社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
TECHNICAL REPORT OF IEICE.
NS2003-359, IN2003-314(2004-03)

サービス指向アーキテクチャを用いたホームネットワークシステムの設計と評価尺度

井垣 宏† 玉田 春昭† 中村 匡秀† 松本 健一†

†奈良先端科学技術大学院大学情報科学研究科 〒630-0192 奈良県生駒市高山町 8916-5

E-mail: †{hiro-iga,harua-t,masa-n,matumoto}@is.aist-nara.ac.jp

あらまし　従来のホームネットワークシステムでは，ネットワーク内の高性能サーバがエンドデバイス(クライアント，PC，ネット家電等の端末)を制御することで，様々なサービスを提供するというシステム構成がとられている．しかし，接続端末数の増加やデバイスの高性能化に伴い，従来の構成では，余剰リソースの増加，サーバへの負荷集中，保守性の低下，スケーラビリティの欠如といった問題が生じると考えられる．そこで我々はWebサービスとサービス指向アーキテクチャを利用して，各サービスのコントローラをエンドデバイスに分散させ，複数のコントローラが協調して動作するような設計法によるホームネットワークシステムを提案した．さらに，ホームネットワークシステムの設計段階における評価のために，(1)システム信頼性，(2)負荷指数，(3)機能複雑度，(4)結合度，という評価尺度を定義し，従来のシステムと提案システムの差異について評価を行った．

キーワード　Webサービス，サービス指向アーキテクチャ，分散システム，ホームネットワーク

A Design and Evaluation Metrics of the Home Network Systems Using the Service Oriented Architecture

Hiroshi Igaki† Haruaki Tamada† Masahide Nakamura† and Ken-ichi Matsumoto†

†Graduate School of Information Science, Nara Institute of Science and Technology 8916-5 Takayama-cho, Ikoma-shi, Nara, 630-0192 Japan

E-mail: †{hiro-iga,harua-t,masa-n,matumoto}@is.aist-nara.ac.jp

Abstract In the conventional home network systems (HNS), a powerful centralized server controls all the end-devices connected (e.g., client terminals, PCs, network appliances etc.) to provide value-added integrated services. However, when the number of the devices increases and the devices become more sophisticated, the conventional architecture would suffer from problems in superfluous resources, maintainability, scalability and reliability. This paper proposes an alternative architecture for HNS, which exploits Web services and the service-oriented architecture. In the proposed architecture, each device is controlled by a Web service in a de-centralized manner. Then, the services autonomously collaborate with each other to achieve the given integrated service scenarios. To evaluate the HNS at the design process, we also present four kinds of evaluation metrics: reliability, load, complexity, coupling. Using these metrics, we conduct a comparative study among the proposed and the previous HNS architectures.

Keyword Web Services, Service-oriented architecture, Home network, distributed system

1. はじめに

コンピュータやネットワーク性能の向上，端末の小型化などに伴い，人々の日常生活に付随するさまざまなものがネットワークによって接続され，制御されるようになってきた．

例えば，家庭にあるエアコンや照明，AV 機器，PC などがネットワークを通して接続され，ホームサーバというコントローラを利用して制御される．それによって，帰宅するとエアコンのスイッチが自動的に入り，DVD を見るときには照明が暗くなる，といった家庭内でのユーザの生活の利便性の向上をはかることができる．このようなシステムはホームネットワークシステム[2]と呼ばれ，企業などで開発が進められている．しかし，現状のホームネットワークシステムでは，中央のホームサーバが全てのデバイスの動作を制御するというアーキテクチャによって設計されており，負荷の集中，機能が複雑化することによる開発コストの増大，スケーラビリティの低下などの問題が避けられない．

以上のような問題を解決するために，本研究ではWebサービスを利用したサービス指向アーキテクチャによってホームネットワークシステムの設計を行い，システムが提供するサービスに応じてコントローラを分散させることで，負荷の分散，機能の単純化をはかった．また，システムの信頼性，負荷，機能の複雑度，

個々のコントローラの結合度などの評価基準を提案することにより，従来のアーキテクチャとサービス指向アーキテクチャを採用したホームネットワークシステムの間の差異について検証を行っている．

2. Web サービスを利用したシステム設計

2.1. Web サービス

本稿で述べる Web サービス(以降 WS)とは，インターネット上の自律したソフトウェア資源を XML ベースの共通のインタフェースを介して提供・再利用する技術，またはそのアプリケーションのことをいう[1]．

WS は以下のような特徴を持っている．

1. XML 形式でデータの交換を行う．
2. 通信プロトコルには SOAP が利用されている（下位プロトコルとしては HTTP などを使用する）．
3. W3C[6]によって標準化された通信なので，プラットフォーム，言語に依存しない．
4. サービスインタフェース（メソッド）は WSDL によって XML 形式で記述され，公開される．

開発者が WS の機能をクライアントから利用する場合，WS のメソッドを通常のメソッド呼び出しとほぼ同じ方法で呼び出すことで遠隔の WS の機能を組み込むことができる．つまり，クライアントの開発者は通信プロトコルやメッセージ書式，対象 WS の内部ロジックを意識せずに(疎結合と呼ばれている)WS の機能を利用することができる．

この疎結合の性質を利用することで，複数の WS を開発者に依存せずに連携させて開発することも容易となる．また，この疎結合を利用したソフトウェアアーキテクチャとして次に述べるサービス指向アーキテクチャが存在する．

2.2. サービス指向アーキテクチャ

サービス指向アーキテクチャ SOA(Service-Oriented Architecture)とは，自律，協調動作が可能なサービス間の疎結合を実現するためのソフトウェアアーキテクチャである[7].ここでいうサービスとは，サービス利用者が必要とする結果を実現するために，サービス提供者が実行する一連のタスクのことをいう．WS を利用した SOA では，サービス利用者の要求に応じて複数の WS を自由に連携，再利用してひとまとまりのサービスとすることが可能となる．

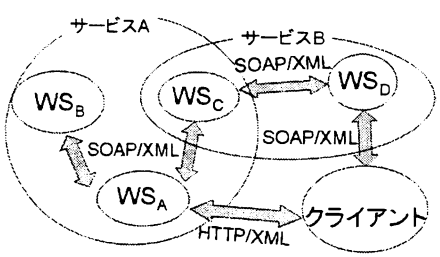

図 1 サービス指向アーキテクチャ例

図 1の例のように，SOAP/XML で通信を行う WS_B,WS_C を連携させて WS_A に集約し，クライアントが要求するタスクを実行するサービスAを実現している．さらに，別の要求に対しては WS_D がサービス A の部分要素である WS_C と連携することでサービス B を実現する．このとき，サービスAを利用するときのクライアントは WS_A, WS_D が公開しているメソッドのみを考慮して振る舞いを決定することが出来る．

このように SOA はクライアントの要求によって，必要とする WS を多様なパターンで連携させる事が非常に容易である．また，直接接続する WS との連携のみを意識すればよいために，制御に要するコストを分散させることが可能になっている．

次章で説明するホームネットワークシステムでは，エアコンや TV といったデバイスがさまざまな状況で利用される．そのため，SOA を適用することで，デバイスを制御する WS をクライアントの要求するサービスに応じて連携し，再利用することが可能になる．

3. ホームネットワークシステムの設計

3.1. ホームネットワークシステム

HNS(Home Network System)は，家庭における生活の利便性を増大させる目的で，家庭内の複数の電化製品や住宅設備，PC 等(これらをデバイスと呼ぶ)をネットワークを介して相互接続し，より高度で付加価値の高い制御を実現するものである．

本論文では，例として以下の 14 個のデバイスから構成される HNS を設計する：DVD プレーヤ，TV，スピーカ，照明(×3)，照度計，ドア，窓(×3)，電話，エアコン，温度計．これらのデバイスを連携させ，以下 8 種類のサービスシナリオ(Service Scenarios, 以下 SS)を実現することを考える．なお，これらの SS は実際に商品化されている HNS を参照して作成したものである．

SS1：照度計を利用して照明の明るさを調整する．
SS2：ユーザがドアから入室すると照明が点く．
SS3：ユーザが DVD を見るときは照明が暗くなり，DVD プレーヤが起動し，TV，スピーカが DVD のモードで起動する．
SS4：ユーザが通常のテレビを見るとき，スピーカが TV 用に起動する．
SS5：ユーザに電話がかかってきたときに，TV がついているとスピーカのボリュームを落とす．
SS6：ユーザがエアコンのスイッチを入れると，温度計の値を利用して動作が調節される．
SS7：ユーザがドアから入室すると，温度計の値を元に，必要に応じてエアコンが起動する．
SS8：ユーザが外出するとき，睡眠時等には，機器の

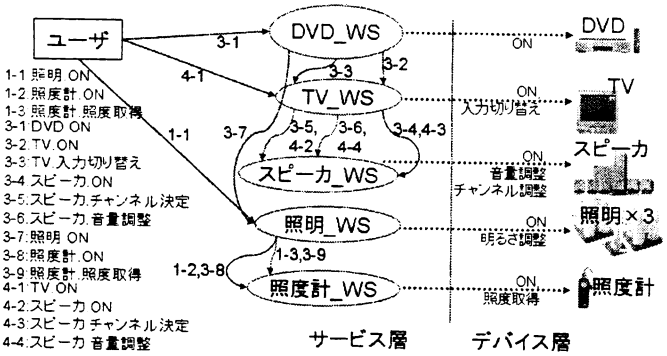

図 3 サービス指向アーキテクチャを利用した HNS の設計

電源を落とし，戸締まりの確認を行う．

3.2. サービス指向アーキテクチャ

本論文のねらいは，各デバイスをソフトウェア制御によって動作する自律的なノードとみなし，それらの相互接続に WS を用いた SOA を適用することにある．各デバイスは，機能制御のためのプロセッサと WS を動作させるサーバを持っているものと仮定する．

SOA に基づいた HNS では，各デバイスはハードウェアとそれを外部から制御するためのサービスから構成される．3.1 で挙げた SS を実現するための WS のメソッドの内容と WS 間の連携の設計を行った．

図 3に，SOA に基づいて設計した HNS のアーキテクチャを示す．アーキテクチャ全体はサービス層とデバイス層に分かれる．各デバイスは，その機能をメソッドとして提供する WS(図中楕円で表す) によって制御される(図中点線矢印)．個々の WS では，制御するデバイスや協調する他の WS のためのサービスインタフェースが定義・公開されている．

3.2 節で述べた SS を実現するために，各デバイスはサービス層において連携を行う．図 3中のサービス層では，ユーザがいくつかの SS を実行する場合の連携を示している．図において，要素 A から B へのラベル L 付の実線矢印は，B が提供するメソッド L を A が利用(実行)することを表している．また，各 WS が提供するメソッドはパラメータを二つ以上取らないものとする．

例として，SS1 を考える(図 3の“1-”で始まるラベル付き矢印)．まず，ユーザは照明_WS に対して照明.ON メソッドを呼び出す．次に，照明_WS は照度計_WS にアクセスして，照度計を ON 状態にし，現在の照度を取得する．最後に，照明_WS は現在の照度を基に照明デバイスに対して最適な照明を設定する．SS4 についても同様で，図 3の“4-”で始まるラベル付き矢印で示される手順でシナリオを実行する．次に，SS3 を考える(“3-”で始まるラベル付き矢印)．ユーザが DVD を ON にすれば，DVD_WS は TV_WS と照明_WS の両方に対して電源 ON のメソッドを実行する．その後，TV_WS と照明_WS はそれぞれ SS4, SS1 のシナリオを実行することで，SS7 を実現できる．このように，いくつかの SS を再利用して組み合わせ，新たな SS を効率よく実現することも可能である．

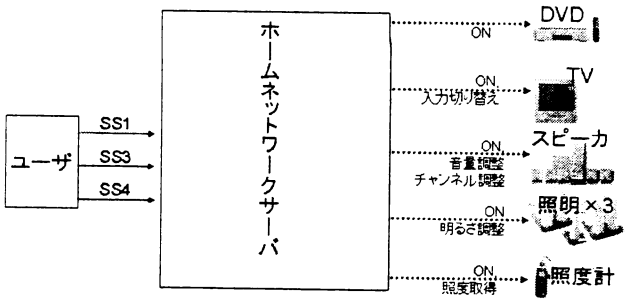

図 2 サーバ中央型アーキテクチャを利用した HNS の設計

SOA に基づいた HNS の特徴は，各デバイスがユーザの要求する SS に応じて分散・協調して制御されること，そのために各サービスの再利用が容易であること，デバイス制御を分散させることで一つ一つの WS の実装がシンプルかつ閉じたものになること，WS による疎結合によって汎用性・相互接続性が高い実装が可能であるということが挙げられる．

3.3. サーバ中央型アーキテクチャ

家庭のネットワーク普及により，近年いくつかの HNS が商品化されてきている(例：[2][3])．これらのシステムでは，家電デバイスに軽量なアダプタを装着し，家庭内のホームサーバ HS(Home Server)から専用のソフトウェアおよびプロトコルを用いて，アダプタに信号を送る方式を採用している．この方式をサーバ中心型アーキテクチャ SCA (Server Centralized Architecture) と呼ぶ

SCA に基づく HNS では，システムの中央に高機能な HS が存在し，全てのデバイスを集中的に制御する．

3.1 節で作成した SS を SCA により設計すると，図 3のようになる．デバイスの連携は，HS 内で密に結合したオブジェクトの連携によって行われる．例えば，ユーザが HS に対して DVD を ON にする処理を要求すると，HS が DVD プレーヤ，TV，スピーカ，照明の各デバイスを直接制御することになる．

このように，SCA に基づいた HNS では，デバイスのインテリジェントな部分を全て HS が担うことで，アーキテクチャ自体は非常にシンプルなものとなる．その一方で，HS に機能が集中するため，HS の複雑化・スケーラビリティ問題が生じること，また，HS に障害が発生すると全ての SS が利用できなくなるといった問題点がある．

4. 設計評価法

ここでは，(a)信頼性，(b)サービス利用頻度に基づく負荷の偏り，(c)WS，HS 等のコントローラの機能複雑度，(d)コントローラの結合度の 4 つの評価指標を定義し，設計段階における評価法を提案する．また，3 章で述べた 2 種類のアーキテクチャに基づく HNS の設計の評価を行う．

4.1. 信頼性評価

2 つのアーキテクチャに基づく HNS の信頼性を評価する．ここで，HNS の n-信頼性とは，デバイスを制御するサービス(SOA では WS，SCA では HS)に障害が発生し得る時，少なくとも n 個のサービスシナリオ SS が正常に稼動する確率であると定義する．

4.1.1. サービス連携グラフの定義

HNS における各サービスシナリオはユーザ，デバイス，WS，HS という 4 種類の構成要素とその間の連携によって性質づけられる．従って，構成要素を頂点，構成要素間の連携を有向辺，メソッドをその連携に用いるラベルとするラベル付有向グラフで表すことができる．

ラベル付有向グラフ G=(N,L,E)とは，N をノードの集合，L を各辺に付けられるラベルの集合，$E \subseteq N \times L \times N$ を有向辺の集合と定義される．今 s をサービスシナリオとするとき，ラベル付き有向グラフ GS=(N,L,E) が以下の全ての条件を満たすとき，GS を s のサービス連携グラフと呼ぶ．

- N は HNS に現れる全ての構成要素の集合
- L は HNS において実行される全てのメソッド，オペレーションの集合(ここでは，WS がユーザまたは他の WS に公開している処理をメソッドと呼び，HS や WS のデバイスに対する処理をオペレーションと呼ぶ)
- s において，p が q のメソッド m を実行する時，$(p,m,q) \in E$

複数のサービスシナリオ s1,s2 の連携グラフ $G_{s1\text{-}s2}$ は，それぞれの個々の連携グラフ G_{s1} と G_{s2} を重ね合わせたグラフと定義する．例えば，図 3において WS，デバイス，ユーザを頂点，全ての矢印を有向辺と見なして得られる有向グラフは $G_{ss3\text{-}ss4\text{-}ss5}$ といえる．

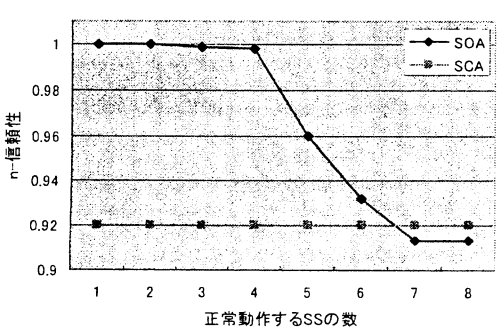

図 4 システムの信頼性と SS の関係

4.1.2. Sum of Disjoint Products

サービスシナリオをグラフで表現することにより，数学的に信頼性を計算できる．本論文では，信頼性の評価アルゴリズムとして Sum of Disjoint Products(SDP)[4][5]を用いた．グラフの頂点と辺の一つ一つに独立した信頼性を与えると，SDP はグラフ内の特定のサブグラフ(の集合)が稼動する確率(信頼性)を，頂点や辺の重なりを考慮して導出する．全てのサービスシナリオの連携グラフを G とすると，サービスシナリオ一つ一つは G のサブグラフで表現できるので，WS や HS に障害が発生する場合の各サービスシナリオの信頼性を計算することができる．

4.1.3. SDP による信頼性評価

3.1 節で述べた 8 つの SS を SOA, SCA それぞれのアーキテクチャで実現する時の n-信頼性を SDP により計算する．今，8 つの SS 全てを含むサービス連携グラフを G とする．8 つの SS のうち少なくとも 1 つが正常に動作する時，即ち HNS の 1-信頼性は，G のサブグラフである $G_{ss1}, G_{ss2}, \ldots, G_{ss8}$ のうちどれか 1 つが稼動する確率である．同様に，2-信頼性は，G のサブグラフ，$G_{ss1\text{-}ss2}$, $G_{ss1\text{-}ss3}, \ldots, G_{ss7\text{-}Gss8}$ のうちどれか 1 つが稼動する確率である．このように，サービス連携グラフの全ての組み合わせを SDP に入力して，n 個 ($1 \leqq n \leqq 8$)のサービスが稼動する n-信頼性 P_n を SOA,SCA それぞれに対して計算できる．

ここでは，アーキテクチャの違いによる信頼性評価を行うため，SOA における WS および SCA における HS の障害のみ考慮に入れる．即ち，ユーザ，デバイス，メソッドの信頼性をそれぞれ 1 とし，WS 単体の信頼性を 0.99，HS 単体の信頼性を 0.92 （SCA では 8 つ全てのシナリオを HS が実行するために WS の正常動作率 0.99 の 8 乗を HS の正常動作率とした）とする．

信頼性評価の結果を図 4と表 1に示す．SCA では全ての SS が中央の HS1 つで制御されるために，HNS 全体の n-信頼性は HS の信頼性と等しくなる．即ち，HS が故障してしまうと，全ての SS が利用できなくなるということを表している．

一方 SOA では，n が 4 までの信頼性が非常に高くなっている．これは，SOA において SS を実現するための WS が分散している，すなわちデバイスや他の WS のコントローラが分散していることが，システム全体の信頼性の向上に寄与しているということを示している(n が 5 以上のときは SS に利用されるサブグラフが

表 1 システムの信頼性

	n	1	2	3	4	5	6	7	8
アーキテクチャタイプ	SOA	0.9999	0.9999	0.9986	0.9982	0.9598	0.9319	0.9135	0.9135
	SCA	0.92	0.92	0.92	0.92	0.92	0.92	0.92	0.92

HNS の構成要素のほぼ全てを必要とするために信頼性が減少し，結果として SCA と同等になっていると考えられる).

4.2. 利用頻度による負荷見積もり

利用頻度による負荷見積もりとは，個々の WS/HS が呼び出される頻度を元に負荷指数として表したものである．

まず，今回例として挙げた SS を元に「もし SS1~8 が実現されている HNS を利用できるとすれば，どの程度の頻度で各 SS を利用するか，現状の生活を踏まえた上で答えてください」という質問を 12 人のユーザ(うち一人暮らしが 8 人，二人家族が 2 人，4 人家族が 2 人)に対して行った．そのアンケート結果を集計し，図 3の WS と図 2の HS が呼び出される週当りの頻度(一家族単位)と標準偏差を示したものが表 2である．

利用頻度指数がそのまま各 WS,HS の負荷指数を意味し，標準偏差が負荷のばらつきを示している．負荷の分散という意図においては， HNS に存在するオブジェクトにおける負荷指数が平均化していることが望ましいといえる(すなわち，標準偏差が小さい)．

例えば SOA においては，照明に関する WS の負荷指数が高くなっている．この負荷指数を下げるためには，照明関連の WS の機能を削減し，新たに削減した機能を実現する WS を別に作成するなどの設計変更が考えられる．一方で SCA の場合，負荷は全て HS に集中する．そのため，標準偏差も SOA と比して非常に大きくならざるを得ないが，負荷を分散する手段は SCA には存在しない．

このように，利用頻度による負荷指数を調べておくことで，WS 等を動かすハードウェアにかかる負荷を想定することが可能になる．結果として，負荷分散を考慮した設計を実現することができる．

4.3. 機能複雑度

HNS において，WS_A が WS_B のメソッド m を実行するとき，m は WS_B の内部に実装される機能(内部機能)であり，WS_A の見地からは外部に存在する機能(外部機能)である．本節では機能複雑度という評価尺度を，

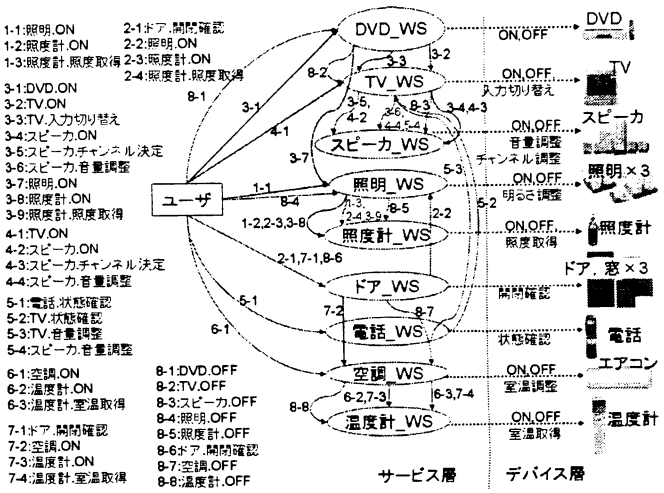

図 5 SOA に基づくサービス連携グラフ G

HNS 実現に求められる各 HS/WS の内部，外部機能数で性質づけることにする．

厳密に言うと，与えられた全ての SS の連携グラフ G=(N,L,E)に対して，$(WS_A,m,WS_B) \in E$ の場合，m を WS_A の外部機能，WS_B の内部機能と呼ぶ．さらに，$w \in N$ について$|\{m|^{\exists}w'(w',m,w)\}|$を w の内部機能数と呼び，inum(w)と書く．同様に，$w \in N$ について$|\{m|\exists w'(w,m,w')\}|$を w の外部機能数と呼び enum(w)と書く．この時，w の機能複雑度 fcomp(w) = inum(w) + enum(w)と定義する．

例えば，図 5は，SOA によって設計された全ての SS を満たす HNS である．ここで，照明_WS の内部機能数と外部機能数は次のようになる．

- inum(照明_WS)=|{1-1:照明.ON,2-2:照明.ON,3-7:照明.ON,8-4:照明.OFF}| = 2
- enum(照明_WS)=|{1-2:照度計.ON,1-3:照度計.照度取得,2-3:照度計.ON,2-4:照度計.照度取得,3-8:照度計.ON,3-9:照度計.照度取得,8-5:照度計.OFF,照明 ON,照明 OFF,照明明るさ調整}| = 6

このようにして SCA,SOA の各設計における個々の WS/HS の複雑度を算出し，表 3とした(外部機能の()は WS が操作するデバイスの機能数を示している)．機能複雑度を表す数値が高ければ，対象となる WS/HS

表 2 利用頻度

(a) SOA

Webサービス	利用頻度/週
DVD_WS	10.7
TV_WS	29.8
スピーカ_WS	29.8
照明_WS	57.4
照度計_WS	57.4
ドア_WS	18.7
電話_WS	3.7
空調_WS	16
温度計_WS	16
標準偏差	18.203

(b)SCA

ホームサーバ	利用頻度/週
HS	86.2
標準偏差	86.2

表 3 システムの複雑度と結合度

	複雑度		結合度	
WS/HS	内部機能	外部機能	利用コンポーネント数	被利用コンポーネント数
DVD_WS	2	6(2)	3(1)	1
TV_WS	5	7(3)	2(1)	3
スピーカ_WS	4	4(4)	1(1)	1
照明_WS	2	6(3)	4(3)	3
照度計_WS	3	3(3)	1(1)	1
ドア_WS	1	4(1)	6(4)	1
電話_WS	1	2(1)	2(1)	1
空調_WS	2	6(3)	2(1)	2
温度計_WS	3	3(3)	1(1)	1
HS	8	23(23)	14(14)	0

に求められる機能数が多くなるということであり，結果として実装が複雑なものになる，実装が複雑になるためにテストやデバッグが困難になる，というデメリットが発生すると考えられる．

例えば，SOAの場合，機能複雑度がもっとも高いものは fcomp(TV_WS)=12 となっている．設計によってこの値が分ることにより，もしこの値が複雑であると考えられる場合には機能の分散をはかることが可能である．一方SCAの場合，HSは単一で全ての機能を実現するためにSOAと比べると，fcomp(HS)=31 という大きな数字になっている．これはそれだけHSの機能が複雑であるということであり，結果として既に述べたようなデメリットが存在する度合いがSOAよりも高いということを表している．

4.4. 結合度

いま，HNSにおいて WS_A が WS_B のメソッドmを実行するとき，WS_A にとって WS_B は利用コンポーネント(HNSの構成要素のうち，ユーザ，WS,デバイス，HSをコンポーネントと呼ぶ)であり，WS_B の見地からは WS_A は被利用コンポーネントである．あるコンポーネントがいくつの利用(被利用)コンポーネントに関連しているかはシステムに与える影響の強さを表すと考えられる．この影響の強さを結合度という評価尺度で表すことにする．

与えられた全てのSSの連携グラフG=(N,L,E)に対して，$(WS_A,m,WS_B) \in E$ の時，WS_A を WS_B の被利用コンポーネントと呼び，WS_B を WS_A の利用コンポーネントと呼ぶ．さらに $w \in N$ について $|\{w'|^3 m(w,m,w')\}|$ をwの利用コンポーネント数と呼び，use(w)と書く．同様に，$w \in N$ について $|\{w'|^3 m(w',m,w)\}|$ をwの被利用コンポーネント数と呼び，used(w)と書く．この時，wの結合度 coupling(w) = use(w) + used(w)と定義する．

例えば，図5における TV_WS の利用コンポーネント数と被利用コンポーネント数は次のようになる．

- use(TV_WS)=|{スピーカ_WS,TV}|=2
- used(TV_WS)=|{ユーザ，DVD_WS，電話_WS}|=3

このようにしてSCA,SOAの各設計における個々のコンポーネント(ユーザは除く)の結合度を算出し，表2の結合度とした(利用コンポーネント数における()はデバイスの数を表す)．

結合度を表す数値が高ければ高いほど，対象のコンポーネントがシステムの他のコンポーネントに与える影響が大きいということであり，障害の発生やコンポーネントの仕様変更によってHNS全体に与える影響が大きくなるということを表す．

例えばSOAでは，照明_WSとドア_WSの利用コンポーネント数と被利用コンポーネント数が多くなっているが，このうちの大半は窓や照明の個数が多いためであり，もし結合度が高いと考えられるならば，特定のWSが利用するコンポーネントを別のWSに振り分けることで結合度の偏りを減らすことができる．一方SCAではHSの利用コンポーネント数が非常に多い．これは全てのデバイスをHSが制御しているためであり，信頼性の節でも述べたように，HSに障害が発生したときのHNS全体に及ぼす影響は非常に大きい．

5. 考察と今後の課題

本論文では，SOAを利用したHNSの設計と設計段階における評価尺度の提案を行い，実際に設計したHNSの評価を行った．この評価尺度を測定することにより，設計プロセス以降のハードウェア設計，実装，テスト，運用，保守といったプロセスにおけるソフトウェア品質の向上をはかることができると考えられる．

例えば，システムの信頼性は運用，保守のプロセスに，負荷分析はハードウェア設計，テストに影響を与える．機能複雑度は実装，テスト，保守のプロセスに，コントローラの結合度は保守プロセスにとって重要なパラメータである．ソフトウェアシステム全体が大きなものになればなるほどこのような影響の度合いは大きくなっていく．

HNSは今後，ユーザの入室管理，音声による機器操作，等々のユーザの利便性を向上させるためのサービスシナリオを実現するために，さらに多くのデバイスを多様な用途で扱うようになっていくと考えられている．そうなったときに，SOAによる設計と，今回提案した評価尺度はさらに重要なものとなると考えられる．

一方で，SOAではコントローラが複数存在するために，デバイスにアクセスする際のセキュリティ問題や複数ユーザが利用する際の競合問題などを解決しなければならない．今後はこれらの問題の解決と実際にWSを動かしてシミュレーションをすることによる定量的評価について研究を進めていく予定である．

文　献

[1] 青山幹雄，“Webサービス技術とWebサービスネットワーク，”信学技報，IN2002-163，pp.47-52, Jan. 2003.

[2] Horaso network service,http://ns.horaso.com/

[3] iReady,http://www.sharp.co.jp/corporate/news/031217-2.html

[4] S. Hariri, C. S. Raghavendra, “SYREL: A Symbolic Reliability Algorithm Based on Path and Cutset Methods,” IEEE Transactions on Computers, Vol.42, No.2, pp.1224-1232, October 1987.

[5] Tatsuhiro Tsuchiya, Tomoya Kajikawa, and Tohru Kikuno, “Parallelizing SDP (Sum of Disjoint Products) Algorithms for Fast Reliability Analysis,” IEICE Transactions on Information and Systems, Vol.E83-D, No.5, pp.1183-1186, May 2000.

[6] W3C Web Service Activity, http://www.w3.org/2002/ws/

[7] What is Service-Oriented Architecture ?, http://webservices.xml.com/pub/a/ws/2003/09/30/soa.html