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.

サービス競合検出・解消システムを用いたホームネットワーク連携サービスの開発

稲田　卓也† 　吉村　悠平† 　池上　弘祐† 　井垣　　宏† 　中村　匡秀†

中北　賢二†† 　竹原　清隆††

† 神戸大学　〒 657–8501 神戸市灘区六甲台町 1–1
†† パナソニック電工株式会社　EMIT プラットフォーム開発センター

あらまし　ホームネットワークシステム (HNS) のアプリケーションの一つとして，複数の家電を連携制御する家電連携サービスの研究が進んでいる．単体では正常に動作する連携サービスでも，複数を同時に実行すると機器や環境に対して衝突を起こしてしまうことがある．我々は，これを「サービス競合」と呼んでいる．HNS のサービス品質を損なわないためにも，このようなサービス競合を検出し解消することが求められている．我々は先行研究において，サービス指向アーキテクチャ(SOA) にもとづくサービス競合検出・解消基盤を提案した．本稿では，これらサービス競合検出・解消基盤を利用し，新しい連携サービス開発プロセスを提案する．提案する開発プロセスでは，連携サービスモデル，競合検出・解消シミュレーション，連携サービスモデルから実装への自動変換の 3 つの開発者を支援する仕組みを導入する．また，この連携サービス開発プロセスを実際の連携サービス開発に適用し，従来の開発プロセスとの比較を行った．その結果，提案する連携サービス開発プロセスが開発効率において優れていることを確認した．
キーワード　HNS, 家電連携サービス, 開発プロセス, メトリクス, モデル駆動開発.

Developing Home Network Services with Feature Interaction Detection and Resolution System

Takuya INADA†, Yuhei YOSHIMURA†, Kosuke IKEGAMI†, Hiroshi IGAKI†, Masahide NAKAMURA†, Kenji NAKAKITA††, and Kiyotaka TAKEHARA††

† Kobe University　Rokkoudaityou 1–1, Nada-ku, Kobe, Hyogo, 657–8501 Japan
†† EMIT Platform Development Center, Panasonic Electric Works Co., Ltd.

Abstract　As one of the major Home Network System, the integrated service of networked home appliances are proposed. When such integrated services are executed simultaneously, functional conflicts can occur between the functions of each appliance. Such conflicts are called Feature Interaction in HNS. In our precedence research, we proposed an feature interaction detection and resolution mechanism based on service-oriented architecture(SOA). In this paper, we propose new software development process for integrated services in HNS. In our process, we adopt a specification template for integrated services, simulation for feature interactions detection and resolution, and automatic transformation service specifications into implementations. In our comparative experiment, we confirmed our development process is more useful to develop reliable integrated services effectively than that of conventional development.
Key words　HNS, Integration Services, Development Process, Metrics, Model-Driven Development.

1. はじめに

ネットワーク技術の発展と共に，一般家庭にある家電機器を家庭内のネットワークに接続して，宅外からの制御や複数機器の連携を実現する**ホームネットワークシステム (HNS)** の研究開発が進んでおり，近年いくつかの製品が商品化されている [1] [2] [3].

HNS では，テレビや DVD レコーダ，エアコン，照明，扇

風機といった複数の家電を連携制御することで，家電単体で利用する場合に比べてより付加価値の高い**家電連携サービス**（以下，連携サービス）を実現することができる．連携サービスは，ユーザの日常生活における快適性・利便性を高める主要な HNS アプリケーションの一つとして研究されている．

下記に連携サービスの例を紹介する．

SS1：DVD シアターサービス ：DVD レコーダ，テレビ，スピーカー，カーテン，照明を連携し，映画館の雰囲気でユーザが DVD を視聴できるサービス．ユーザがサービスを要求すると，自動的に DVD レコーダとテレビが再生モードで ON になり，カーテンが閉まり，照明が暗くなり，スピーカーの 5.1ch が選択され，DVD が再生される．

SS2：おかえりサービス ：ユーザが帰宅した際に，照明を明るくするサービス．サービスは照明を 5 分間制御し続けることができる．

HNS にこのような連携サービスが提供されると，ユーザの利便性が向上する反面，**サービス競合**という新たな問題が発生することがある [4] [5]．サービス競合とは，単独で正常に動作する連携サービスが複数同時に実行されることで，互いに干渉・衝突を起こし，ユーザの意図した通りに動作しなくなる現象である．HNS におけるサービス競合はユーザの快適性・利便性を損ない，HNS の品質を低下させる要因となるため，HNS による**検出・解消**が求められている．

上記の連携サービス例においても，同時に実行されることでサービス競合が発生する．例えば，あるユーザ A が DVD シアターサービスを実行しているときに別のユーザ B が帰宅し，おかえりサービスを実行したとする．ここでユーザ A は DVD を見ているために照明を暗くしておきたいのだが，おかえりサービスが実行されてしまうことで照明が明るくなってしまう．これは照明機器において 2 つの連携サービスの要求が衝突してしまったことによるサービス競合である．この例においては，ユーザ B がおかえりサービスを実行したときに，照明機器に対する競合を検出し，解消する仕組みを HNS に導入しておく必要がある．

しかしながら，連携サービスや機器の種類が増大するに伴い，開発時に全ての競合を把握し，競合検出・解消が適切に行われる仕組みを実装することは非常に困難である．また，家庭ごとに異なる HNS の構成要素や機器構成の変化といった多様性と柔軟性への対応はさらに難しい．

そこで本研究では，サービス競合の検出・解消を考慮した新しい連携サービス開発プロセスを提案する．我々はこれまでの研究において，**サービス指向アーキテクチャ(SOA)** にもとづくサービス競合検出・解消基盤を提案した [9]．この基盤では，競合検出，競合解消，連携サービス実行管理という 3 つの構成要素を対象にサービス化を行った．我々が提案する連携サービス開発プロセスでは，これらの各基盤サービスを導入することで，開発効率の改善とバグの削減を実現した．開発者を支援するための主要なプロセス要素を以下に示す．

連携サービスモデル： 連携サービスは，実行される機器および機器メソッドのシーケンスとして表現される．サービス競合の検出・解消までを行う場合には，メソッドのシーケンスのみでは不十分である．そこで我々は連携サービスにおけるサービス競合の検出・解消に関するこれまでの研究成果 [10] にもとづき，**連携サービスモデル**を構築した．我々の連携サービスモデルにもとづくテンプレートによって，ステークホルダーから抽出すべき要件が明確になり，適切な要求分析が可能となった．また，テンプレートを元に連携サービスシナリオを記述したものを連携サービス記述と呼ぶ．

競合検出・解消シミュレーション： 競合検出・解消並びに実行管理の各サービス基盤を利用し，競合サービスモデルにもとづいた競合検出・解消に関する回帰テスト環境を提供する．開発者は連携サービスごとの連携サービス記述を入力・編集するだけで，連携サービス記述にもとづくあらゆる検出・解消結果を容易に確認することが可能となる．

連携サービスモデルから実装への自動変換： 提案プロセスでは，連携サービス記述の完成に伴い，実装が生成される．詳細設計および実装の工程を完全に自動化することで，人的要因によるミスの削減を図る．

本稿では，これらの仕組みを導入した連携サービス開発プロセスを実際の連携サービス開発に適用し，従来の開発プロセスと比較を行った．その結果，連携サービスの新規開発，開発済み連携サービスの改訂，既存連携サービスへの新たな連携サービスの追加，の全てにおいて，要した工数とバグ数の改善が見られた．特に新規開発においては工数にして 9.9 倍，バグ数において 11.7 倍もの差があることがわかった．

以降では，2 章でホームネットワークシステムにおけるサービス競合と既存のサービス競合検出解消システムについて，3 章では従来の連携サービス開発プロセスについて，4 章で我々が提案する連携サービス開発プロセスについて説明する．さらに，5 章で実際に行った連携サービス開発実験について述べ，その結果について考察する．

2. 準　　備

2.1 HNS におけるサービス競合

我々は先行研究において，HNS におけるサービス競合の**静的競合検出手法** [4] と**動的競合検出・解消手法** [8] を提案した．これらの手法では，各家電を状態 (**機器プロパティ**) と操作 (**機器メソッド**) を持つオブジェクトとしてみなすことで **HNS のモデル化**を行った．ここで機器メソッドは，機器プロパティを参照・更新するものとしてモデル化される．メソッドとプロパティの関係は以下のように定義される．

- メソッドの実行に必要な機器に関する事前条件
- メソッド実行後に成立する機器に関する事後条件

例えば TV の場合，機器プロパティとして power， channel，volume を持ち，power は true(電源 ON) か false(電源 OFF) である boolean 型，channel と volume は整数の int 型で与えられる．ここで，TV の機器メソッドとして，真偽値を返す vol(int volume) があるとする．このメソッドは事前条件として power プロパティが true でなければならず，実行後 (事後条件) に volume プロパティの値が vol メソッドの引数で与えられた

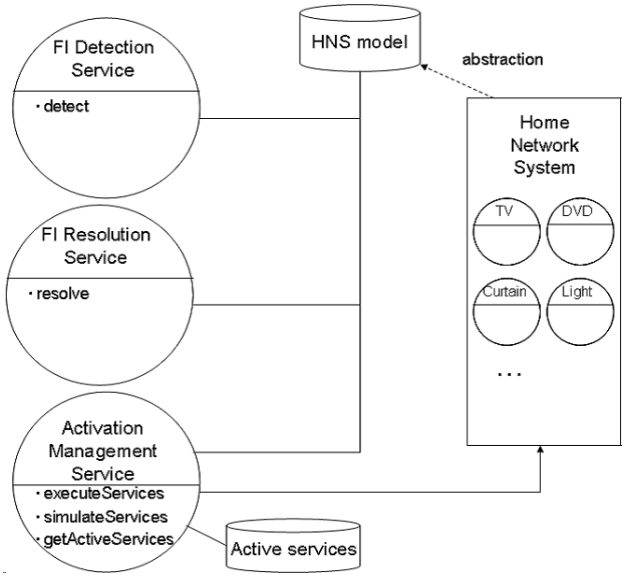

図 1　SOA にもとづく競合検出・解消基盤のアーキテクチャ

値で更新されていなければならない．

連携サービスは任意の機器メソッドを組み合わせたものであり，機器メソッドの系列を**連携サービスシナリオ**と呼ぶ．例えば 1. 章で述べた SS1:DVD シアターサービスは DVDRecorder.on(), TV.on(),Curtain.close() 等のメソッドから構成されている．また，連携サービスは通常実行時に呼び出される**開始シナリオ**と終了時に実行される**終了シナリオ**を含んでいる．

我々の静的競合検出手法では，サービス競合を「あるサービスに含まれるメソッド m と別のサービスに含まれるメソッド m' について，m の事後条件と m' の事後条件あるいは m' の事前条件を同時に満たすことができない場合に発生する」と定義している．与えられた連携サービス群が含む全ての機器メソッドのペアに対して，条件を評価することで，潜在的な全てのサービス競合を静的に検出することができる．検出されたサービス競合に優先順位等の競合解消手段を適用することで，サービス競合を考慮した連携サービスを開発することが可能となる．

実際に，我々は先行研究 [9] において，HNS および静的競合検出のモデルを利用し，実行時に競合検出・解消を行う手法およびシステムを提案した (図 1)．我々の開発した競合検出・解消基盤は，現在の連携サービスの実行状態を管理する**実行管理サービス**，新たに実行される連携サービスと現在実行中の連携サービスとの間の競合を検出する**競合検出サービス**，競合検出結果にもとづいて競合を解消する**競合解消サービス**の 3 つのサービスから構成される．

本研究ではこの競合検出・解消基盤を利用し，連携サービス開発プロセスを構築した．次節ではまず従来の HNS 連携サービス開発手法とその課題について説明する．

3.　従来の連携サービス開発プロセス

図 2. にウォーターフォールモデルにもとづく従来の連携サービス開発プロセスを示す．各工程で行われるタスクとその成果物について以降で説明する．

3.1　ソフトウェア要求分析

要求分析工程は作成する成果物に応じた 2 つの工程に分割される．仕様書を作成する工程と競合表を作成する工程である．仕様書作成工程では，企画書や顧客の要求にもとづいて，開発する連携サービスとその連携サービスが含む機器とその機能が仕様書に記述される．競合表作成工程では，仕様書に記述された全ての連携サービスを対象として，連携サービス間の競合を分析する．競合表には，発生する競合と競合発生時の解消内容が記述される．

3.2　ソフトウェア方式設計

ソフトウェア方式設計で作成される方式設計書には，作成した仕様書と競合表にもとづいたサービス優先度，サービス実行処理，機器制御方法が記述される．ここで機器制御方法は個別の機器を呼び出すための手順を記述したもので，サービス実行処理は連携サービスごとの競合検出・解消フローを含むフローチャートを示している．

3.3　ソフトウェア詳細設計

開発者はソフトウェア詳細設計において，詳細設計書 (クラス図とシーケンス図) を作成する．ここでは，競合が発生しない場合のシーケンス図と競合が発生する場合のシーケンス図を必要な数だけ作成し，その処理にあわせたクラス図を作成する．

3.4　ソフトウェアソースコード作成および単体・統合テスト

ソフトウェアソースコード作成工程では詳細設計書に基づいて，実際に実装を行う．テスト工程では以下の 3 つのテストを順に行う．

（1） 詳細設計書にもとづいたクラスの単体テスト

（2） 方式設計書にもとづいた競合を考慮しないときの連携サービス単体の動作テスト

（3） 方式設計書にもとづいた連携サービス間の競合検出・解消のテスト

3.5　サービスデプロイおよびシステム結合テスト

デプロイ工程では，実際に連携サービスを HNS アプリケーションとして配置し，実行可能な状態にする．さらにシステム結合テストでは，実際の機器の動作状態を確認しながら仕様書および競合表との対応を確認する．具体的には，まず連携サービス単体で動作させ，家電機器が仕様書どおりに動作しているか検証する．次に，2 つの連携サービスを組み合わせ実行させ，競合検出・解消結果が競合表と合っているかを検証する．この作業を仕様書に記述された連携サービスの全てのペアにおいて行う．

4.　競合検出・解消基盤を用いた連携サービス開発

本稿では先行研究 [9] で提案した競合検出・解消基盤を利用した新しい連携サービス開発プロセスを提案する．図 3. に我々が提案する連携サービス開発プロセス (新規開発) を示した．提案プロセスでは，開発者は**連携サービス記述テンプレート**を利用して，連携サービス記述を作成する．作成された連携サービス記述は競合検出・解消基盤によって**模擬実行**され，発生する可能性のある全ての競合が競合表として生成される．開発者は，仕様書にもとづいて，自動生成された競合表の検証を行う．競

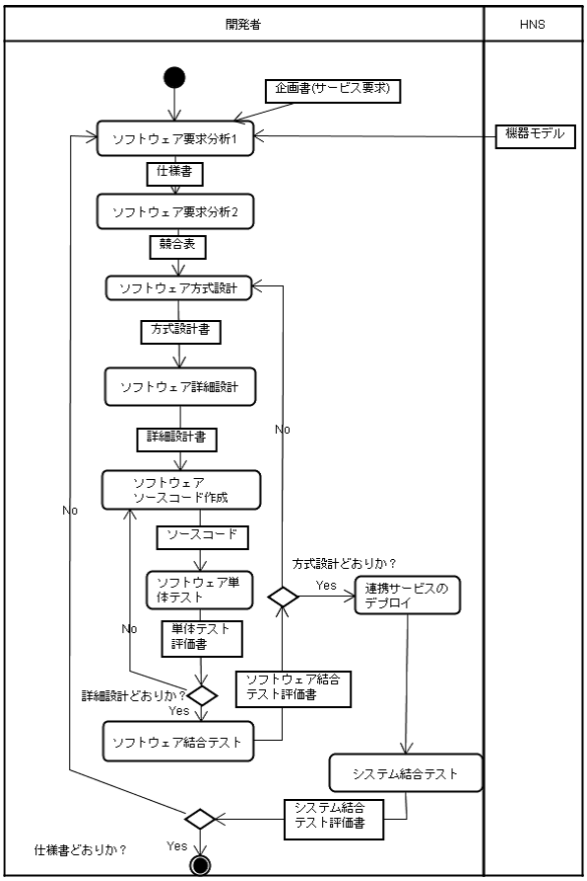

図 2　従来の連携サービス新規開発プロセス

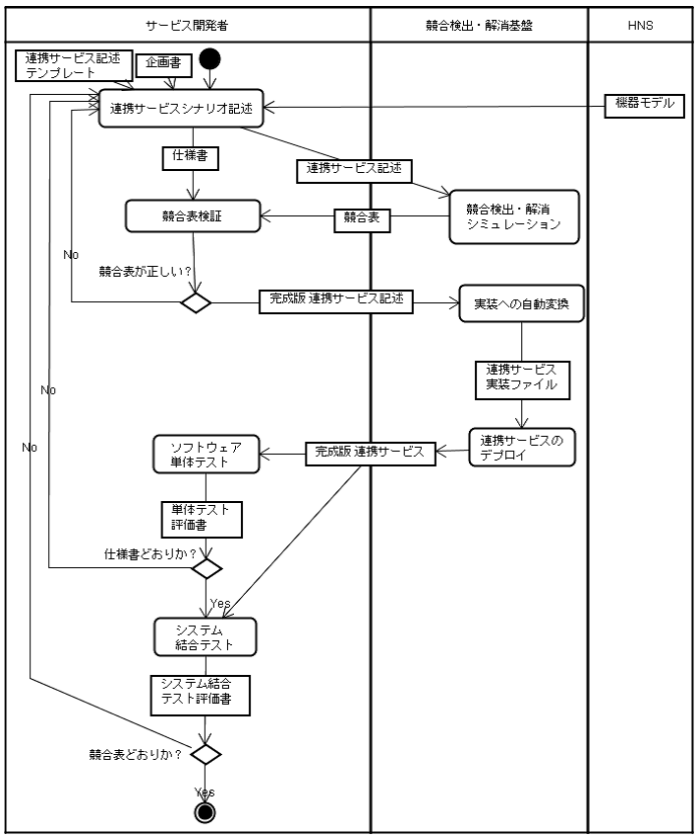

図 3　競合検出・解消基盤を用いた連携サービス新規開発プロセス

合表検証後，完成した連携サービス記述を競合検出・解消基盤を利用して**自動的に実装コードに変換**し，サービスとして配備される．競合表の作成や実装コードの自動生成を行うことで，従来開発で混入しがちであった人的要因によるバグを削減し，開発効率の改善が可能であると考えている．

以降では，提案プロセスの詳細について説明する．

4.1　連携サービスシナリオ記述

連携サービスは実行される機器および機器メソッドのシーケンスとして表現される．しかしながら，競合検出・解消を考慮する場合，メソッドのシーケンスのみでは不十分である．そこで我々は連携サービスにおけるサービス競合の検出・解消に関するこれまでの研究成果 [10] にもとづき，**連携サービスモデル**を構築した．我々の連携サービスモデルでは全ての連携サービスは開始シナリオと終了シナリオの二つの機器メソッドのシーケンスを持ち，機器メソッドごとに以下の項目を持つ．

優先度　：連携サービス間で競合が検出された場合，競合解消時には優先度の高い連携サービスが優先して実行される．

必須・通常　：サービス内で必ず実行されなければならない重要なメソッドか否かを指定する．競合解消時に，もし必須メソッドが停止されるようなことがあれば連携サービス全体が停止される．

中断・再開　：競合解消時にサービスを一時的に中断するか否かを指定する．中断可能な連携サービスは，競合する連携サービスが終了したときに自動的に再開される．

アクチベーション　：連携サービスの有効期間を示す．有効期間には，以下の 3 種類があり，その有効期間にある連携サービスのみが競合検出の対象となる．

ACT1:BeginEnd　ユーザによって開始された連携サービスは，同じようにユーザによって終了されるまで有効期間が続く．

ACT2:Timer　開始された連携サービスは，開始時に指定された時間が経過すると自動的に有効期間が終了する．

ACT3:Instant　連携サービスの有効期間は開始時の一瞬のみ．開始された後は有効期間範囲外となる．

本研究では，上記の連携サービスにもとづく**連携サービス記述テンプレート**を作成した．表 1 に，テンプレートに準拠して作成された DVD シアターサービスのサービス記述を示す．連携サービスシナリオ記述工程でこのテンプレートを利用することにより適切な要求分析が容易に実現可能となった．

4.2　競合検出・解消シミュレーションおよび競合表検証

3. 章で述べたように，従来法では要求分析工程において開発者自身が競合表を作成していた．我々が提案する開発プロセスでは，連携サービス記述にもとづいて競合検出・解消基盤が連携サービスを**模擬実行**し，競合表を自動生成する．開発者は作成した連携サービス記述をシミュレータに登録後，出力された競合表の検証作業を行う．競合表が仕様書や顧客の意図したものでなければ，連携サービス記述を修正し，再度シミュレータに登録を行う．競合表が自動生成されることで，競合検出時のバグの混入を防ぐことができる．また，連携サービス記述から

表 1　連携サービスモデルにもとづく DVD シアターサービス記述

各機器のWSDL		開始シナリオ						終了シナリオ			アクチベーション
機器名	WSDL	機器名	メソッド名	引数	優先度	必須・通常	再開可・不可	機器名	メソッド名	引数	
TV	http://1	TV	on		3	必須	可	TV	downVolume	10	BeginEnd
LIGHT	http://1	TV	upVolume	10	3	通常	可	TV	off		
CURTAIN	http://1	TV	changeInputScreen	6	3	必須	可	LIGHT	setBrightness	5	
DVD_RECO	http://1	LIGHT	setBrightness	1	3	通常	可	CURTAIN	open		
		CURTAIN	close		3	通常	可	DVD_RECORDER	changeHDDMode		
		DVD_RECORDER	on		3	必須	可	DVD_RECORDER	off		
		DVD_RECORDER	changeDVDMode		3	必須	可				
		DVD_RECORDER	play		3	必須	可				

競合表検証までの工程を繰り返し行うことが非常に容易であるため，結果として質の高い連携サービス記述の作成が可能となると考えられる．

4.3　連携サービスモデルから実装への自動変換

適切に作成された連携サービス記述は連携サービス開発に必要な情報を全て含んでいる．そこで我々は連携サービス記述をもとに，連携サービス実装を自動生成する仕組みを導入した．これにより，従来法で必要であったソフトウェア方式設計からソフトウェアコード作成までの工程が提案開発手法では不要となった．生成された実装コードは同じく自動的にサービスとしてデプロイされる．

4.4　ソフトウェア単体テストおよびシステム結合テスト

テスト工程における従来法との最大の違いは，実装への変換およびサービスデプロイが終わった時点で，既に連携サービスが正常に動作する状態になっているということである．開発者が関与する余地がないため，クラス単位での単体テストも行わない．ソフトウェア単体テスト工程では，デプロイされた連携サービスが単独で仕様書どおりに動作するかが確認される．同様にしてシステム結合テストでは，既に配備された連携サービスを対象に競合表および仕様書どおりに連携サービスが動作するかの確認を行う．

我々はこれらの従来連携サービス開発手法および提案開発手法を実際の連携サービス開発に適用し，比較を行った．次章では開発実験について詳述する．

5.　開 発 実 験

5.1　実 験 概 要

本開発実験では，新規連携サービス開発，連携サービスの追加，連携サービスの改訂という 3 つの事例を対象に従来開発プロセスおよび提案開発プロセスを実際に行った．開発者は 1 名．

新規に開発した連携サービスは以下の 3 種類である．

SS1：DVD シアターサービス　:1. 章に記述

SS2：おかえりサービス　:1. 章に記述

SS3：長期不在省エネサービス　：30 分以上の不在を感知すると，TV，DVD レコーダ，エアコン，リビング照明の電源を OFF にする．

連携サービス開発企画書には，上記の連携サービスの情報と，サービスの優先順位についての情報が記述されている．本実験での新規サービス開発時の優先順位は DVD シアターサービス = おかえりサービス > 長期不在省エネサービス とする．

連携サービスの追加開発では，以下の連携サービスを開発し，既存連携サービスと組み合わせる．

SS4 セキュリティビデオ監視サービス　：人感センサが感知すると，TV にセキュリティカメラの映像を映し出し，DVD レコーダで録画する．

連携サービスの追加によって，サービスの優先順位はセキュリティビデオ監視サービス > DVD シアターサービス = おかえりサービス > 長期不在省エネサービス となるものとする．

最後に連携サービスの改訂では，既存の連携サービスの優先順位の変更を行う．新しいサービスの優先順位は，セキュリティビデオ監視サービス > 長期不在省エネサービス > DVD シアターサービス = おかえりサービス となるものとする．

開発実験では共通の企画書をもとに，従来開発プロセスおよび提案開発プロセスで開発を行い，工程単位で工数およびバグ数を計測した．

表 2　工 数 比 較

	従来法			提案法		
工程	新規	追加	改訂	新規	追加	改訂
ソフトウェア要求分析(仕様書)	0.5	0.5	0.5	3	0.4	0.05
ソフトウェア要求分析(競合表)	6.5	1.5	0.5	1.5	0.1	0.05
ソフトウェア方式設計	12	1	0.5			
ソフトウェア詳細設計	13	1	0.5			
ソフトウェアソースコード作成	6.5	1.5	1			
ソフトウェア単体テスト	9	1.5	0.5	0.5	0.5	0.5
ソフトウェア結合テスト	1.5	1	0.5			
システム結合テスト	5.5	1.5	7	0.5	0.5	2
合計工数(人時)	54.5	9.5	11	5.5	1.5	2.6

表 3　従来法バグ数 (新規開発時)

		その工程で発見したバグ数							
		要求分析 1	要求分析 2	方式設計	詳細設計	ソースコード作成	単体テスト	SW 結合テスト	SYS 結合テスト	合計バグ数
工程	ソフトウェア要求分析(仕様書)	0	0	0	0	0	0	0	1	1
	ソフトウェア要求分析(競合表)		6	0	0	0	0	0	0	6
	ソフトウェア方式設計			6	1	0	0	0	0	7
	ソフトウェア詳細設計				4	6	0	0	0	10
	ソフトウェアソースコード作成					0	7	2	2	11
	ソフトウェア単体テスト						0	0	0	0
	ソフトウェア結合テスト							0	0	0
	システム結合テスト								0	0
	合計バグ数	0	6	6	5	6	7	2	3	35

表 4　提案法バグ数 (新規開発時)

		その工程で発見したバグ数				
		連携サービスシナリオ記述	競合表検証	ソフトウェア単体テスト	システム結合テスト	合計バグ数
工程	連携サービスシナリオ記述	1	2	0	0	3
	競合表検証		0	0	0	0
	ソフトウェア単体テスト			0	0	0
	システム結合テスト				0	0
	合計バグ数	1	2	0	0	3

5.2　実 験 結 果

表 2 に工数比較表，表 3 に従来法バグ数表，表 4 に提案法バグ数表を示す．表 2 には，各工程で費やした工数 (人時) が記されている．従来法と提案法では，工程が異なるため単純な比較は難しいが，従来法のソフトウェア要求分析 (仕様書) およびソフトウェア要求分析 (競合表) が提案法の連携サービスシナリオ記述および競合表検証にそれぞれ対応するものとして比較を

行っている．工数比較表が示すように，従来法と提案法では新規開発において 9.9 倍，追加開発において 6.3 倍，優先度改訂において 4.2 倍の差が観測されている．

表 3 と表 4 は，見出し行の工程で開発者が発見した，見出し列に示した工程で混入したバグ数を示している．例えば，ソフトウェア要求分析 (仕様書) と SYS 結合テストが交わるところに 1 と書いてあるが，これはシステム結合テスト時に開発者が発見したソフトウェア要求分析 (仕様書) 工程で混入したバグの数が 1 件であることを示している．両プロセスが必要とする工程が異なるため，工程ごとの単純な比較は難しいが，新規開発時の開発プロセス全体では，バグ数にして 11.7 倍の差が出ている．

5.3 考　　察

開発に要した工数という観点では，提案法の工程で従来法よりも工数が多くなっているものは連携サービスシナリオ記述工程のみである．この工程において提案法の工数が多くなっている最大の理由は，提案法における仕様書 (連携サービス記述) を作成する工程が手戻りを許容するプロセスであるということである．提案法では，連携サービスシナリオ記述工程および競合表検証工程は繰り返し行うことによって良い連携サービス記述を作成することがその目的となっている．競合表検証工程も同様にして繰り返し実行されるが，自動生成による効率の改善によって，工数の削減が実現されている．他の工程では，提案法，従来法ともに行うシステム結合テストにおいて，提案法のほうが工数が少なくなっている．これは従来法におけるバグの混入によるところが大きい．工数のバランスという観点では，提案法は設計および実装工程を削減し，全体の 55%を連携サービス記述の作成にあてている．提案法では，連携サービス記述の品質が連携サービスの品質にそのままつながる為，最も多くの工数が割かれるべきであると考えている．

今回の連携サービス開発実験では，従来法で発見されたバグの 80%がソフトウェア方式設計以降の工程で混入したものである．そのため，本提案法における連携サービスシナリオ記述から実装コードの自動生成は，バグの削減に非常に有効であることが確認された．また，バグの検出による手戻りは特にウォーターフォールモデルにおける開発コストの増大を招く．従来法におけるバグ発生による手戻りは全体の 54%のバグで発生しており，特にシステム結合テストで発見された仕様書のバグは開発作業全体に大きな影響を及ぼした．このバグは SS2：おかえりサービスの処理において，ユーザが帰宅時に一定時間照明を明るいまま制御し続けるという有効期間に関する処理を開発者が誤解していたことが原因で発生した．サービスの有効期間については，提案法ではアクチベーションという項目がテンプレート上に存在するため，同種のバグは提案法では混入しにくいと考えられる．バグによる手戻りについては，連携サービスシナリオ記述と競合表検証という繰り返し実行されることを想定している工程間でしか提案法では発生していない．これは開発工数全体の半分以上を連携サービスシナリオ記述にあてたことと，自動生成される競合表による連携サービス実行のシミュレーション機能が有効に働いたためであると考えられる．

6. おわりに

本稿では，競合検出・解消基盤を利用した新しい連携サービス開発プロセスの提案を行った．また従来法と提案法を比較する開発実験を行い，開発効率における提案法の優位性を示した．

今後は，発生する競合の検出や解消について対処方法をより容易に設計することができるように，競合表の見せ方を工夫するなど，さらなる改善を進めていきたいと考えている．

謝辞　この研究の一部は，科学技術研究費 (若手研究 B 20700027，21700077)，および，パナソニック電工株式会社の助成を受けて行われている．

文　　献

[1] 日立ホーム&ライフソリューション株式会社,“ ホラソネットワーク ”, http://www.horaso.com/

[2] パナソニック電工株式会社,“ ライフィニティ ”, http://denko.panasonic.biz/Ebox/kahs/

[3] 東芝,“ 東芝ネットワーク家電 Feminity ”, http://www3.toshiba.co.jp/feminity/about/index.html

[4] M. Nakamura, H. Igaki, and K. Matsumoto, ”Feature Interactions in Integrated Services of Networked Home Appliances -An Object-Oriented Approach-,” In Proc. of Int’l. Conf. on Feature Interactions in Telecommunication Networks and Distributed Systems (ICFI’05), pp.236-251,2005.

[5] M. Wilson, M. Kolberg, and E. H. Magill. Considering side effects in service interactions in home automation - an online approach. In *Proc. Int’l. Conf. on Feature Interactions in Software and Communication Systems (ICFI’07)*, pages 172–187, 2007.

[6] M. Nakamura, A. Tanaka, H. Igaki, H. Tamada, and K. ichi Matsumoto. Constructing home network systems and integrated services using legacy home appliances and web services. *International Journal of Web Services Research*, 5(1):82–98, January 2008.

[7] 吉村悠平, 井垣宏, 中村匡秀, ”ホームネットワークシステムにおけるサービス競合の動的検出・解消システムの設計と実装 ”, 電子情報通信学会技術研究報告,Vol.108, No.136, pp.35-40,July 2008.

[8] 吉村 悠平, 池上 弘祐, 井垣 宏, 中村 匡秀, ”ホームネットワークシステムにおける家電連携サービスのための競合解消方式の考察 ”, 電子情報通信学会技術研究報告, Vol.108, No.458, pp.439-444, March 2009.

[9] 吉村 悠平, 稲田 卓也, 井垣 宏, 中村 匡秀, “ SOA にもとづくホームネットワーク サービス競合検出・解消基盤の提案, ”情報処理学会研究報告, vol.2009-SE-166, no.4, November 2009.

[10] Masahide Nakamura, Hiroshi Igaki, Yuhei Yoshimura, and Kousuke Ikegami, ”Considering Online Feature Interaction Detection and Resolution for Integrated Services in Home Network System, ”In 10th International Conference on Feature Interactions in Telecommunications and Software Systems (ICFI2009), pp.191-206, June 2009.