

多言語サービス基盤のための連邦アーキテクチャ

村上 陽平^{†a)} 林 冬恵^{†b)} 石田 亨^{†c)} 田仲 正弘^{††d)}

Federation Architecture for Multilingual Service Infrastructure

Yohei MURAKAMI^{†a)}, Donghui LIN^{†b)}, Toru ISHIDA^{†c)}, and Masahiro TANAKA^{††d)}

あらまし サービスグリッドはサービス指向集合知のためのミドルウェアである。サービスを連携させるために、サービスレポジトリやサービス合成、アクセス制御、モニタリングなどの基盤機能群を備えている。このミドルウェアを用いて多言語サービス基盤を構築するには、多様な言語サービスがサービスグリッド上で接続される必要がある。しかしながら、単一の運営者だけでは、多様な言語を網羅して言語サービスを体系化し集約するのは困難である。そこで、地域ごとにサービスグリッドを構築しそれらを接続することで集合知を形成する必要がある。具体的には、汎用的なサービスグリッドを言語ドメインのサービスの管理に特化させるために、サービスのメタモデルを設計した。更に、サービスレポジトリを共有し、サービスグリッド間を越えてアクセス制御を行いながらサービスを連携させるためにサービスグリッドの連邦アーキテクチャを開発した。最後に連邦アーキテクチャを京都とバンコクの言語グリッドの連邦制運営に適用することでその有効性を検証した。

キーワード サービスグリッド, 連邦制, SOA, 集合知

1. ま え が き

集合知とは、多くの参加者によって作成された知識を集約し組み上げて新しい価値を形成するものである。Wikipediaのように、インターネット上で文章、写真、動画などコンテンツを集約したコンテンツ指向の集合知は世界規模で取り生まれ、人類の知識基盤を生み出している。一方、インターネット上には多様な言語資源（データ及びプログラム）があるにもかかわらず[1]、これらを組み上げて多言語サポートを行う集合知は形成されていない。これは、機械翻訳や用例対訳、対訳辞書といった複雑な知財を伴う言語資源を連携させる必要があるためである。これらの言語資源は所有権を保持したい提供者によって提供されるため、言語資源の代わりに言語資源にアクセスするサービスを集約し

て組み上げるサービス指向の集合知が有効である。

これまでもサービス連携のミドルウェアとしてESB (Enterprise Service Bus) が提案されている。ESBではサービスをプラグインするだけで、サービス間の通信が可能になる。ただし、ESBは企業内の閉じた環境を想定した基盤であるため、ESBの運営者はあらかじめサービス提供者と提供ポリシーを合意し、ESBにサービスを組み込む。一方、サービス指向の集合知は、サービスの集約を加速させるために、よりオープンな環境で参加者となるサービス提供者が自由にサービスを登録できる必要がある。オープンな環境では利用者をあらかじめ特定できないため、サービス提供者は利用者ごとに適宜提供ポリシーを設定できなければならない。

この要件を満たしサービスの集合知を形成する枠組みをサービスグリッドと呼び、我々はそのためのミドルウェアを開発している。このミドルウェアは、オープンな環境で利用者によるサービス連携やサービスの共有を可能にするための基盤機能を提供している[2]。オープンな環境では同じ機能を提供しているサービスでもインタフェースが異なる可能性があるため、サービスごとに連携プロセスを記述しなければならず、共有されるサービスの数が増えてもなかなか集合知が形

[†] 京都大学大学院社会情報学専攻, 京都市
Department of Social Informatics, Kyoto University, Kyoto-shi, 606-8501 Japan

^{††} (独) 情報通信研究機構, 京都市
National Institute of Information and Communications Technology, Kyoto-fu, 619-0289 Japan

a) E-mail: yohei@i.kyoto-u.ac.jp

b) E-mail: lindh@i.kyoto-u.ac.jp

c) E-mail: ishida@i.kyoto-u.ac.jp

d) E-mail: mtnk@nict.go.jp

成されない。このような問題を解決するために、サービスグリッドでは運営者がサービスタイプごとのインタフェースの標準化を行う。ただし、単一運営者だけでは、あらゆる言語サービスに精通することができないため、集合知の形成に必要な多様な言語サービスの体系化が期待できない。更に、言語サービスは言語の局所性から提供者が各地域に分散しており、単一運営者では各地域の言語サービス提供者の参加を募り多様な言語サービスを収集することが困難である。

そこで、本研究では地域ごとに言語サービスのサービスグリッドの運営を分担し、サービスグリッドの連邦制を組むことでサービス指向集合知に基づく多言語サービス基盤を構築することを目的とする。この目的を達成するために取り組んだ課題は以下の2点である。
サービスメタモデルの設計 複数の運営者の協働によって言語サービスを体系化するには、ドメイン固有の情報をアーキテクチャから切り離し、言語サービスドメインのサービス体系を定義するメタモデルが必要である。これにより言語サービスドメイン固有のモデルをデータとして交換することでサービスグリッド間の言語サービスの相互運用を実現する。

サービスグリッド連邦アーキテクチャの開発 サービスグリッド間の連携を実現するための連邦アーキテクチャが必要である。サービスグリッドごとに運営が異なるため、各サービスグリッドが管理する情報の共有範囲や、それぞれの責任範囲を明確にしたアーキテクチャにしなければならない。

2. サービス指向集合知

サービス指向の集合知とは、多くのサービス提供者によって作成された多様なサービスを集約して組み上げ、新たな価値を付加したサービスを提供するものである。サービスには、データやプログラムなどの資源へのアクセスを可能とする原子サービスと、複数の原子サービスを連携させた複合サービスがある。資源ではなくサービスを集約することで、提供者による資源の所有を妨げることなく集合知を形成することができる。集合知の基本性質である、独立性 (independence)、分散性 (decentralization)、多様性 (diversity)、集約性 (aggregation) のうち独立性と分散性はサービスの特性で満たされるため、サービス指向集合知の形成には多様性と集約性の実現が重要である [3]。

2.1 ユースケース

サービス指向集合知の多様性と集約性を実現するために、集合知の運営者は、サービスを集約し連携させる基盤を運営し、ユーザの追加、削除を行う必要がある。この基盤に求められるユースケースを図1に示す。

多様なサービスが共有されるように、運営者がサービスを収集し登録するのではなく、サービス提供者が自発的に原子サービスを登録できる必要がある。同様に複合サービスを登録するために、原子サービスを連携させるプロセスもプロセス設計者が自発的に登録できる必要がある。更に、コンテンツを紡ぐコンテンツ指向集合知では、運営者が統一的なライセンスを示していたが、サービス指向集合知ではサービス提供者のインセンティブを高めるためにもサービスの利用条件に関して提供者が個別に設定できなければならない。一方、サービス利用者は、提供者によって定められた利用条件の範囲内で、連携プロセスに従ってサービスを自由に組み合わせて集約できなければならない。

2.2 サービスグリッドアーキテクチャ

サービス指向集合知を形成するために、我々はサービスグリッドアーキテクチャに基づくミドルウェアを開発している [2]。図2は、サービスマネージャ、サービススーパーバイザ、グリッドコンポーザ、サービスデータベース、複合サービスコンテナから構成されるサービスグリッドアーキテクチャを示している。

サービスマネージャは、サービスグリッドに登録されるノードや、ユーザ、サービス、資源に関する情報を管理する。サービスに関する情報には、そのサービスのアクセス制御ポリシーやアクセスログも含まれる。これらの情報は全てサービスマネージャを介して登録されるため、サービスマネージャはサービス呼び出し以外のあらゆる機能のフロントエンドとしての役

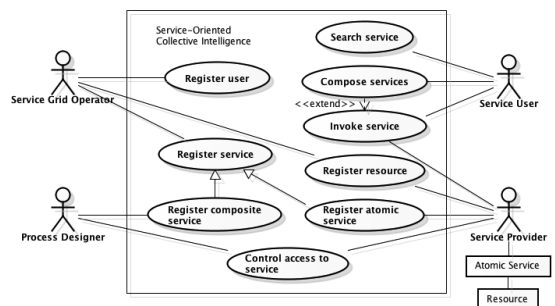


図1 サービス指向集合知のユースケース
 Fig.1 Use cases of service-oriented collective intelligence.

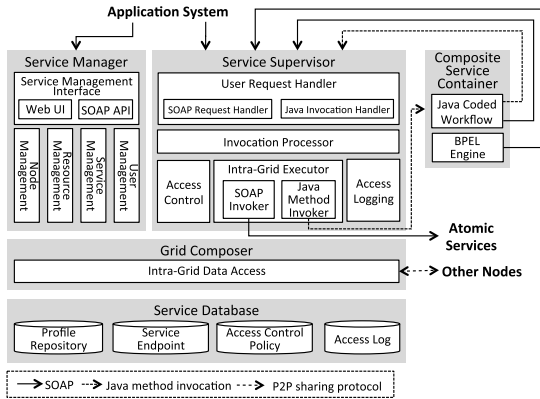


図 2 サービスグリッドアーキテクチャ [2]
Fig. 2 Service grid architecture [2].

割を担う。サービススーパーバイザは、サービス利用者のサービス呼び出しリクエストがサービス提供者によって設定されたアクセス制御ポリシーを満たしているかどうか検証するとともに、同種のサービス間の切り替えを行うことで連携プロセスの再利用性を高め、サービスの集約性を促進する。グリッドコンポーザは、サービスグリッド内の他のノードと接続し P2P グリッドネットワークを形成し、ノード間での情報の同期を行う。サービスデータベースは、サービスマネージャを通して登録された多様な情報やアクセスログを各レポジトリに格納する。最後に複合サービスコンテナは、配備された複合サービスのプロセスに従って、複数の原子サービスを呼び出して複合サービスを実行する。

3. サービスメタモデル

言語サービスの集合知を形成するためにサービスごとに関連プロセスを作成しては、集合知の形成が加速されない。特にサービスが増加するとその問題は顕著となる。そこで、集合知形成を促進するために、言語サービスのドメインモデルを定義するためのサービスメタモデルを運営者に提供する。これにより、各運営者は言語サービスを機能単位に分類してサービスタイプを定義し、各サービスタイプのインタフェースを標準化することができる。このサービスのインタフェースの標準化により、プロセス設計者の作成する連携プロセスの再利用性が高まり、サービス利用者は連携させるサービスを選択するだけで新たな価値を享受できるようになる。更に、作成された各言語サービスドメインモデルを相互に参照することで、サービスグリッドは多様な言語サービスを管理することができる。

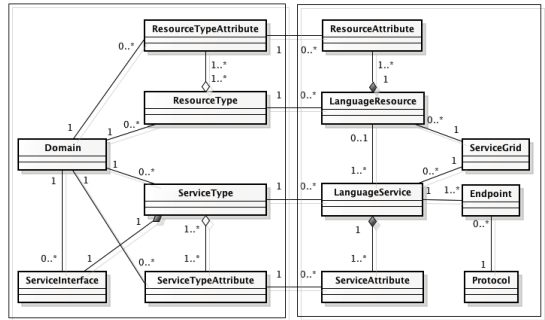


図 3 サービスメタモデルとインスタンスの関連
Fig. 3 Relation between service metamodel and instances.

図 3 の左部に示すように、サービスグリッドのメタモデルは、大きく分けてサービスタイプと資源タイプの定義に分けられる。資源タイプの定義には、資源タイプに属する資源の属性を定める資源タイプ属性が一つ以上存在する。この資源タイプ属性は、複数の資源タイプで利用される可能性がある。一方、サービスタイプの定義にも同様に、サービスタイプに属するサービスの属性を定めるサービスタイプ属性が一つ以上存在する。更に、サービスタイプの定義には、サービスのアクセス方法を定めたサービスインタフェースが必ず一つ含まれている。このインタフェースは、サービス提供者が自分の資源をサービス化する際や、プロセス設計者がサービスの連携プロセスを作成する際に用いられ、サービスの再利用を促進する。ただし、インタフェースの標準化により各サービスの独自性や多様性を損なわないように、既存サービスタイプを拡張した機能を有するサービスが存在する場合は、そのインタフェースを継承した新たなサービスタイプを定義するよう、ドメインモデル構築時に注意しなければならない。インタフェースを継承することで、多様性や独自性を確保しつつ、継承先のサービスタイプに属するサービスも継承元のサービスタイプの候補として再利用可能となる。なお、同じ機能を有しても入出力のデータ構造に互換性のないサービスは別のサービスタイプとして定義する必要がある。

一方、図 3 の右部に示すように、サービスグリッドで管理される言語サービスや言語資源のインスタンス情報は、メタモデルで定義された言語サービスドメインモデルと関連付けられる。言語資源は自身へのアクセスを提供する言語サービスとの関連を複数もち、言語サービスはインタフェースに準拠したリクエストを

受信するエンドポイントとの関連を複数もつことができる。前者は、一つの言語資源が異なるサービスタイプの言語サービスを複数提供できることを意味し、後者は、負荷に応じてサービスのエンドポイントを冗長化することができることを意味する。更に、そのエンドポイントがリクエストの送信方法を定めたプロトコルとの関連をもつため、サービスに対して多様なプロトコルへの拡張が可能である。

3.1 言語サービスドメインモデル

多言語サービス基盤を構築するために、サービスメタモデルを用いて定義した言語サービスドメインモデルの資源タイプとサービスタイプを表1と表2に示す。なお、これは初期のモデルであり、運営者は運営を通して新しいタイプを順次追加することができる。

言語サービスドメインモデルでは、13種の資源タイプと18種のサービスタイプを定義している。サービスタイプの方が多いのは、複合サービスのサービスタイプ (BackTranslation, MultihopTranslation, TranslationWithTemporalDictionary) が含まれるためである。また、どちらのタイプにも言語資源や言語サービスが処理できる言語や画像・音声のファイルタイプを設定するための supportedLanguages や supportedLanguagePairs, supportedLanguagePaths, supportedImageTypes, supportedAudioTypes, supportedVoiceTypes といった属性を定義している。一方、サービスタイプは、BilingualDictionary や ConceptDictionary などの言語データに検索機能を備えているため、利用可能なマッチング手法を指定するための matchingMethod を属性として更に定義している。なお、対訳辞書サービスには、収録語に最長一致する用語を入力テキストから全て抽出する機能を有するものがあるため、BilingualDictionary のインタフェースを継承した BilingualDictionaryWithLongestMatchSearch も定義している。一方、係り受け解析サービスには、入力テキストのものと形態素集合のものがあり、データ構造に互換性がないため、それぞれ DependencyParse と MorphemeBasedDependencyParse を定義している。

この言語サービスドメインモデルを用いて、言語資源や言語サービスのプロファイル情報及びインタフェースが管理される。例えば、言語資源である係り受け解

表1 言語サービスドメインモデル (資源)
Table 1 Language service domain model (Resource).

ResourceType	ResourceTypeAttribute
BilingualDictionary	supportedLanguagePairs
ConceptDictionary	supportedLanguages
DependencyParser	supportedLanguages
DialogCorpus	supportedLanguages
LanguageIdentification	supportedEncodings, supportedLanguages
MorphologicalAnalyzer	supportedLanguages
ParallelText	supportedLanguagePairs
Paraphraser	supportedLanguages
PictogramDictionary	supportedLanguages, supportedImageTypes
SimilarityCalculator	supportedLanguages
SpeechRecognition	supportedLanguages, supportedAudioTypes, supportedVoiceTypes
TextToSpeech	supportedLanguages, supportedAudioTypes, supportedVoiceTypes
Translator	supportedLanguagePairs

表2 言語サービスドメインモデル (サービス)
Table 2 Language service domain model (Service).

ServiceType	ServiceTypeAttribute ^(注1)	ServiceInterface ^(注2)
BackTranslation	supportedLanguagePaths	backtranslate
BilingualDictionary	matchingMethods	search
BilingualDictionaryWithLongestMatch	matchingMethods	search, searchLongestMatchingTerms
ConceptDictionary	matchingMethods	searchConcepts, getRelatedConcepts
DependencyParse		parseDependency
DialogCorpus	matchingMethods	search
LanguageIdentification		identify
MorphemeBasedDependencyParse		parseDependency
MorphologicalAnalysis		analyze
MultihopTranslation	supportedLanguagePaths	multihopTranslate
ParallelText	matchingMethods	search
Paraphrase		paraphrase
PictogramDictionary	matchingMethods	search
SimilarityCalculation		calculate
SpeechRecognition		recognize
TextToSpeech		speak
Translation		translate
TranslationWithTemporalDictionary	supportedLanguagePairs	translate

析プログラム CaboCha は、supportedLanguages 属性として日本語が設定されている。一方、言語サービスとしての CaboCha は、係り受け解析サービスタイプと形態素解析サービスタイプの両方のインスタンスである。これは CaboCha が形態素解析プログラムの機能を内包しており、形態素解析の機能もサービスとして提供できるためである。

(注1)：サービスタイプ用に新たに追加された属性のみ記載

(注2)：オペレーション名のみ記載

4. サービスグリッド連邦アーキテクチャ

サービスメタモデルを用いて各運営者が言語サービスドメインモデルを定義すると、サービスグリッドの連邦制によりそれらを相互に参照することで多様な言語サービスを共有でき、集合知の多様性を加速させることができる。一方のサービスグリッドへ参加している提供者に他方への参加を求めることもできるが、その交渉のコストが高いため、運営者が代表して連携するサービスグリッドの連邦制が妥当である。そこで、我々はサービスグリッド連邦アーキテクチャを提案する。本アーキテクチャは、自身のサービスグリッド上のサービスのアクセス制御やサービス連携に責任をもちつつ、連邦制を介して他のサービスグリッドへの透過的なアクセスを提供する。既存のグリッドミドルウェア (Globus Toolkit) は複数組織のサービスを連携可能とする仮想組織を形成するが、ここでは仮想組織が一つのサービスグリッドに対応する。したがって、サービスグリッドの連邦制は仮想組織の連邦制を意味し、このような機能は既存のグリッドミドルウェアにはない本アーキテクチャ独自のものである。

図 2 のサービスグリッドアーキテクチャを拡張したサービスグリッド連邦アーキテクチャを図 4 に示す。既存アーキテクチャにグリッド間実行部 (Inter-Grid Executor) とグリッド間データアクセス部 (Inter-Grid Data Access) が追加されている。また、サービスデータベースにも新たにサービスドメインモデルと連邦リストの二つのレポジトリが追加されている。

グリッド間実行部は、受信したリクエストの宛先が他の連携サービスグリッド上のサービスであった場合

に、サービスグリッド間を越えてサービスを呼び出す役割を担う。これは同一サービスグリッド内のサービスを呼び出すグリッド内実行部 (Intra-Grid Executor) との大きな違いである。一方、グリッド間データアクセス部は他の連携サービスグリッドとの情報共有の役割を担う。共有する情報は、ユーザや資源、サービスなどのプロファイル情報と言語サービスドメインモデルといった全ステークホルダーに公開されたパブリックな情報のみである。これは利用者の認証情報や利用履歴、サービスのエンドポイント情報、アクセスログ、アクセス制御ポリシーといった同一サービスグリッド内に閉じたプライベートな情報を他のノードと共有するグリッド内データアクセス部 (Intra-Grid Data Access) との大きな違いである。

以降では、サービスグリッド間の連邦関係の確立時に互いに発行される証明書を用いた、連邦制での情報共有、サービス連携の詳細について説明する。権限委譲によるサービス連携のためにプロキシ証明書を用いていた OGSA (Open Grid Service Architecture)^(注3)とは異なり、サービスグリッドの連邦制では、各運営者が利用者の認証やサービスの認可に責任をもつため、リクエストの送信元の利用者ではなく連携サービスグリッドの運営者かどうかの証明が重要であり、リクエストごとに連邦関係の証明書を用いる。

4.1 サービスグリッド間の情報共有

連邦関係の依頼元のサービスグリッドは、依頼先のサービスグリッドの管理するプロファイル情報 (ユーザ、資源、サービス)と言語サービスドメインモデルを取得し、自分のサービス利用者 (連携利用者) に対して各種プロファイル情報を提供する。これにより連携利用者は依頼先のサービスグリッドのサービスの指定や言語サービスドメインの参照が可能になる。一方、依頼先の運営者は、依頼元の運営者の管理するユーザプロファイル情報のみを取得し、自分のサービス提供者にその情報を提供する。これによりサービス提供者による連携利用者のアクセス制御が可能になる。

このように、依頼元の運営者は自分の管理するサービス情報を依頼先のサービスグリッド運営者に提供せず、非対称な情報共有を行うことで、互いの運営者の立場の違いを明確にしている。これは運営者間で目的が異なり、各サービスグリッドで必要なサービスが等

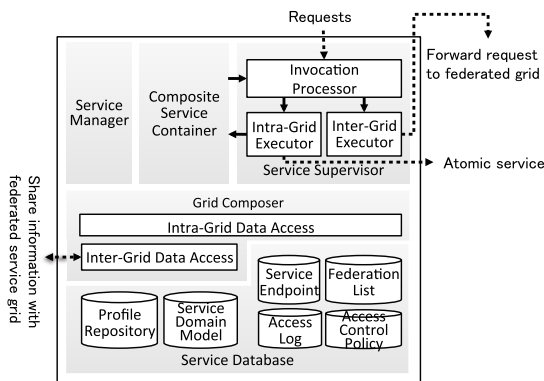


図 4 サービスグリッド連邦アーキテクチャ
Fig. 4 Service grid federation architecture.

(注3) : <http://toolkit.globus.org/alliance/publications/papers/ogsa.pdf>

しいとは限らないためである。したがって、両サービスグリッド間でのサービスの相互利用を可能にするには、双方向に連邦関係の依頼を行う必要がある。

4.2 サービスグリッド間のサービス連携

連邦制による言語サービスドメインモデルやプロファイル情報の共有によって収集された多様な言語サービスを集約するために、サービスグリッドを跨ぐサービス連携が必要である。図5に、サービスグリッドを跨ぐサービス連携時のメッセージングを示す。この図ではサービスグリッドの応用システムが、サービスグリッドと連邦関係にある連携サービスグリッド (Affiliated Service Grid) 上の複合サービスにサービスグリッド上の原子サービスをバインドして呼び出している。

サービスグリッドを跨ぐサービス連携を行う場合、まずは互いのサービススーパーバイザ内の呼び出し処理部 (Invocation Processor) が、呼び出し元が自身のサービスグリッドの利用者か、連邦関係にある連携サービスグリッドかの認証を行う。次に呼び出し先のサービスに応じてグリッド内実行部とグリッド間実行部を使い分け、サービスを連携している。呼び出し先のサービスが自身のサービスグリッドに属する場合、グリッド内実行部によってサービスを呼び出す。一方、異なる場合は、グリッド間実行部が呼び出し先のサービスを管理している連携サービスグリッドにリクエストメッセージを転送する。

グリッド内実行部では、呼び出し先サービスのエンドポイントリストを取得し、負荷の少ないエンドポイントをラウンドロビンや応答時間の平均を用いて選択してサービスを呼び出す。一方、グリッド間実行部は、呼び出し先サービスの属する連携サービスグリッドのURLを連邦リストから取得して、リクエストメッセージを転送するだけである。アクセス制御やエンドポイ

ントの選択は、アクセス制御ポリシーやサービスのエンドポイント情報を管理している呼び出し先の連携サービスグリッドに委譲する。このとき、連邦関係にある連携サービスグリッドからの転送であることを証明するために、連邦制の証明書をヘッダーに挿入する。

5. 応用及び考察

本章では、サービスグリッドの連邦アーキテクチャによる多言語サービス基盤の有効性を検証するために、言語グリッドの構築事例を説明するとともに、言語サービスの集合知形成の効率性について考察する。

5.1 言語グリッド

言語の壁を越えるために、インターネット上の言語資源のアクセシビリティやユーザビリティを向上させる多言語サービス基盤として「言語グリッド」が提案されている [4]。

言語グリッドは、言語サービスを共有し、それらを組み合わせて異文化コラボレーションを支援するためのサービスを構築することを可能にする。オープンソースとして公開された言語処理プログラムのパイプライン処理によって大量のテキストを解析することが目的であった UIMA [5] などの既存のフレームワークとは異なり、言語グリッドは異文化コラボレーションを支援するために、機械翻訳や用例対訳、対訳辞書といった複雑な知財を伴う言語資源を連携させる必要がある。これらの言語資源は所有権を保持したい提供者によって提供されるため、言語グリッドは、サービスを自由に連携させたい利用者のインセンティブと、特定の利用者だけに利用させたい提供者のインセンティブを調整しつつ、言語サービスの集合知を形成する。

5.2 言語グリッドの連邦制運営

言語グリッドの5年間の運営経験から様々な知見が得られている。連邦制運営の重要性を認識したこともその一つである。いかに国際的な活動を意識しても、運営者の近くから参加組織が増えていく。運営者が日本にいるために、言語グリッド利用者全149組織のうち70~80%が日本のグループである。しかし、このような局所性は地域性に留まらない。運営者が大学であるために、言語グリッド利用者の70~80%が大学や研究機関などの学術機関で占められている。こうした事実は、言語グリッドの運営者が世界中に、また様々な機関に分散し連携する必要があることを示している。サービスグリッドの連邦アーキテクチャの提案はこの経験に基づいている。実際、2011年2月からタ

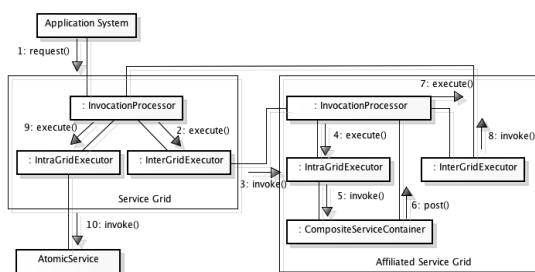


図5 連邦制サービスグリッドにおけるサービス連携
Fig. 5 Service composition in service grid federation.

イの国立研究所 (NECTEC) が新たな運営者となりバンコクに言語グリッドを立ちあげ、日本と連携して連邦制の運営を開始している。この結果、これまで登録されていなかったベンガル語やラオス語、モンゴル語、シンハラ語など東南アジアの言語に関する概念辞書へのアクセスも可能になっている。

5.3 集合知形成の効率性

サービスグリッドの連邦アーキテクチャを導入することで、言語サービスの集合知形成がどの程度効率化されたかを考察する。

集合知形成の効率性は工数とサービスの組み合わせのバリエーションによって評価することができる。京都とバンコクの言語グリッドの連邦制運営を例に考察する。現在、59 言語 178 サービスが両言語グリッドの参加組織によって共有されている。このうち京都の言語グリッドには 51 言語 139 サービス、バンコクの言語グリッドには 14 言語 35 サービスが登録されている。これらのサービスの実行をサービスの登録されているサービスグリッドにグリッド間実行部を介して委譲することで、バンコクの言語グリッド運営者は、京都に登録されている言語資源の原子サービスの開発及び配備の工数をかけずに、139 サービスをバンコクの言語グリッドの利用者に提供することができている。一方、京都の言語グリッド運営者はこれまで収集が困難であった東南アジアの言語資源を、バンコクの言語グリッドを介することで、原子サービスの開発及び配備の工数をかけずに 35 サービスも利用することができ、新たに 8 言語に対応できるようになっている。

また、連邦制により多様な言語やサービスをカバーすることで、サービスの組み合わせにも大きな影響を与える。例えば、専門辞書、形態素解析、機械翻訳サービスの三つを組み合わせたドメイン特化翻訳サービスでは、中国語、タイ語、タガログ語、インドネシア語、マレー語と英語間の翻訳時に、それぞれ京都言語グリッドで 170 通り、28 通り、48 通り、8 通り、3 通り、バンコク言語グリッドで 0 通り、8 通り、0 通り、0 通り、0 通りの実現方法があるのに対し、連邦制によって相互利用が可能になることで、お互いに足りない言語サービスを補うことで、255 通り、126 通り、64 通り、16 通り、6 通りの実現方法に数を大幅に増やしている。これらの結果から、連邦アーキテクチャがサービス指向集合知の形成を大幅に効率化していることが分かる。

6. 関連研究

本論文で提案したアーキテクチャは、複数のサービス連携基盤を接続することで、大規模かつ多様なサービスの連携を実現する。これまでも、グリッドコンピューティングの分野では、利用者の要求を満たすのにローカルリソースが十分でないときに、連邦制をとった他のクラスターを共有し、透過的に計算資源を利用可能とする試みが行われてきた [6], [7]。本研究では、計算資源ではなくサービスを共有し透過的に利用可能とする点で異なる。

一方、サービスの分野では、サービスの再利用性を高めるために、UBR (UDDI Business Registry) のような中央集権的なレジストリが存在しない状況で、個別に管理されたサービスレジストリを連邦制で接続させる、DIRE と呼ばれる試みが行われている [8]。連邦したレジストリ間で、登録されたプロファイル情報を publish することで情報を共有し、必要なサービスのタイプを Subscribe することでレジストリは興味のあるサービスのプロファイル情報を収集する。この手法は、連邦したサービスレジストリ間でサービスのプロファイル情報を効率的に伝搬させるが、本研究で提案されたように連邦したサービス連携基盤間を越えてサービスの連携実行を実現するものではない。

SOA4All では、トップダウンなアプローチで全ての ESB を連結し、グローバルなサービスクラウドの実現を目指している [9], [10]。ただし、SOA4All では、多様なサービスの連携を実現できる環境はあるものの、実際に連携させるプロセスは利用者が個別に作成しなければならぬ。サービス指向の集合知では、より手軽に利用者が新しい価値を享受できるように、プロセスの再利用を促進するサービスインタフェースの標準化の枠組みが必要である。

7. むすび

本論文では、地域ごとに言語サービスドメインのサービスグリッドを構築し、複数のサービスグリッドの連邦制により言語サービスの集合知を拡大させ、多言語サービス基盤を構築する方法を提案した。これを実現するために、我々の取り組んだ課題は以下の二つである。

サービスメタモデルの設計 言語サービスドメインモデルを定義するためのサービスメタモデルを構築した。これにより複数のサービスグリッド運営者が、サービ

スグリッド上で管理する言語資源や言語サービスのタイプとその属性、そして言語サービスの標準インタフェースを協働で定義することができる。その結果、多様な言語サービスの体系化が可能であり、言語サービスの集合知の形成を促進している。

サービスグリッド連邦アーキテクチャの開発 サービスグリッド間の連携を実現するサービスグリッド連邦アーキテクチャを構築した。これにより、言語サービスドメインモデルをサービスグリッドに適用した言語グリッドの連邦制運営を実現することで、複合サービスの実現方法の組み合わせを大幅に増加させることに成功している。

謝辞 本研究は、日本学術振興会科学研究費基盤研究(S) (24220002, 平成24年度～28年度)及び科学技術振興機構「問題解決型サービス科学研究開発プログラム」の補助を受けた。

文 献

- [1] K. Choukri, "Recent activities within the European language resources association: Issues on sharing language resources and evaluation," Proc. Fourth International Conference on Language Resources and Evaluation (LREC'04), pp.933-936, 2004.
- [2] Y. Murakami, D. Lin, M. Tanaka, T. Nakaguchi, and T. Ishida, "Service grid architecture," in The Language Grid: Service-Oriented Collective Intelligence for Language Resource Interoperability, ed. T. Ishida, pp.19-34, Springer, 2011.
- [3] J. Surowiecki, The Wisdom of Crowds: Why the Many Are Smarter Than the Few and How Collective Wisdom Shapes Business, Economies, Societies and Nations, Doubleday, 2004.
- [4] T. Ishida, "Language grid: An infrastructure for intercultural collaboration," Proc. IEEE/IPSJ Symposium on Applications and the Internet (SAINT'06), pp.96-100, 2006.
- [5] D. Ferrucci and A. Lally, "UIMA: An architectural approach to unstructured information processing in the corporate research environment," J. Natural Language Engineering, vol.10, pp.327-348, 2004.
- [6] R. Ranjan, A. Harwood, and R. Buyya, "A case for cooperative and incentive-based federation of distributed cluster," Future Generation Computer Systems, vol.24, no.4, pp.280-295, 2008.
- [7] M.D. Assuncao, R. Buyya, and S. Venugopal, "InterGrid: A case for internetworking island of grids," Journal Concurrency and Computation: Practice & Experience, vol.20, no.8, pp.997-1024, 2008.
- [8] L. Baresi and M. Miraz, "A distributed approach for the federation of heterogeneous registries," Proc. International Conference on Service Oriented Computing (ICSOC-06), pp.240-251, 2006.

- [9] F. Baude, I. Filali, F. Huet, V. Legrand, E. Mathias, P. Merle, C. Ruz, R. Krummenacher, E. Simperi, C. Hammerling, and J. Lorre, "ESB federation for large-scale SOA," Proc. ACM Symposium on Applied Computing (SAC-10), pp.2459-2466, 2010.

- [10] F. Lecue, R. Gonzalez, Y. Gorrongoitia, M. Radzinski, and M. Villa, "SOA4All: An innovative integrated approach to services composition," Proc. IEEE International Conference on Web Services (ICWS-10), pp.58-67, 2010.

(平成25年9月6日受付, 12月24日再受付)



村上 陽平 (正員)

2003 京大大学院修士課程了。2006 同大学院博士課程了。博士(情報学)。現在、京都大学大学院特定研究員。電子情報通信学会サービスコンピューティング研究専門委員会委員長を務める。サービス指向の集合知形成に取り組む。



林 冬恵 (正員)

2005 上海交通大学修士課程了。2008 京大大学院博士課程了。博士(情報学)。現在、京都大学大学院特定助教。サービスコンピューティング、異文化コラボレーション、クラウドソーシング、サービス科学の研究に従事。



石田 亨 (正員：フェロー)

1976 京大・工・情報工学卒, 1978 同大学院修士課程了。同年日本電信電話公社電気通信研究所入所。現在、情報学研究科社会情報学専攻教授, この間、ミュンヘン工科大学, パリ第六大学, メリーランド大学, 上海交通大学, 清華大学客員教授などを経験。工博。情報処理学会, IEEE フェロー。デジタルシティ, 言語グリッド, 異文化コラボレーションなど情報技術と社会をつなぐ研究プロジェクトを推進。



田仲 正弘 (正員)

2005 京大大学院修士課程了。2009 同大学院博士課程了。博士(情報学)。現在、独立行政法人情報通信研究機構研究員。サービスコンピューティングによる大規模情報分析基盤の研究に従事。