

ウェブサイト運営における“Make or Buy”

—クラウドと多元的取引費用—

An Overview of the Make-or-Buy Decision in Website Management

—Cloud Computing and Multiple Transaction Costs—

棚橋 豪*・長瀬 雄貴

Takeshi TANAHASHI* and Yuki NAGASE

要旨

法人組織のウェブサイト運営には、ハード面とソフト面の異質なタスクが共存している。具体的には、いわゆるサーバサイドとフロントエンドの事を指すが、これらは取引費用経済学に準えると、組織の内部で賄う場合（Make）と外部の委託事業者からサービスを購入する場合（Buy）に対応する。この文脈において、ウェブサイト運営者は、サーバサイド／フロントエンドの諸々について、取引費用節約の観点から Make or Buy の取引様式を選択を行っていることになる。

そこで法人組織の動向について明らかにするために、我が国の上場企業と大学を対象に、Netcraft Site Report にスクレイパーを介在させて、各ウェブサイトのサーバ情報を自動取得する。そして、このデータを手掛かりにして、法人組織のウェブサーバのクラウド化（サーバサイドの Make/Buy）とコンテンツ・マネジメント・システムの採用状況（フロントエンドの Make/Buy）の傾向を明らかにした。

一方、この実態調査と並行して、理論面で批判的考察と概念拡張を行っている。異質な要素からなるタスクに取引様式が多角的に絡み合った場合、従来の取引費用経済学のようなワンショットの意思決定論では説明がつかない。法人組織は諸タスクの取引様式の複合的選択に加えて、これに最適な組織編制が必要となる。さらに、これらは現時点において最適であるだけでなく、将来の能力開発や不確実性についての見通しを伴わなくてはならない。結果、変化を伴う環境のなか、意思決定それ自体の断続的なモニタリングが必要となるだろう。これらを考慮に入れたコスト体系を我々は「多角的取引費用」と命名し、既存の取引費用経済学との差別化を図った。

キーワード：クラウド、アウトソーシング、スクレイピング、取引費用経済学、中間組織

I. 導入 そのウェブサイトは何処にあるのか？

1. ホームページの裏方

今、あなたは、さまざまに奈良公園を訪れている。スマホを片手に、奈良公園に関する観光案内やグルメ情報を検索している。あなたは、奈良公園の外れにある迷宮のように入り組んだ古民家街の一角に、めばしい喫茶店があることを知り、その店の公式サイトにアクセスする。そこで気になる情報は、次のようなものだろう。店内の写真、ドリンクやケーキの種類、価格、営業時間などだ。ひょっとしたら、その店主のプロフィールまで探ろうとするかもしれない。

あなたは、このカフェのことが気に入り、正確な所在地を確かめようとする。しかし、あなたがどれだけこの店の熱狂的なファンになったとしても、そのウェブサイトの物理的な所在地について興味を抱くことはないだろう。そのウェブサイトのホストサーバは、店内の奥にあるいわゆる「自宅サーバ」だろうか？ それともカンガルーの国のデータセンターだろうか？ などと思い巡らすことは、まずもって、ない。公園のシカ達と戯れて疲れ果てて、一刻も早くアイスカフェオレとピスタチオケーキにありつきたいあなたが知りたいのは、他でもないその店舗の場所であって、ウェブサーバの設置場所ではないだろう。

だが、ウェブサイト運営者にとって、ウェブサーバの所在は看過できない重要課題となる。サーバを所有して自身で保守管理していくのか（オンプレミス）、それとも外部に委託するのか（クラウド）によって、マネジメントのあり方が大きく異なるからだ。

2. ウェブサーバの所在地を探る

もっとも、零細店舗がホームページのために自宅や店内にウェブサーバを構築することは稀だろう。サーバの購入、そしてこれを維持する費用は計り知れない。事実、これを避けるために、外部の事業者からサーバをレンタルするか、プロバイダ契約に付随するホームページスペースを利用するのが一般的だ。それでは大企業なら、ウェブサーバは本社ビルないしは直轄のデータセンターに存在するのだろうか？

仮にこのような疑問を持ったとしても、公表されているドメイン名では、ウェブサーバの所在地を探る手掛かりにはならない。だが、ウェブサイトのサーバ情報が閲覧可能なサービスが存在しており、これを活用すれば、ウェブサーバの在処が推測可能になる。

このサービスの代表格が Netcraft Site Report（以下 Netcraft）である。一例として Netcraft でソニーのウェブサイト www.sony.com を見てみよう。サーバのオーナー情報を示す項目 Netblock Owner は Amazon Technologies Inc. とある。念を押して、サーバ名が記載されている項目 Reverse DNS を確認すると amazonaws.com とある。これより www.sony.com のウェブサーバの所有者はクラウド事業者としてのアマゾン社であり、ソニー社内にはウェブサーバが存在しないことが推察できる¹。

3. Make or Buy の意思決定論

先のソニーの例からも分かるように、企業はすべての製品・サービスを自社内で生産しているわけではない。組織内部で生産するべきか、それとも外注するべきかを決定するのは、言うまでもなく企業の「どちらが得か」というコスト意識である。以下に、これを理論化したフレームワークの由来に触れておこう。

新古典派経済学をはじめとした主流派において、意思決定論はすべからく市場取引を自明視するところからはじまる。ここでは、市場システムの価格メカニズムのみが注目されるが、「見えざる手」がシステム維持のタスクを前に手を抜いたりミスを起こしたりすることは考慮されない。また、市場を逸脱したものは無視される。それらは「市場の失敗」や「外部性」と呼ばれ、厚生経済学が扱う環境問題のような周縁的問題として扱われてきた。

しかし、この外部性には、環境問題とは全く別のタイプも存在する。それは意外にも「企業」という取引制度である。1937年、ロナルド・コースは、市場制度は完全なものではなく、市場の利用それ自体に費用が発生することを指摘した。そして彼は、この費用を節約する形で企業組織が編制されることを説いた。ただし、この論文が注目を集めるのは、コース自身も認めるように、それから半世紀近く経ってからのことである²。

1970年代、オリバー・ウィリアムソンは、コースを再評価した上で、経済主体が「階層組織と市場」という二

1 さすがに Amazon のクラウドサーバの所在地までは特定することはできない。あくまでもウェブサーバの所有者が判明する程度だが、本論の関心からすればこれで十分である。

2 Coase (1937, 1988)

つの取引制度を選択するという理論枠組みを提唱した³。これは取引費用経済学（Transaction Cost Economics, 以下 TCE）と呼ばれ、これ以降、経済学、法学、経営学の垣根を超えた一大フレームワークとなっていく。TCE において、経済主体 X がある財を調達する場合、X は「組織で内製する費用」と「市場取引で発生する費用」を比較することになる。階層組織内の命令・権限のもとで内製するのか（Make）、市場から購入するのか（Buy）という Make or Buy を前にして、取引費用節約の観点から最適な取引制度を選択するのである。

先のウェブサーバの所在地がオンプレミスなのかクラウドなのかという問題は、そのままウェブサーバの Make or Buy 選択問題として考えることができる。過去に取引費用論からクラウドを論じたものに棚橋（2010）がある⁴。以後 10 年間は、クラウド化が一つのトレンドとなるが、Makhlouf（2020）は、企業のクラウド導入時やそれ以後の顛末について取引当事者へのインタビュー調査を行い、クラウド導入後に想定外の取引費用が発生、または潜在していることを明らかにしている。

本論は、インタビュー調査とは別の方法を探り、我が国の企業と大学におけるウェブサイトのクラウド化の現状についてマクロデータから迫りたい。また、Makhlouf（2020）の分析は、表向きは取引費用論的な分析を試みているが、皮肉にも企業の意思決定における「取引費用節約の失敗」の事例集になっている。その理由の一つは、既存の TCE がクラウドを語るには素朴すぎるからである。取引費用概念それ自体の考察を介して、あらためてウェブサイト・マネジメントに臨む必要があるだろう。

4. リサーチ対象とその方法

本論は、上場企業 3867 社と大学 800 校について、Netcraft からウェブサーバ情報を取得した。尚、この手順に関するスクレイピング技法は別途 APPENDIX に記載している。

この取得データを基にして、II 章では、オンプレミス/クラウドの比率、主要クラウド事業者の傾向をグラフ化する。これに加えて、コンテンツ・マネジメント・システム（Contents Management System, 以下 CMS）の採用状況にも注目する。これはウェブサイト運営におけるソフト面であるフロントエンドについて、これの内部化の程度を推し測る指標となり得る。

III 章では、ソフト/ハード、内部/外部の組み合わせから、4 象限のマトリックスを構成し、II 章のデータを再プロットする。ここでは既存の取引費用論を超えて、時間や組織再編成といったさらなる取引費用問題が扱われる。最後の IV 章において、本論の意義と残された課題などに簡単に触れておきたい。

II. 法人組織ウェブサイトにおけるクラウド化と CMS 採択状況

1. 企業ウェブサイト編

上場企業 3867 社の公式ウェブサイトについて、ウェブサーバのオーナー情報について調査した。データ取得は自動で行ったが、類型化自体は手動で行っている。この方法について簡単に触れておこう。サーバのオーナー情報を示す Netblock Owner が重複している場合は、クラウド事業者である可能性が高い。ゆえに、検索結果が単体

3 Williamson (1975, 1996)

4 ゼロ年代、「クリック&モルタル」という二分法のなかで、ウェブ店舗「クリック」は実店舗「モルタル」に比べて限界費用が限りなくゼロに近いことが強調されたが、しかし現実には「クリック」で集客を募るにはサーバという「モルタル」の保守管理費が無視できないことが明らかとなっていく。この論文が記された 2010 年頃からクラウドサービスが一般にも知られるようになっていたが、このクラウドサービス（cloud）について、企業はサーバ保守管理の Make or Buy 問題に直面し難渋すること、そしてもう一つのクラウドである群衆（crowd）との取引は反システムの戯れの範疇にあり、経済学的な説明枠組みから逸脱していることを説いた。

尚、後者の crowd の議論は、現在の文脈では SNS のマーケティングに相当するだろう。SNS 内のコミュニケーションではビジネス色が強くなるほど一般ユーザは冷めて離れてしまうという矛盾がある。結果として、企業や個人事業主側も「マーケティングしない、というマーケティング」とも言うべきアクロバティックな戦略が不可避となっている。

のものに絞り、これが当該企業名と合致しているかを確認してオンプレミスかどうかを判別した。

わずかな例外として、クラウド事業者のウェブサイトが自社クラウドに存在する場合は、自社内のサーバにあるもの、すなわちオンプレミスとしてカウントした（例：さくらインターネット株式会社）。また別法人でも権限関係が明確な子会社のクラウドにある場合もオンプレミスとした（例：ウェブサイト/クラウド、トナミホールディングス株式会社/トナミシステムソリューションズ株式会社）。ただし、かつての子会社であっても関連性が希薄なものや巨大資本グループ内でのクラウド化は別法人へのアウトソースとみなしている（例：川崎重工業株式会社/クベニックソリューション、近鉄百貨店/近鉄ケーブルネットワーク）。

以上の集計ルールから導かれた企業ウェブサイトのクラウド化の傾向は、クラウド化した企業は3686社、社内でも所有する企業（オンプレミス）は181社だった。上場企業内の95%が、ウェブサーバをクラウド事業者にアウトソースしていることが判明した。

シェアの高いクラウド事業者についても触れておく。採択件数50件以上のクラウド事業者15社をピックアップした。この15社だけでクラウド事業者全体の78%を占めている。採択数トップは、Amazon系（845件）である。ただし、これは以下4社のAmazon.com Inc.（415件）、Amazon Data Services Japan（369件）、Amazon Technologies Inc.（55件）、Amazon Web Services Elastic Compute Cloud EC2 JP（6件）を一括している。もっとも、仮にこれらを独立してカウントしても、Amazon系2社がツートップである。

続いて国内クラウド事業者のSAKURA Internet Inc.（367件）、KDDI Web Communications Inc.（283件）、XSERVER Inc.（221件）が並ぶ。次のAkamai Technologies Inc.（169件）だが、この結果を額面通り受け取るのは差し控えた方が良くかもしれない。アカマイ社はCDN事業者だからである。正確なウェブサーバ情報の取得方法については、スクレイピング技法上の今後の課題であろう⁵。尚、Server Hosting Service（140件）とOpen Computer Network（77件）はNTTコミュニケーション系列であり、両者を合わせて一つと見なすならば、3番手のXSERVER Inc.（221件）に並ぶ勢力となっている。（図1）

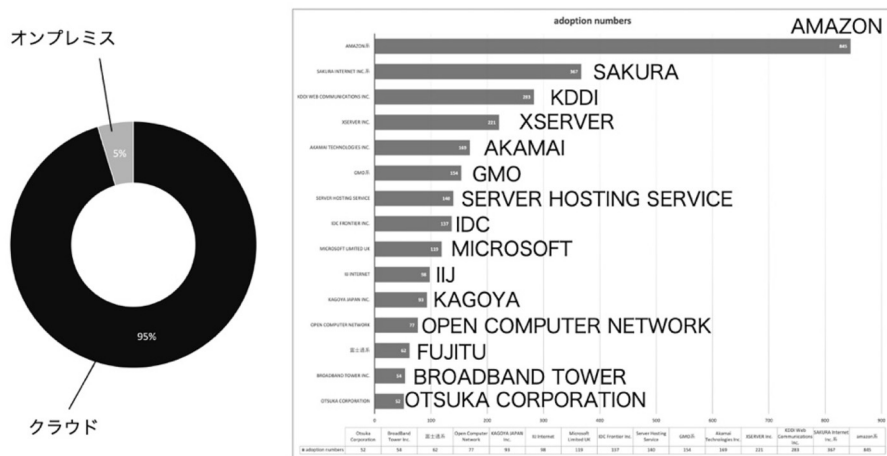


図1 企業ウェブサイトにおけるクラウド比率とクラウド事業者シェア

Netcraftは、ウェブサーバにインストールされたソフトウェア技術の詳細をも明らかにしてくれる。ここでは

5 大手ウェブサイトのNetcraft検索結果に、しばしばNetblock OwnerがAkamaiと表示されることがある。アカマイ社は、サーバとクライアント間を円滑につなぐコンテンツデリバリーネットワーク（CDN）事業者でありクラウド事業者ではない。したがってこの場合、NetcraftはウェブサーバではなくCDNのキャッシュサーバを検知している可能性がある。尚、アカマイ社の発展は、インターネット環境の歴史そのものと言ってもよいだろう。これの詳細については、あきみち・空閑（2011）の7章と小川（2014）を参照せよ。

CMS 採用の有無に注目したい。CMS とは、HTML や CSS などの専門知識がなくてもウェブページ内のテキストや写真が更新可能となるアプリケーションである。したがって、ある企業のサーバに CMS が存在するという事は、ウェブページの更新において、外部委託事業者だけでなく内部の社員も参画可能であることを示唆している。Ⅲ章の論点先取りを許せば、本論において CMS 採択の如何は、ソフト面・フロントエンドの内部化／外部化に対応している。

企業ウェブページにおける CMS 採用状況と CMS シェアを下記に示す。CMS 採用企業は、3867 社中 1315 社となり、全体の 34%を占めている。さらに、1315 社中 1116 社が WordPress を採用しており、WordPress が圧倒的シェアを誇っていることを裏付ける結果となった。(図 2)

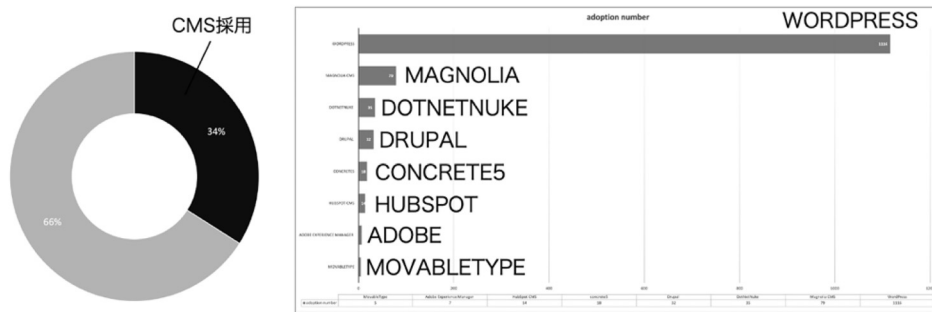


図 2 企業ウェブサイトにおける CMS 採用比率と CMS シェア

2. 大学ウェブサイト編

企業編と同様に、大学ウェブサイトについても考察する。国公立大学・私立大学 800 校のウェブサイトについて、ウェブサーバのクラウド化比率、クラウド事業者シェア、CMS 採択比率、CMS シェアをグラフ化した。

大学ウェブサイトにおけるクラウド化の比率であるが、クラウド化した大学は 800 校中 620 校となり、全体の 77%を占めている。上場企業のクラウド化が 95%であることを考えると、相対的に大学ウェブサイトのクラウド化は低い。この背景には、歴史的にインターネットが大学を基点に拡充したこと、また情報基盤センターを有する大学が一定数存在することとも関係があると思われる。

10 件以上採択されているクラウド事業者を抽出したところ、SAKURA Internet Inc. (134 件) がトップで、これに Amazon 系 (71 件)、KDDI Web Communications Inc. (65 件)、XSERVER Inc. (50 件) が連なる。企業編と同じく Amazon 系は、以下 3 件の Amazon Data Services Japan (38 件)、Amazon.com Inc. (31 件) そしてその他 (2 件) を合算している。留保事項として、Japan Network Information Center (12 件) については、データ自動取得時の結果は 39 件だったが、Hosting company 名が「.ac.jp」で終わるものは、ウェブサーバがオンプレミスであると見なして、これから除外している。(図 3)

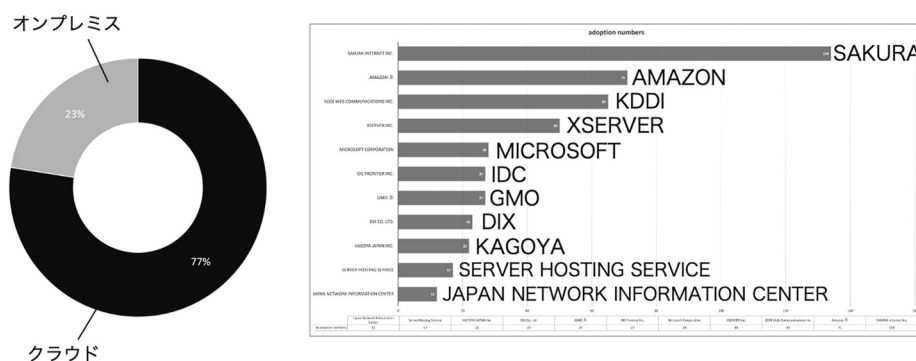


図 3 大学ウェブサイトにおけるクラウド比率とクラウド事業者シェア

大学ウェブサイトの CMS 採用状況については、800 校中 284 校が採用しており、全体の 35%を占めている。これは先に見た上場企業の CMS 採用状況と同程度である。やはり WordPress が CMS 採用 284 校中 252 校に選ばれており、大学ウェブサイト運営においてもデファクトスタンダードとなっていることが判明した。(図 4)

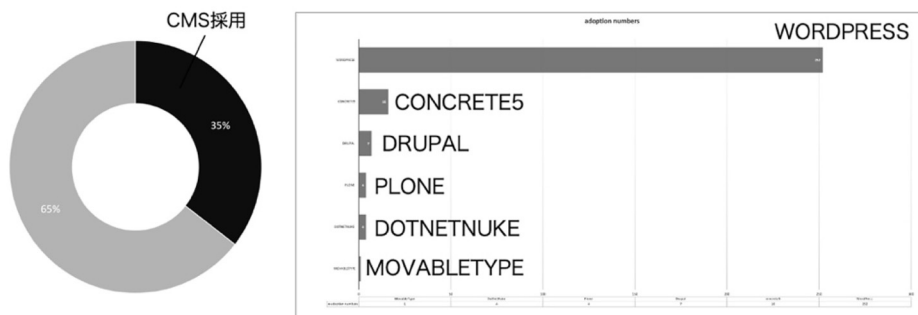


図 4 大学ウェブサイトにおける CMS 採用比率と CMS シェア

Ⅲ. クラウドと多元的取引費用

1. 市場と階層組織のあいだ

TCE に対する典型的な批判の一つに、Make or Buy という二元論が現実離れしている、という指摘がある。社会学者マーク・グラノヴェッターは、この極端な二元論を批判した上で、第三の関係性として「信頼」を掲げた⁶。本来の経済取引は社会関係に埋め込まれたものであり、交換は本来的にコミュニケーションである。そうであるにもかかわらず、経済分析は合理性や計算高さに終始しており社会的側面を見落としている、という論点である。

ウィリアムソンはこれへの反論において、自身の TCE を拡張している。取引費用を連続的な尺度として考えれば、市場と階層組織の「あいだ」にあるような取引制度を導くことができる。これを「ハイブリッド」と呼び、ウィリアムソンにとって「信頼」とは、この「ハイブリッド」となる。この意味での「信頼」は、もはや経済学を超えた批判理論ではなく、あくまでも打算的な関係性の一つでしかない。本論の立場としては、グラノヴェッターの意を汲みながらも、取引費用経済学の中にとどまり、しかしウィリアムソンとは別の方策を探りたい。

その糸口として、もう一つの「あいだ」としての「中間組織」に注目しよう。今井賢一他（1982）は、2軸の独立した取引費用を直行させて、市場と階層組織の二つの極他に、中間領域が存在することを示した。この中間組織論は、「ケイレツ」などに代表される、かつての日本企業の特異性やあいまいな取引関係を説明するモデルとして注目された。この中間組織論は、本論の主題とは直接の関連はないが、2軸の取引費用を直行させてマトリックスを構成する着想に注目したい。なぜならこのアイデアは、ハード面とソフト面が絡んだウェブサイト運営の考察に有効だからである。

そこで本論は、ウィリアムソンの一元的な取引費用ではなく、諸々の独立した取引費用が絡み合ったコスト体系を「多元的取引費用」(Multiple Transaction Cost)と命名する。もっとも、このマトリックスは、あくまでも多元的取引費用の端緒にすぎない。実際には、このマトリックス上の制度選択に伴い、さらなる多元的取引費用が連なっていく。その一つが、複合的な取引形態に応じた組織の再編成に伴うコストである。組織再編成の放棄は、選択された取引制度が期待されたパフォーマンスを発揮しない可能性があり、別種の取引費用を招くかもしれない。

6 Granovetter (2017) は社会ネットワーク論の泰斗であるが、今井賢一他 (1982) の中間組織論もまた今井賢一・金子郁容 (1988) に代表されるように産業組織のネットワーク論へ転換していく。そして、「ネットワーク」モデルに可能性を見ながらも、その曖昧模糊とした「きわどい」概念に飲み込まれ、肝心の理論構築は霧散化していったように思われる。ただし、点と線から市場や社会の構造に迫ることは、現在でも理論的魅力をなんら失ってはいないだろう。

さらに時間の問題がある。契約後や組織再編成後も、環境変化が激しい IT 環境では断続的なモニタリングが必要になる。また、仮に軌道修正の必要性に気づいたとしても、契約期間などの拘束条件により、機敏な意思決定には制約がかかるだろう。これらも多角的取引費用の一つである。

以上のように、多角的取引費用とは、既存の TCE のように「場当たりに取引費用を節約する」だけでは捕捉できないデリケートな状況を射程に収めた概念である。

2. 多角的取引費用とウェブサイト運営

まずは先のマトリックスに関して、一般的なモデルを示しておこう。ある製品 X において、これが全く異なった特性を持つデバイスやサービスから成り立っているとき、一元的な取引費用のメトリックから、これを内製化するのかそれとも外注するのかといった二者択一の意味決定はナンセンスである。その属性に応じた多角的取引費用のなかで、それらに応じた内製と外注を適切に選り分ける必要がある。場合によっては、内製と外注が混在した取引形態になるかもしれない。

より具体的に、次のような条件を与えて考察する。製品 X は「ハード」と「ソフト」という二つの独立した属性から成る。このとき、ハード／ソフトそれぞれに、「内製」と「外注」を意味する内部／外部の選択が存在する。結果、製品 X について次の 4 タイプの取引形態があり得る。ハードもソフトも外部である純粋な「市場」、ハードもソフト内部である純粋な「階層組織」、ハードは内部—ソフトは外部である「中間組織 a」、ハードは外部—ソフトは内部である「中間組織 b」である⁷。これらは次の 4 象限に示される。

		ハード	
		外部	内部
ソフト	外部	II 市場	I 中間組織 a
	内部	III 中間組織 b	IV 階層組織

ここで、ウェブサイト運営もまた、ハードとソフトからなる複合的なタスクだと言える。ウェブサイト運営におけるハード面とはウェブサーバの保守管理に対応し、またソフト面とはウェブコンテンツの更新作業に相当する。そして各々を組織内部で処理するのか、外部へアウトソースするのかという意思決定があり得る。II 章で見たように、内部／外部の指標として、サーバサイドではクラウド採択状況、フロントエンドでは CMS 採用状況が手がかりとなる。

もっとも、CMS 採用の有無がそのまま、ソフト面の外部／内部の実態だと考えることには、議論の余地があるだろう。確かに、ある組織のウェブサイトが CMS を採用していなくても、組織内部の一部のシステムエンジニアが HTML や CSS のコードを直接書き換えて運営している場合も考えられる。他方、CMS を採用していても、組織内部のスタッフは一切関与せずに外部委託事業者だけがこれを使用していることも考えられる。

本論が定義する CMS の意義は、実態を超えた側面を含んでおり、それはリスクや将来に備えた潜在能力に関係している。例えば、組織内部のシステムエンジニアが退職や急病で不在になった場合、CMS なくして内部の一般職員が迅速にこれを代行することは難しい。また、かつての書類の決裁スタンプラリーのように、ウェブ更新依頼メールが社内中でたらい回しになった挙句の果てに、外部委託事業者がこれをすべて代行している状況があったとする。これを憂慮したトップが、フロントエンド更新作業の内部化を推進しようとするとき、すでに CMS が存在すれば、内部化に伴う取引費用は社員の CMS 研修程度で済むだろう。反対に CMS が不在であれば、研修以前に抜本的なウェブページ自体の再構築が必要となる⁸。

7 今井賢一他（1982）の中間組織論の現代的意義は、市場と階層組織の「あいだ」を単に指摘したことではなく、この「あいだ」に複数のタイプが存在するという点である。

そこで前章のデータを、サーバサイド/フロントエンド、内部/外部からなるマトリックス内のカテゴリー別に再集計をかけて、4象限の各件数とその割合を算出した。

企業編のマトリックスでは、もともとウェブサイトのクラウド化が圧倒していたので、実質的に前章の企業CMS採用比率と大差はなく、第Ⅱ象限（市場）と第Ⅲ象限（中間組織 b）の二極化がみられる。企業のウェブサイト運営において、クラウド採用は自明だが、CMS採用はそれほど進んでいない。ただし、これが均衡的な状態なのか移行中なのかは、今後の時系列分析によって明らかにされるべき論点である。（図5）

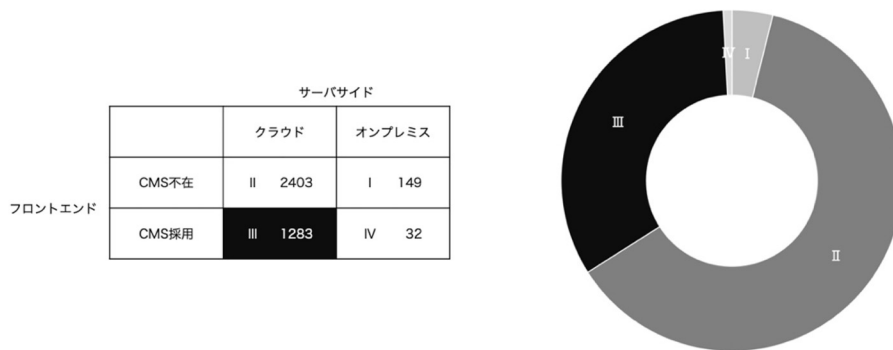


図5 企業ウェブサイトにおけるマトリックスとカテゴリー別シェア

大学編マトリックスでは、各象限が比較的分散して共存している。その中で、数値だけを見ればサーバもフロントエンドも外注する第Ⅱ象限（市場）がトレンドのように見える。

しかしながら、多元的費用論の立場からすれば、これを最適な意思決定の結果と断定するのは早計である。インフラ面での転換は迅速な意思決定ができないからである。また、仮に第Ⅲ象限へ移行したとしても、これに応じた組織デザイン、例えばCMSを使いこなす内部職員の育成などを怠れば片手落ちの意思決定となる。このマトリックスが語るのは、あくまでも表向きの取引形態であって、組織編制がこれにチューニングされているかどうかは、別次元の問題として潜在している。（図6）

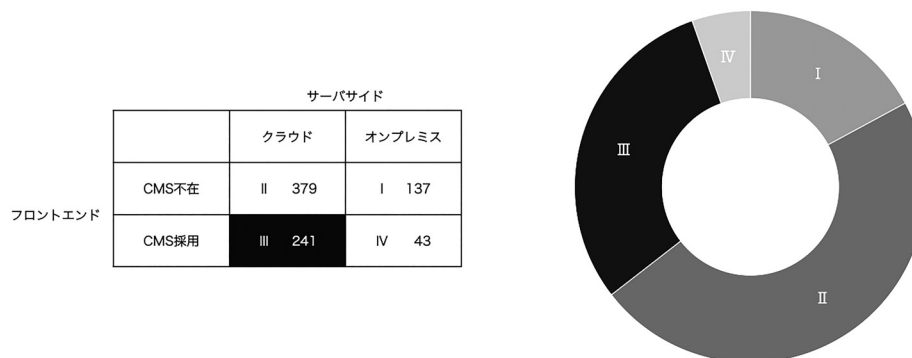


図6 大学ウェブサイトにおけるマトリックスとカテゴリー別シェア

以上、一般的にウェブサイトの運営と言えば、表層的なウェブデザインなどに関心が向きやすい。しかしながら、ウェブサイト運営の本質は、多元的取引費用を前にした制度選択であると同時に組織デザインでもある。この事を

8 ここでのCMSとは、現状デファクトスタンダードであるWordPressを想定しているが、もともとブログ管理用のソフトであったことから、WordPressを採用しているからといってウェブサイト全体がWordPress上で構築されているという保証はない。「新着情報」や「IR」といった一部のページだけに適用されているケースもあり得るだろう。

看過した Make or Buy の意思決定は「取引費用節約の失敗」に終わり、予期せぬ取引費用に難渋することになるだろう。

IV. 結語と残された課題

ウェブサイトの元は、物理的にどこに在るのだろうか？これが本論執筆の発端となる素朴な疑問だった。幸いにもこの疑問に応えられるウェブサービスが存在しており、データ取得を自動取得するスクレイパーをこれに介在させて、我が国の上場企業と国公立・私立大学をほぼ網羅したウェブサーバのリストを入手した。この方法の詳細は APPENDIX に記している。

企画当初は、より多角的な分析のために、組織の規模を加味した考察も目論んでいた。しかしながら、従業員数や学生数などを扱った基本的なオープンデータが容易に見つからず、これに関する分析は見送っている。この点は今後の課題だが、このままでも今後の時系列調査の端緒とはなり得る。また本論は、サーバ情報に加えて、追加的に CMS にも注目したが、これが「ハード/ソフト」と「内部/外部」のマトリックスから取引形態を類型化することに繋がり、理論面でのさらなる考察が可能となった。

尚、本論は集計結果としてのマクロデータのみを扱ったが、網羅的なリストがある以上、特定大学や独自のカテゴリーに限定した、よりミクロな動向調査が可能である。本学園の経営情報部がこの詳細なリストと手法のレシピを所有していることは、一研究を超えて実務面においても意義深いものがあるだろう。

最後に、現代ではウェブサイトを衛星のように取り囲む SNS の存在も無視できなくなっている。これらについては稿を改めて考察したい。

文献

- 1) あきみち・空閑洋平 (2011) 『インターネットのカタチ：もろさが織りなす粘り強い世界』オーム社.
- 2) 今井賢一他 (1982) 『内部組織の経済学』東洋経済出版社.
- 3) 今井賢一・金子郁容 (1988) 『ネットワーク組織論』岩波書店.
- 4) 小川晃通 (2014) 『アカマイ：知られざるインターネットの巨人』KADOKAWA.
- 5) 棚橋豪 (2010) 「二つのクラウドと取引費用」『日本情報経営学会誌』 Vol.30, No.4, pp.81-90.
- 6) Coase, R. H. (1937) “The Nature of the Firm,” *Economica*, Vol.4, pp.386-405 (宮澤健一他訳 (2020) 『企業・市場・法』筑摩書房所収).
- 7) Coase, R. H. (1988) *The Firm, the Market, and the Law*, University of Chicago Press (宮澤健一他訳 (2020) 『企業・市場・法』筑摩書房).
- 8) Granovetter, M. (2017) *Society and Economy: Framework and Principles*, Belknap press (渡辺深訳 (2019) 『社会と経済：枠組みと原則』ミネルヴァ書房).
- 9) Makhlof, R. (2020) “Cloudy Transaction Costs: A Dive into Cloud Computing Economics,” *Journal of Cloud Computing: Advances, Systems and Applications*, Vol.9, No.1, pp.1-11.
- 10) Williamson, O. E. (1975) *Markets and Hierarchies: Analysis and Antitrust Implications*, Free Press (浅沼萬理・岩崎昇訳 (1980) 『市場と企業組織』日本評論社).
- 11) Williamson, O. E. (1996) *The Mechanisms of Governance*, Oxford University Press (石田光男・山田健介訳 (2017) 『ガバナンスの機構：経済組織の学際的研究』ミネルヴァ書房).

APPENDIX

1. 大学・企業 URL データの作成

本研究にて調査対象とした大学 URL の一覧データは、文部科学省が「教育機関関連リンク集」¹として公開する内容を加工し作成した。また企業 URL の一覧データは「日本株の配当金データベース」というウェブサイトにて「国内上場企業の Web サイト URL 一覧」²として公開されている内容を加工し作成した。大学・企業 URL の一覧データは1列目に大学・企業名、2列目に大学・企業ウェブサイトのルートディレクトリにあたる URL を記述した、CSV 形式のファイルである。大学・企業 URL のデータについて、形式例を図1に示す。

```
北海道大学,http://www.hokudai.ac.jp/  
北海道教育大学,http://www.hokkyodai.ac.jp/  
室蘭工業大学,http://www.muroran-it.ac.jp/  
小樽商科大学,http://www.otaru-uc.ac.jp/  
帯広畜産大学,http://www.obihiro.ac.jp/  
:  
東日本国際大学,http://www.shk-ac.jp/  
福島学院大学,http://www.fukushima-college.ac.jp/
```

図1 大学・企業 URL のデータ形式例(中略)

作成した CSV は、大学は 2022 年 8 月 5 日時点で掲載されていた、国立大学・公立大学・私立大学を対象にした 806 大学分³、企業は 2022 年 9 月 21 日時点で掲載されていた、上場企業を対象にした 3,879 社分⁴のデータである。

2. Netcraft Site Report

ウェブサイトの解析にはイギリスのネットワーク関連企業である Netcraft 社が公開する「Netcraft Site Report」^{5 6}を使用した。同サービスは調査対象のウェブサイトの URL を指定することで、対象のウェブサイトがホストされているネットワークや IP アドレス・DNS の情報、ウェブサイト上で使用されている技術・プログラム・ライブラリの推定情報、SSL/TLS 証明書の状態、トラッカーの稼働状況などを一般利用者視点から調査し、調査結果をサービス利用者に提供する。

¹ https://www.mext.go.jp/b_menu/link/1294885.htm

² <https://dividend-jp.com/tool/12118/>

³ 806 大学の内、情報取得結果がエラー・不明のデータを除外し、有効データは 800 大学分とした

⁴ 3,879 社の内、情報取得結果がエラー・不明のデータを除外し、有効データは 3,867 社分とした

⁵ <https://sitereport.netcraft.com/>

⁶ Netcraft 社の利用規約 (<https://www.netcraft.com/terms/>、2022/10/6 閲覧)ではロボットの仕様について全面禁止はしていないが、Netcraft 社の裁量でアクセスをブロックすることが可能である旨の記載がある。本研究では、サービスに過負荷をかけないようにアクセス間隔を十分に長く確保する等の配慮を実施した。

3. データ取得・分析プログラムの動作環境

大学・企業 URL データの各ウェブサイトの情報を取得し、データを解析するため、Python3 でスクリプトを記述した。またウェブサイトの情報取得には Google Chrome ブラウザをプログラムで操作できる ChromeDriver を使用した。使用したソフトウェア・ライブラリとバージョン・用途を表 1 に示す。

表 1 本研究のデータ取得で使用したソフトウェア

ソフトウェア	バージョン	用途
Python3	3.10.7	メインスクリプトの記述
ChromeDriver	105.0.5195.52	Google Chrome をプログラムから自動制御
Selenium	4.4.3	ChromeDriver を制御・取得データからの抽出
BeautifulSoup4	4.11.1	HTML をパースし指定した要素からデータを抽出

4. サイトデータ取得スクリプト

大学・企業 URL データを元に Netcraft Site Report の調査結果を取得する Python3 スクリプトを示す。入力 CSV ファイルは変数 csvfilename で指定する。また結果出力は Netcraft Site Report の中からネットワーク情報・ウェブサイト技術情報を含む部分の HTML ソースを抽出し、大学・企業名をファイル名としたプレーンテキスト形式で保存する。

```

from selenium import webdriver
from selenium.webdriver.common.by import By
import time
import urllib.parse
import random

chromedriver_path = "./chromedriver105.exe"
csvfilename = "./調査対象一覧.csv"
wait_netcraft_show = 15
wait_interval = 20
netcraft_url = "https://sitereport.netcraft.com/?url="

# CSVの読み込み
f = open(csvfilename,encoding="utf-8")
l = f.readlines()

# ChromeDriverの初期化
chrome_service = webdriver.chrome.service.Service(executable_path=chromedriver_path)
chrome = webdriver.Chrome(service=chrome_service)

# メインループ:CSVの行ごとにループ
for i in range(len(l)):
    name,url = l[i].split(",")
    print("Target:" + name + "(" + url + ")")

    # ページを読み込み、表示完了まで待機
    chrome.get(netcraft_url + urllib.parse.quote(url))
    time.sleep(wait_netcraft_show)

    # メイン部分のみHTMLソースコードをプレーンテキスト形式で保存
    f2 = open(name+".txt",mode="w",encoding="utf-8")
    f2.write(chrome.find_element(By.CLASS_NAME,"page-content").get_attribute("innerHTML"))
    f2.close()

    # アクセスが連続しないようランダム時間待機
    r = random.randint(wait_interval-5,wait_interval+5)
    print("waiting for "+str(r)+" seconds...")
    time.sleep(r)

f.close()
chrome.quit()
print("done")

```

5. ネットワーク情報抽出スクリプト

Netcraft Site Report から取得したデータを格納した、ウェブサイトの調査結果のテキストファイル群から、ネットワーク情報についての結果を抽出し、単一の CSV ファイルに集計出力する Python3 スクリプトを示す。本研究ではネットワーク情報のうち、“Site”、“Netblock Owner”、“Hosting company”、“Hosting country”、“IPv4 address”、“Organisation”の 6 項目を出力対象とした。

```

from pathlib import Path
from bs4 import BeautifulSoup
import glob
import re

school = {}
r = r'^[/]+%.txt$'
targetnetdata = ["Site", "Netblock Owner", "Hosting company",
                 "Hosting country", "IPv4 address", "Organisation"]

# 取得データを格納したテキストファイルを収集
filelist = glob.glob("./*.txt")
# 各テキストファイルの調査
for file in filelist:
    # HTMLソースの読み込み
    print("Progress in %s" % file + "%s...")
    f = open(file, encoding="utf-8")
    src = f.read()
    f.close()

    # 組織名を取得、全組織分のデータを格納する変数の中にデータ枠を作成
    m = re.search(r, file)
    schoolname = m.group().replace(".txt", "")
    school[schoolname] = {}
    for i in targetnetdata:
        school[schoolname][i] = ""

    # HTMLソースからネットワーク情報を格納する要素を抽出
    soup = BeautifulSoup(src, "html.parser")
    dd = soup.find(id="network_table_section")
    dd2 = dd.find(class_="table--list")
    dd3 = dd2.find_all(name="tr")

    # 項目を順に調べ出力対象とした項目の部分記録
    for data in dd3:
        colname = data.find(name="th").text
        if colname in targetnetdata:
            # データ記録時にはHTMLソース・出力結果に含まれる不要文字を除去する
            school[schoolname][colname] = (data.find(name="td").text
                                           .replace("\n", "")
                                           .replace(" ", "")
                                           .replace("%xa0", "")
                                           .replace(" (VirusTotal)", "")
                                           .replace(", ", ""))

# 出力データのヘッダを作成
out = "schoolname"
for i in targetnetdata:
    out = out + "," + i
out = out + "\n"

# 出力データの中身を作成
for sch in school.keys():
    out = out + sch
    for i in targetnetdata:
        out = out + "," + school[sch][i]
    out = out + "\n"

# 結果をCSVファイルに出力
f2 = open("network.csv", mode="w", encoding="utf-8_sig")
f2.write(out)
f2.close()
print("done.")

```

6. ウェブサイト技術情報抽出スクリプト

Netcraft Site Report から取得したデータを格納した、ウェブサイトの調査結果のテキストファイル群から、ウェブサイトが使用している技術情報についての結果を抽出し、単一の CSV ファイルに集計出力する Python3 スクリプトを示す。使用している技術はウェブサイトごとに異なるため、まずは全ての組織のデータを調べて使用されている技術情報を収集した上で、各組織の技術情報の有無を出力とした。

```
from pathlib import Path
from bs4 import BeautifulSoup
import glob
import re

techname = {} # いずれかのウェブページで登場した技術グループ・技術名を記録する
school = {}
r = r'^[/]+\%.txt$'

# 取得データを格納したテキストファイルを収集
filelist = glob.glob("./*.txt")
# 各テキストファイルの調査
for file in filelist:
    # HTMLソースの読み込み
    print("Progress in %%" + file + "%"...")
    f = open(file,encoding="utf-8")
    src = f.read()
    f.close()
    # 組織名を取得、全組織分のデータを格納する変数の中にデータ枠を作成
    m = re.search(r,file)
    schoolname = m.group().replace(".txt","")
    school[schoolname] = []
    # HTMLソースから技術情報を格納する要素を抽出
    soup = BeautifulSoup(src,"html.parser")
    data = soup.find(id="technology_table_section")
    if data != None:
        data2 = data.find_all(name="li")
        for i in range(len(data2)):
            techtype = data2[i].find(name="h3").text
            # 抽出した技術グループが初出であれば記録
            if techtype not in techname:
                techname[techtype] = []
            data3 = data2[i].find(class_="technology_table")
            data4 = data3.find_all(name="span")
            # Technology,Description,Exampleの順に並んだ表記のTechnology部分を抽出
            for j in range(0,len(data4),3):
                # 抽出した技術名が技術グループの中で初出であれば追加記録
                if data4[j].text not in techname[techtype]:
                    techname[techtype].append(data4[j].text)
                # 調査対象のウェブページで使用されていた技術名として追加記録
                school[schoolname].append(data4[j].text)

# 出力データのヘッダを作成
out = "schoolname"
for tech in techname.keys():
    for i in techname[tech]:
        out = out + "," + tech + ":" + i
out = out + "\n"
# 出力データの中身を作成
for sch in school.keys():
    out = out + sch
    for tech in techname.keys():
        for i in techname[tech]:
            # 出力対象の組織のウェブページに、技術名があれば○、なければ×を出力
            if i in school[sch]:
                out = out + ",○"
            else:
                out = out + ",×"
    out = out + "\n"
# 結果をCSVファイルに出力
f2 = open("output.csv",mode="w",encoding="utf-8_sig")
f2.write(out)
f2.close()
print("done")
```

