**アプリケーションの使用・導入・負荷状況把握支援システムの開発**

**Development of Support System for Grasping the Situations on Application Use, Installation and Load**

神長研究室　　210810052　　菅野聖矢

1. **はじめに**

近年計算機は多機能高性能化により一台で複数の分野の活動を行うことが多い.それらの活動ごとに必要となるアプリケーションが異なる場合,必要に応じてユーザがインストールしていくため,結果として一台の計算機内に多数のアプリケーションが同時にインストールされる事になる.

その結果として,計算機の処理速度の低下や保存領域の圧迫などといった問題が生じ,計算機の利便性を損なうことがある.

それを避けるためにはこまめなアプリケーションの整理が必要であるが,先に述べたようにアプリケーションの数が多く整理が手間であるという点や,アプリケーション間の依存関係の把握が困難といった点からユーザにとっては難しい作業であるといえる.

その問題を解決する研究として"system information for windows"や"プログラムの追加と削除"といったものがある.

しかし,これらのソフトでは前者には表示される情報が多く理解が難しく,後者については最終起動日と簡単な頻度が表示されるのみであり,要・不要の判断基準としては情報が少ない.また,どちらのソフトについても依存関係については考慮されていないため,アプリケーションを削除した後に,他のアプリケーションの動作に何かしらの影響が出る可能性がある.

そこで,これらの状況を的確に判断しユーザに提示することにより安全に不要なアプリケーションを削除できるようなシステムの開発を目指す.

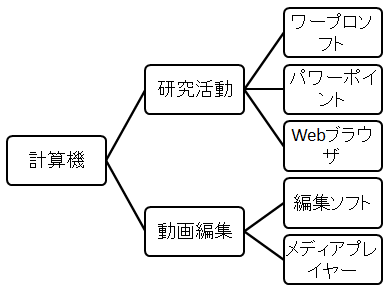


図１：計算機活動とアプリケーションの関係性

1. **問題点及び支援方法**

# **問題点**

問題点を整理すると,アプリケーションの数の多さによる要・不要の判断と表示される結果の確認の煩雑さ（問題点１）.複数のアプリケーション間の依存関係の確認の分かりにくさ（問題点２）.判断基準となる情報収集の面倒さとシステムとユーザの判断する必要度の相違（問題点３）.既存の研究では表示される情報が簡単で少ない情報か,専門的な知識が必要な大量の情報か,のように両極端である（問題点４）と言った点があげられる.

**2.2.支援方法**

これらの問題を解決する方法として,システム側で全アプリケーションについてまとめて処理してくれるシステムの開発（解決策１）.各アプリケーションについてユーザの判断に近い要・不要の判断が可能な手法の開発（解決策２）.アプリケーションごとに判断基準となる情報の収集・蓄積が可能な手法の開発（解決策３）.表示された多量のアプリケーションについて,アプリケーションごとの情報を容易に確認・理解が容易にできるような表示手法の開発（解決策４）.を行い、以上のようなアプローチより,問題点の解決をはかる.

1. **支援システム設計**

**3.1システム要件**

システム要件については,全てのアプリケーションについて一括で走査・分析が可能であること（要件１）.起動回数や使用時間等の使用状況やインストール先等の導入状況など多数の情報を収集し,アプリケーションごとに蓄積することが可能であること（要件２）.結果の表示が一目でわかりやすく,必要な情報についても容易に引き出せるような表示方法であること（要件３）.各アプリケーションについて,ユーザ自身の判断との差異が少なくなるように適確に判断が可能であること（要件４）.これらをもとにシステムを作っていく.

**3.2システム概要**

本システムの概要については,アプリケーションの状況について,使用状況や導入状況などの複数の情報から把握し,使用時間などの動的な情報についても逐次収集・蓄積する.その後,必要に応じてそれらの情報とユーザの設定より各アプリケーションについてユーザがそれをどの程度必要としているかの必要度や他のアプリケーションにどの程度の影響を与えるかといった影響度,計算機にそのアプリケーションがどの程度の負荷を与えているかといったリソース使用率等を表示する.これにより,計算機内部のインストール済みの全アプリケーションについて簡単な設定で様々な情報からユーザ自身の判断基準に近い判断が可能であり,表示される情報についても項目を絞り,短時間で多くのアプリケーションについての確認を可能なようにするシステムである.

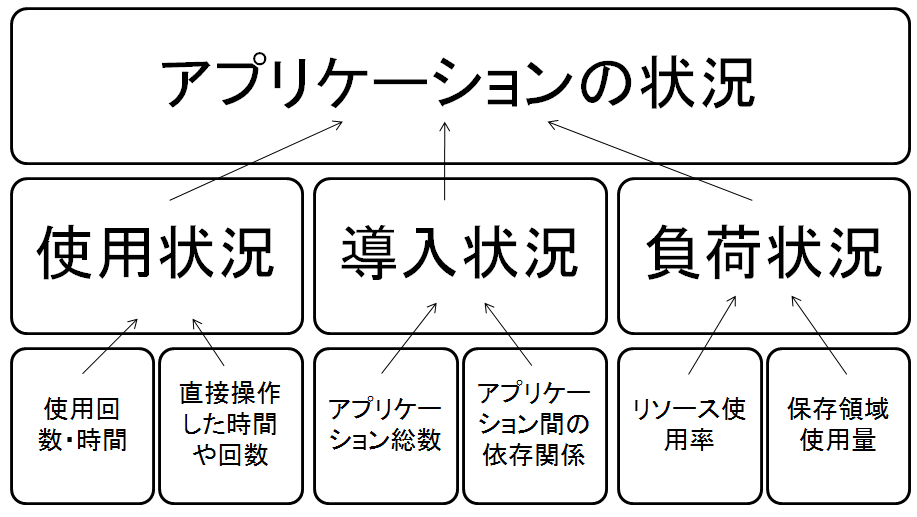


図２：アプリケーション把握の概要

このシステムの対象としては,計算機を長期間にわたり使用するなどして,一台の計算機内部に多数のアプリケーションがインストールされている計算機を使用しているユーザを対象としている.

このシステムを用いることにより,計算機内部に多数存在するアプリケーションについても,簡単な操作で全てのアプリケーションについて判断が可能である.

また,依存関係についても表示されているため,あるアプリケーションを消した際に他のアプリケーションに及ぼす影響を事前に把握可能であきるため,不要であっても削除した場合の影響が大きいため削除しない,影響が少ないならば気にせず削除する,といった判断が可能になり,予期せぬ問題が発生しないように,確実にアプリケーションを削除し,計算機のリソースを確保することができるようになる.

1. **状況把握支援システムの流れ**

**4.1情報収集**

最初の段階として必要な情報の収集を行う.これはアプリケーションが起動・終了した日時や,実行していた時間等の情報を収集し,計算機内部にアプリケーションごとに蓄積していく必要があるので,計算機の起動と同時にシステムも起動し,必要な情報を常時収集・蓄積していく.

**4.2設定入力**

設定入力では,ユーザがどの情報を重視して必要度を判断するかの指標となる情報ごとの重みづけや,蓄積された情報の内,どの期間の情報を参照するか,結果表示の際にどの情報を表示するかといった項目を入力していく.

**4.3処理分析**

前項で設定した期間や重みづけに従い,各アプリケーションについて必要度やリソース使用率などを算出する.

具体的には各アプリケーションについて,前項で設定した期間内の情報より情報ごとに基準評価点を算出し,基準評価点を重みづけに従い増減させ,それの合計をアプリケーションの最終評価点とする.最終評価点より必要度を算出する.

リソース使用率についても参照する情報の種類は変わるが,手順については同様の手順で算出する.

**4.4結果表示**

処理分析の結果算出した必要度やリソース使用率,依存関係による影響やアプリケーションの名前といった分かりやすい情報のみを少数表示する.それによりユーザの理解を容易にし,アプリケーションの数が多くとも短時間で確認可能にする.また,必要に応じて詳細な情報についても随時閲覧可能にし,システム側の判断の根拠の確認や自分での再判断が簡単に行えるような表示方法にする.

また、結果を確認後不要と判断したものや,あまり使わないと判断したアプリケーションに関しては単純な削除だけでなく,普段は一時的に圧縮し,ユーザからの要求に応じて即座に解凍し実行,というような動作を行うことも可能にする.

1. **おわりに**

今後はこれらの機能を本当に必要かどうか考えつつ,より簡潔に分かりやすくした機能を実際のシステムとして実装を進めていきたい.

# **参考文献**

[1] System Information for Windows

(URL:http://www.gtopala.com/)

[2] Microsoft Windows XP

(URL:http://windows.microsoft.com/ja-JP/windows/products/windows-xp)

[3] 宇佐美 晶宏, 神長 裕明, 森本 康彦, 横山節雄, 宮寺康造, 中村勝一 ” システム制約に適応的な研究環境構築のための可搬型支援システム” 電子情報通信学会技術研究報告, Vol.110, No.453, pp 157-162, 2011.