

L-013

無効サービス検出機能を備える携帯端末向け検索システムの提案 The Proposal of Mobile Search System with Invalid Service Detection Method

藤井 邦浩[†] 長沼 武史[†]
Kunihiro Fuzii Takefumi Naganuma

深澤 佑介[†] 倉掛 正治[†]
Yusuke Fukazawa Shoji Kurakake

1. はじめに

近年、携帯端末からインターネットに接続し、インターネット上のコンテンツやサービスを利用するユーザが増加しており、携帯端末向け検索システムの重要性が高まりつつある。

インターネット上のサービスを発見する手段として、一般的に、キーワード検索システム [1] やカテゴリ検索システム [2] がある。これら検索システムでは、Web ロボット [7] が定期的に収集するインターネット上のコンテンツを基にインデックス化や検索処理する。そのため、サービスを提供するサーバの輻輳や停止、あるいは途中経路のネットワークの不調などを原因とする無効なサービスを検索結果に含む場合がある [4]。

ところで、携帯端末は PC と比較して操作性や帯域が制限され、検索時におけるユーザの負担が大きく、通信費用が発生する場合もある。それゆえ、無効サービスを検索結果としてユーザに提示すると、検索システム自体の評価低下につながる。

そこで我々は、検索結果に無効サービスを含まない携帯端末向け検索システムの実現を目的とし、無効サービスの検出手法を提案する。

2. 関連研究

従来手法における無効サービスの検出手法や検出時期について述べる。

まずキーワード検索システムやカテゴリ検索システムでは、Web ロボットが定期的に収集するコンテンツをインデックス化し、検索処理する。よって、Web ロボットの巡回周期で無効サービスを検出する。

次に Squid [6] を代表とする Proxy サーバは、サービス提供元が保持するコンテンツを Proxy サーバ上にキャッシュする。そしてキャッシュの状態を最新に保つため、コンテンツの有効時間を規定しその有効時間に基づいてオリジナルのコンテンツとキャッシュしたコンテンツを比較する [5]。よって、キャッシュの有効時間を周期とし無効サービスを検出する。

以上のように、従来手法では特定の周期でサービス提供元へアクセスし無効サービスを検出するため、無効サービスの反映に遅延を生じる。また検出周期を短くすると、サービス提供元のサーバや、サービス提供元への経路となるネットワーク (ルータや回線など) へ負荷を与える。

ところで、無効サービスはサーバの輻輳や停止、設定ミス、ネットワーク器機・回線の不調など予期せぬ原因で発生し、予測や発生周期の特定は困難である。そのため、従来手法のような特定周期で検出する手法では、検索結果に無効サービスを含む可能性がある。

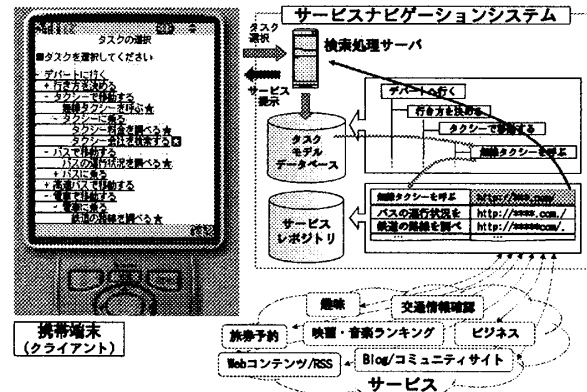


図 1: サービスナビゲーションシステム概要

3. 実現方式と課題

検索結果から無効サービスを排除するため、検索結果となるサービスに対するサービス提供元の状態を確認し、無効サービスを排除した後に、ユーザへ検索結果として提示する方式を採用する。

しかし課題として、無効サービス検出のための処理時間が必要となり、ユーザへの検索結果の提示が遅れる可能性がある。さらに、検索処理の毎にサービス提供元へアクセスする方式はサービス提供元のサーバや、サービス提供元への経路となるネットワークに負荷を与える。

そこで、検索結果としてユーザへ提示するサービスを事前に予測でき、さらにサービスを絞り込み可能な検索システムを採用する。これにより、無効サービスの検出処理を事前に開始可能とし、また状態確認するサービスの対象数を削減する。さらに重複するサービス状態確認を削減することでサービス状態確認頻度を軽減し、課題解決を目指す。

4. 提案方式

4.1 サービスの絞り込みと状態確認の制御

検索結果としてユーザへ提示するサービスを事前に予測でき、さらにサービスを絞り込み可能な、検索システムとしてサービスナビゲーションシステム [3] の利用を提案する。

サービスナビゲーションシステムは、サービスの存在を知らない、あるいは検索サービスに不慣れなユーザでもやりたいこと (タスク) を選択していただく適切なサービスを発見可能とする。

サービスナビゲーションシステム概要を図 1 に示す。タスクモデルデータベースはユーザの目的や行動 (タスク) をモデル化し、検索・再利用が可能な知識として蓄積する。サービスレポジトリはタスクに関連するサービスを利用する際に、サービス提供元へアクセスするため

[†]NTTDoCoMo ネットワーク研究所

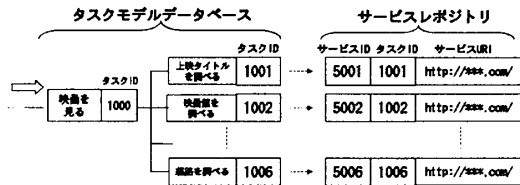


図 2: タスクに関連するサービスの抽出

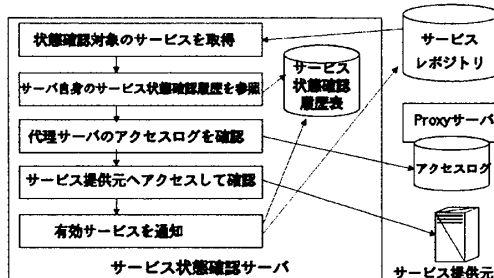


図 3: サービス状態確認サーバの処理

の参照情報を蓄積する。ユーザは携帯端末からサービスナビゲーションシステムにアクセスし、適切なタスクを順次選択する。そしてユーザが選択したタスクに関連するサービスをサービスレポジトリから抽出し、検索結果としてユーザに提示する。

ここで、ユーザが選択したタスクに関連するサービスを抽出する例を図2に示す。タスクモデルデータベースはユーザの行動モデルを木構造で表現する。タスクはリーフに近づくにつれて具体化・詳細化する。また、各タスクは識別子である「タスクID」を持つ。サービスレポジトリは、タスクIDとそのタスクに関連するサービスを応対付けて蓄積する。そして、ユーザが選択したタスクのタスクIDに対応するサービスをサービスレポジトリから抽出し、ユーザへ提示する。

また無効サービスの検出処理時間を確保するため、ユーザが選択したタスクに対してリーフ側のタスクを先読み処理する。例えば図2において、ユーザがタスクID「1000」に到達した時点で、タスクID「1001」から「1006」に関連するサービスの状態確認処理を開始する。

4.2 アクセスログの利用

サービスナビゲーションシステムが導いた検索結果となるサービスについて、履歴情報やアクセスログを利用してサービス状態確認の頻度を軽減する方式を提案する。なお前提条件として、携帯端末とサービス提供元の通信を仲介するProxyサーバが存在し、Proxyサーバのアクセスログが参照可能であるとする。またProxyサーバはサービスナビゲーションシステムを利用していないユーザのアクセスログも蓄積する。

無効サービスの検出処理は、サービス状態確認サーバで実施する。検出処理の概要について図3に示す。まず、サービスレポジトリから状態確認対象のサービスを受信する。次にサービス状態確認履歴表を参照し、有効期間内の履歴表において対象とするサービス提供元へ正常にアクセスした記録があるか調査する。サービス状態確認履歴表は、サービス状態確認サーバ自身がサービス提供

元へアクセスし、サービス状態確認した履歴情報を蓄積する。そして、対象とするサービス提供元へ正常にアクセスしていた場合、そのサービスは有効であると判断する。これにより、他の装置やネットワークに負荷を与えずに、サービス状態確認を実現する。なお、履歴の有効期間は、過度に重複するサービス状態確認の防止を目的とするパラメータであり、十分に短い時間を設定する。

サービス状態確認サーバの履歴で判断できなかった場合、次にProxyサーバのアクセスログを参照し、有効期間内のアクセスログにおいて、いずれかの携帯端末が対象とするサービス提供元へ正常にアクセスしたか調査する。これにより、Proxyサーバ以外の装置やネットワークに負荷を与えずにサービス状態確認を実現する。

Proxyサーバのアクセスログで判断できなかった場合、次に、サービス状態確認サーバはサービス提供元のサーバへ直接アクセスしサービス状態を確認する。これにより、サービス状態確認を確実に実施する。また状態確認した結果は、サービス状態確認履歴表へ記録する。

5. おわりに

携帯端末は検索時におけるユーザの負担が大きいため、無効サービスを検索結果としてユーザに提示すると、検索システム自体の評価低下につながる。そこで、検索結果となるサービスについてサービス状態を確認し、有効なサービスのみユーザへ提示する方式を採用する。しかし、サービス状態確認に伴う処理時間が発生し、さらにサービス提供元のサーバやネットワークへ負荷を与える。そこで、サービス状態確認の対象となるサービスを予測した事前検出処理や、対象サービスの絞り込みを実現するため、サービスナビゲーションシステムを利用する。またサービス提供元へのサービス状態確認の頻度を軽減するため、アクセスログを利用する手法を提案した。今後の課題として、実装とサービス状態確認に伴う検索処理全体の処理時間やサービス提供元へ与える負荷の評価を予定する。

参考文献

- [1] <http://www.google.com>
- [2] <http://www.yahoo.com>
- [3] T.Naganuma and S.Kurakake. "A Task Oriented Approach To Service Retrieval in Mobile Computing Environment," Artificial Intelligence and Applications, 2004
- [4] Diomidis Spinellis. "The decay and failures of web references," Communications of the ACM Vol.46, No.1, pp.71-77, Jan.2003
- [5] James Gwertzman and Margo Seltzer. "World Wide Web Cache Consistency," Proceedings of the USENIX 1996 Annual Technical Conference San Diego, California, Jan.1996
- [6] <http://www.squid-cache.org/>
- [7] <http://www.robotstxt.org/wc/robots.html>