間 正崇，清水 久恵，山下 政司，木村 主幸，有澤 準二，有澤 博明，松崎 智哉，坂 丈敏，“オープンソース・ソフトウェアを使用した医療機器管理支援パッケージソフトウェア”，生体医工学シンポジウム 2012，CD-R（4-3-05），2012.

研究会報告

1. 渡邉 翔太郎，北間 正崇，木村 主幸，有澤 準二，有澤 博明，樋口 雅人，松崎 智哉，坂 丈敏，“オープンソース・ソフトウェアを使用した臨床用医療機器管理用データベースサーバの構築”，電子情報通信学技法研究報告，MBE2010-62，pp41-45，2010.

2. 渡邉 翔太郎，守田 憲崇，北間 正崇，木村 主幸，有澤 準二，有澤 博明，松崎 智哉，坂 丈敏，“オープンソース・ソフトウェアを使用した医療機器管理支援パッケージソフトウェアによる物品管理”，電子情報通信学技法研究報告，MBE2012-70，pp33-37．2012.

3. 泉 朋伸，渡邉 翔太郎，守田 憲崇，北間 正崇，清水 久恵，山下 政司，木村 主幸，有澤 準二，有澤 博明，松崎 智哉，坂 丈敏，“データベースアプリケーションの連動による透析回診・医療材料管理支援システムの構築”，電子情報通信学会技術研究報告，MBE2012-70，pp39-43，2012.

学会発表

1. 渡邉 翔太郎，岡田 恵一，菅野 将也，北間 正崇，黒田 聡，木村 主幸，有澤 準二，“ネットワーク IP センサを用いた貸出機器の位置情報取得システム”，第 19 回北海道臨床工学会，2008.

2. 渡邉 翔太郎，北間 正崇，木村 主幸，有澤 準二，“IP センサと電流センサを用いた医療機器稼動時間時間取得システム”，平成 21 年度電気・情報関係学会北海道支部連合大会，2009.

3. 渡邉 翔太郎，二口 伊郎，菅原 俊継，黒田 聡，木村 主幸，有澤 準二，有澤 博明，樋口 雅人，松崎 智哉，坂 丈敏，“データベースを用いた臨床用機器管理システム”，第 20 回北海道臨床工学会，2009.

4. 渡邉 翔太郎，菅原 俊継，黒田 聡，木村 主幸，有澤 準二，有澤 博明，樋口 雅人，松崎 智哉，坂 丈敏，“オープンソース・ソフトウェアを用いた臨床用機器管理システム”，第 20 回日本臨床工学技士会，2010.

5. 渡邉 翔太郎，北間 正崇，木村 主幸，有澤 準二，有澤 博明，樋口 雅人，松崎 智哉，坂 丈敏，“オープンソース・ソフトウェアを使用した臨床用医療機器管理支援システムの構築に関する研究”，第 49 回日本生体医工学会北海道支部大会，2010.

6. 有澤 博明，樋口 雅人，松崎 智哉，坂 丈敏，渡邉 翔太郎，有澤 準二，“Web サーバ用データベースを用いた血液透析装置の管理”，第 55 回日本透析医学会，2010.

7. 有澤 博明，樋口 雅人，松崎 智哉，坂 丈敏，渡邉 翔太郎，有澤 準二，“データベースを用いた血液透析装置の管理”，第 10 回北海道病院学会，2010.

8. 阿部 義啓，佐々木 諒，渡邉 翔太郎，泉 朋伸，北間 正崇，木村 主幸，有澤 準二，“e-learning を用いた学習支援システムに関する研究~第 2 種 ME 技術実力検定試験及び臨床工学技士国家試験対策”，第 21 回北海道臨床工学技士会，2010.

9. 阿部 義啓，佐々木 諒，渡邉 翔太郎，泉 朋伸，北間 正崇，木村 主幸，有澤 準二，“国家試験及び ME2 種対策用 e-learning システム”, 第 21 回日本臨床工学会, 2011.

10. 泉 朋伸，渡邉 翔太郎，北間 正崇，木村 主幸，有澤 準二，有澤 博明，松崎 智哉，坂 丈敏，“データベースを用いた透析回診業務支援システム”，第 22 回北海道臨床工学会，2011.

11. 渡邉 翔太郎，北間 正崇，清水 久恵，山下 政司，木村 主幸，有澤 準二，有澤 博明，松崎 智哉，坂 丈敏，“オープンソース・ソフトウェアによる医療機器管理支援ソフトウェアのパッケージ化”，第 51 回日本生体医工学会，2012.

12. 泉 朋伸，渡邉 翔太郎，北間 正崇，清水 久恵，山下 政司，木村 主幸，有澤 準二，有澤 博明，松崎 智哉，坂 丈敏，“データベースを用いた透析中の回診業務支援システム”，第 51 回日本生体医工学会，2012.

13. 泉 朋伸，渡邉 翔太郎，守田 憲崇，北間 正崇，清水 久恵，山下 政司，木村 主幸，有澤 準二，有澤 博明，松崎 智哉，坂 丈敏，“データベースを用いた透析回診支援と物品管理システム”，第 51 回日本生体医工学会北海道支部大会，2012.

14. 渡邉 翔太郎，守田 憲崇，北間 正崇，木村 主幸，有澤 準二，有澤 博明，坂 丈敏，“オープンソース・ソフトウェアを使用した医療機器管理システムの構築”，第 52 回日本生体医工学会北海道支部大会，2013.

付　録

本研究で開発した Open-MEMS における医療機器管理用 Web アプリケーションのプログラムソースコードを付録として添付する．なお，類似するプログラムソースコードについては省略する．

● 医療機器管理用 Web アプリケーションのプログラムソースコード

・データベース接続コード
・スタイルシート
・トップ画面
・情報管理メニュー
・保守管理メニュー

・データベース接続コード

[me_db.properties]

```
/*設定*/
/*Tomcat5-common-classesに置く*/
con = jdbc:postgresql://192.168.11.3/me_db?user=master&password=arisawa
jdbc = org.postgresql.Driver
```

・スタイルシート

[style.css]

```
/*設定*/
/*<link rel="stylesheet" type="text/css" href="stylesheet.css">*/
/*背景*/
body {background-color:lemonchiffon;}
/*見出し1*/
h1.t {background-color:darkslateblue;color:white;text-align="center";} /*TOP*/
h1.j {background-color:royalblue;color:white;text-align="center";}/*情報管理ページ*/
h1.h {background-color:lightseagreen;color:white;text-align="center";}/*保守管理ページ*/
/*見出し2*/
h2.t {background-color:darkslateblue;color:white;text-align="center";} /*TOP*/
h2.j {background-color:royalblue;color:white;text-align="center";}/*情報管理ページ*/
h2.h {background-color:lightseagreen;color:white;text-align="center";}/*保守管理ページ*/
/*文字*/
p.1 {font size="18pt";}/*大きい文字*/
p.2 {font size="14pt";}/*通常使う文字*/
p.3 {font size="12pt";}/*小さい文字*/
/*リンク*/
a.1 {font size="16pt";}
/*表*/
table
th.j1 {background-color:royalblue;color:white;font size="12pt";font-weight:blod;white-space:nowrap;
width="150"}/*縦表示*/
th.j2 {background-color:royalblue;color:white;font size="12pt";font-weight:blod;white-space:nowrap;}/*横表示
*/
td.j1 {font size="12pt";}
td.j2 {background-color:deepskyblue;color:black;font size="12pt";font-weight:blod; white-space:nowrap; }
th.h1 {background-color:lightseagreen;color:white;font
size="12pt";font-weight:blod;white-space:nowrap;width="150"}/*縦表示*/
th.h2 {background-color:lightseagreen;color:white;font size="12pt";font-weight:blod;white-space:nowrap;}/*横
表示*/
td.h1 {font size="12pt";}
td.h2 {background-color:skyblue;color:black;font size="12pt";font-weight:blod; white-space:nowrap; }
/*インライン-色付け*/
span.1 {background-color:cyan;}
span.2 {background-color:magenta;}
```

トップ画面

[トップ画面 - 00_top.jsp]

```
<!--ディレクティヴ-->
<%@ page contentType="text/html;charset=UTF-8" import="java.sql.*,
java.io.*,java.util.*,java.util.Date,java.text.*" %>
<html>
 <head>
  <title>医療機器管理用アプリケーション</title>
  <link rel="stylesheet" type="text/css" href="stylesheet.css">
  <script src="script.js" type="text/javascript"></script>
 </head>
 <body onLoad="DayWatch()">
 <center>
 <h1 class="t">医療機器管理用アプリケーション</h1>
 <form name="watch">
  <p align="right"><input type="submit" name="watch1"></form></p>
  <hr size="5" noshade="noshade">
 <h2 class="t">メニューを選択して下さい</h2>
  <br>
  <a href="j0_top.html"><img src="img/j1.png"></a>
  <a href="h0_top.html"><img src="img/h1.png"></a>
  <br>
  <br>
  <hr size="5" noshade="noshade">
 <h2 class="t">Copyright(C)<br>北海道工業大学　医療工学部　医療福祉工学科　有澤ゼミ研究室
</h2>
 </center>
 </body>
</html>
```

[ログイン認証フォーム - Web.xml]

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE web-app
 PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"
 "http://java.sun.com/dtd/web-app_2_3.dtd">
<web-app>
 <security-constraint>
  <display-name>Example Security Constraint</display-name>
   <web-resource-collection>
   <web-resource-name></web-resource-name>
    <url-pattern>/j0_top.html</url-pattern>
    <http-method>DELETE</http-method>
    <http-method>GET</http-method>
    <http-method>POST</http-method>
    <http-method>PUT</http-method>
    </web-resource-collection>
     <auth-constraint>
     <role-name>tomcat</role-name>
     </auth-constraint>
    </security-constraint>
   <login-config>
    <auth-method>DIGEST</auth-method>
    <realm-name>me_db</realm-name>
    </login-config>
    <security-role>
     <role-name>tomcat</role-name>
    </security-role>
</web-app>
```

情報管理メニュー

[機器情報登録 －j1_me_insert.jsp]

```
<!--ディレクティヴ-->
<%@ page contentType="text/html;charset=UTF-8" import="java.sql.*,
java.io.*java.util.*java.util.Date,java.text.*" %>
<%
        //現在の日時を取得
         Date dt = new Date();
         SimpleDateFormat today = new SimpleDateFormat("yyyy/MM/dd HH:mm");
%>
<html>
 <head>
  <title>医療機器管理用アプリケーション</title>
  <link rel="stylesheet" type="text/css" href="stylesheet.css">
  <script src="script.js" type="text/javascript"></script>
 </head>
 <body>
  <center>
   <h1 class="j">医療機器情報新規登録</h1>
   <hr size="5" noshade="noshade">
   <p class="1">新規に登録する機器情報を入力して下さい。</p>
    <form action="j1_me_insert01.jsp" method="post" onSubmit="return CONFIRM()">
     <table border="3" cellpadding="3">
      <tr>
       <th class="j1">登録日時</th>
       <td class="j1"><%=today.format(dt)%></td>
       <input type="hidden" name="reg_date" value="<%=today.format(dt)%>">
      </tr><tr>
       <th class="j1">管理番号</th>
       <td class="j1"><input type="text" name="me_no" size="15" onkeyup="chkCode(this)">
「00-00-00」の形で入力して下さい.</td>
      </tr><tr>
       <th class="j1">JMDNコード</th>
       <td class="j1"><input type="text" name="jmdn_code" size="15"></td>
      </tr><tr>
       <th class="j1">機器名称</th>
       <td class="j1"><input type="text" name="me_name" size="15"></td>
      </tr><tr>
       <th class="j1">メーカ</th>
       <td class="j1"><input type="text" name="maker" size="15"></td>
      </tr><tr>
       <th class="j1">モデル</th>
       <td class="j1"><input type="text" name="model" size="15"></td>
      </tr><tr>
       <th class="j1">購入年月日</th>
       <td class="j1"><input type="text" name="buy" size="15"> 「yyyy-mm-dd」の形で入力して下さ
い.</td>
```

```
      </tr><tr>
       <th class="j1">廃棄年月日</th>
       <td class="j1"><input type="text" name="scrap" size="15"> 「yyyy-mm-dd」の形で入力して下さい.</td>
      </tr><tr>
       <th class="j1">設置区分</th>
       <td class="j1"><input type="text" name="set" size="15"> 「中央管理」,「各部署常駐」,etc.</td>
      </tr><tr>
       <th class="j1">機器状態</th>
       <td class="j1">
        <select name="state">
         <option value="使用不可(未点検)">使用不可(未点検)</option>
         <option value="使用不可(未修理)">使用不可(未収理)</option>
         <option value="使用可">使用可</option>
        </select>
       </td>
      </tr><tr>
       <th class="j1">全景</th>
       <td class="j1"><input type="file" name="view" size="15"> 画像ファイルを参照して下さい.</td>
      </tr><tr>
       <th class="j1" rowspan="3">添付文書</th>
      </tr><tr>
       <td class="j1"><input type="text" name="append" size="15"> 保管場所を入力して下さい.</td>
      </tr><tr>
       <td class="j1"><input type="file" name="append_pdf" size="15"> PDFファイルを参照して下さい.</td>
      </tr><tr>
       <th class="j1">特記事項</th>
       <td class="j1"><textarea name="note" cols="50" rows="3">特記事項があれば記入して下さい。</textarea></td>
      </tr>
     </table>
    <br>
     <hr size="5" noshade="noshade">
     <input type="submit" value=" 登録 ">
     <input type="reset" value=" リセット " onClick="return RESET()">
    </form>
   </center>
  </body>
</html>
```

[保有機器詳細情報 – h1_me_info.jsp]

```
<!--ディレクティヴ-->
<%@ page contentType="text/html;charset=UTF-8" import="java.sql.*,java.io.*,java.util.*,java.text.*" %>
<%
        //データベースの設定・接続
          ResourceBundle rb=ResourceBundle.getBundle("me_db");
          Class.forName(rb.getString("jdbc"));
          Connection me_db=DriverManager.getConnection(rb.getString("con"));
        //エンコーディングの設定
          request.setCharacterEncoding("UTF-8");
        //ステートメントオブジェクトの取得
          Statement stdb1 = me_db.createStatement();
          Statement stdb2 = me_db.createStatement();
          Statement stdb3 = me_db.createStatement();
          Statement stdb4 = me_db.createStatement();
          Statement stdb5 = me_db.createStatement();
          Statement stdb6 = me_db.createStatement();
          Statement stdb7 = me_db.createStatement();
          Statement stdb8 = me_db.createStatement();
          Statement stdb9 = me_db.createStatement();
          Statement stdb10 = me_db.createStatement();
          Statement stdb11 = me_db.createStatement();
          Statement stdb12 = me_db.createStatement();
        //データの取得
          String me_no = request.getParameter("me_no");
        //SQLの作成
          String sql_me_info = "SELECT * FROM me_info_tbl WHERE me_no = '"+ me_no +"'";
          String sql_opening = "SELECT * FROM me_opening_check_info_tbl WHERE me_no = '"+ me_no +"'
ORDER BY opening_check_no DESC";
          String sql_opening1 = "SELECT COUNT(*)-1 AS 総始業点検回数 FROM
me_opening_check_info_tbl WHERE me_no = '"+ me_no +"'";
          String sql_using = "SELECT * FROM me_using_check_info_tbl WHERE me_no = '"+ me_no +"' ORDER
BY using_check_no DESC";
          String sql_using1 = "SELECT COUNT(*)-1 AS 総使用中点検回数 FROM me_using_check_info_tbl
WHERE me_no = '"+ me_no +"'";
          String sql_closing = "SELECT * FROM me_closing_check_info_tbl WHERE me_no = '"+ me_no +"'
ORDER BY closing_check_no DESC";
          String sql_closing1 = "SELECT COUNT(*)-1 AS 総終業点検回数 FROM me_closing_check_info_tbl
WHERE me_no = '"+ me_no +"'";
          String sql_regular = "SELECT * FROM me_regular_check_info_tbl WHERE me_no = '"+ me_no +"'
ORDER BY regular_check_no DESC";
          String sql_regular1 = "SELECT COUNT(*)-1 AS 総定期点検回数 FROM me_regular_check_info_tbl
WHERE me_no = '"+ me_no +"'";
          String sql_repair = "SELECT * FROM me_repair_info_tbl WHERE me_no = '"+ me_no +"' ORDER BY
repair_no DESC";
          String sql_repair1 = "SELECT COUNT(*)-1 AS 総修理回数 FROM me_repair_info_tbl WHERE me_no
= '"+ me_no +"'";
```

```
            String sql_ava = "SELECT * FROM me_ava_info_tbl WHERE me_no = '"+ me_no +"'   ORDER BY
ava_no DESC LIMIT 1";
           //SQLのセット・実行
            ResultSet rs1 = stdb1.executeQuery(sql_me_info);          rs1.next();
            ResultSet rs2 = stdb2.executeQuery(sql_opening);            rs2.next();
            ResultSet rs3 = stdb3.executeQuery(sql_opening1);           rs3.next();
            ResultSet rs4 = stdb4.executeQuery(sql_using);              rs4.next();
            ResultSet rs5 = stdb5.executeQuery(sql_using1);             rs5.next();
            ResultSet rs6 = stdb6.executeQuery(sql_closing);            rs6.next();
            ResultSet rs7 = stdb7.executeQuery(sql_closing1);           rs7.next();
            ResultSet rs8 = stdb8.executeQuery(sql_regular);            rs8.next();
            ResultSet rs9 = stdb9.executeQuery(sql_regular1);           rs9.next();
            ResultSet rs10 = stdb10.executeQuery(sql_repair);           rs10.next();
            ResultSet rs11 = stdb11.executeQuery(sql_repair1);          rs11.next();
            ResultSet rs12 = stdb12.executeQuery(sql_ava);              rs12.next();
%>
<html>
 <head>
  <title>医療機器管理用アプリケーション</title>
  <link rel="stylesheet" type="text/css" href="stylesheet.css">
  <script src="script.js" type="text/javascript"></script>
 </head>
 <body>
  <center>
  <h1 class="h">機器詳細情報</h1><hr size="5" noshade="noshade">
   <table border="2" cellpadding="2" width="700">
    <tr><th class="h1">登録日時</th>
     <td class="h1" width="250"><%=rs1.getString("reg_date")%></td>
     <th class="h1" width="300" align="right">全景</th>
    </tr><tr>
     <th class="h1">管理番号</th>
     <td class="h1" width="250"><%=rs1.getString("me_no")%></td>
     <td rowspan="10" align="center" width="300">
     <img src="img/<%=rs1.getString("view")%>" class="Photo"   alt="<%=rs1.getString("view")%>">
    </tr><tr><th class="h1">JMDNコード</th>
     <td class="h1" width="250"><%=rs1.getString("jmdn_code")%></td>
    </tr><tr><th class="h1">機器名称</th>
     <td class="h1" width="250"><%=rs1.getString("me_name")%></td>
    </tr><tr><th class="h1">メーカ</th>
     <td class="h1" width="250"><%=rs1.getString("maker")%></td>
    </tr><tr><th class="h1">モデル</th>
     <td class="h1" width="250"><%=rs1.getString("model")%></td>
    </tr><tr> <th class="h1">購入年月日</th>
     <td class="h1" width="250"><%=rs1.getString("buy")%></td>
    </tr><tr><th class="h1">廃棄年月日</th>
     <td class="h1" width="250"><%=rs1.getString("scrap")%></td>
    </tr><tr><th class="h1">添付文書</th>
     <td class="h1" width="250"><%=rs1.getString("append")%>   |  <input type="button" value="閲覧する"
onClick="window.open('img/<%=rs1.getString("append_pdf")%>','','toolbar=no,location=no,directories=no')
"></td></tr><tr>
     <th class="h1">設置区分</th><td class="h1" width="250"><%=rs1.getString("set")%></td>
    </tr><tr>
```

```
    <th class="h1">機器状態</th><td class="h1" width="250">
        <%          //色つけ
         String a = "使用可";
         if(a.equals(rs1.getString("state"))){%>
         <span class="1"> <%=rs1.getString("state")%> </span></td>
         <%} else{%>
         <span class="2"> <%=rs1.getString("state")%> </span></td>
         <%} %>
   </tr><tr>
    <th class="h1">特記事項</th>
    <td class="h1"   colspan="2" height="50"   width="550" valign="top"><%=rs1.getString("note")%></td>
   </tr></table>
<br>
 <table border="2" cellpadding="2" width="700">
 <tr><form action="h1_me_list>me_info>me_opening_check_info.jsp" method="post">
    <input type="hidden" name="me_no" value="<%=rs1.getString("me_no")%>" />
    <th class="h2" colspan="6" align="center" height="30">日常点検</th>
 </tr><tr>
    <td class="h2" align="center">始業点検回数</td>
    <td class="h1" align="center" width="50"><%=rs3.getString("総始業点検回数")%>回</td>
    <td class="h2" align="center">最終始業点検日時</td>
    <td class="h1" align="center"><%=rs2.getString("opening_check_day_now")%></td>
    <td class="h1" align="center"><input type="submit" value="  始業点検履歴  " ></td>
  </form></tr><tr>
  <form action="h1_me_list>me_info>me_using_check_info.jsp" method="post">
  <input type="hidden" name="me_no" value="<%=rs1.getString("me_no")%>" />
  <td class="h2" align="center">使用中点検回数</td>
  <td class="h1" align="center" width="50"><%=rs5.getString("総使用中点検回数")%>回</td>
  <td class="h2" align="center">最終使用中点検日時</td>
  <td class="h1" align="center"><%=rs4.getString("using_check_day_now")%></td>
  <td class="h1" align="center"><input type="submit" value="使用中点検履歴" ></td>
  </form></tr><tr>
  <form action="h1_me_list>me_info>me_closing_check_info.jsp" method="post">
  <input type="hidden" name="me_no" value="<%=rs1.getString("me_no")%>" />
  <td class="h2" align="center">終業点検回数</td>
  <td class="h1" align="center" width="50"><%=rs7.getString("総終業点検回数")%>回</td>
  <td class="h2" align="center">最終終業点検日時</td>
  <td class="h1" align="center"><%=rs6.getString("closing_check_day_now")%></td>
  <td class="h1" align="center"><input type="submit" value="  終業点検履歴  " ></td>
 </form></tr><tr>
 <form action="h1_me_list>me_info>me_regular_check_info.jsp" method="post">
 <input type="hidden" name="me_no" value="<%=rs1.getString("me_no")%>" />
 <th class="h2" colspan="6" align="center" height="30">定期点検</th>
 </tr><td class="h2" align="center">総定期点検回数</td>
 <td class="h1" align="center" width="50"><%=rs9.getString("総定期点検回数")%>回</td>
 <td class="h2" align="center">最終点検日時</td>
 <td class="h1" align="center"><%=rs8.getString("regular_check_day_now")%></td>
 <td class="h1" align="center"><input type="submit" value="  定期点検履歴  " ></td>
</form></tr><tr>
<form action="h1_me_list>me_info>me_repair_info.jsp" method="post">
<input type="hidden" name="me_no" value="<%=rs1.getString("me_no")%>" />
 <th class="h2" colspan="6" align="center" height="30">修理</th>
```

```
    </tr>
    <td class="h2" align="center">総修理回数</td>
    <td class="h1" align="center" width="50"><%=rs11.getString("総修理回数")%>回</td>
    <td class="h2" align="center">最終修理日時</td>
    <td class="h1" align="center"><%=rs10.getString("repair_day_now")%></td>
    <td class="h1" align="center"><input type="submit" value="  修理履歴  " ></td>
   </form></tr></table>
    <table border="2" cellpadding="2" width="700">
    <tr><th class="h2" colspan="7" align="center" height="30">データ分析</th>
    </tr><tr>
    <td class="h2">MTBF</td>
    <td class="h1"><%=rs12.getString("mtbf")%>分</td>
    <td class="h2">MTTR</td>
    <td class="h1"><%=rs12.getString("mttr")%>分</td>
    <td class="h2">Availability</td>
    <td class="h1"><%=rs12.getString("ava")%></td>
    <td class="h1" align="center">
    <form action="h1_me_list>me_info>me_ava.jsp" method="post">
    <input type="hidden" name="me_no" value="<%=rs1.getString("me_no")%>" >
    <input type="submit" value="  詳細  " >
    </td></form></tr></table><br>
    <hr size="5" noshade="noshade">
    <table><tr><th><form>
          <input type="button" value="  一つ前へ戻る  " onClick="history.back()" />
        </form></th></tr></table>
   </center>
  </body>
</html>
```