Workflow-as-a-ServiceのためのGlobal Social Service Networkを使ったWeb-Scale Functional Mapの構成

Tetsuya Tashiro, Wuhui Chen, Incheon Paik

School of Computer Science and Engineering University of Aizu

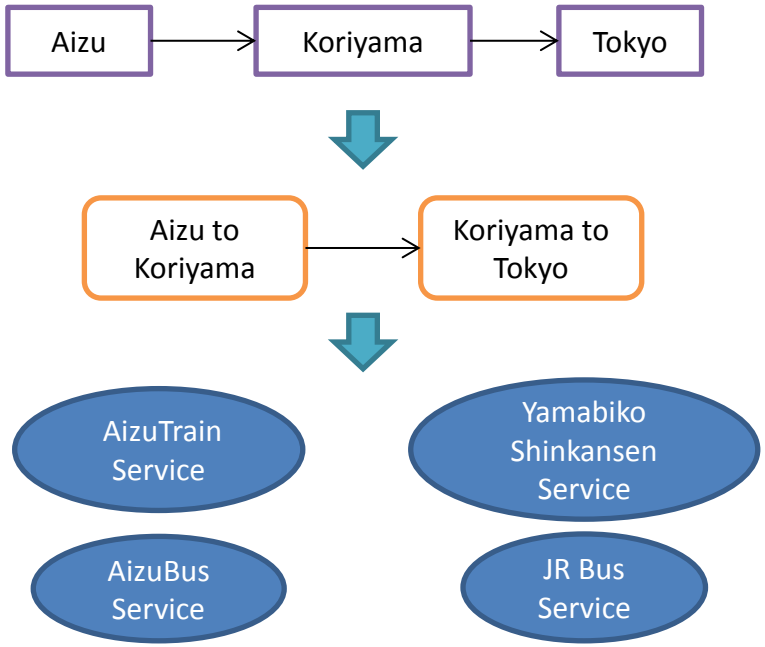
Outline

- Motivation
- Introduction
- Previous Work
  - Automatic Service Composition
- Linked Social Service
  - Social Link
  - Linked Social Service-specific principles
- Global Social Service Network
- Web-Scale Functional Map
  - Input/Output tree
- Workflow as a Service
- Example for Scenario
- Conclusion
- Future work

Motivation

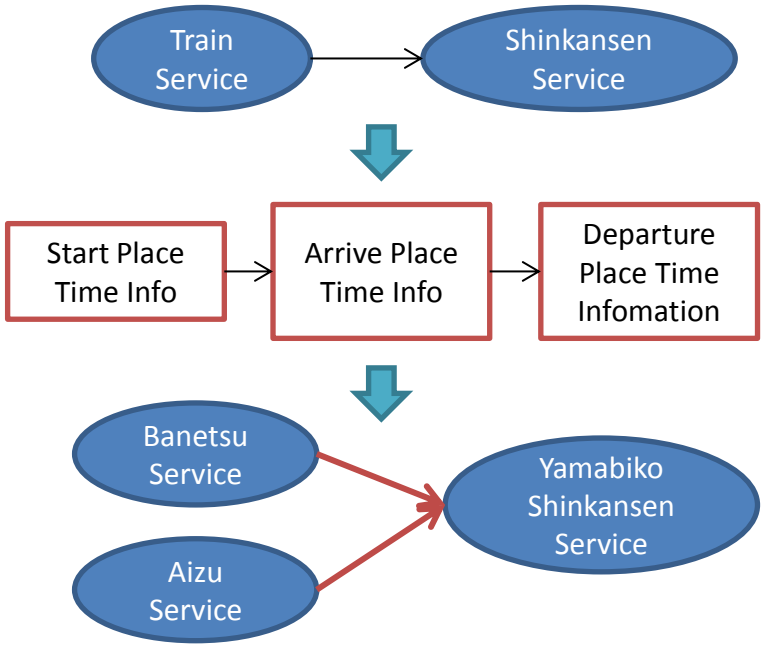
Traditionally approach

今までのアプローチでは、サービス同士の関係と関係なくサービス合成をしていた。したがって、サービスは孤立していてサービスの発見やサービス合成を妨げています。

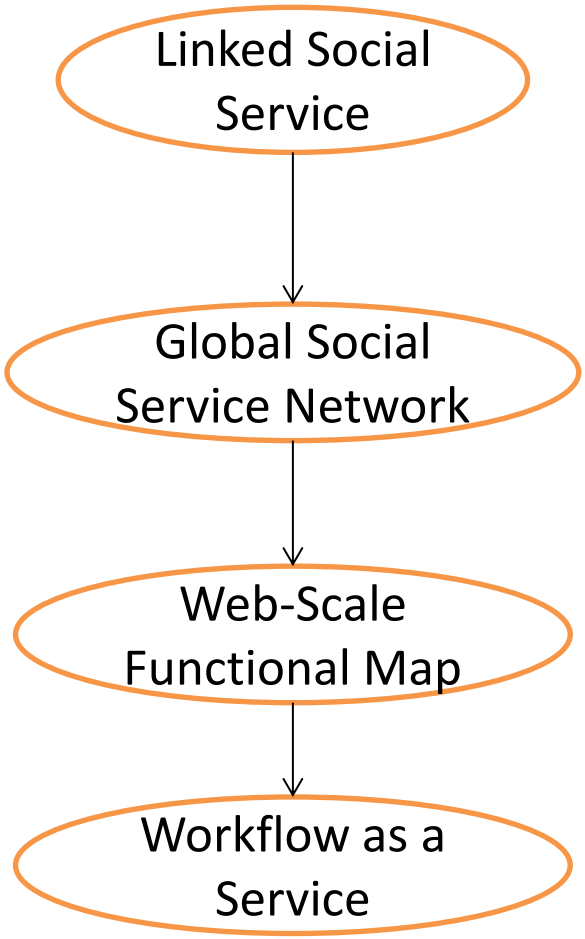


Our approach

私たちのアプローチではサービス同士の関係を基にサービス間にリンクを付けました。その結果私たちのアプローチでは、サービスの発見、合成が簡単にできるようになりました。



Entire Architecture of our approach



これは私たちのアプローチの全体の構造になります。私たちの研究は、サービス合成のためのabstract workflow を構成する新しい方法を提案します。

1. 最初に、サービス同士の関係を基にサービス同士をリンクし、Linked Social Serviceを構成します。
2. 次に、Linked Social Serviceを基にGlobal Social Service Networkを構成します。
3. 次に、Global Social Service Network を基にサービス合成をするために、Web-Scale Functional Map を構成します。
4. 最後に、サービス合成のためにworkflowを構成します。

Linked Social Service

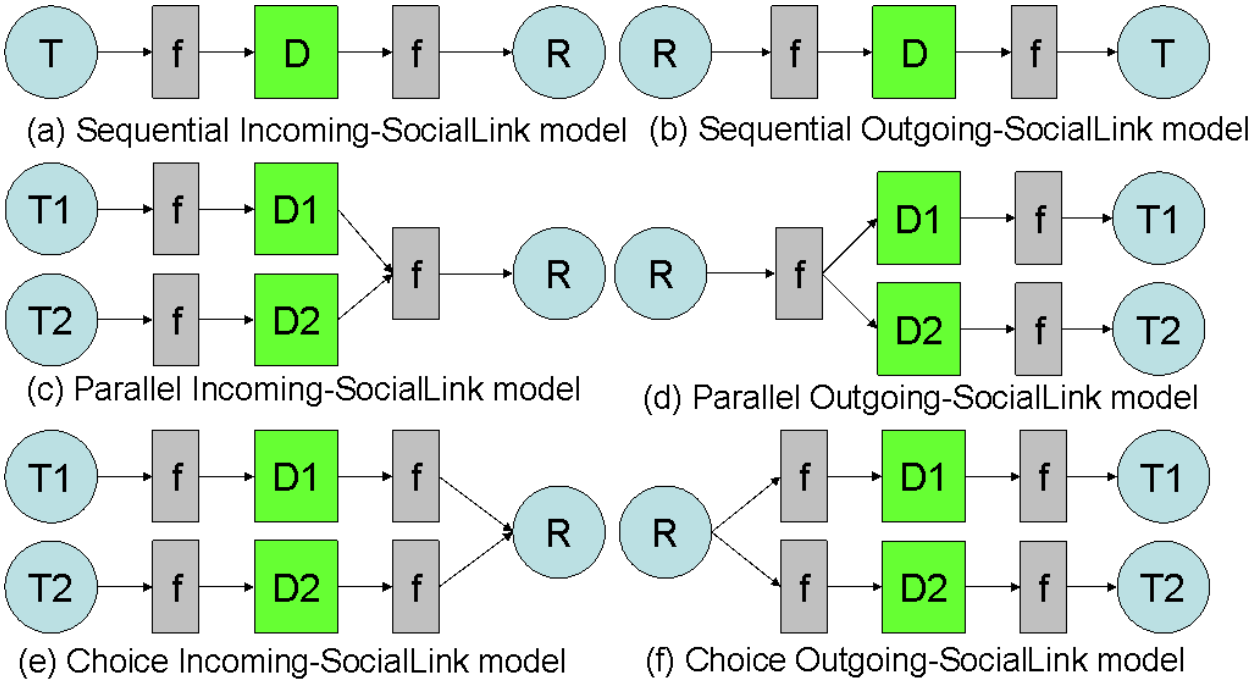
- Linked Social Service-specific principles

- Social Link

Linked Social Service-specific principles

1. サービスはlinked data principlesのように公開されます。
2. サービスはサービス同士の機能性から関係あるサービスにリンクします。
3. input/outputのresource servicesとoutput/inputのtarget servicesの間の関係をSocialLinkと呼びます。
4. linked social serviceのinputとoutputの関係をServiceLinkと呼びます。
5. Sequenceのlinks, SocialLink とServiceLinkの両方,は意味のあるワークフローの一種である。

Social Link



Difinition: An SocialLink model is quintuple (Ti,Ri,D,f,L(Ti,Ri))

Ti are a set of target services, which may refer to a set of nodes on the open web, i is the number of services.

Ri are a resource service, which may refer to a known service, i is the number of services.

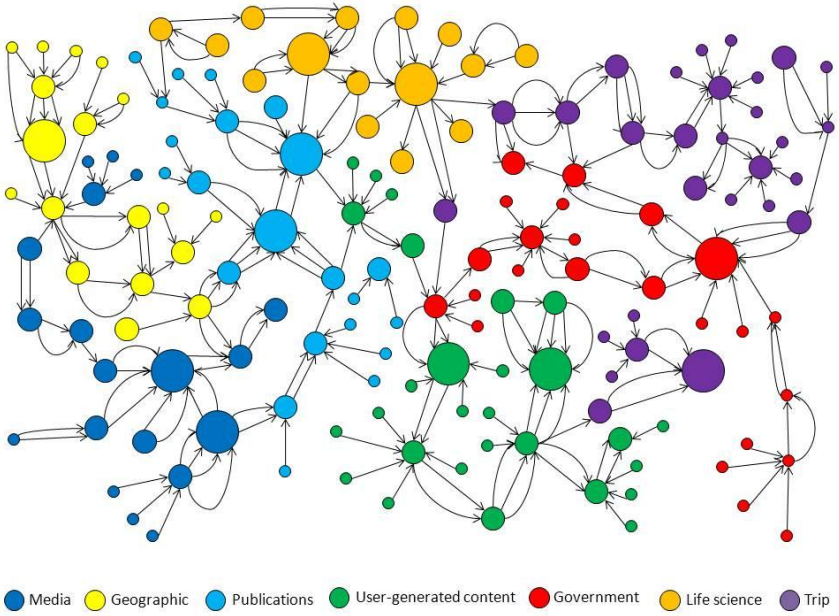
D is a set of data, which maybe is a arbitrary datatype.

f is a transform that maps input/output to D.

L(Ti,Ri) is a set of SocialLink, which is the data correlation relationship of Ti and Ri.

Global Social Service Network

- social linksを使って, social services communityを構成し、global social service networkと呼びます。
- Global social service networkは有効グラフです。



Web-Scale Functional Map

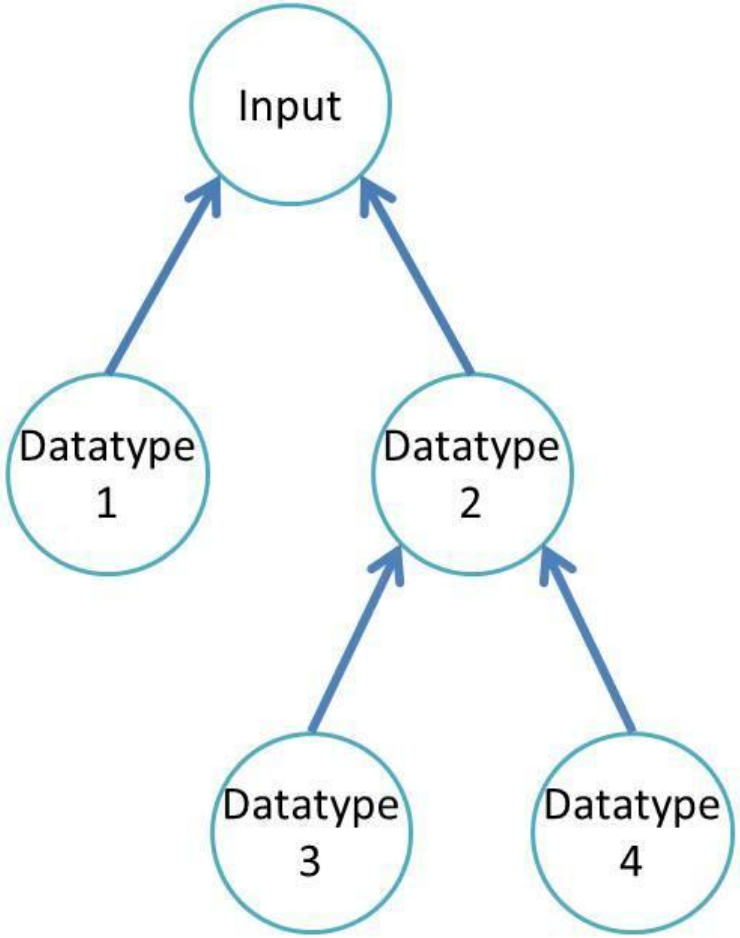
- 私たちはサービスを合成してworkflowを作るために、global social service network を抽象化し、Web-Scale Functional Map にする方法を提案する。
- サービスの消費者はWeb-Scale Functional Mapを使うことで、直観的にサービスがどういう合成をしているか、知ることができます。
- web-scale functional mapを構成するために, 私たちはinput/output treeを定義します。次に、input/output tree を使ってweb-scale functional mapを定義します。

Input/Output tree

Definition (Input/Output tree).

The input tree and output tree is a hierarchical tree structure with a set of linked nodes $\mathrm{T} = \left(N_r, N_p, N_c, L\right)$, where

- N_r is root node, its value is input or output.
- N_c is child node, its value represents a primary datatype or concrete datatype.
- L is a link between N_p and N_c, it represents the relationship of N_p and N_c, such as subclass of, aggregation, generation.

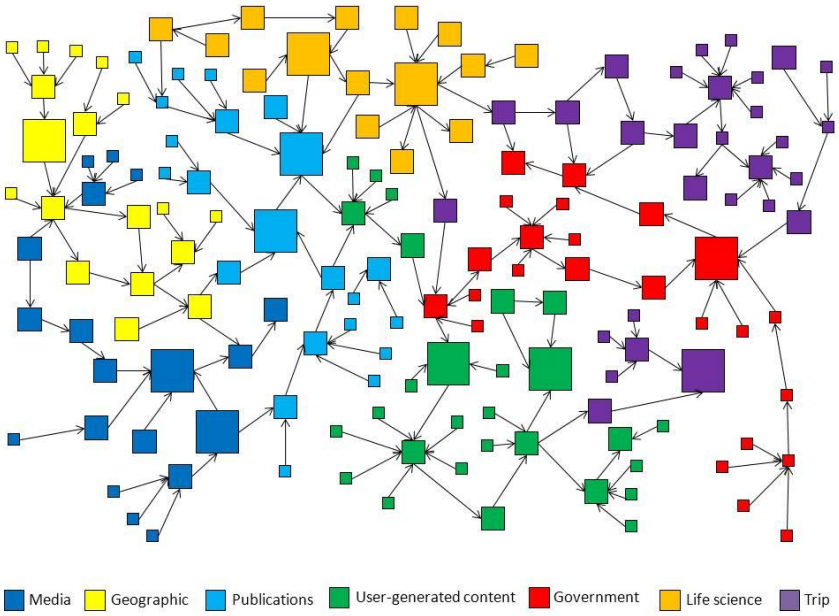


Web-Scale Functional Map Definition

Definition 6(Web-Scale Functional Map).

Web-Scale Functional Mapはglobal social service networkの機能性を抽象化したものです。Web-Scale Functional Map は有効グラフ $G'^{(V',E')}$

- *V' はノードの集合を示しています。各ノードはlinked social serviceの input/output tree で定義されます。*
- *E' は有向辺の集合を示しています。各辺はlinked social serviceのinputとoutput の関係を示すService Link と一致します。*
- Note that there is a service template for each pair (V'_i, E'_j, V'_j), where E'_j link from V'_i to V'_j.



Constructing Functional Map Rules

1. ノードはglobal social service network の中のlinked social serviceからinput/output treeを抽出することで構成します。
2. ノード間の関係はsocial linkと同じように定義します。
3. ノードのタイプがsequenceまたはchoiceの時に、ノードはWeb-Scale Functional Mapに追加されます。
4. ノードのタイプがparallelの時、ノードとパートナーとなるノードが追加されます。
5. 辺はinput treeとoutput treeのリンクによって構成されます。

Workflow as a Service Definition

- Given some known services $S_n(1 < n < N)$ including original service S_o and destiny service S_d, such as given an uncompleted workflow fragment. Workflow as a Service is to find a subnetwork N=<V, E> which starts with S_o and ends with S_d based on global social service network, such that :
  1. V is a finite set of services $W_m(1 < m < M, S_n \subseteq W_m)$.
  2. E is a set of Peer Social links.
  3. for each $W_i(0 < i < M)$, $W_i \cdot l \in \bigcup_1^M W_m \cdot O$
  4. The total cost Cost(Sd) is minimal.

Workflow as a Service

1. Web-Scale Functional Map を隣接行列にマッピングします。
2. 到達可能性を計算します。
3. Workflow-Search Algorithmを使いworkflowを生成します。

Mapping Web-Scale Functional Map to adjacency matrix

隣接行列$\mathrm{A}(G')$は(1)のように定義されます。

$$\mathbf{A}(\boldsymbol{G'}) = \begin{pmatrix} \boldsymbol{a_{11}} & \cdots & \boldsymbol{a_{1j}} \\ \vdots & \ddots & \vdots \\ \boldsymbol{a_{i1}} & \cdots & \boldsymbol{a_{ij}} \end{pmatrix} \tag{1}$$

(2) は$a_{ij}.\, a_{ij}$の定義を示します。$a_{ij}.\, a_{ij}$ はweb-scale functional map 内のsocial link の関係を表しています。

$$\boldsymbol{a_{ij}} = \begin{cases} \mathbf{0},\ \boldsymbol{without}\ \boldsymbol{L}\left(\boldsymbol{FM_i}, \boldsymbol{FM_j}\right) \\ 1, \mathbf{L}\left(\boldsymbol{S_i} \leftarrow \boldsymbol{S_j}\right)\ \boldsymbol{or}\ \boldsymbol{L}\left(\boldsymbol{S_i} <\!\| \boldsymbol{S_j}\right) \\ \quad \mathbf{or}\ \mathbf{L}\left(\boldsymbol{S_i} \rightarrow \boldsymbol{S_j}\right)\ \boldsymbol{or}\ \boldsymbol{L}\left(\boldsymbol{S_i} \|> \boldsymbol{S_j}\right) \\ \frac{\mathbf{1}}{\boldsymbol{n}}, \mathbf{other} \end{cases} \tag{2}$$

Calculating reachability matrix

- 隣接行列を使って、到達可能性を計算します。
- 到達可能性はWorkflow-Search Algorithmで計算コストを減らすために使われます。

Calculating reachability matrix

$\mathrm{A}(G')$ for $\mathrm{G}' = \langle V', E' \rangle$ with M vertices $S_m (1 < m < M)$ was defined as $\mathrm{A}(G')$, then L power of $\mathrm{A}(G')$ can be denoted as (3).

$$(\mathrm{A}(G')\,)^L = A^L = \begin{pmatrix} a_{11}^{(L)} & \cdots & a_{1j}^{(L)} \\ \vdots & \ddots & \vdots \\ a_{i1}^{(L)} & \cdots & a_{ij}^{(L)} \end{pmatrix} (L \geq 2) \qquad (3)$$

Where $a_{ij}^{(L)}$ represents the number of L step connection (or paths of length L) from S_i to S_j, can be defined by (4).

$$a_{ij}^{(L)} = \sum_{k=1}^{M} a_{ik}^{(L-1)} \cdot a_{kj}^{(1)}, A^1 = A \begin{pmatrix} a_{11}^{(L)} & \cdots & a_{1j}^{(L)} \\ \vdots & \ddots & \vdots \\ a_{i1}^{(L)} & \cdots & a_{ij}^{(L)} \end{pmatrix} \qquad (4)$$

We define reachability matrix such as (5) and (6).

$$R(G') = \begin{pmatrix} r_{11} & \cdots & r_{1j} \\ \vdots & \ddots & \vdots \\ r_{i1} & \cdots & r_{ij} \end{pmatrix} \qquad (5)$$

$$r_{ij} = \begin{cases} 1 & if \ \sum_{n=1}^{M-1} a_{ij}^{(n)} > 0 \\ 0 & otherwise \end{cases} \qquad (6)$$

Workflow-Search Algorithm

1. Find{
2. K is initialized with the terminate node S_d
3. Until K is empty{
4. Remove the next node n in K with the smallest $g(G'' \cdot WF(n), n)$;
5. If n is the starting node S_o
6. Break;
7. reachability(child node of n);
8. J=expand(S_n) with reachability
9. If(J!=null){
10. for each $S_j \in J${
11. if $L(S_i \leftarrow S_j)$ *or* $L(S_i <\| S_j)$ or $L(S_i \rightarrow S_j)$ *or* $L(S_i \| > S_j)${
12. if $g(G'' \cdot WF(S_j), S_j) > E(S_j) + g(G'' \cdot WF(S_n), S_n)${
13. $g(G'' \cdot WF(S_j), S_j) = E(S_j) + g(G'' \cdot WF(nS_n), S_n)$;
14. MarkedParent(S_j)=n;
15. }
16. if (S_j has not been visited and $S_j \notin K$){
17. add S_j to K;
18. Label S_j as known;
19. }
20. }
21. else{
22. if all S_j's parents are known {
23. $g(G'' \cdot WF(S_j), S_j) = \sum_{over\ j's\ Parent\ p} \left(g(G'' \cdot WF(S_p), S_p)\right) + E(S_j)$
24. if (S_j has not been visited and S_j∉K){
25. add j to K;
26. Label S_j as known;
27. }
28. }
29. else
30. Record that one more parent of S_j (i.e., n) is known;
31. }
32. } //for each child of S_j
33. } //if J != null
34. n is optimized (solved);
35. } //end of until
36. if S_o is optimized (solved), the minimal cost of the solution graph of S_o is g(G^''·WF(S_o),S_o)else report that no solution can be found;
37. }

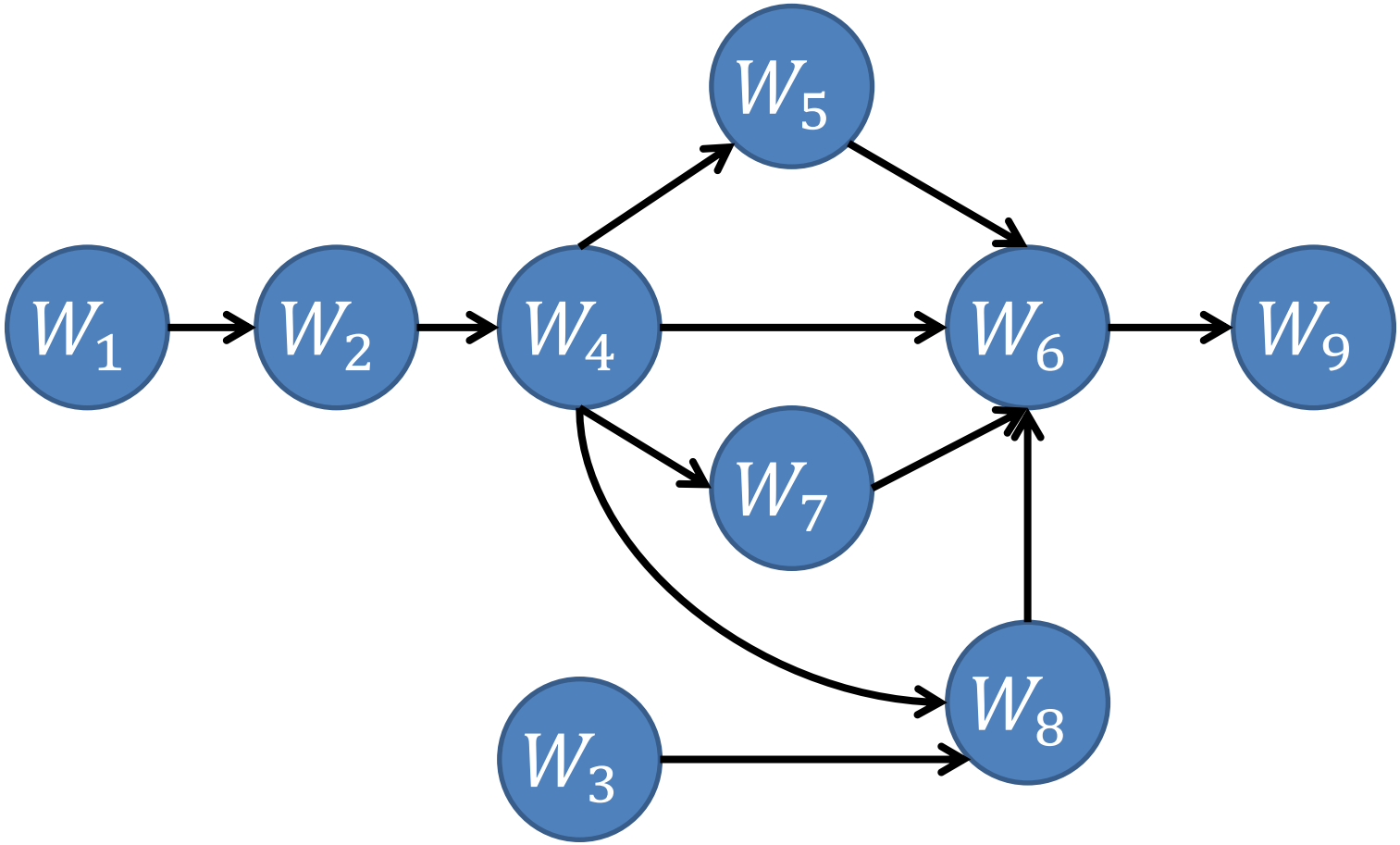
Example Scenario

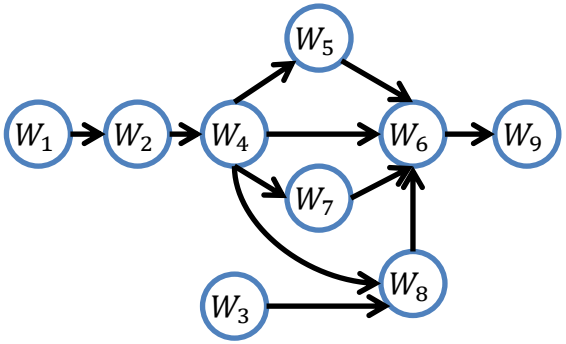
- 一人の男性が観光に行くとします。初めに彼はどんな観光地があるか知りたいはずです。
- 次に観光地の詳細や近くの宿泊地やレストランの情報が必要になります。
- 最後に、宿泊地やレストランへ行く方法の情報が必要になります。
- この要求を満たすためには3つの異なったサービスが公開されたサービスプロバイダーが必要になります。

Example Service

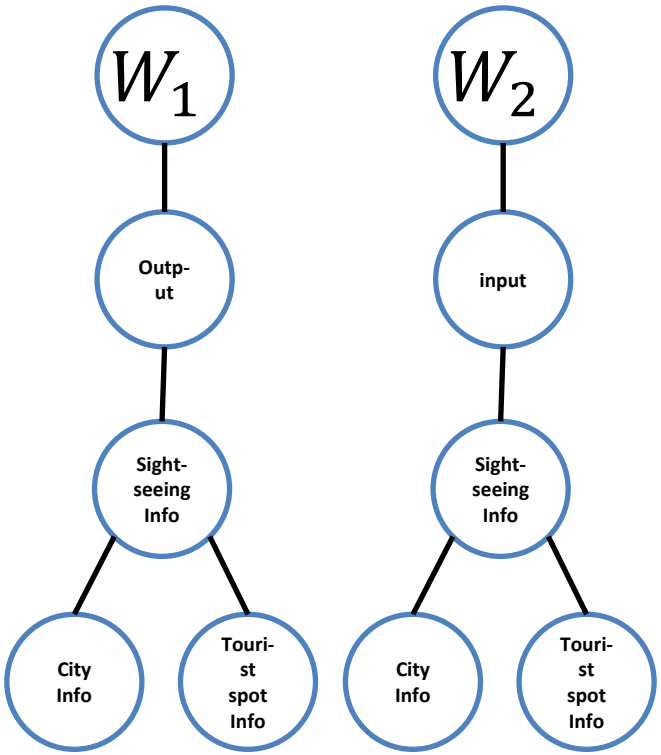
	Service	Input	Output
W_1	SightseeingService	area information	sightseeing information
W_2	getRelatedSightseeing Information Service	sightseeing information	address type, tourist spot information
W_3	getAddress Information Service	Sightseeing community information	tourist spot Address
W_4	findSightseeing Related Community Service	address type, tourist spot information	zip code, tourist spot Address
W_5	getTour Service	zip code	tourAddress
W_6	findDirection Service	two addresses	mapImages, driving map
W_7	getHotelInformation Service	zip code	hotelAddress
W_8	getRestaurantService	tourist spot Address	restaurant Address
W_9	getWeatherService	mapImages	weather information

Global Social Service Network





Rule 1



Rule 2

• • •

Dbpedia: Sightseeing community Info

Dbpedia: Area Info

Dbpedia: Sightseeing Info

Dbpedia: Sightseeing Info

Dbpedia:Tourist spot Address

Dbpedia:Tourist spot Information

Dbpedia: Zip code

Dbpedia: Tourist spot Address

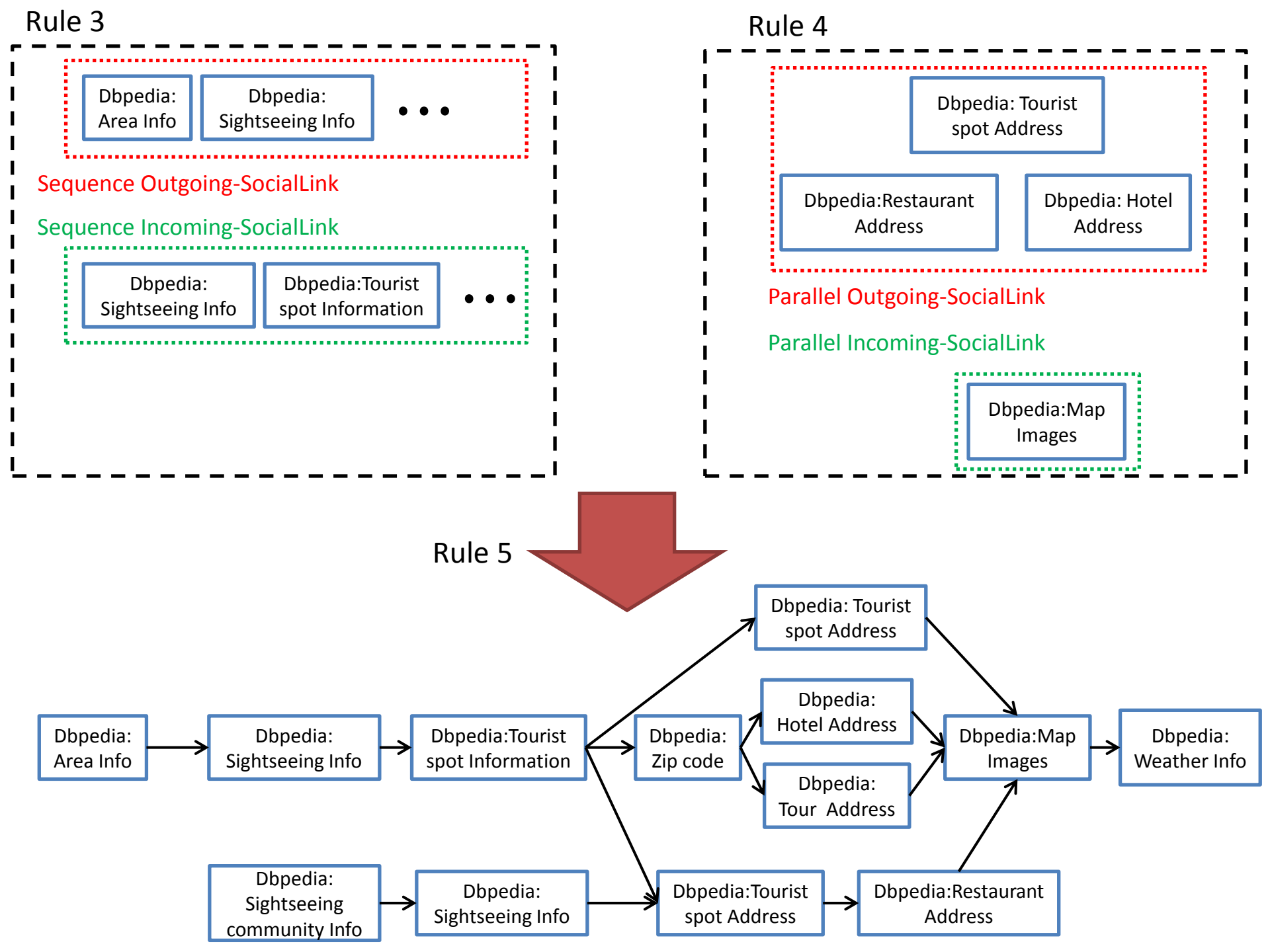
Dbpedia: Tour Address

Dbpedia:Restaurant Address

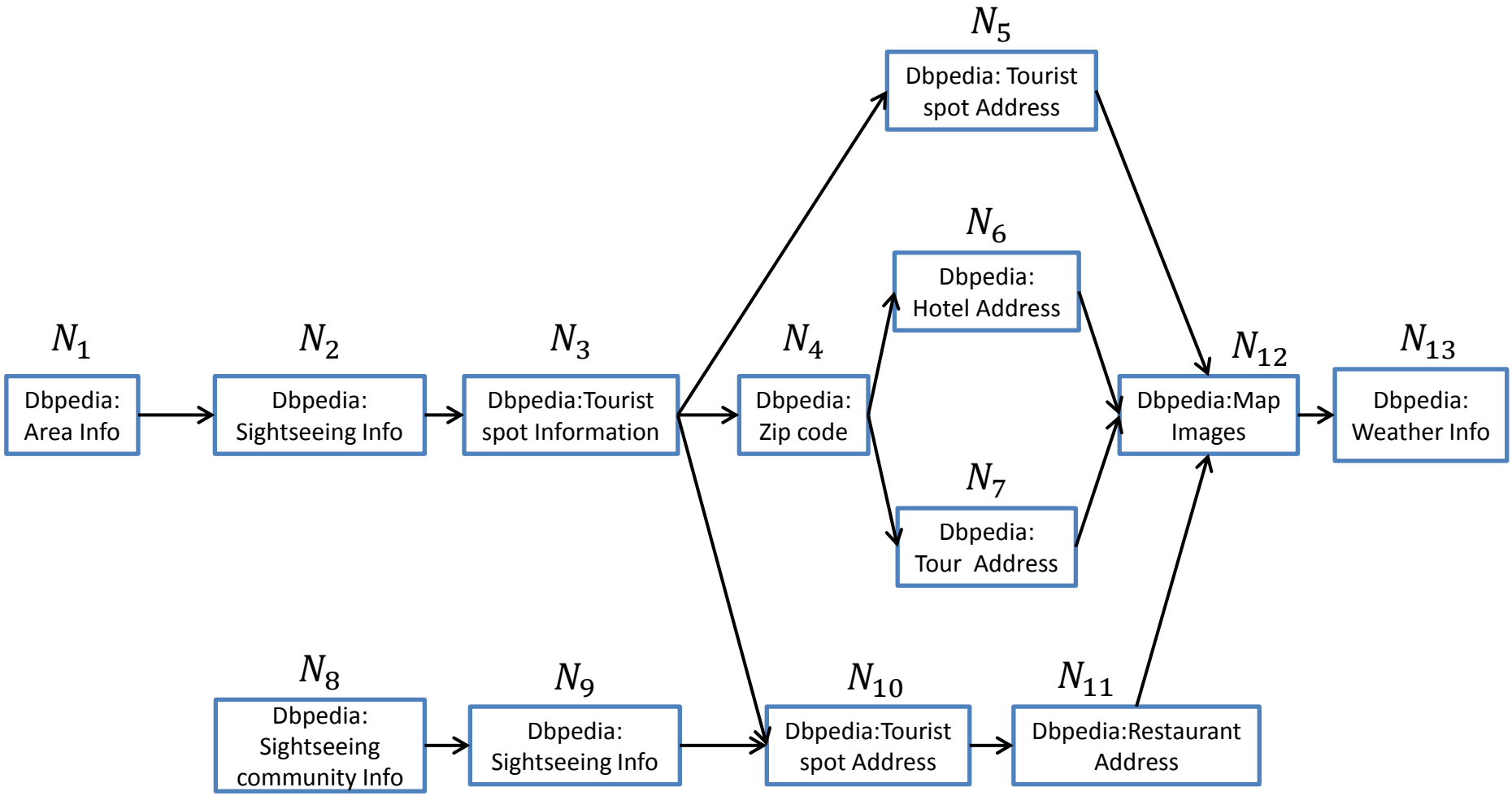
Dbpedia: Hotel Address

Dbpedia:Map Images

Dbpedia: Weather Info


Rule 3
Dbpedia: Area Info
Dbpedia: Sightseeing Info
Sequence Outgoing-SocialLink
Sequence Incoming-SocialLink
Dbpedia: Sightseeing Info
Dbpedia:Tourist spot Information
Rule 4
Dbpedia: Tourist spot Address
Dbpedia:Restaurant Address
Dbpedia: Hotel Address
Parallel Outgoing-SocialLink
Parallel Incoming-SocialLink
Dbpedia:Map Images
Rule 5
Dbpedia: Area Info
Dbpedia: Sightseeing Info
Dbpedia:Tourist spot Information
Dbpedia: Zip code
Dbpedia: Tourist spot Address
Dbpedia: Hotel Address
Dbpedia: Tour Address
Dbpedia:Map Images
Dbpedia: Weather Info
Dbpedia: Sightseeing community Info
Dbpedia: Sightseeing Info
Dbpedia:Tourist spot Address
Dbpedia:Restaurant Address

Functional Map for the Scenario



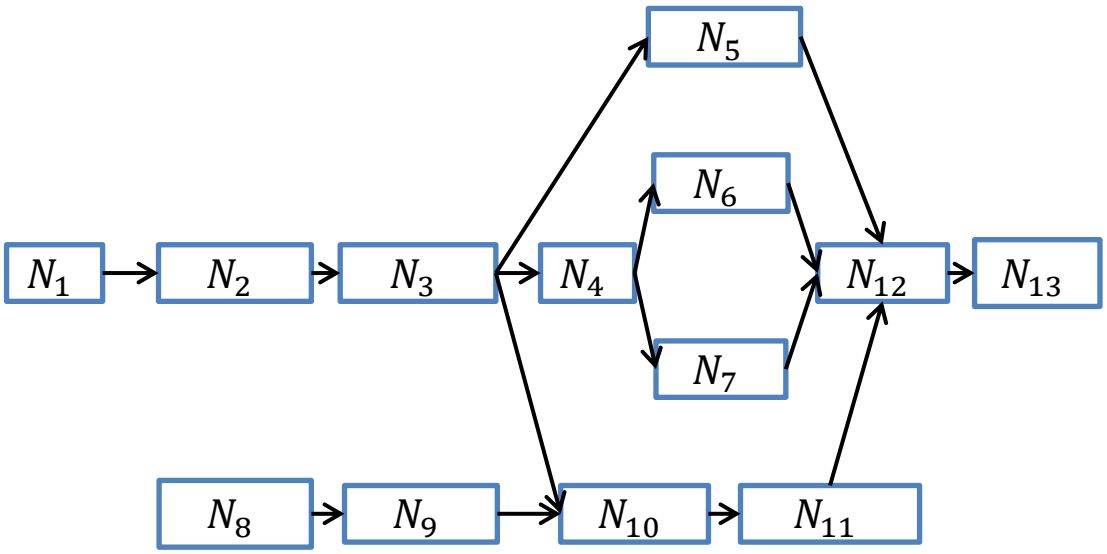
Adjacency matrix

	N_1	N_2	N_3	N_4	N_5	N_6	N_7	N_8	N_9	N_{10}	N_{11}	N_{12}	N_{13}
N_1	0	1	0	0	0	0	0	0	0	0	0	0	0
N_2	0	0	1	0	0	0	0	0	0	0	0	0	0
N_3	0	0	0	1	1	0	0	0	0	1	0	0	0
N_4	0	0	0	0	0	1	1	0	0	0	0	0	0
N_5	0	0	0	0	0	0	0	0	0	0	0	1	0
N_6	0	0	0	0	0	0	0	0	0	0	0	1	0
N_7	0	0	0	0	0	0	0	0	0	0	0	1	0
N_8	0	0	0	0	0	0	0	0	1	0	0	0	0
N_9	0	0	0	0	0	0	0	0	0	1	0	0	0
N_{10}	0	0	0	0	0	0	0	0	0	0	1	0	0
N_{11}	0	0	0	0	0	0	0	0	0	0	0	1	0
N_{12}	0	0	0	0	0	0	0	0	0	0	0	0	1
N_{13}	0	0	0	0	0	0	0	0	0	0	0	0	0

Reachability Matrix

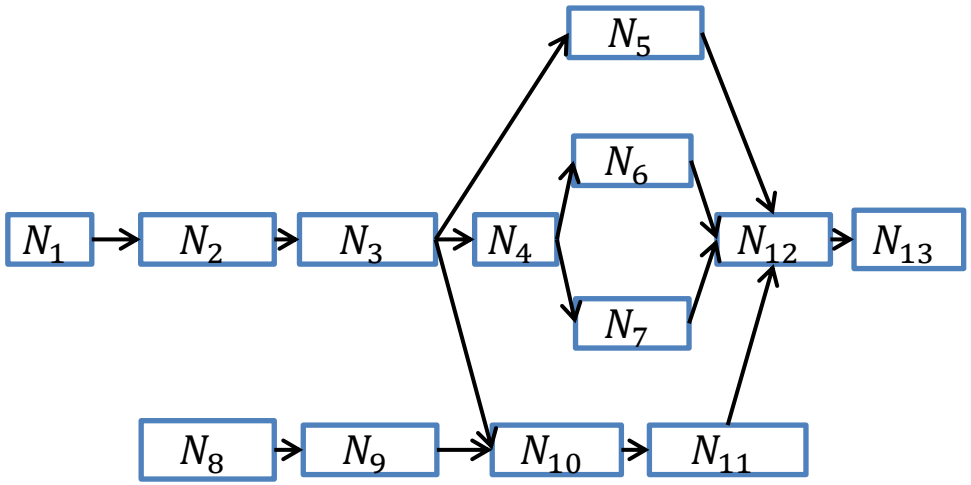
$$R(G') = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Suppose cost of nodes



service	cost
N_1	1
N_2	2
N_3	3
N_4	1
N_5	3
N_6	2
N_7	1
N_8	2
N_9	2
N_{10}	1
N_{11}	1
N_{12}	2
N_{13}	3

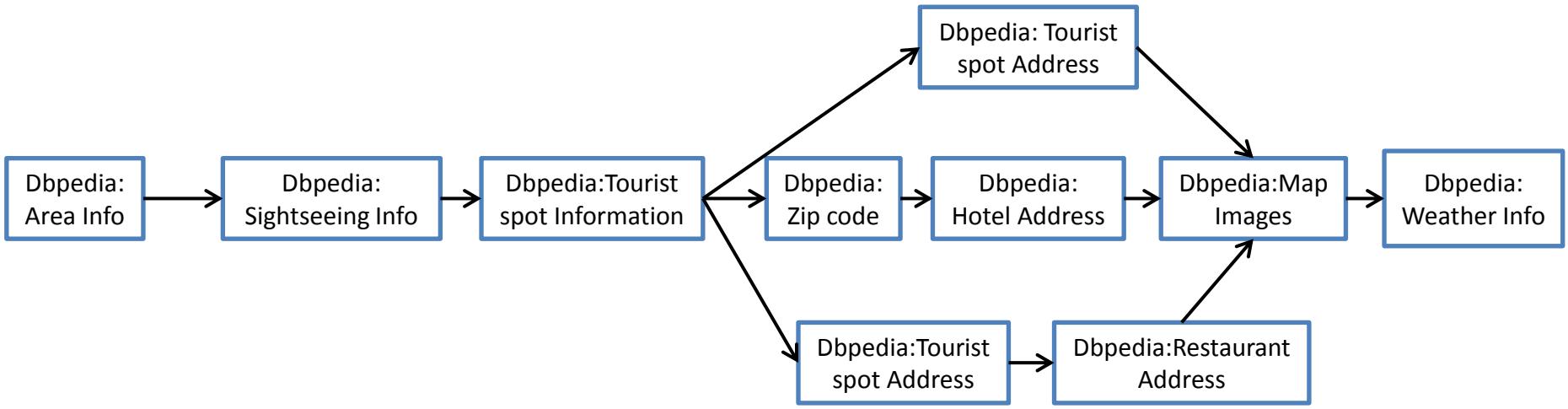
Workflow-Search Algorithm



n	K	J
N_{12}	N_{12}	N_5, N_6, N_7, N_{11}
N_6	N_5, N_6, N_7, N_{11}	N_4
N_5	N_4, N_5, N_7, N_{11}	N_3
N_{11}	N_3, N_4, N_{11}	N_{10}
N_{10}	N_3, N_4, N_{10}	$N_3,$ ✖
N_4	N_3, N_4	N_3
N_3	N_3	N_2
N_2	N_2	N_1
N_1	Return workflow	

N_9 dose not have reachability

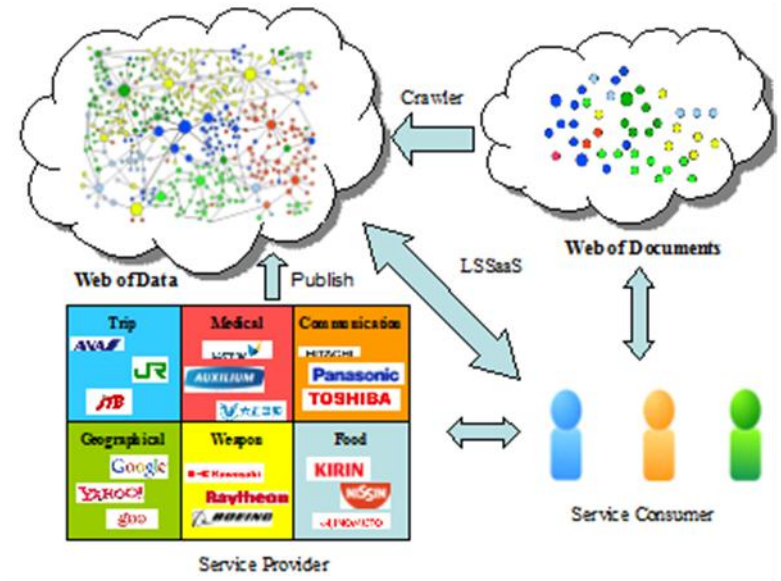
Workflow as a Service

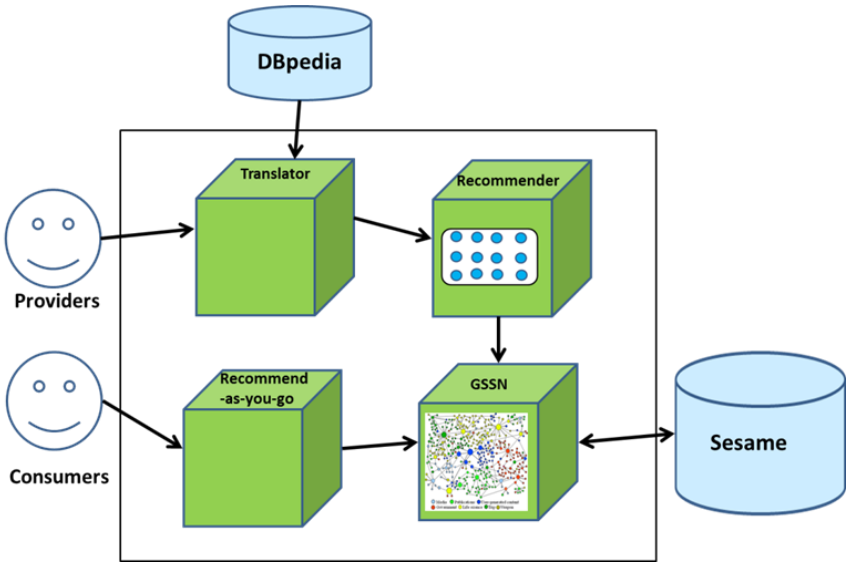


Implementation

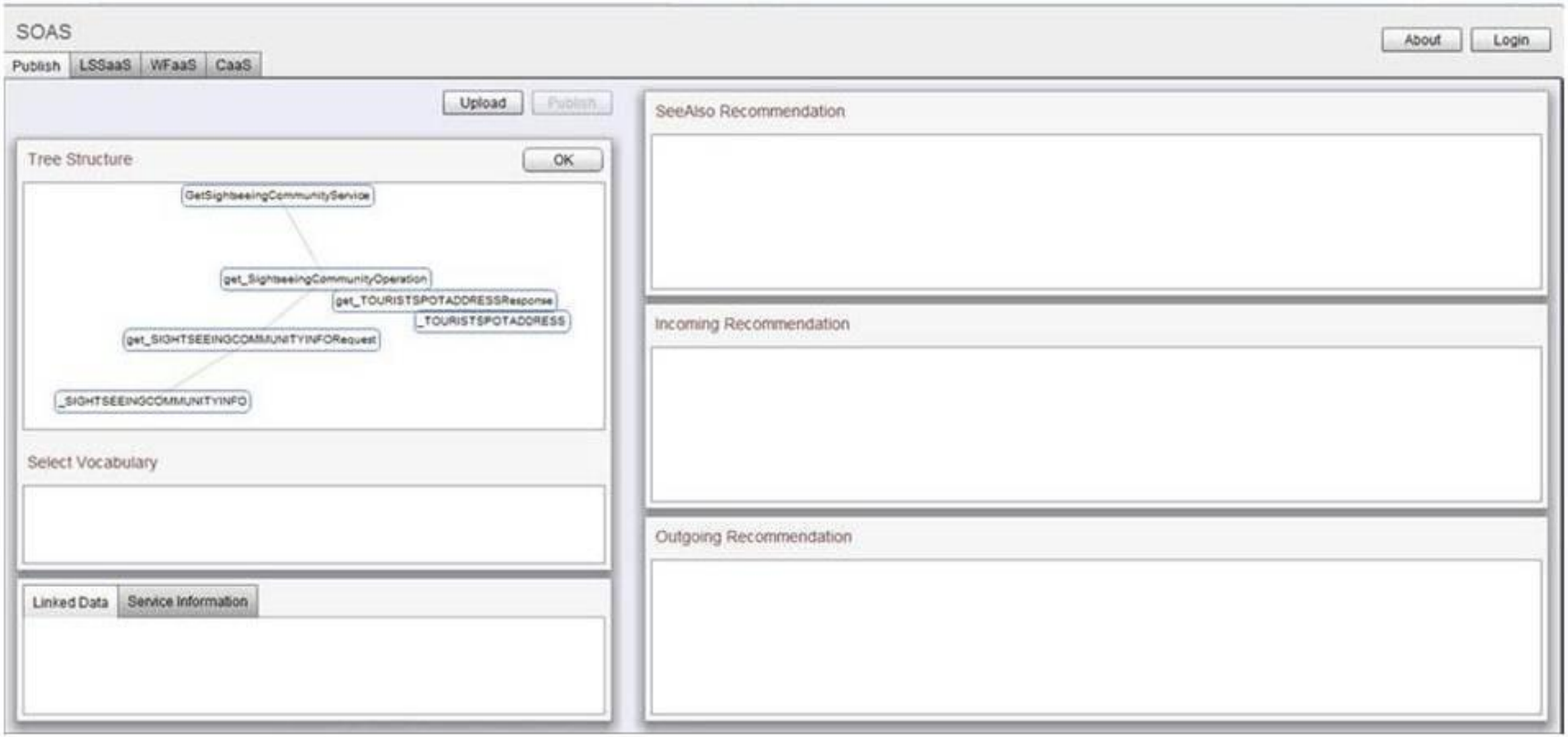
- 私たちの研究の実装はService of a Service (SOAS)モデルを使いました。
- SOASはglobal social service networkを構成するプラットフォームです。
- このプラットフォームの構造は図のようになります。(Left: High level Architecture Right: Lower-level Architecture).

Implementation

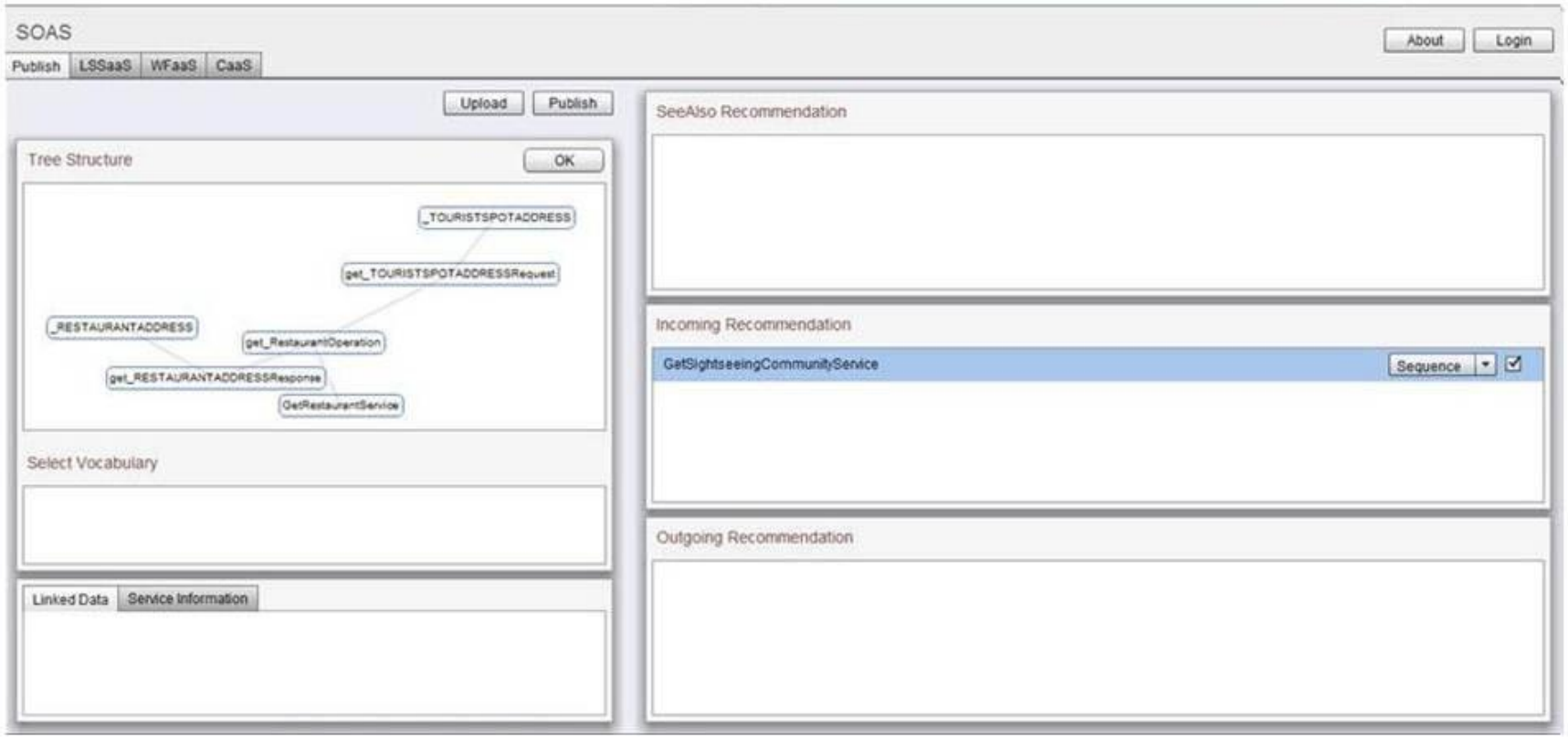




Implementation



Implementation



Conclusion

- 私たちの研究ではサービス合成や探索の今までのモデルの限界を解決するために、サービス間の関係を基にサービス同士をリンクした。
- したがって、サービス同士の機能性の関係を基にサービス同士がつながり、サービスからサービスへの探索が可能になった。
- Workflow-Search Algorithmを使い、到達可能性を考慮しサービスの探索を行うことでより効率の良いサービスの探索ができるだろう。
- 私たちはweb-scale functional mapを使うことで、social linkを基にした新しいサービス自動構成の方法を提案し、実装をした。

Future work

- Future workはこのアプローチの評価をすることと、global social service networkを基にしたworkflow as a service がどれくらいよくなるかの研究になります。