

エージェントベースシミュレーションを用いた CPSの検証環境の構築

○齋藤 健太郎 (東京工業大学) 市川 学 (芝浦工業大学)
原田奈穂子 (宮崎大学), Chang Shuang, 出口 弘 (東京工業大学)

Verification of Cyber Physical System using agent based simulation

*K. Saito (Tokyo Institute of Technology), M. Ichikawa (Shibaura Institute of Technology),
N. Harada (Miyazaki University), C. Shuang, H. Deguchi (Tokyo Institute of Technology)

概要— 2010年代のICTの急速な発展により、モノがインターネットに接続可能になるIoT(Internet of Things)デバイスが増加している。IoTデバイスを利用してサイバー空間と現実世界が複雑に絡み合ったCyberPhysical System(CPS)が構築されている。しかしCPSの機能が正確に稼働することを評価する検証をCPS全体に行うことは難しい。そこでCPSが稼働すると思われる環境を現実世界に実装したノードとシミュレーション上に展開したものを合わせた環境で構築することを考える。この手法をパーシャルデプロイメントと呼ぶ。本研究ではこのパーシャルデプロイメントにて作成した環境上で、被検証CPSである避難所情報マネジメントシステムを検証する。これによりISO/IEC25010の品質モデルのうちどのような性質があるかを評価し、パーシャルデプロイメントによる検証方法について論ずる。

キーワード: CPS, エージェントベースシミュレーション, IoT, パーシャルデプロイメント, システム検証, マイクロサービス

1 序論

2010年代から話題になってきたIoT(Internet of Things)は現在になってもその勢いに衰えはなくむしろ増加している。この理由として社会的な面では、Society5.0やIndustry4.0など様々な国でIoTに関する政策が打ち出されていることが挙げられる。また技術的な面では、多種多様なIoTデバイスが益々廉価化していることがある。このような理由で我々の社会にIoTデバイスが入り込んでいき、サイバー空間と現実世界の境界が複雑になっていくと思われる。その結果、サイバー空間の処理(コンピューティング)と現実世界の物理的処理の境界も曖昧になり、これらを統合されたものとみなすことをEdwardら¹⁾によればCyberPhysical System(CPS)という。CPSの考え方に則り、現実世界の情報をIoTデバイスで収集し、サイバー空間上でデータ処理をすることで得られた結果を元に、正確な制御や高品質なサービスの提供が期待されている。CPSの事例としてはAmazonGO²⁾やスマートグリッド³⁾が挙げられる。

CPSも情報処理システムと同じように、正しく稼働するかについて評価する検証が必要である。しかし実際のCPSが導入先の環境にて仕様通りに正確に稼働することについての検証は困難であると思われる。この理由として、稼働する環境の構築とパラメータ調整に莫大な手間がかかると思われるためである。前者については、1テストのためにIoTデバイスを用意し配置をすることは小規模であれば可能であろうが、少しでも規模が大きくなってしまうと現実的に行うことは不可能であると思われる。後者についてはたとえテスト用の環境が構築できたとしても、温度や湿度などの現実世界の環境を変化させ100以上ものテストケースを実行することは非常に時間がかかり現実的ではないと考えられる。このような理由でCPSの検証は難しいと言えるのだが、我々の社会は情報処理システムに対す

る依存度を高めており、そのシステムが正しく稼働するかどうかという検証の重要性は上昇している。そこで現在のCPS検証方法例を2つ取り上げる。

先行検証事例1: スマートパーキングシステム

Fraiferら⁴⁾はCCTVカメラを用いたスマートパーキングシステムを提案し、その検証をした。このCPSは、駐車場において空いている駐車スペースを探すために無駄な燃料や時間を消費しているという現状を解決するために考案された。駐車場にカメラを取り付け、そのカメラが送信してきた画像を解析することで空きスペースを認識する。この空きスペースの情報を探している車のドライバーのスマートフォン等に通知することで、現状の解決を試みたCPSである。このCPSの検証は、車の小型模型を用いて、スケールダウンした駐車場を再現することで行われ、機能が正しく動くかどうかの評価をした。

先行研究事例2: エネルギーマネジメントシステム

Alら⁵⁾は、電力センサーを用いたエネルギーマネジメントシステムを提案し、その検証をした。このCPSは著者の地域では、非常に高温で空調管理のために用いられる電力が多く、電力マネジメントを適切に行うことへの重要性が背景にある。そこで各家庭に設置した電力センサーから得た値を解析することで、気温と消費電力の関係など統計情報のグラフ等を作成しそれを表示するCPSを構築している。このCPSの検証はAlらの研究室に電力センサーを設置し、そこで機能が正しく稼働するかを評価している。またPub/Sub通信に用いるブローカのテストを別途に行い多対多通信のスケラビリティを利用して、研究室だけでなく他の場所を混ぜた稼働予定の環境でも正しく稼働すると論じている。

事例1はCPSが稼働する環境を小型模型で再現することで検証し、事例2は各サーバの評価を別途に行い、稼働予定より規模が小さいシステムで検証した。そし

てシステムの拡張性から稼働予定規模のシステムでも機能が正しく稼働すると述べた。事例1では稼働予定の環境よりノード数が不足しているという課題が存在する。システムノード数が不足していることで、例えば全てのノードから同時にデータが来た時にどのようなかについて懸念がある。事例2ではブローカへのテストは行ったが、そのブローカに関する単体テストをただけであり、システム全体をテストしたとは言い難い。では現実世界にセンサー等を設置する手間やパラメータ調整の手間を減らすために、CPSが稼働する環境をシミュレーションで再現することを試みる。シミュレーションで再現できれば、どのようなノード数でも作成が可能になり様々な手間を取り除けられるため、CPSの検証が可能になると考えられる。このためにはシミュレーション上の環境とCPSをうまく統合する必要がある。これはCPSのアーキテクチャをPub/Sub通信⁶⁾を用いたマイクロサービスアーキテクチャ⁷⁾にすることで、エージェントベースシミュレーション⁸⁾を利用することで可能になる。このシミュレーションと現実世界を合わせた環境でCPSを構築することをパーシャルデプロイメント⁹⁾と呼ぶ。本研究ではパーシャルデプロイメントを用いてCPSを構築し、そこでCPSを検証する。ISO/IEC25010の品質モデル¹⁰⁾のうちどのような性能が評価できるかを論じ、パーシャルデプロイメントを利用したCPS検証方法について述べる。

2 パーシャルデプロイメント

出口⁹⁾によるとシステムのノードの一部をシミュレーション上のノードで代替したもので実装することをパーシャルデプロイメントという。パーシャルデプロイメントを実現するためには、エージェントベースシミュレーションを用いることと、Pub/Sub通信⁶⁾を用いたマイクロサービスアーキテクチャ⁷⁾が必要である。エージェントベースシミュレーションとは、システムを独立して意思決定できるエージェントとその相互作用から構成されるものとし、このモデル上でエージェントに意思決定をさせシミュレーションをするものである⁸⁾。エージェントベースシミュレーションによって経済、人口動態、感染拡大、人流といったもののシミュレーションが可能になり、エージェントに対する政策によってどのようにシステム全体が変化するかを分析することが可能になる。このエージェントベースシミュレーションの応用として、市川ら¹¹⁾は教育等に利用されるゲーミングシミュレーションの環境の構築を行なった。これはエージェントの意思決定をエージェントが持つウェブサイトから人間が操作できるようにすることで実現された。つまり人間とエージェントを入れ替えることで参加型シミュレーションを行なった。そこで人間だけではなくモノやソフトウェアもエージェントと入れ替えれば、ヒト・モノ・ソフトウェアの参加型シミュレーション環境が構築できる。これを可能にするのがPub/Sub通信を用いたマイクロサービスアーキテクチャである。システムを小さいサービスを提供するもの(マイクロサービス)の集合体とみなし、その相互通信にPub/Sub通信を用いるものである。これによってシステムのノードは自律的に意思決定をするものとみなし、市川らのゲーミングと同じように、モノやソ

フトウェアと、入れ替える先のワークフローを擬似的に実行するエージェントとを入れ替えることが可能になる。これによりシステムの一部をエージェントに代替したもので実装することが可能になり、実際に稼働する規模のシステムが構築できる。

例えば災害避難所に支援内容を提案するCPSを導入し、その検証のプロセスについて述べる。本検証に用いるパーシャルデプロイメントCPSはFig.1のようになる。避難所では現実世界に実装されたセンサーとアセ

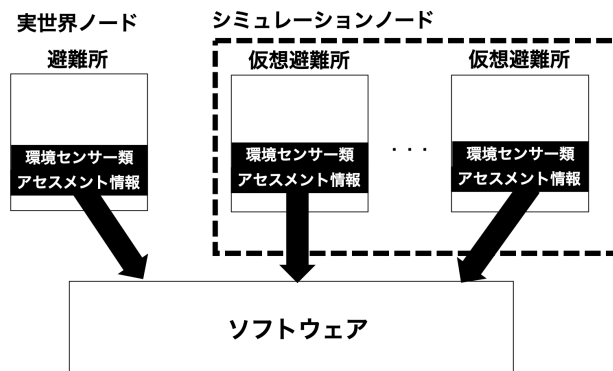


Fig. 1: パーシャルデプロイメント

メント者の持つデバイスがPublishをしている。また仮想避難所はシミュレーション上の仮想センサーやアセスメント者によるPublishが行われている。この環境でソフトウェアを稼働させることで、CPSが意図した通りに正しく動くか、すなわちCPSの機能や性能を検証する。検証する内容として、例えば支援内容として受け取った避難所のランク付けを行ない一番環境が悪いと思われる避難所に対して有効な支援案を推薦するシステムを考える。仮想避難所の値はランダムで決定されるとして、得たデータをもとに人間によって検討された結果と実際に出力されたものを比較することを繰り返せば良い。また1つの避難所の持つ値として、ランダムではなく2因子間を網羅するような組み合わせを選択するようなアルゴリズムをシミュレーションに導入しておくことで、組み合わせテストができる。これによりソフトウェアが正しく稼働するかの評価が正確にできる。このCPSは事例1のように模型で表すことが難しく、また実災害時は1医療圏に最大300箇所以上の避難所が開設する可能性があるため事例2のように小規模なCPSでの検証では想定外のエラーが発生する懸念が残る。しかしパーシャルデプロイメントを利用したCPSを検証することで、実災害規模での検証が可能になり、より多くのエラーを洗い出せる。

3 実装

3.1 実装システムについて

本研究でパーシャルデプロイメントを利用したCPSを検証するにあたり、被検証CPSである避難所情報マネジメントシステムの概要を示す。避難所情報マネジメントシステムは、避難所情報収集機能、表示機能、分析機能、メッセージング機能を持つPub/Sub通信を用いたマイクロサービスアーキテクチャで構築される。

3.2 システムの目的と背景

避難所情報マネジメントシステムの目的は、避難所アセスメントの結果をまとめ分析結果を提供することで、災害対策の1つである避難所への支援をサポートすることである。この目的設定の背景は避難所の環境整備にはアセスメントの実施が必須であるが、そのアセスメントを定期的に数百の避難所で行い、結果の入力分析の全ての過程を人の手で行うには限界があったという教訓がある¹²⁾。そもそもなぜ避難所にて健康を損ねる者が出るのかについては、暑さによる熱中症、睡眠不足による体力減少、トイレの不衛生を理由とする感染症や、複合的なストレスによる食欲の減少など様々な要因で体力が失われ、中長期的には災害関連死という帰結を招きかねないからである。特に急性期では混乱しているため、災害関連死を招きうる重要な因子のみを調査するラピッドアセスメントが行われる。このような背景の下、避難所情報マネジメントシステムに求められる機能とは、避難所の重要な因子をすばやく集め、災害関連死を招きうる重要な課題を分析し、まとめて表示し自治体防災の意思決定を支援すること。そしてそれを現場の支援者へ迅速に伝達することである。

3.3 システムの制約条件

避難所情報マネジメントシステムは災害時に稼働するシステムであり、CPSを含む一般的な情報処理システムとは異なる部分が多く存在する。そのため守らねばならない制約条件が生じ、これらは避難所情報マネジメントシステムの機能要件や非機能要件の定義について重要である。畑山ら¹³⁾によれば、災害時に用いられる地理情報システムの満たすべき条件は以下の5つである。

1. 平常時に使用していること
2. 専門家でなくとも使用ができること
3. 可搬型であること
4. 複数システム間の情報統合ができること
5. 最新の地域データベースが作れること

本研究で用いる避難所情報マネジメントシステムの制約条件もこれに則ることとする。ただし本システムでは地理情報システムは用いないため、「最新の地域データベースが作れること」は除くことにする。また畑山らの頃より通信技術が発達し、災害時では00000JAPAN¹⁴⁾といった無線LANサービスが行政から提供されることは避難所情報マネジメントシステムにとって重要な点である。

3.4 システムの機能

システムの目的から避難所情報マネジメントシステムに求められる機能を定義する。避難所情報マネジメントシステムでは、避難所情報収集機能、避難所情報分析機能、WEBサイト構築機能、メッセージング機能の4つを定義した。

避難所情報収集機能

この機能を実現するにあたり、避難所は気温・湿度・照度・CO2濃度の物理情報と電気・ガス・水道・食事などのライフライン情報、避難者数・手洗い環境など

の衛生環境情報、インフルエンザや透析患者数などの医療需要情報からなるものとした。そこで物理情報はIoTセンサーから取得される。IoTセンサーは可搬できるようトランクにキットとして包む形で実装した。またセンサーから取得できないものについては人によるアセスメントをする。アセスメント結果をアプリケーションに入力したり、光学文字認識(Optical Character Recognition:OCR)対応の帳票に記入することで収集をする。

避難所情報分析機能

避難所情報分析機能では、受け取った避難所情報を元にルールベースで発生している課題を表示する。今回は分析対象課題として、熱中症リスク、水不足リスク、感染症リスク、睡眠不足リスク、トイレリスク、報告漏れリスクの6つの課題を選択した。各課題の表示条件をTable1に示す。Table.1におけるWBGT¹⁵⁾と

Table 1: 分析サービスにおけるリスク表示条件

リスク名	条件	表示
熱中症リスク	$WBGT(t, RH) \geq 31$	暑さ警戒度: 危険
	$WBGT(t, RH) \geq 28$	暑さ警戒度: 嚴重注意
	$WBGT(t, RH) \geq 25$	暑さ警戒度: 警戒
	$WBGT(t, RH) \geq 21$	暑さ警戒度: 注意
水不足リスク	$t > 30 \wedge RH \leq 40$ かつ「飲料水」がCまたはD	水消費量増大リスクあり
感染症リスク	$t < 20 \wedge RH < 40$ かつ感染症の患者が存在する	感染症蔓延リスクあり
睡眠不足リスク	$t < 15 \wedge lux > 400$ かつ「寝具」がCまたはD	睡眠不足リスクあり
トイレリスク	「寝具」がCまたはDかつ「トイレ掃除」がCまたはDかつ「下水」がCまたはD	トイレのリスクあり
情報収集不安定リスク	以前帳票を受け取った日時と今回の日時が2日以上離れている	情報収集不安定リスクあり

は暑さ指数のことである。そして「飲料水」がCまたはDとは、帳票における飲料水の状況についてのアセスメント結果がCもしくはDの評価であることを意味する。帳票の評価水準は数字、AからDまでのアルファベット、チェックボックスが用いられている。なおこれらの評価基準は、国際的に用いられている難民や災害支援の最低基準であるスフィア基準における関連する基本指標に基づいている¹⁶⁾。この分析機能により課題が抽出されることで、データの使い方がわからない時、すなわち専門家でなくとも重要なものに関しては現場で利用が可能になる。

WEBサイト構築機能

避難所情報収集機能と分析機能で得た結果を表示するWEBサイトを構築する。WEBサイトは医療圏ごとに構築できるようにし、非災害時でも使用できるようにモードによる区別をする。

メッセージング機能

自治体防災からの意思決定を迅速に現場で伝達するために、WEBサイトからメッセージが送信できる機

能を持つ。これによりさらに効率的に避難所の支援ができ、目的達成のために重要な役割を發揮する。

3.5 システムアーキテクチャ

以上の機能を持つものを Pub/Sub 通信を用いたマイクロサービスアーキテクチャで構築する。この理由はマイクロサービスアーキテクチャには技術特異性、回復性、スケーリング、デプロイの容易性等の利点が存在するためである¹⁷⁾。また CPS のシステムアーキテクチャは絶えず変化するため、Pub/Sub 通信を用いることで対応できる点も理由の1つである。例えば避難所情報マネジメントシステムでは避難所情報のやり取りが非同期的に行われるためシステムアーキテクチャは常に変化することが想定される。

マイクロサービスアーキテクチャには上述の利点が存在するのだが、どのようにマイクロサービスに分割するかが課題となる。例えば Microsoft Azure チームら¹⁸⁾によれば、ビジネスドメインから疎結合かつ高凝集性を満たすような境界がその候補になりうるとある。しかし出口¹⁹⁾によれば、マイクロサービスアーキテクチャを用いて CPS を構築するとき、マイクロサービス間のデータのやり取りをデータの流れ(データフロー)として捉え、このフィルターをマイクロサービスの単位にするとのある。避難所情報マネジメントシステムもこれに習い、マイクロサービスをフィルターとみなす。

以上の考えを元に機能をマイクロサービスに分割すると避難所収集機能の部分がセンシングサービス、アセスメントサービス、OCR サービスになり、その他については機能がそのままマイクロサービスになる。これよりデータフローを定義すると、WEB 到達前後でそれぞれ Fig.2, Fig.3 になる またマイクロサービス同士の通信

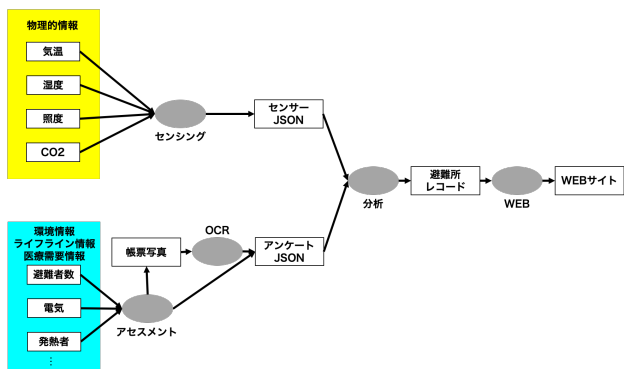


Fig. 2: インput部分データフロー

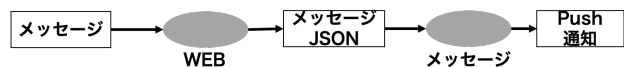


Fig. 3: アutput部分データフロー

に用いられる Pub/Sub 通信のトピック設計を Table.2 に示す。Table.2 におけるヘッダ部とは、"/saito/モードコード/イベントコード"であり、モードコードは、training や disaster が用いられる。またモードコードをデータベースオブジェクトの識別子として用いている。そしてイベントコードは年と通し番号2桁からなる。例えば 201901 である。これはデータベースのコレクションに識別子に用いる。このトピック設計によ

て医療圏で異なるイベントコードを用いれば、それぞれ固有な WEB サイトを獲得できるようになった。また訓練時、テスト時、災害時の振り分けも可能になり、日常的に使用ができるという制約条件も満たされる。サーバサイドの処理をするマイクロサービスに対しては1つのサービスを1つのマシンに実装する。以上の通りに避難所マネジメントシステムを構築するとアーキテクチャ図は Fig.4 のようになった。避難所情報

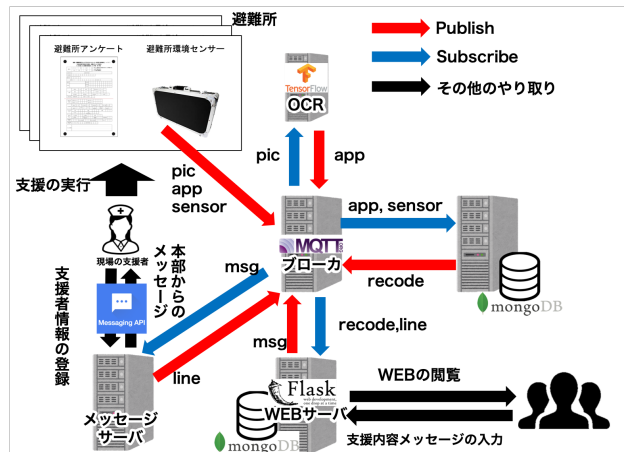


Fig. 4: システムアーキテクチャ

マネジメントシステムでは Pub/Sub 通信の実装プロトコルとして MQTT を利用した²⁰⁾。さらに MQTT ブローカは emqtt²¹⁾を用いている。MQTT のメッセージ部分は、様々なプログラミング言語に対応しているという理由で JSON を採用した²²⁾。そしてデータベースには JSON と相性がよい NoSQL 型の MongoDB を用いた²³⁾。各マイクロサービスは ubuntu18.04 をインストールした仮想マシン(メモリ 1GB, 1CPU)上に実装をした。なおホストコンピュータの性能はメモリ 16GB, 6CPU である。

3.6 マイクロサービス

マイクロサービスの詳細をまとめる。前節よりセンシングサービス、アセスメントサービス、OCR サービス、分析サービス、WEB サービス、メッセージングサービスの6つのマイクロサービスが構築される。

センシングサービス

センシング部分では、(株)パイケーキ作成のトランクキットをベースにしたセンサーキットを利用した。このトランクキットは気温、湿度、照度、CO2 濃度が測定可能である。このトランクキットに他のブローカに Publish するようなプログラムを加えたものを災害避難所情報マネジメントシステムで用いた。このプログラム起動時は避難所ごとに与えられる避難所 ID の入力が必要である。1時間に1回それぞれの物理的パラメータを JSON にまとめて Publish する。

アセスメントサービス

避難所に派遣される保健師等が避難所の状態をアセスメントし、避難所の環境情報・ライフライン情報・医療需要情報が、スマートフォンアプリへの結果入力を通して JSON データに変換され Publish される。もしくはアセスメント結果をアンケート帳票に記入し、その画像データを Publish する。

Table 2: トピック設計

サービス名	PubTopic	SubTopic
センシングサービス	[ヘッダ部]/sensor/[避難所コード]	
OCR サービス	[ヘッダ部]/app/[避難所コード]	[ヘッダ部]/pic/[避難所コード]
アンケートサービス	[ヘッダ部]/app/[避難所コード]	
分析サービス	[ヘッダ部]/recode/[避難所コード]	[ヘッダ部]/app/[避難所コード] [ヘッダ部]/sensor/[避難所コード]
WEB サービス	[ヘッダ部]/msg/	[ヘッダ部]/recode/[避難所コード] [ヘッダ部]/line/[避難所コード]
メッセージサービス	[ヘッダ部]/line/[避難所コード]	[ヘッダ部]/msg/#

OCR サービス

アセスメントサービスで得られたアンケート帳票の画像データを受け取り、それをOCRする。このサービスではOCRモデルに畳み込みニューラルネットワーク(Convolutional Neural Network: CNN)²⁴⁾を用いており、数字・アルファベット・チェックボックスからなる帳票項目を約90%ほどの精度で認識できる。OCR終了後アセスメントサービスのアプリケーションと同じ形式のJSONがPublishされる。

分析サービス

センシングデータもしくはアンケートデータを受け取ることで、同じ避難所IDを持つレコードをデータベースから探し、そのレコードと受け取った情報を合わせて発生課題を分析する。使用するデータベースの選択はトピックによって制御される。分析終了後、データベースを更新したのち最新のレコードデータをPublishする。

WEB サービス

分析サービスから送信された避難所レコードをデータベースに格納しその数値を表示するWEBサイトを構築する。WEBサービスも分析サービスと同様にトピックによる使用データベースの制御を行っており、それがWEBサイトのURLにも利用される。

またメッセージングサービスに送信したいメッセージをPublishする役割も担う。メッセージの送信先に文章を入力することによって、メッセージの送り先IDとメッセージの内容のJSONが作成されメッセージサービスに送信される。

メッセージングサービス

WEBサービスからのメッセージ情報を受け取り、メッセージをデバイスに送信する。本サービスでは、LINEMessageAPI²⁵⁾を用いてPUSH通知を指定された送信先へメッセージを送信する。

4 システムの評価

4.1 評価方法

4章にて述べた避難所情報マネジメントシステムの避難所情報収集機能(センサーとアプリケーション)、避難所情報分析機能、避難所情報表示機能の評価をパッチャルデプロイメントを用いて行う。CPSの評価指標としてはISO/IEC25010¹⁰⁾を用いる。ISO/IEC25010は、a)機能適合性、b)性能効率性、c)互換性、d)使用性、e)信頼性、f)セキュリティ、g)保守性、h)移植性の8つの特性にそれぞれ2個から5個ほどの副特性が

付いたものからなる。ISO/IEC25010はソフトウェアに関する評価モデル指標であるが、Karnouskos²⁶⁾らの議論によると、CPSにISO/IEC25010を用いることは適しているとある。Karnouskosらによると、CPSの評価基準について他に基準となりうる指標は存在せず、さらにKarnouskosらのシステムはソフトウェアエージェントを統合したものであるため、利用できることある。本研究における避難所情報マネジメントシステムはソフトウェアエージェントを統合したものであるため、評価指標にISO/IEC25010を用いることとした。またマイクロサービスアーキテクチャを用いることでc)互換性、e)信頼性、g)保守性、h)移植性については高い評価が期待される。d)使用性とf)セキュリティについては専用のテストが必要であるため、本研究ではa)機能適合性とb)性能効率性を評価することに注目する。

評価は3段階の実験で行う。1段階目の検証(実験1)では、ランダムで値をPublishするエージェントを用い、処理速度やCPU使用率を測定し通常運用時の性能効率性を評価する。また実際に本番に近い状態で稼働してみることで、システムの試験的利用も可能になる。そこで機能がユーザの目的に対して適切であるかどうかを分析し改善点を抽出する。2段階目の検証(実験2)では、1つの避難所で完結する機能である分析機能の検証を組み合わせたテストで行う。組み合わせテストについては2因子間を網羅するテストケースを利用する。最後に3段階目の検証(実験3)では、実験1での追加機能としてのセンサー異常検知と因果関係分析機能を検証する。これらの機能は大量のセンサーデータから知見を得るといふCPSの典型的な機能であり、入力の組み合わせが指数オーダーになる。これらの機能もエージェントの構成を工夫して行い、機能が正確に動くことを評価する。

実験1

実験1では、ランダム値を出力するPublishするエージェント299体を稼働させ、現実世界に実装したセンサー系1つを稼働させたパッチャルデプロイメントによるCPSを作成した。現実世界の30秒をシミュレーション世界の1日とした。実験1の目的として、センサーデータがWEBサービスサーバまでに届くまでの処理時間、各サービスが稼働する仮想マシンの平均CPU使用率の性能を評価すること、機能が目的を達成するために不足していることを分析することが目的である。時間計測のためのブローカは芝浦工業大学の市川研究室に設置されたものを用い、その後他のサービスと同様に仮想マシン上に設置したものに移行した。ネット

ワークは一般的な家庭用の LAN を用いた。仮想マシン同士はホストオンリーアダプタにより同じネットワークに接続されている。

実験 2

実験 2 では避難所情報マネジメントシステムの因子水準を洗い出しそのうち 2 因子間の組み合わせを網羅するようなテストケースをエージェントに持たせ、CPS が正確に稼働することを評価する。分析機能のルールベースから避難所情報マネジメントシステムは Table.3 の因子と水準を得る。

しかしこの設定項目の中には到底起こりえない状態も含む。例えば気温が 15 度以下で湿度が 40% 以下であるときに、WBGT 式が 31 以上であるものは存在しない。これを禁則という。この禁則を考慮すると気温と湿度と WBGT 式は合わせて 16 水準となる。この 16 水準を Table.4 に示す。したがって、16 水準 1 因子と 2 水準 8 因子の組み合わせテストをすれば良い。全通りすることを考えると 4096 通りの入力が必要となりテストは難しくなる。そこで 2 因子間を網羅した組み合わせで行うことを考える。Kuhn²⁷⁾ らによれば、1 因子で生じる欠陥数と 2 因子で生じる欠陥数の合計は 70% から 97% であり、2 因子間網羅することでかなりの数の欠陥を取り除けるとある。そこで Pairwise 法²⁸⁾ を用いて 2 因子間を網羅するようなテストケースを作成した。なお作成には Microsoft による Pict²⁹⁾ というソフトウェアを用いた。必要なテストケースは合計 33 項目である。このテストケースを用いて避難所の分析が正確に行われることを評価する。評価のために本実験で用いるプログラムには、特定の 2 因子選択されると表示するはずだった内容を消してしまうバグを意図的に混ぜている。具体的には WBGT 式が 25 以上かつ、情報収集ができていないときである。また比較のため特定の 3 因子が選択されると同様に内容を消してしまうバグを混ぜている。具体的には WBGT 式が 21 未満かつ、照度が 400 より大きく、かつ寝具等が C または D のときである。

実験 3

実験 3 では、実験 1 にて避難所情報マネジメントシステムが目的をより正確に達成するために必要であると判断された機能を検証する。すなわち大量のデータを処理するという CPS に期待されている機能の一例として、センサーの異常検知とアセスメント結果の因果関係の分析機能が正確に動くことを評価する。センサーの異常検知については、他のセンサーから受け取ったデータの統計を取り比較するというホディング理論による統計的な手法を用いた³⁰⁾。この機能は分析サービスに追加した。アセスメント結果の因果関係分析機能はリアルタイムで全ての避難所のデータから相関関係のヒートマップと高い相関係数を持つ組を表示する機能である。これは WEB サービスに追加し、ヒートマップを表示するページを構築するようにした。これらの機能に対する入力、考えられる 1 箇所あたりの避難所のパラメータの組み合わせ数を l 、避難所数を n とすれば、 l^n 通り発生し 2 因子間網羅等行っても膨大な数である。そこでエージェントの構成を調整することでテストシナリオを作成し、それを用いて検証をする。実験 3 ではセンサー異常を温度計にのみ発生するもの

とし、センサーとアセスメント結果それぞれに通常値と異常値を Publish するものの組み合わせ、計 4 種類のエージェントからなるパーシャルデプロイメントによる CPS を構築した。本研究ではセンサー異常を出力するものを 100 箇所につき 1 箇所、アセスメント異常値を出力するものを 50 箇所につき 1 箇所とし、センサー値とアセスメント結果の両方に異常値を出力するものを必ず 1 つは用意するようにした。避難所数の合計は実験 1 と同様に 300 箇所とした。

4.2 検証結果

実験 1

各マイクロサービスの平均 CPU 使用率は約 90% で、実世界に実装されたセンサー値が Web サーバにたどり着くまでの時間は平均 121.81ms であった。パーシャルデプロイメントで代替したノードを実際のモノやソフトウェアに戻しても、CPS 全体から考えると大きな変化はないためこれは本番に近い値であると考えられる。ただし、災害時におけるネットワークに依存する部分が多いためネットワークに関しては懸念が残る。またこの状態で実際に避難所情報マネジメントシステムを使用してみると、ユーザの目的達成のために不足している部分が発見された。これは 300 ある避難所が無秩序に表示されていたため知りたい避難所の情報を得るのに時間がかかること、また -40 度といったあり得ない値も混ざっておりノイズが多かったことの 2 点である。この不足している部分はいわゆる要件定義が足りなかった部分である。そこで新たに機能要件に秩序するように避難所を表示すること、異常値に対応するためセンサーの異常検知とアセスメント結果の因果関係分析の機能を避難所情報マネジメントシステムに加えることにした。

実験 2

当初の意図通り、暑さ指数が 25 以上かつ、昨日未受信であるとき、表示がされていないエラーを検出できた。組み合わせテストができたと言える。一方 3 因子間のエラーであるものは検出できなかった。これは今回のテストケースの 3 因子間網羅率が 27% と低いことが原因である。このパーシャルデプロイメントの CPS でシステムテストができた。

実験 3

最初に検証をしたところ、分析部分のマイクロサービスの計算が追いつかなく、単体テストで評価されたのにもかかわらずシステムテストではうまく機能しなかった。そこで、計算をスレッドにて行うように実装し直したところ、当初の意図通り、異常な温度を Publish する避難所を抽出できた。また因果関係もエージェントに入れたものについては出力できた。

4.3 考察

実験 1 ではランダム値を出力するエージェント 299 体と 1 箇所の実世界の避難所を利用して、避難所情報マネジメントシステムにおける性能効率性の評価と可能な範囲での機能が適切かどうかの評価を行なった。実験 2 では被検証システムの機能から避難所のとりうる値と水準を分析し、2 因子間網羅をするテストケースを用いた CPS の組み合わせテストを行なった。CPS のモジュールを統合した状況で組み合わせテストをする

Table 3: 避難所情報マネジメントシステムの設定項目

気温	湿度	WBGT 式	照度	飲料水	感染症患者	寝具	生活用水	トイレ掃除	下水	昨日受信
$t \geq 30$	$RH > 40$	$WBGT \geq 31$	$lux > 400$	AorB	存在	AorB	AorB	AroB	AorB	受信
$20 < t < 30$	$RH \leq 40$	$WBGT \geq 28$	$lux \leq 400$	CorD	存在しない	CorD	CorD	CorD	CorD	未受信
$15 < t \leq 20$		$WBGT \geq 25$								
$t \geq 15$		$WBGT \geq 21$								
		$WBGT < 21$								

Table 4: 避難所情報マネジメントシステムの気温湿度 WBGT のとりうる組み合わせテストケース

番号	気温_湿度_WBGT 式	(気温, 湿度)
1	$t \geq 30_RH > 40_WBGT \geq 31$	(30,72)
2	$t \geq 30_RH > 40_WBGT \geq 28$	(30,70)
3	$t \geq 30_RH \leq 40_WBGT \geq 31$	(40,20)
4	$t \geq 30_RH \leq 40_WBGT \geq 28$	(40,0)
5	$t \geq 30_RH \leq 40_WBGT \geq 25$	(30,24)
6	$t \geq 30_RH \leq 40_WBGT \geq 21$	(30,10)
7	$20 < t < 30_RH > 40_WBGT \geq 31$	(29,80)
8	$20 < t < 30_RH > 40_WBGT \geq 28$	(29,60)
9	$20 < t < 30_RH > 40_WBGT \geq 25$	(29,41)
10	$20 < t < 30_RH \leq 40_WBGT \geq 25$	(29,39)
11	$20 < t < 30_RH \leq 40_WBGT \geq 21$	(29,20)
12	$15 < t < 20_RH > 40_WBGT \geq 21$	(20,80)
13	$15 < t \leq 20_RH > 40_WBGT < 21$	(20,41)
14	$15 < t \leq 20_RH \leq 40_WBGT < 21$	(20,30)
15	$t \leq 15_RH > 40_WBGT < 21$	(15,100)
16	$t \leq 15_RH \leq 40_WBGT < 21$	(15,30)

ことでシステムが正確に機能することを評価した。実験3ではCPSに期待される大規模なデータ処理をする機能としてセンサーの異常検知とアセスメント結果の因果関係分析機能の検証を行なった。これは構築するエージェントの構成を変えることでシナリオベースのテストケースを作成することで行われた。パーシャルデプロイメントによるCPSを検証することで、機能適合性の一部と性能効率性の評価ができる。

1節で取り上げた現在のCPS検証方法と比較して、実際のシステムノード数で評価ができた点、パラメータ操作や用意の手間が少ない点が優れている。また模型を用いる事例1の方法と比較して、CPSが稼働する環境の制限は少ないと思われる。そして小規模CPSで検証を代替する事例2の方法と比較して、実際に稼働するCPSでのシステムテストをできたため、さらに多くの不具合は抽出できていると思われる。

しかしパーシャルデプロイメントによる検証の課題点も存在する。まず代替した部分の検証が不足する点である。実験1を実行中の際CO2センサーの値が0を示すエラーが発生したため、センサー系に対してのテストが十分にされなかった懸念が残る。避難所情報マネジメントシステムはPub/Sub通信を用いたマイクロサービスアーキテクチャを採用しているため全体への影響は少ないと思われるが、例えば同時に同じMQTTのクライアントIDを利用してしまおうといったエラーが想定可能である。次に検証できない機能が存在する点である。例えば1節で取り上げた車の画像を認識して空きスペースを認識するといった機能はパーシャルデプロイメントを利用した方法では難しいと思われる。

5 結論

本研究では、代表的なCPSとして避難所情報マネジメントシステムを構築し、センサー値や避難所アセスメント結果をPubSubするエージェントを用いてパーシャルデプロイメントを利用したCPSの検証を行なった。これにより機能が正確に稼働することと実際の稼働時に近い性能効率性を評価できた。また可能な範囲でのユーザの目的達成度合いについても評価ができた。これらの評価は実際に検証環境を構築し、そこでパラメータを操作するものに比べて少ない手間で実施できたため、パーシャルデプロイメントによるCPSの検証は有効な手段と言える。

しかし、パーシャルデプロイメントによるCPSの検証方法はエージェントで代替した部分の検証が不足してしまうので、その点を解決するような工夫が必要である。また画像認識を用いる機能に関しては難しく、すべての機能の検証に用いられるわけではない。

今後は本研究では扱わなかった使用性の部分に注目し、ユーザによるテストを実施することが挙げられる。ユーザテストを通して機能がユーザの目的を網羅しているかを評価したり、使いやすいかどうかについての評価が行えると思われる。これを通してパーシャルデプロイメントを利用したCPSの検証方法についてより研究をする必要がある。

謝辞

本研究は総合科学技術・イノベーション会議のSIP(戦略的イノベーション創造プログラム)「レジリエントな防災・減災機能の強化」(管理法人:JST)によって実施されました

参考文献

- Edward A Lee: Cyber physical systems: Design challenges. In *11th IEEE Symposium on Object Oriented Real-Time Distributed Computing (ISORC)*, 363/369. IEEE, (2008)
- Nick Wingfield: Amazon Moves to Cut Checkout Line, Promoting a Grab-and-Go Experience. *New York Times*, (2016)
- NEDO: スマートグリッド. <https://www.nedo.go.jp/content/100079453.pdf>, (2010)
- Muftah Fraifer and Mikael Fernström: Designing a Smart Car Parking System (PoC) Prototype Utilizing CCTV Nodes: A vision of an IoT parking system via UCD process. *Advances in Science, Technology and Engineering Systems Journal*, 2, 3, 755/764, (2017)

- 5) Abdul-Rahman Al-Ali, Imran A Zualkernan, Mohammed Rashid, Ragini Gupta, and Mazin Alikarar: A smart home energy management system using IoT and big data analytics approach. *IEEE Transactions on Consumer Electronics*, **63**, 4, 426/434, (2017)
- 6) Patrick Th Eugster, Pascal A Felber, Rachid Guerraoui, and Anne-Marie Kermarrec: The many faces of publish/subscribe. *ACM computing surveys (CSUR)*, **35**, 2, 114/131, (2003)
- 7) James Lewis and Martin Fowler: Microservices. <https://martinfowler.com/articles/microservices.html>, (2014)
- 8) Nigel Gilbert: *Agent-based models*. 153. Sage, (2008)
- 9) 出口弘: IOE 時代の P2M 支援環境としての実世界 OS 国際 P2M 学会誌, **9**, 2, 99/122, (2015)
- 10) Organización Internacional de Normalización: *ISO-IEC 25010: 2011 Systems and Software Engineering-Systems and Software Quality Requirements and Evaluation (SQuaRE)-System and Software Quality Models*. (2011)
- 11) 市川学, 出口弘: 社会シミュレーションプラットフォーム SOARS を用いた教育と研究. 横幹連合コンファレンス予稿集, (2017)
- 12) 石井正: 石巻赤十字病院の東日本大震災対応の経験から見てきた大災害時における被災地域の保健医療福祉提供体制のあり方. *保健医療科学*, **62**, 4, 374/381, (2013)
- 13) 畑山満則, 松野文俊, 角本繁, 亀田弘行: 時空間地理情報システム DiMSIS の開発. *GIS-理論と応用*, **7**, 2, 25/33, (1999)
- 14) 無線 LAN ビジネス推進連絡会: 災害用統一 SSID 0000JAPAN (ファイブゼロジャパン). <https://www.wlan-business.org/customer/introduction/feature>, (2018)
- 15) 環境省: 環境省熱中症予防情報サイト 暑さ指数とは? <http://www.wbgt.env.go.jp/wbgt.php>
- 16) 難民支援協会: スフィア・ハンドブック 2011 年版 (日本語版). https://www.refugee.or.jp/sphere/The_Sphere_Project_Handbook_2011_J.pdf, (2011)
- 17) Sam Newman: マイクロサービスアーキテクチャ. オライリージャパン, (2016)
- 18) Microsoft: Azure でのマイクロサービスの設計, 構築, および操作. <https://docs.microsoft.com/ja-jp/azure/architecture/microservices/>, (2018)
- 19) 出口弘: IoE 時代のマイクロサービスとワークフローのアーキテクチャデザイン. 国際 P2M 学会研究発表大会予稿集, 70/101, (2016)
- 20) Urs Hunkeler, Hong Linh Truong, and Andy Stanford-Clark: MQTT-S—A publish/subscribe protocol for Wireless Sensor Networks. In *Communication systems software and middleware and workshops, 2008. comsware 2008. 3rd international conference on*, 791/798. IEEE, (2008)
- 21) EMQ project: EMQ - The Massively Scalable Open Source MQTT Broker. <http://emqtt.io>, (2012)
- 22) Douglas Crockford: The application/json media type for javascript object notation (json), (2006)
- 23) Kristina Chodorow: *MongoDB: The Definitive Guide: Powerful and Scalable Data Storage*. "O'Reilly Media, Inc.", (2013)
- 24) Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton: ImageNet Classification with Deep Convolutional Neural Networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, 1097/1105. Curran Associates, Inc., (2012)
- 25) LINE: Messaging API - LINE Developers. <https://developers.line.biz/ja/services/messaging-api/>, 2018
- 26) Stamatis Karnouskos, Roopak Sinha, Paulo Leitão, Luis Ribeiro, and Thomas I Strasser: Assessing the integration of software agents and industrial automation systems with ISO/IEC 25010. In *2018 IEEE 16th International Conference on Industrial Informatics (INDIN)*, 61/66. IEEE, (2018)
- 27) D Richard Kuhn, Dolores R Wallace, and Albert M Gallo: Software fault interactions and implications for software testing. *IEEE transactions on software engineering*, **30**, 6, 418/421, (2004)
- 28) 土屋達弘, 菊野亨: ペアワイズテストソフトウェアテストの効率化を求めて一. *電子情報通信学会論文誌 D*, **90**, 10, 2663/2674, (2007)
- 29) Jacek Czerwonka: Microsoft/pict: Pairwise Independent Combinatorial Tool. <https://github.com/Microsoft/pict>, (2018)
- 30) 井手剛: 入門 機械学習による異常検知 - R による実入門 機械学習による異常検知. コロナ社, (2015)