DEOS-FY2013-RA-01J

© 2013 産業技術総合研究所

JST-CREST

研究領域

「実用化を目指した組込みシステム用ディペンダブル・オペレーティングシステム」

DEOS **プロジェクト**

D-Case のロボット応用

～日本科学未来館フロア移動ロボットを題材として～

（第２版）

独立行政法人**産業技術総合研究所**

デジタルヒューマン工学研究センター編

AIST　Digital Human Research Center

「実用化を目指した組込みシステム用ディペンダブル・オペレーティングシステム」（DEOS プロジェクト）は科学技術振興機構（JST）の戦略的創造研究推進事業 CREST の研究領域のひとつです。

1. 技術報告書の目的

本技術報告書は、異なるステークホルダ間の合意について、サービスロボットを例に取り、議論を進めながら合意形成を行う DEOS プロセスのループを回した例を示す。対象のロボットは、日本科学未来館の展示フロア用のサービスロボットであり、ロボットのサービスとしては、展示フロアの中を人や障害物を避けて巡回しながら、来館者との対話、デモの予定時刻と内容の宣伝などを行うことを対象としている。

D-Case は主張を証憑から導く議論を記して、安全・安心を申し立てるアシュランス・ケースと呼ばれる手法の一つである。本技術報告書では既に D-Case の手法は既知の読者を想定しており、詳細については参考文献を参照されたい。このサービスロボットにおいて、ロボットの設計者とサービス提供者が設計から実証に至る合意形成を行うために、機能、運用、安全、説明責任、改善の 5 項目に関して作成した D-Case 木による議論と合意形成について述べる。

図 1　科学未来館内部の様子と、要素機能の検証実験中の様子

2. 対象となるロボットサービスの概要説明

ロボットは展示フロアを人や障害物を避けて巡回しながら、来館者との対話、デモの予定時刻と内容の宣伝などを行うことを目的としている。また安全に配慮するために、スタッフがジョイパッドを持ちロボットの後方から随行し、非常停止やサービス継続のための操作を担当する。

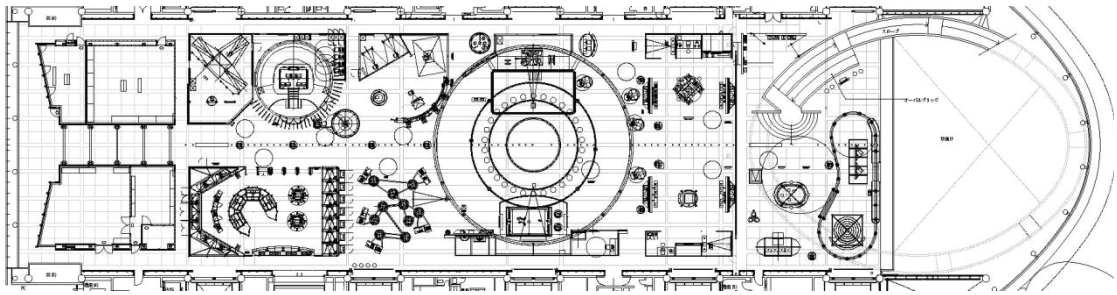

図 2　日本科学未来館の３F 展示フロア図面

図 1 に示したフロアは３階の展示部分約 30 x 130m の領域である。ここに一日当たり１万人程度の来館者が訪れる。

ロボットのハードウエアは２輪駆動のモバイルベース Pioneer3DX をプラットフォームに、多層型のレーザー距離センサ Velodyne HDL-32e と Microsoft® KINECT センサ、マイクロフォンアレイを搭載したものである。人が怪我をしないように、車体にセンサを取り付けたバンパー部と、ウレタンフォームで覆われた車体、および光学センサ部分は透明なシールドにより覆った構造となっている。

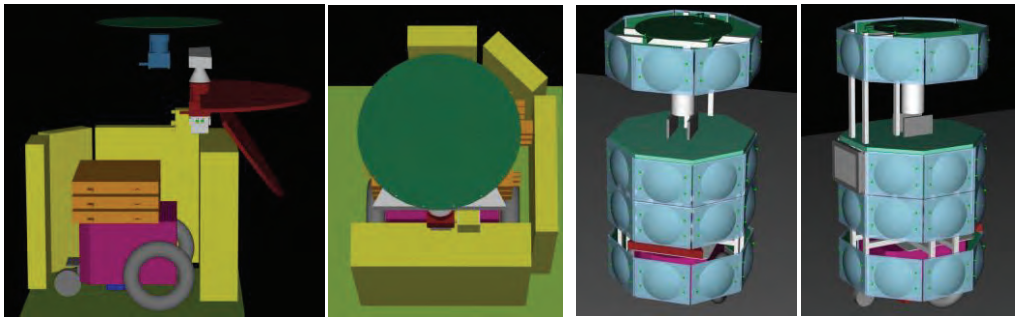

図 ３　議論によるロボットの構成の変化（左右２枚ずつは各々同一）

ソフトウエアは、各種のセンサや機能をロボット用ミドルウエア ROS で統合している。ソフトウエアのモジュールとしては、認識、計画、制御の各部に分かれている。認識部では、多層型レーザー距離センサからの位置認識・地図作成と人などの移動障害物追跡機能、静止障害物発見と走行可能領域認識機能、KINECT センサからは静止障害物発見と人の姿勢検出機能、マイクロフォンアレイからは音源定位・分離・認識機能などを行う。計画部では地図上で認識した自己位置から目的位置まで、障害物を避け、安全性を高く保ちながらなるべく早く目的地に移動できるような経路を計画する。計画部では移動する人の進路を監視し、必要な場合には適応的に回避する経路を計画する。制御部では計画した経路に沿うように車体を制御しながら、静止障害物や移動障害物との相対速度を監視し、速度を適切に制御する。

3. 作成した D-Case 木の解説

作成した D-Case は、機能、運用、安全、説明責任、改善の 5 項目に関して議論を行ったものである。要素数はゴール 71、エビデンス 37、ストラテジ 31、コンテクスト 15、アンディベロプド 5、モニターノード 17 の合計 176 ノードである。

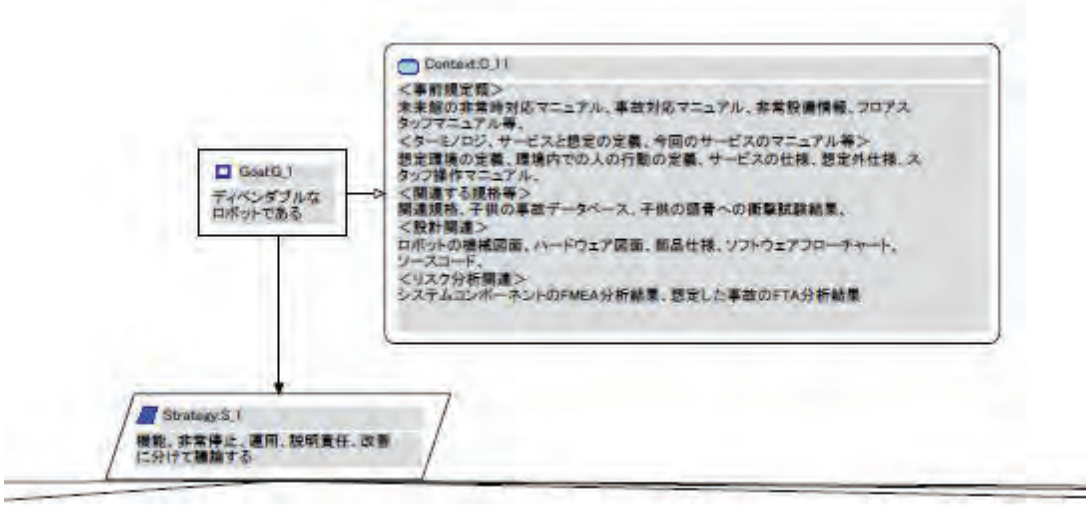

図 4　トップゴールとそのコンテクスト

3.1. 機能に関する議論

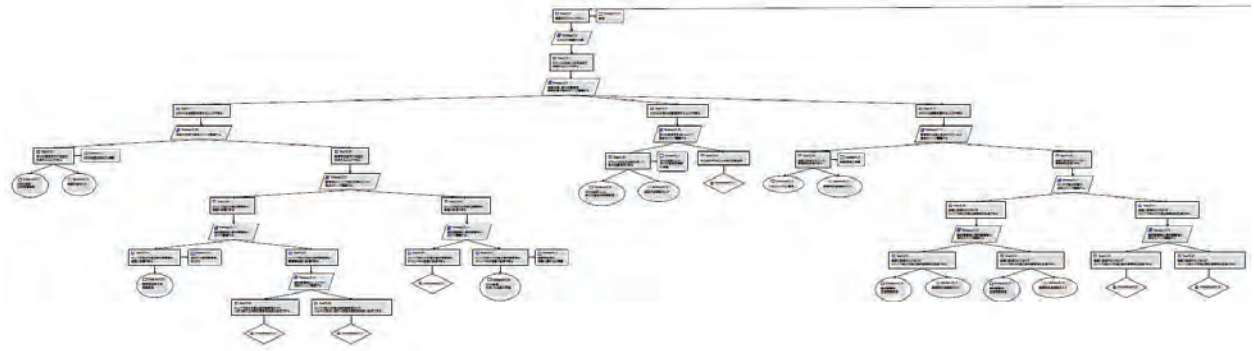

図 5　機能の議論

認識・計画・制御部のそれぞれの機能についてロボットのサービスの観点からの議論を行った。認識部においては、下記の議論を行った。

- **フロアの地図が精度良く作成できる**
- **地図を用いてその中で精度良くロボットの位置が認識でき、人に取り囲まれてセンサの視界を一定程度塞がれていても、その位置認識の最大の誤差が許容範囲内である**
- **椅子、掲示板、子供用のスツールなどの静止障害物の定義と、それを精度良く発見して地図に登録できること。また発見できない床面上の静止障害物の最大値に乗り上げた時に、ロボットが転倒などの事象が発生しないこと**
- **人などの移動障害物の定義と、それを精度良く地図上に記載でき、隠れが生じた時にもロバストに追跡でき、複数人が隣接しているときにも精度良く別々に推定できること**

- 人の姿勢の定義とこれを精度良く検出できること、および見えない部位の推定結果の取り扱いなどの議論を行った。

計画部においては、下記の議論を行った。

- 得られた地図情報を入力として、与えられたゴールに対して、安全性が高くなおかつ可能な限り短い時間で到達可能な経路を計算コストを小さく導き出せること、
- 巡回経路にたいして、来館者が経路上に留まっているなどの理由で経由点に到達不可能な場合には、そこを避けた新規のゴールを設定できること、
- 移動障害物や静止障害物を新たに発見した場合にはそれを地図に登録しながら新たな経路をオンラインで計画出来ること。

最後に制御部においては、下記の議論を行った。

- 与えられた経路に対して、認識した自己位置とその誤差の共分散行列を基に、ロボットの旋回と前進の精度を決定し、精度良く追従すること、
- 検出した、または地図に記載している静止障害物からの離隔距離を一定値以上に保つこと、
- 認識した移動障害物の将来の進路を損なわないように、ローカルな回避行動を行うこと

これらの議論においては、言葉の定義、その原理、ソフトウエアの仕様などがコンテクストとして示され、一方で実際の動作試験による性能評価結果などがエビデンスとして示されることにより、スコープの明示化が行われている。

エビデンスを獲得するために毎週１回、閉館後の実際の環境を利用して実験走行を１年間行い、ソフトウエアの入力と出力をモニターノードを利用して記録した。ART-Linux の複数コア利用機能により、任意の数のモニターノードをシステムの性能低下を恐れずに配置することができる。これらのモニターノードは、実際の運用時には説明責任や改善の議論を行うために利用される。

一方で想定する想定外についての議論を行った。検出不能な物体に関する議論を例にとると、

- 他の障害物の背後から飛び出してきた場合などで、観測時間が短すぎて進路が推定できない場合、
- 対象の光学特性が全反射あるいは無反射に近く、観測が難しい場合
- 対象が小さすぎる、あるいは細すぎて観測が難しい場合

これらには、発見不可能な対象物であるとして、スタッフにより地図への手動での登録を行うなど運用の議論により解決するもの、バンパーなどの安全の議論により解決するものなどの議論を行った。同様に計画や制御レベルでの想定外の定義と、その対応に関する議論も行った。

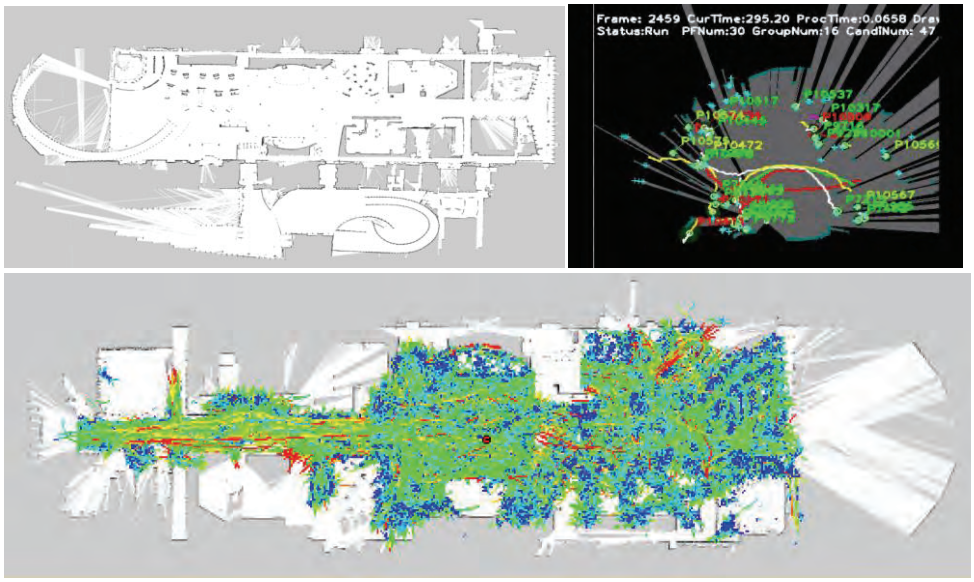

図 6　ロボットが取得した二次元地図（左上）、ロボットの周囲の人流計測結果（右上）、および館内の 1 日の人流解析結果

3.2. 運用に関する議論

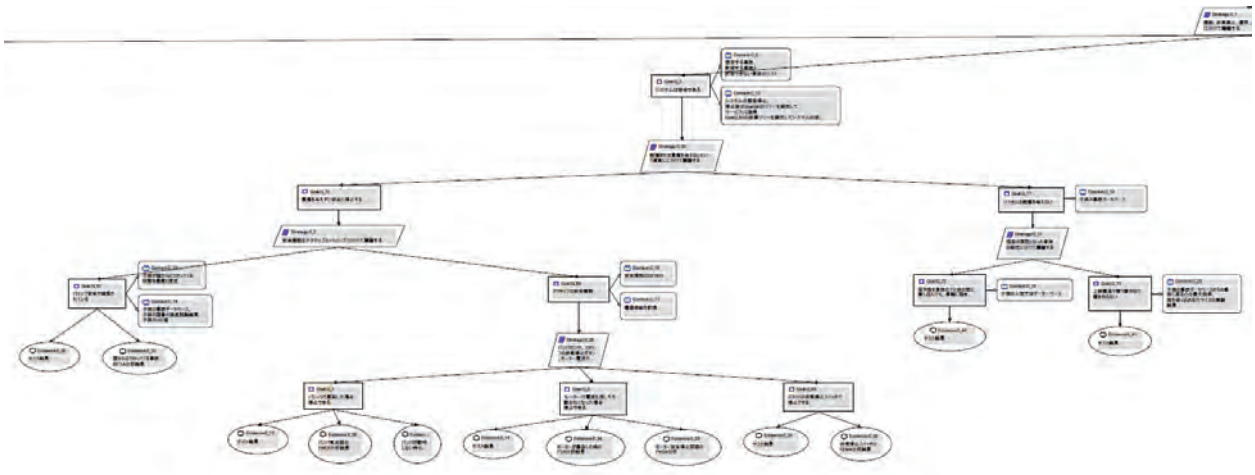

図 7　運用の議論

日本科学未来館では、フロアに専任のサイエンスコミュニケータとボランティアの説明員を配置している。これらのスタッフと、運用に係る担当者、および来館者が直接のステークスホルダーとなる。運用の方法を決定するために、運用に関する議論を行った。

- 安全にかかわる部分および機能に係る部分における、システムの想定する、あるいは想定外のエラーに関して、サービスを継続するための介入のタイミングと方法、
- 事故が生じた際の対応

議論のスコープとして、コンテクストに記載した運用のマニュアルを作成し、これを引用することにより、議論とマニュアル間の対応を取った。またこのようなスタッフの介入に対してシステムがプリエンプティブに応答する仕様の議論を行い、実際にシステムの動作試験の結果がエビデンスとして付随する。

運用における重要な問題として、事故などの想定外が発生した場合や、サービスが継続できない場合には、変化対応サイクルでの改善を行う手順が必要となる。これの事象に関する議論もあわせて行い、こちらもマニュアルに記載した。

3.3. 安全に関する議論

ロボットの安全については、パッシブおよびアクティブな安全について議論を行った。特にパッシブな安全装置としては、衝突が避けられなかった場合に、システムが停止することについて議論している。これはバンパスイッチおよびモーターの電流センサとエンコーダを用いた衝突検出システムを用いており、これらはロボットのソフトウエアシステムからは独立なハードウエアとして準備されており、電源の供給を前提に、ロボットが何かに衝突した際に人にダメージを与えずに安全に停止する。また衝突が起きた時に人間に及ぼす被害を最小化するためのエラスティックな緩衝材が存在する。

この議論のコンテクストとして機能安全に関する基準や、ロボットの安全基準、想定されるリスクと想定しないリスク（例えば人がロボットの上に落ちてくる等）、子供の頭部への衝撃と障害の関係の知見などをスコープに入れて議論を行った。個々のシステムに関しては FMEA 分析による故障モード解析の結果や、実際のシステムが設計通りに動作するかどうかの試験結果、緩衝材の硬さ試験の結果がエビデンスとして付随する。

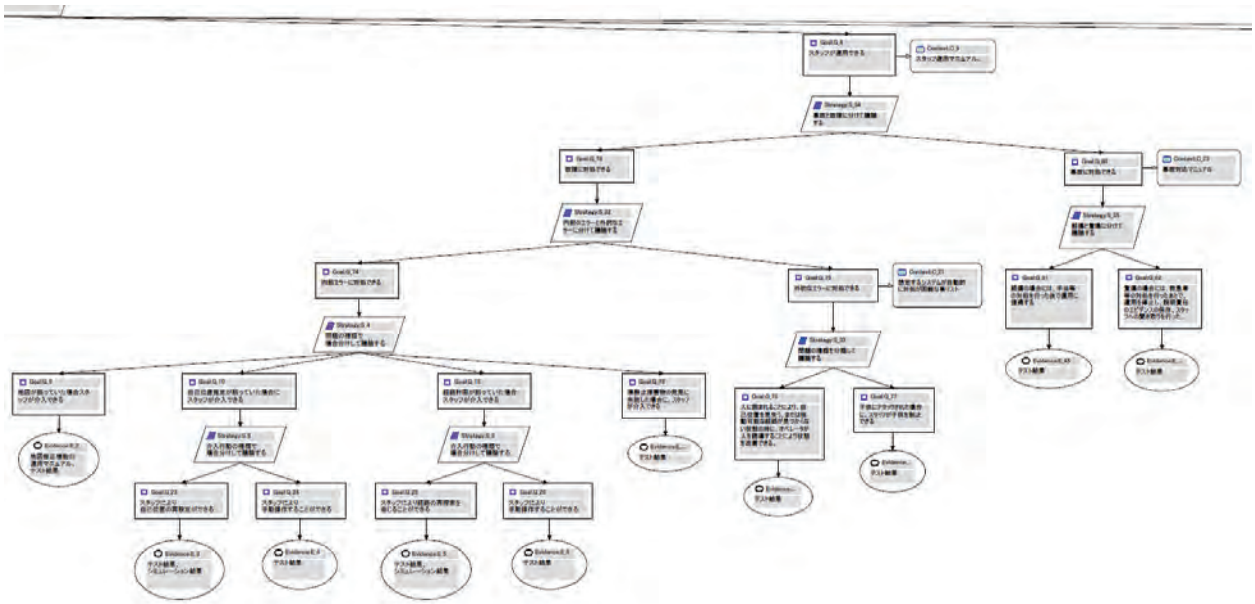

図 8　安全の議論

説明責任に関する議論

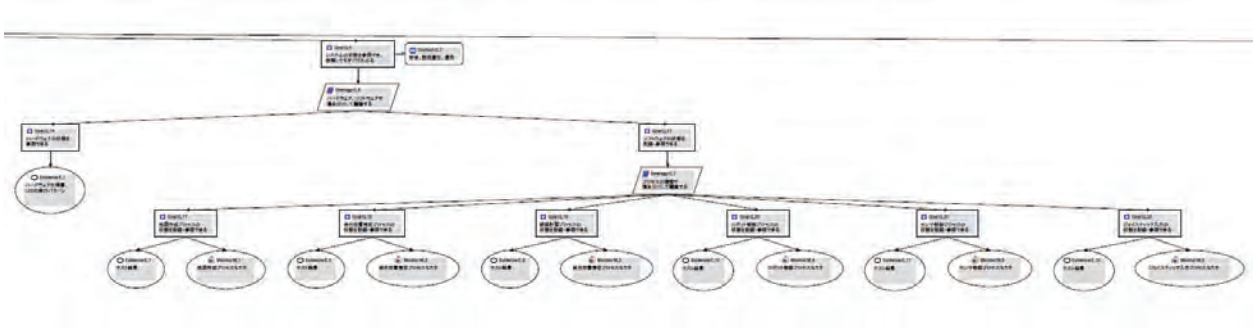

図 9　説明責任の議論

システムが正常に動作していること、想定外に係る何らかの異常が発生したことなどを示すためには、システムの個々の要素の入力や出力の記録が実時間で記録され、保存されていることが重要である。ART-Linux の D-RE 機能により、モニターノードのデータを共有メモリにコピーだけで、別プロセッサ上で実行されているロギングシステムがデータを保存する。このために必要に応じてデータを記録することができる。本ロボットでは、毎秒数十 MB の圧縮データを本システムを通じてモニタしながらロギングする。システムの個々の要素や、外界の現象において異常の発生を検知すると、モニターノードとして設置したデータ収集システムがアラートを発行し、システムは安全に停止状態に移行すると共に、その記録を保存することにより、何が起きたか、どの要素が異常を示したのかを示すことができ、改善のループに移行することができる。

　このためにどのような異常を想定し、記録を行うかの議論を行い、すべてのセンサやモーターへの入出力、各機能毎の入出力、運用担当者の持つジョイパッドの操作入力など合計１７か所のモニターノードを配している。各異常に対して正常に記録が取れることを示したテスト結果がエビデンスとなっている。

3.4. 改善に関する議論

システムが想定外の事故を起こしたり、事故に至らないまでも異常な挙動を起こしたり、センサやモーター等のサブシステムが故障したりした場合において、現場の運用担当者によるオペレーションだけではサービスの継続が不可能な場合には、サービスを停止し、変化対応サイクルと呼ぶ改善のループに入ることになる。故障、想定した想定外、想定しない想定外などの個々の事象について議論した後で、故障と想定した想定外に対しては、前述のモニターノードにより何が起きたのか再現し、システムの改善等の対処を行う。一方で想定しない想定外に対しては、モニターノードが適切に配置されているかどうかは不明である。各種センサの記録、追従していたスタッフからの聞き取りなどから現象の再現を試み、改善を試みるという議論が

なされている。

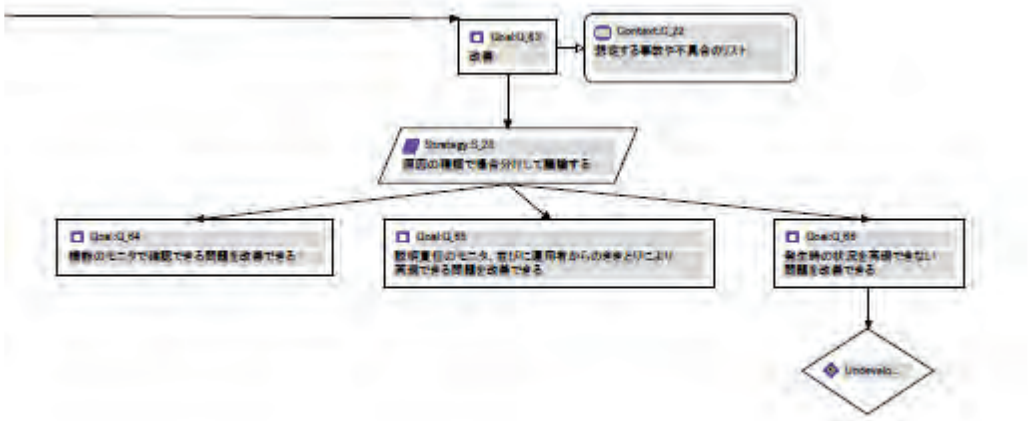

図 １０　改善の議論

4. D-RE 機能を提供する実時間 OS：ART-Linux

実世界に対応するロボットでは、複数の固有周期を持つ実時間サイクルをジッタ少なく実行できる必要がある。典型的なロボットの制御ループは 1ms の周期を持ち、それに対してせいぜい数%のジッタしか許されない。そのため非実時間の Windows や Linux のような OS では不十分である。このような目的のために、実時間処理に特化した組み込み目的の OS として VxWorks や QNX のような専用設計の OS が存在する。

一方で D-Case のモニターノードをシステム内に多数配置することにより、通常のシステムに比べてデータのコピーやモニタリングのオーバヘッドが発生する上に、異常検出のためにはモニタリングのレイテンシも最小に抑えたいと言う要求がある。

我々は知能ロボットの研究に重要なライブラリやミドルウエアの存在、デバイスへのサポートの多さなどの利点を鑑み、Linux をベースにして、実時間処理の制約を満たした ART-Linux と呼ぶ実時間 OS を 1998 年より開発してきている。ART-Linux は Linux をベースとした実時間 OS であり、ユーザー空間のプログラムに対して実時間システムコールを提供することにより、10us オーダーの実時間性能を提供している。本プロジェクトでは、D-RE 機能を提供するために、複数の CPU コアを、いくつかのコアにバインドされた実時間 AMP(Asymmetric Multi-Processing)と複数の CPU コアを利用する通常の非実時間 SMP(SymetricMulti-Processing)の Linux が任意の割合で混在可能な ART-Linux を設計し、開発を行った。このようなシステムは、非実時間 SMP システムの側でオープンソースの汎用ソフトウエアやデバイスを利用できる一方で、実時間 AMP の側では、専用の IO を利用しながら、制御系、安全系、監視系、高信頼のための二重系などのディペンダビリティー機能を、お互いに非干渉な形で独立に実装できるという利点がある。

ART-Linux を用いた D-System Monitor は本システムの持つ共有メモリを利用して、独立した CPU に配置された各サブシステムから、システムのデータをモニタリングしたり、記録を

保存することが可能であり、即時性、耐事故性などに優れている。

図 11 に、開発した ART-Linux により可能となるシステムの構成例を示す。この図では現在利用可能な CPU の例として 8 個のコアがあるものを挙げ、そのうちの 3 個で通常の非実時間の Linux が動作し、また残りの 5 つのコアに対してそれぞれ独立に実時間 OS である ART-Linux が動作している例を示している。

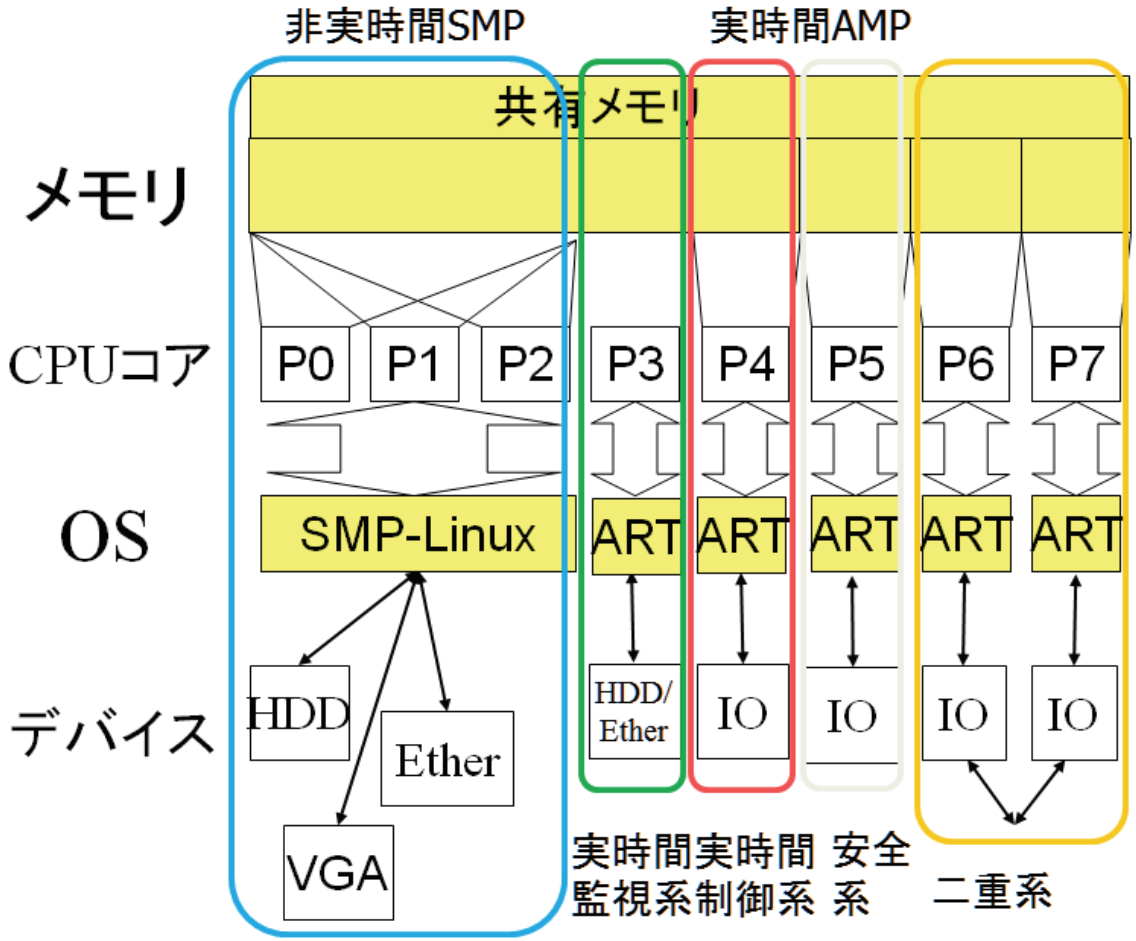

図 11　ART-Linux 構成例

非実時間 SMP-Linux: 図 11 では、P0～P2 のプロセッサに非実時間の SMP システムを配置した例を示している。図中に示すように、SMP の Linux にはデバイスとして通常のディスクやグラフィックス、ネットワークなどをアサインしている。このシステムの役割としては下記のようなものが挙げられる。

· ユーザーインターフェース等。

· グラフィックハードウエア、イーサネットを始めとしてロボットの直接制御に用いない IO を行うハードウエアの処理。これらは多数の割り込みを発生させることから実時間システムにとっての障害となりやすい。

· 長期のプランニング、地図作成、モデルマッチングによる認識処理などの、ロボットの知能処理のうちでも非実時間の処理を行う。これらの結果は、仮想ネットワークあるいは共有メモリを通じて、実時間制御系などに伝える。
· 実時間用途に開発されていない多くのオープンソースライブラリの利用。
本システムはメモリアサインメント部を除けば、ほぼ通常の SMP-Linux であるために、通常の Linux のために開発されたものが、改変なしにそのまま動作するという特徴がある。

実時間監視 AMP-Linux: 図 11 では、P３ のプロセッサに AMP 実時間システムを配置した例を示している。この実時間監視系はログを保存するためのディスクや外部のシステムに警告するための別のネットワークデバイスをアサインしている。
　本システムは他の SMP&AMP システムの内部状態を共有メモリを通じて監視し（他のシステムは自分で共有メモリに状態を書き出す必要がある）、ログを残すとともに、他のシステムの異常をリアルタイムで検知し、後に述べる非常系に知らせるなどの機能を果たす。

実時間制御 AMP-Linux: 図 11 では、P４ のプロセッサに AMP 実時間システムを配置した例を示している。制御のために必要な IO ボードをアサインしている。システムはマルチコアの他のプロセッサを利用することにより、他のオーバーヘッドがなく、制御に専念することが出来るために、低いジッターによる高い実時間制御性能や、リソース配分の簡単化によるシステム設計の容易さと障害の起きにくさが期待できる。次に述べる実時間監視系のために、内部状態を共有メモリに書き出すことによりシステムの監視をオンラインで行うことができる。

安全用 AMP-Linux: 図 11 では、P5 のプロセッサに AMP 実時間システムを配置した例を示している。安全系も独自の IO システムをアサインしている。前述の実時間監視系を通じて、あるいは共有メモリや仮想ネットワークを通じた他システムからの通知や他システムの監視、あるいは IO を介して得られるセンサの値などから、システムの異常を検知し、モーター電源断を始めとする緊急停止などの安全策を行う。
　このシステムを他のシステムから独立にすることにより、非常システムが他の実時間システムに阻害されて動作しないなどの影響を避けることができる。また一方で、例えばヒューマノイドロボットの歩行、また高速で走行している車、アームで人の上に重量物を持ちあげている、などのいきなり電源断すると被害が大きくなるような状況では、その状況を回避するための計算機能と能力を有している必要がある。そのような状況に備えて、実時間計算能力を他のシステムから独立に温存することが、本システムの目的である。

二重系用 AMP-Linux:　システムが複雑になり、実世界において安全に行動するためには、二重系や多重系による信頼性の確保が重要になる。この際にハードウエアを含めた二重系を構成する方法は良く用いられているが、本システムでは OS を含めたソフトウエアをアイソレーションして実現する二重系が構成可能である。
　図 11 では、P6～P7 のプロセッサに同一の AMP 実時間システムを配置している例を示している。２つのシステムは、それぞれ独立に IO ボードをアサインすることも、IO ボードへ

のアクセスを排他的に制御することも可能であり、柔軟な二重系を構成することが可能である。（ただし排他的アクセスには IO ボードの制限も存在する）

5. まとめ

図 １２　ロボットが稼働している様子

本報告書では、具体的な事例として日本科学未来館の展示フロアでサービスを行うロボットの障害対応サイクルと変化対応サイクルを司る機能、安全、運用、説明責任、改善の５つの要素の議論と、それを実現するロボットのシステム構成について、設計者と運用者の２者のステークホルダ間で合意を形成するために議論した D-Case 木について述べた。これらの議論は、詳細においては具体的なセンサやアクチュエータなどの部品や、具体的なレギュレーションなどが表れてくるため、本技術報告書では議論の大枠を示すにとどめているが、スコープ、レギュレーション、エビデンス、設計書、試験結果、マニュアルなどを統合した議論が行えるという特性を示せていると考えている。

またこれらの議論は、設計から運用まで、システムのライフサイクルを通じて常時更新されていくと共に、その動作の記録としてのモニターノードが常時記録され、不具合あるいは更新の対象の議論を行うことができるようになっている。

D-Case 木により議論を行いながら合意形成を行うことには、次のような利点が存在する。

- 問題全体のスコープの明確化
- 障害対応サイクルと変化対応サイクルの遷移条件の明示化
- 議論に用いる資料の明示化
- 個々の FTA,FMEA 等の望ましくない事象のリスク分析結果の統合
- 既存の安全や技術規格との適合判定との統合
- 個々のゴールに対する問題設定と合意条件の議論の記録
- モニターノードにより得られる説明責任情報の明示化
- 運用、整備などのマニュアルの明示化
- 想定する想定外の明示化

- ターミノロジやパラメータの定義の明示化
- 精度などの条件の明示化

一方で、D-Case 木の議論の完全性、木の部分改変における木全体の一貫性の保証、各ゴールに誰が合意したか、上のゴールに合意するための下の方の木の理解、などは利用者に任されており、使用上の注意が必要である。（このうちいくつかの課題は D-ADD により解決されている）

また今回のサービスでは運用と改善のフェーズ間の遷移はスタッフが行うことになっており、また障害対応サイクル内での対応はすべてプログラミングされており、これらを補助する D-Script を用いていない。D-Script を用いることにより柔軟な制御が可能になると期待されている。

参考文献

[1] Tokoro (ed), Open Systems Dependability - Dependability Engineering for Ever-Changing Systems, CRC Press, 2012.

[2] D-Case 入門 ~ディペンダビリティ・ケースを書いてみよう！~松野裕、高井利憲、山本修一郎 ISBN 978-4-86293--079-8

[3] Yutaka Matsuno, Jin Nakazawa, Makoto Takeyama, Midori Sugaya, Yutaka Ishikawa: Towards a Language for Communication among Stakeholders. PRDC 2010: 93-100

[4] D-Case Editor http://www.dependable-os.net/osddeos/index.html

[5] White Paper（DEOS プロジェクト文書） DEOS-FY2011-WP-03J ：DEOS Project White Paper Version 3.0

[6] ART-Linux http://sourceforge.net/projects/art-linux/

[7] 複数コアを SMP・AMP 分割利用可能な ART-Linux の設計と開発、加賀美 聡, 石綿 陽一, 西脇 光一, 梶田 秀司, 金広 文男, 尹 祐根, 安藤 慶昭, 佐々木 洋子, サイモントンプソン, 松井 俊浩、第 17 回ロボティクスシンポジア講演論文集, pp.521--526, 山口県萩市, 萩本陣, Mar., 2012.

[8] シングル CPU 用 ART-Linux 2.6 の設計と開発、石綿 陽一, 加賀美 聡, 西脇 光一, 松井 俊浩、日本ロボット学会誌, 第 26 巻, 6 号, pp.546--552, Sep., 2008.


Dependable Embedded Operating Systems
Dependability Engineering for Open Systems
DEOS

DEOS プロジェクト