

構成要素に基づく製品間の影響関係可視化

A Visualization Method for Componential Influence Relations among Industrial Products

河田 裕成*¹
Hironari Kawada

赤石 美奈*²
Mina Akaishi

*¹ 法政大学大学院情報科学研究科

*² 法政大学情報科学部

Graduate School of Computer and Information Science, Hosei University Faculty of Computer and Information Science, Hosei University

Industrial products have influenced each other during their evolution. We can identify this process by analyzing relations of the components between products. However, it takes much effort and time to calculating and visualizing the differences of those manually. Therefore, in this paper, we propose a method for visualizing componential influence relations between products automatically from product information data. In our method, we define the product information data model that is generated from inputted product information. Then, the differences of attributes between a pair of components in each product information are calculated while taking into account of elements such as developer of a product to reduce redundant relations. Finally, these differences are visualized as directed graphs with products as nodes and edges showing influence relation of a product to another. The result of an experiment shows that the proposed method visualized influence relations include the existing method cannot.

1. はじめに

工業製品は互いに影響しあいながら発展してきた。この影響関係を把握することで、製品の成り立ちや後発製品に対して大きな影響を及ぼした製品、特定時期における工業製品のトレンドなど、製品発展史を様々な視点から理解するための知識を得ることができる。

しかし、製品間の影響関係は様々な分野に跨っており、一つの製品が複数の製品に影響し、その関係は複雑になる場合がある。従って、製品間の影響関係を図やグラフを用いて可視化することで、こうした関係の把握を支援することが有用である。

また、ある製品間の影響関係を把握するための一つの手法として、製品毎にその部品や機能などの構成要素を抽出し、複数製品間で比較することが考えられる。しかし、分析する製品数が非常に多い場合、この作業には多大な手間と時間がかかる。

そこで、本研究では複数の製品データから自動的にその構成要素を比較し、その差異を製品間関係としてジャンルや開発者を考慮しつつ可視化する手法を提案する。これにより、時系列的な製品発展史の分析と理解の支援を目指す。

2. 関連研究

複数製品間の関連や、製品自身の属する技術分野に対する影響の分析を行った研究として、小林ら[小林 2005]の研究がある。小林らは、システムやハードウェア名などのキーワードが出現する文書を分析し、キーワード間の関連度や技術分野への影響度などを時系列的に可視化した。

また、工業製品の発展構造の分析に関する研究として、太田ら[太田 2013]の研究がある。この研究では、複数製品間におけるその構成要素の類似度に注目し、製品の発展を系統図として可視化した。これらの手法では製品の大まかな発展や、製品システムの推定を行うことができる。その一方、製品間の詳細な差異を可視化できない。

本研究では個々の製品に対して構成要素の継承や削除、追加に注目することにより、ある一連の製品を構成する要素が、時間と共に少しずつ変化していく様子を詳細に可視化することを目的としている。

3. 提案手法

本研究では、部品や機能等、製品が発展していく中で変化する構成要素を属性として、製品間での属性の継承や新規追加、削除といった影響関係を可視化する。この際、製品を特定のグループに分類する要素として製品ジャンルや開発者名を考慮することで、共通の開発者等を持つ一連の製品群の漸進的な発展と、それに対する他製品群からの影響に注目した可視化を可能とする。そのために、製品を製品名、属性、発売年、開発者名、製品ジャンル名を保持するデータとしてモデル化した。

実際に処理される製品データは人手で作成した外部ファイルから作成される。この際、製品の属性もまた製品として見做しうることを考慮し、製品の各属性についても製品モデルに基づいてデータ生成を行い、製品に関連付ける。こうした製品データを元に、製品間での属性比較と、製品の属性間での属性値比較を行い、得られた結果を製品間の影響関係とする。その後、製品をノード、影響関係とその詳細をノード間のラベル付き有向辺として有向グラフを用いて可視化する。以下にその具体的な手法を述べる。

3.1 製品・属性データの生成

可視化用の製品と属性のデータは、表 1 に示したような製品情報を XML 形式で記述した外部ファイルから生成され、有向グラフ上のノードとして扱われる。ここで、属性の詳細な比較を可能とするために、属性の詳細を属性値としてリストで表現した。製品の各属性は、属性の発売年情報を「属性の初登場年」、開発者やジャンル情報を「属性と関連付けられた製品の開発者やジャンル」とし、属性値リストの各要素を属性として持つようなノードとして生成される。

表1 製品情報例

属性	属性値
製品名	iPhone6s
発売年	2015
開発者	Apple
ジャンル	Smartphone
OS	["iOS", "Apple製"]
CPU	["Apple A9", "2core", "64bit", ...]
RAM	["2GB", "LPDDR4 SDRAM"]
⋮	⋮

3.2 属性の比較

3.1.節で生成された製品データについて、2製品間でその保持する属性を比較し、属性継承等の関係を分析する。

製品間での属性比較は、製品同士もしくは「CPU」等のような製品に関連付けられた属性の属性値同士で行う。したがって、製品同士での属性差異が製品間での属性自体の継承や削除、追加を表すのに対し、属性同士での属性値の差異は、属性間におけるディスプレイの規格等のような属性を構成する要素の影響関係を表現することができる。

ここで、属性比較について、製品の漸進的な発展を可視化するためには、同一シリーズ製品間での大規模な属性継承を考慮する必要があると考えた。また、現実の製品開発においては、ある開発者の複数製品間で共通の部品を使用する、モジュール化設計が利用される場合があることに注目した。

以上のことから、共通した開発者と製品ジャンルをもつ製品群を製品系列とし、同一系列内の製品間での属性構成の変化と、系列外の製品からの属性の継承や追加に注目した属性比較手法を提案する。

(1) 開発者情報等を考慮した属性比較

まず、製品データ群から、重複のない組合せ製品ペアを生成する。次に、各ペアに属する製品間で属性の比較を行い、属性の影響関係を求める。このとき、ある製品を含む複数の製品ペアにおいて、その属性差異に重複した部分が存在することが考えられる。こうした重複がある場合は、製品の開発者や製品ジャンル、発売年情報を考慮して重複部分をラベルに含んだ有向辺を描画する製品ペアを決定する。決定の際に、同一系列に属する発売年が近い製品ペアが描画候補の上位になりやすく、異なる系列に属する発売年がより遠いペアの優先順位がより描画候補の下位になるようにすることで、同一系列の製品間では複数製品間で継承されやすい属性を含む連続した大規模な属性継承が、異なる系列の製品間では小規模な属性継承が可視化される。以下に、製品ペアの優先順位の決定法を述べる。

ある製品*i*を含む製品ペア*i, j*群において、各ペアの属性差異に重複した部分がある場合、以下の①～③の優先順位に基づいて、重複部分が描画される有向辺を決定する。ただし、製品*p*の開発者名を開発者(*p*)、製品ジャンル名を製品ジャンル(*p*)、発売年の値を発売年(*p*)とする。

- ① 開発者(*i*) = 開発者(*j*)
- ② 製品ジャンル(*i*) = 製品ジャンル(*j*)
- ③ $\min(|\text{発売年}(i) - \text{発売年}(j)|)$

また、本研究ではある系列の製品群の発展における他系列製品からの影響に注目した可視化を目的とするため、製品情報

表2 実験用製品データ

製品系列	製品ジャンル	開発者	製品数
iPod touch	PDA	Apple	8
iPhone	Smartphone	Apple	15
Nexus	Smartphone	Google	7
合計			30

に全く共通点を持たない製品間での有向辺は描画しない。したがって、製品*i, j*間の属性差異を有向辺として描画するのは、*i, j*が以下の条件I～IIIのいずれかを満たす場合のみとする。

- I. $(\text{開発者}(i) = \text{開発者}(j)) \wedge (\text{製品ジャンル}(i) = \text{製品ジャンル}(j))$
- II. *i, j*間に属性の継承関係が存在する
- III. *i, j*間に属性の追加関係が存在する

例えば、異なる系列に属する製品ペアの各属性構成に全く共通点がなく、時系列的にその製品で初めて出現する属性が存在しない場合、これらの製品間に有効辺は描画されない。

3.3 ユーザ支援機能

入力された製品データの数が非常に多い場合、可視化結果は煩雑なものとなる。そこで、ユーザが興味を持つ製品群と関連するノードと有向辺や、特定の種別の影響関係のみを表示する機能を実装した。これは、属性比較後の製品ノード群を、ユーザが指定した条件を用いてフィルタリングするものである。ここで指定可能な条件は製品名、開発者名、製品ジャンル、製品間の影響関係種別の4つを用意した。この条件を組合せて用いることで、例えばユーザの指定する製品系列と影響関係を持つ製品のみを可視化することができる。

また、有向辺のラベルを表示するか選択する機能も実装した。これを利用することで、製品間の影響関係の種別のみを表示した、より見やすい可視化結果を得られる。

3.4 可視化

属性比較処理によって求められた製品間における属性の影響関係は、製品をノード、製品間の属性差異を有向辺とした有向グラフを用いて可視化する。

まず、各製品ノードを発売年情報に基づき、図の上部から昇順に配置する。この際、開発者情報を用いてノードの図横軸上の位置を決定することで、同開発者の製品が上下方向に整列して配置されるようにする。各ノードには製品名、開発者名、発売年、製品ジャンル名が表記される。次に、各ノード間において3.2.節で求めた有向辺情報を参照し、ノード間に影響関係が存在する場合は、属性の「継承」や「削除」などの種別に応じて色分けした有向辺を引く。この際、各有向辺には製品間の各影響関係と、それに属する属性差異のラベルが付与される。

テストデータとして表2に示したような3系列の計30製品について製品情報を用意し、可視化を行った。図1は得られた可視化結果のうち、製品全体を表示したものを示す。図2は図1の一部を抜粋、拡大したものを示す。

図1と図2左側に示すように、開発者情報を考慮して属性比較を行うことによって、ディスプレイやCPU等の多数の製品で共通するラベル表記を整理し、同系列の製品間での大規模な属性継承と、製品系列間での属性「サブカメラ」の追加など、小規模な属性継承や追加を有向辺として可視化できている。

「iPod touch」系列製品

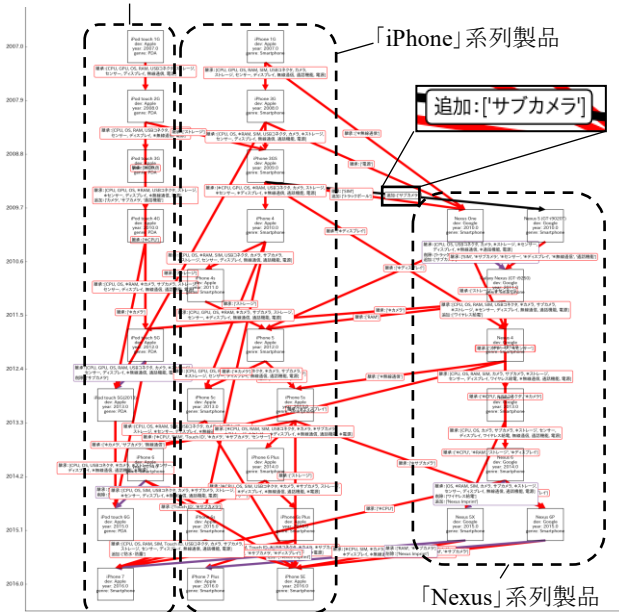


図1 開発者情報等を考慮した可視化結果

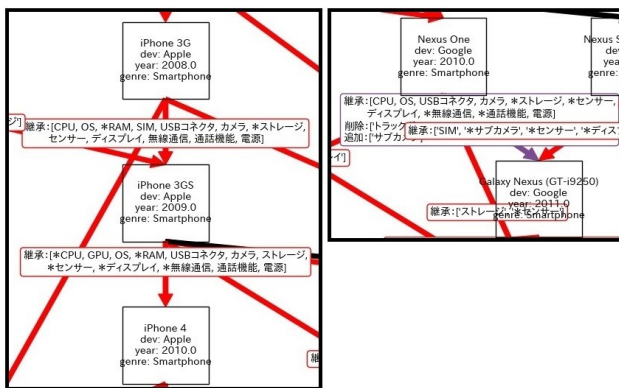


図3 開発者情報等を考慮した可視化結果(一部抜粋)

以上より、提案手法では製品系列に注目した可視化を提供できる。ここで、各製品系列はシリーズ製品など、それぞれ共通する開発者とジャンルをもつような似た製品のグループである。従って、製品系列の漸進的な発展と系列間の特徴的な影響関係を共に表示できる提案手法は、長い期間における多数の製品の発展を俯瞰する際に有効であると考えられる。

4. 適用実験

提案した手法と2章で述べた太田らの手法をそれぞれ表2に示す30製品に対して適用し、可視化を行った。各可視化結果を比較し、提案手法の有効性検証を行う。その後、本研究における開発者情報等を考慮した属性比較手法による可視化結果について、現実における製品属性の差異を基に可視化の妥当性を検証する。

4.1 視覚化手法の比較

提案手法による可視化のうち、開発者情報等を考慮した属性比較を用いたものを太田らの手法による可視化と比較し、手法の有効性を検証する。



図3 太田らの手法による可視化結果(一部抜粋)

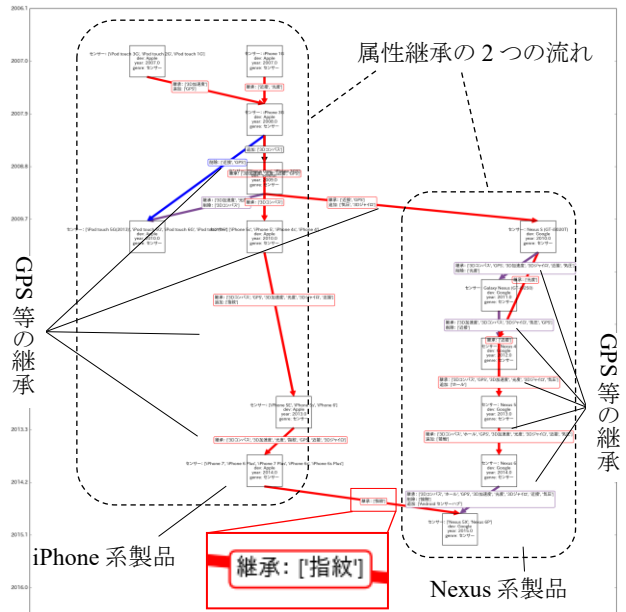


図4 属性「センサー」に関する可視化結果

(1) 先行手法の再実装

太田らの手法は、製品間で共通する属性の数を用いてその類似度を計算し、類似度が閾値より高ければそこに系統関係があると判定する。ここで、2つの製品 p_1, p_2 間における類似度は以下の式により求められる。

$$\text{類似度}(p_1, p_2) = \frac{p_1 \text{ と } p_2 \text{ の共通属性数}}{\text{全属性数}} \quad \dots (1)$$

太田らの手法では、分析対象とするのは単一のジャンルに属する製品のみとしている。一方、本研究では様々な製品ジャンルに属する製品を分析対象としているため、太田らの手法を本研究のシステムにそのまま用いることができない。そこで、複数の製品ジャンルに対応した2製品 p_1, p_2 間の類似度計算の式(2)を定義し、太田らの手法を本研究向けに実装し直した。具体的には、分析対象が多様な製品ジャンルに属していた場合、太田らの類似度計算手法における計算式(1)の分母である全属性の数が非常に大きくなる可能性が高いため、これを類似度計算の対象となる2製品の属する製品ジャンル情報を用いた式に置き換えた。

$$\text{類似度}'(p_1, p_2) = \frac{p_1 \text{ と } p_2 \text{ の共通属性数}}{N(G(p_1) \cap G(p_2))} \quad \dots (2)$$

ただし、 $G(p)$ を製品 p の製品ジャンルに属する製品集合、 $N(G(p))$ は $G(p)$ のもつ属性集合の要素数とする。

(2) 可視化結果の比較

図3に表2のデータを太田らの手法により可視化した結果を示す。なお、系統関係を判定する閾値は、太田らの実施した実験で最も製品系列を可視化できていた0.9とした。実験の結果、iPod touch系列の第4から第6世代までの4製品間に有向辺が引かれた。これらの製品は、それ以外のお他製品からの有向辺を持たなかった。また、互いに共通する属性の多い第1から第3世代のiPod touchの間でも、有向辺は描画されなかった。

太田らの手法では製品間の類似度に基づく系統関係のみを可視化するため、製品間の類似度は高くはないが属性の継承や削除等の関係が存在する場合に、それらの影響関係を表現できない。一方、図1と図2に示す本研究での提案手法による可視化では同一系列の製品間を示す大規模な属性継承や、「Nexus One」と「Galaxy Nexus」間での属性「トラックボール」の削除といった属性的部分的な変化を可視化できる他、入力された製品だけでなく、その各属性について影響関係を可視化できている。

4.2 可視化結果の分析

製品の各属性に対する可視化結果を分析し、現実の製品間における構成要素の影響関係と比較することで、提案手法の妥当性を検証する。

属性「センサー」について、3系列全ての製品を対象に影響関係を可視化した結果を図4に示す。全体的に、同じ開発者をもつ発売の近い製品は1つのノードに関連付けられる傾向がある。また、ノード間における属性継承の関係の多くは同じ開発者の製品間で描画されるため、2つの大規模な属性継承の流れが可視化されている。ここで、有向辺のラベルに注目すると、「GPS」など、2つの属性継承の流れにおけるそれぞれのラベルの表示には共通する属性の表記が多数ある。このことから、分析対象の製品群は、その開発者や製品ジャンルにかかわらず、似た構成のセンサー群を装備していたと推測できる。実際、これらの製品には共通する機能が多いため、そのためのセンサーの構成も類似するものとなっている。

他にも、2つの属性継承の流れをまたいで引かれた有向辺に注目することで、異なる開発者の製品間での小規模な影響関係が分析できる。例えばApple社のiPhoneは、2013年に発売された製品から指紋センサーを新たに装備している。一方でGoogle社のNexusは、2015年の製品で初めて指紋センサーが追加されており、可視化結果でもこれらのノードの間で有向辺が引かれている。ここから、Google社が2015年にNexus系列の新製品を開発する際、1年以上前に発売されたiPhoneの存在が影響を及ぼしたと推測することができる。

5. 課題

本研究で提案した製品間関係の可視化手法や、それを実装したシステムに見られる課題について述べる。

・可視化結果の表示方式

本研究によるシステムでは有向グラフを用いて製品間関係を可視化する。このとき、グラフの有向辺を直線的な矢印で表示するため、可視化結果において有向辺やノードの表示が重複し、有向辺のラベル等が読みづらくなってしまうことがある。よって、有向辺の表示が重なる場合は直線の代わりに曲線を使うなど、より視認性を上げるための工夫が必要である。

・属性比較手法

提案手法による可視化結果では、同一系列の製品が同時に出現する場合、有向辺のラベルに正確な属性差異が表示されなかった。また、幾つかの製品間において、製品発展史を理解する上で不必要な有向辺の表示があった。この問題に対して、属性差異の重複を排除する際の条件を変更するなどの対策が考えられる。

・製品情報の正確性や粒度

提案手法は、人手によって作成した製品情報ファイルを利用して、製品間関係の可視化を行う。このため、製品情報の正確さは作成者の分析対象に対する知識や、参照した文書の信頼性に依存する。また、提案手法は製品の属性比較の際にその粒度を考慮しないため、複数製品間で異なる粒度の同じ属性が存在した場合は、それらを正しく比較できない。例えば、ある製品では「センサー群」を属性としてもつが、別の製品ではセンサー群を構成する要素を各属性としてもつような場合、これらは全く異なる属性とみなされるため、属性の詳細な比較が行われない。以上の問題の解決策として、パターンマッチングや係り受け表現などを用いて自動的に文書から製品情報を抽出し、製品オントロジー等によって製品情報を一定の粒度で管理することが考えられる。

6. おわりに

本研究では、製品発展史の理解を支援するために、製品間での属性比較によって得られた属性差異を製品間関係として可視化した。この際、製品の開発者などの情報を考慮することによって、ある一連の製品の発展における他製品の影響に注目した可視化を目指した。

実験の結果、開発者名などの製品情報を考慮した属性比較により、重複している余分な情報を整理し、製品系列を可視化することができた。また、製品の各属性に対する影響関係の可視化結果を現実の関係と合わせて分析し、その妥当性を示した。そのほか、既存手法との比較の結果、提案手法による可視化は既存手法では表現できない関係を可視化できることを示した。

その一方で、提案手法では、属性比較において余分な情報を完全には排除できていないことがあった。また、製品情報の分析と製品間関係の可視化は自動化できたものの、製品情報自体は人手で用意しなければならない。従って、今後の課題としては、属性比較手法を改良することによる可視化精度の向上や、分析用の製品情報の自動収集によるユーザ負担の減少、オントロジーを用いた製品情報の管理などが挙げられる。

以上のことから、本研究で提案した手法を用いることで、製品群の発展を時系列的な図で可視化することができる。この可視化結果は製品の属性構成を基に生成されており、製品の開発者に対する取材等によらない。よって、開発者の意図に基づく部分の大きい現実の製品開発における他製品の影響を確実に可視化することはできないものの、製品の属性構成に基づいた製品系列の発展分析など、製品群全体の時系列的な変化を俯瞰する場合において特に有用であると考えられる。

参考文献

- [小林 2005] 小林慎一, 白井康之, 比屋根一雄, 桑野文洋, 犬島浩: インターネットリソースを用いた技術動向の時系列的分析, 電気学会論文誌C, Vol.125, No.5, pp.720-729, 2005.
- [太田 2013] 太田章悟, 武藤敦子, 犬塚信博: 年代順を考慮した工業製品の進化系統図作成法, 情報処理学会 第75回全国大会講演論文集, pp.457-458, 2013.