

# インターネットにおけるデバイス環境の多様化，サーバ環境の大規模化，プラットフォームのエコシステム化への対応

森 正弥<sup>†a)</sup>

Internet Technologies with Diverse User Devices, Large-Scale Servers, and Ecosystems on Service Platforms

Masaya MORI<sup>†a)</sup>

あらまし インターネットはその発祥からこれまでの発展を通して，着実に我々の日々の生活や社会の隅々に浸透し，多様かつ巨大，複雑かつ高度に組織化した存在へと変わりつつある．本論文では，このようなインターネットにおける変化の中，今後，どのような技術的対応がビジネスとして必要となるのかについて考察する．特に，多様化するデバイス環境を支える基盤技術，情報爆発に対応する分散処理技術，プラットフォームでのエコシステムの開発を中心に，今一度重要性が増してくるであろう代表的な研究事例にも言及しつつ，考察する．

キーワード モバイルエージェント，レコメンデーション，大規模分散，分散並列コンピューティング，プラットフォーム戦略

## 1. ま え が き

インターネットはその発祥からこれまでの発展を通して，着実に我々の日々の生活や社会の隅々に浸透し，近年，多様かつ巨大，複雑かつ高度に組織化した存在へと変わりつつある．

従来の PC や携帯電話でのインターネット利用だけでなく，iPhone に代表されるスマートホンや，各種インターネット接続デバイスの登場でデバイス環境は非常に多様化している．

一方，サーバ環境に視点を移してみると，情報爆発に代表されるトランザクション，データ量の増大に伴い，大規模処理を実現するための技術基盤が求められており，それがクラウドコンピューティング [1] 等の新しいインフラ環境の整備につながっている [2]．これによりサーバ側のインフラも大規模化している．

デバイス環境，サーバ環境問わず，インターネットにおける技術環境は大幅に変わってきている．インターネット上でサービスを展開し，ビジネスを行う企

業においては，技術環境の変化に合わせて，新しいニーズ・課題が生まれており，それらへの技術的対応が必要になってきている．

このビジネス的観点からの技術的対応において重要なのは，サービスの利用者であるユーザの視点から見たニーズと課題であり，サービス提供者の技術的対応によって，ユーザの求めているものをいかに実現するか，あるいは，ユーザの求めていること，やりたいことをいかに止めないか，ということである．

以降では，2. においてインターネットにおける技術環境の全体像を示し，3. においてデバイス側の多様化とそれに対応するための技術，4. においてサーバ側の大規模化とそれに対応するための技術，5. においてプラットフォームのエコシステム化とそれに対応するための技術について示した後，6. において本論文のまとめを行う．

## 2. インターネットにおける技術環境の全体像

ビジネス的観点でのインターネットを取り巻く技術環境は，図 1 に示すように，以下の三つに分けて考えることができる．

(1) クライアント側のデバイス環境

<sup>†</sup> 楽天株式会社 楽天技術研究所，東京都  
Rakuten Institute of Technology, Rakuten Inc., 4-12-3  
Higashishinagawa, Shinagawa-ku, Tokyo, 140-0002 Japan  
a) E-mail: masaya.mori@mail.rakuten.co.jp

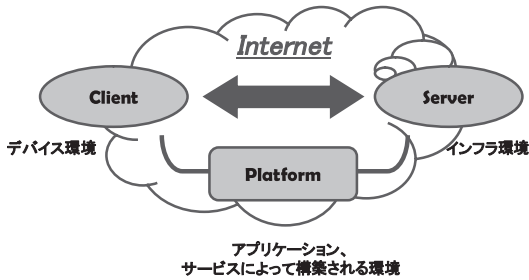


図 1 インターネットを取り巻く技術環境

Fig. 1 Technology Environments around the Internet.

(2) サーバ側のインフラ環境

(3) 多数のクライアントとサーバの参加によって構築されたアプリケーション, サービスのプラットフォーム環境

インターネット上でのサービスはサービス授受の最小構成単位に着目すると, クライアント・サーバ型モデルであり (1) と (2) の技術環境を想定することができる。しかし,  $N$  対 1 のモデルである従来のクライアント・サーバ型モデルと比べ, インターネット上のクライアント・サーバ型モデルは,  $N$  対  $N$  という点において異なる。これにより, 多数のクライアントとサーバが参加し, まるで生態系のような, 参加者が集まってみんなでコンテンツやサービスを作り上げていくプラットフォームとしての性質をもっている。それが (3) に該当する。近年ではこのプラットフォームが集合的サービス開発を行う環境としてインターネット上に存在することが普及しつつある [3]。

### 3. クライアント側の多様化

従来の PC や携帯電話だけでなく, iPhone に代表されるようなスマートホンや, 各種インターネット接続デバイスの登場でクライアント側のデバイス環境が多様化している。

従来, インターネットの利用は PC での利用が一般的だった。それに加えて, 一部の先進的なユーザがモバイル端末や携帯端末でのインターネット利用を行っていた。しかし, もう状況は変わってきている。

総務省の H20 年度通信利用動向調査 [4] によると, 平成 14 年末に, インターネットを利用している人のうち, PC を用いている人の割合は 82.4% であり, 携帯電話のようなモバイルデバイス (PDA 等含む) によって利用している人の割合は, 40.2% であった。し

かし, 平成 20 年末では, PC での利用は 90.8% と伸びているものの, モバイルの伸びの方が大きく, 利用は 82.6% となっている。PC とモバイルの利用割合が並ぶのも時間の問題ともいえる。

インターネット接続におけるモバイル利用の割合も進展しているが, 昨今の変化はもはやそれだけに限らなくなってきている。今までにインターネットに接続してこなかった様々な機器がインターネットへの接続機能を有するようになってきている。

#### 3.1 インターネット接続デバイスの多様化

従来, インターネットに接続してこなかった機器で, 今やインターネット接続が当たり前になっているものの代表例はデジタルテレビである。画像もデジタルデータであり, インターネットについでブラウザでコンテンツを見る機能が欠かせない。日本の主要メーカーの薄型テレビのブランドのほとんどがインターネットへの接続機能を有する。

ほかにいろいろな電子機器を見てみても, 様々な機器がインターネットに接続可能なことに気が付く。例えばデジタルカメラやビデオカメラ, ビデオレコーダー, Blue-ray プレイヤー, 任天堂の Wii やソニー・コンピュータエンタテインメントの PlayStation 3 のようなゲーム機等。中には, ここ数年で一般家庭でも普及してきたデジタルフォトフレームのような機器においてもインターネット接続の製品が登場してきている。更には, Amazon の Kindle のような電子ブックリーダー, Apple の iPad のような新しいインターネット接続デバイスも登場し, かつて M. Weiser が唱えた多様なデバイス形態 [5] が実現されつつある。今後, このようなインターネット接続可能なデバイスの多様化はますます加速していくと考えられる。

#### 3.2 デバイスの多様化に伴う開発の困難さ

デバイスの多様化に対しては, もちろんサービス提供者としては, サービスを個々のデバイス技術に対応させていく必要がある。

例えば, ドコモの i-mode の携帯端末は昨年から Web ブラウザの機能を大幅に拡張している [6]。その中で, Ajax や Flash, 動画の機能を拡充させており, 従来のモバイルアプリケーションでは実装できなかった UI を実現させている。今までにはできなかったよりリッチかつダイレクトなサービスが実現できるようになっている。アプリが非常に人気があり, 普及している iPhone においては, マルチタッチスクリーンや三軸加速度センサを有し, キー操作では実現しなかつ

たより直感的に操作できるアプリケーションが可能になっている。Google が提唱しているモバイルプラットフォームの Android は、i-mode や iPhone とは異なり、従来のコンピュータ型アプリケーションの開発にはなるが、それでもタッチスクリーンや各種センサへの対応があり、通常アプリケーションの開発技術とは異なる個々の知識が必要となる。更にはインターネット接続のゲーム機やデジタルテレビも併せて考えていくと、個別に技術環境に対応していくのは大変でもあり、非効率である。従来も、モバイル機器の機種ごとの搭載 Web ブラウザや表示仕様等の差異に合わせてサービスのコンテンツを自動変換していく仕組みが考えられ、主にモバイルコンテンツを提供する企業にて実装されてきた [7]。そのようなデバイスごとの HTML 文書のフォーマットやアプリ形式を変換していく技術においても今後新しい技術が必要であるだろう。しかし、昨今の多様化が進むデバイスごとの違いは HTML 文書のフォーマットやアプリ形式の違いだけでなく、プラットフォームの違いにも及び、そのような変換していく技術では差異を吸収できないところもある。そのため、個別のデバイスに対応したサービスの開発生産性を高めていく仕組みや基盤等も必要となっていくと考えられる。

### 3.3 多様なデバイス間のシームレスな利用

一方、もう一つ重要なことがある。それは、PC やモバイル、あるいは、他の様々なデバイス機器の垣根を越え、多様化しているあらゆるデバイスを活用しながらも、ユーザがインターネット上のサービスをシームレスにどこにおいても、継続して利用し続けることができるか、ということである。

デバイスはどんどん多様化しており、PC やモバイルだけでなく、デジタルテレビ等のデバイスも含めれば、一人のユーザが保有する端末数はどんどん増えてきている。今後は、多くのサービスがこのようなユーザのその時点での利用デバイスによらない、常に一貫したデータ、サービスを提供し、シームレスな利便性を実現していくのが求められるだろう。

このようなユーザが自由にデバイスを切り換えて用いるという利用状況の進展を考えると、デバイス間でのタスクの移動が必須になってくると思われる。例えば、ネットワーク上のコンピュータを移動しながら処理を進めるプログラムである、モバイルエージェント [8], [9] や、その一つであるモバイル計算機環境下でのプロセスマイグレーション、セッションマイグレーション等の技術は、

今現在のデバイス環境の多様化の状況下では、今一度重要性が増してくるだろう。

またもう一つは、従来その時点でのユーザの状況を示す閲覧履歴や行動履歴等の履歴情報や位置情報、すなわち、コンテキストはクライアント側で保持されていたが、今後はクライアント側では保持せずに、サーバ側でコンテキストを保持、把握しつつ、それに基づいてデータやサービスを提供するようなシステムのあり方に切り換わってくるだろう。そもそも、多様化している様々なデバイスは、ユーザのデータを十分に格納できない。そしてユーザのそのときの都合により利用されるデバイスは様々切り換えられていく。こういう状況でユーザへ価値を発揮するには、サーバ側でユーザのコンテキストを適切に保持し、継続して一貫したサービスを提供していくことが重要になる。従来は PC が利用の中心であったため、Cookie 等の技術を利用し、ユーザのコンテキストはクライアント側の PC 内に保持していたが、今後は多様なデバイスにおいて共通して利用できるコンテキスト基盤を開発し、保持していくことが重要だろう。従来からもサーバ側におけるセッション管理の技術等を応用することで多様なデバイスにおいて共通して利用できる仕組みの提供はあったが、多数のユーザが利用するシステムにおいてはサーバ側の負荷により限定的な利用にとどまっていた [10]。なぜなら、すべてのユーザに対応したそれぞれのコンテキストを、サーバ側で保持し、高速に入出力するには大規模なトランザクション処理を可能にする技術基盤が必要であるからである（例えば、Yahoo!Japan は月間のアクティブなユーザ数で 2400 万を超え、1 日平均 PV が約 15 億あり、ミクシィでは月間のアクティブなユーザ数が 1000 万を超え、1 日平均 PV が約 10 億ある [11] ~ [13]。どのページを過去に見たかという Web ページの閲覧履歴等に代表されるコンテキスト情報を、これほどのアクセス規模において保持するにはサーバ側の負荷が大きい）。これは、以降で述べるサーバ側の処理の大規模化につながる大きな原因の一つともいえる。

## 4. サーバ側の大規模化

デバイス環境の多様化に対応したサービスやビジネスのために、アプリケーションの開発効率をいかに高めることや、デバイス間でのタスクの移動方法、コンテキストを保持しデバイスに依存しないシームレスなサービス提供が重要とされている。特に最後のテーマ

においては, コンテキスト基盤を構築し, バックエンドに配置していく必要があるということになるが, 単にコンテキストのデータストアを用意すればよいという話にとどまらず, サーバ側での変化に対応した技術が必要となってくる.

#### 4.1 情報爆発

UC パークレイ校による調査によれば, 人類の創成する情報量は爆発的に増大している. いわゆる「情報爆発」である [14], [15].

では, なぜ情報が爆発しているか. 基本的には, ビジネスや身の回りの様々な技術のコンピュータ化, コンテンツのデジタル化によって, 情報量が増大しているわけだが, やはりここ十年のスパンで見たとき, インターネットの発展によるところが大きい. インターネット上で, 毎日, 大量に生成されるニュース, ユーザが作成する掲示板, ブログ, Twitter 等での書込みにより日々の世界の出来事が常に追加されていくようになっている. 更には, 前述のデバイス環境の多様化により, そのデータの追加に拍車がかかっている.

人類のデジタル化された情報量はこれからもどんどん増えていく, しかも, その増加の割合は幾何級数的に, 急激に, 上昇している. その増え方は, 今までのシステムやインフラの増強等では対応できないレベルである. 情報の増加ではなく, まさに爆発である.

#### 4.2 ビジネスにおけるデータの活用

インターネットビジネスの裏側においてもデータが爆発している. インターネット上には, 多様なサービスがある. そして, それぞれのサービスの裏側には, 蓄積されたビジネスデータベースがある. 来るべき情報爆発時代に向け, このようなデータを集約させ, 分析する手法を構築することはインターネットサービス企業においては重要なことである. 多くの企業は, ビジネス上のデータを集約しデータウェアハウスを構築している. 特に, ユーザの属性や行動情報をデータベースに集約し分析することで, ユーザの利便性の向上施策, 事業施策へと活用することを行っている. それにより, 例えば, ユーザー一人ひとりにマッチしたコンテンツやサービスの提供が可能となっている.

そこで多くの企業では, レコメンデーションエンジン, パーソナライズエンジンを開発し, これらの保有データを十分に生かす試みを行っている. レコメンデーションエンジンによって顧客のニーズを推測して商品やサービスを提案し, パーソナライズエンジンによって顧客ごとに応じた情報提供やサービスの提供を

カスタマイズしている. レコメンデーションにおいては, Goldberg ら [16] や Resniek ら [17] の研究から協調フィルタリングの手法が開拓され, その適用が長く研究されてきた. 現在は, 協調フィルタリングに限らず, 様々なアルゴリズムのレコメンデーションエンジンやパーソナライズエンジンが作られている [18].

しかし, ここにおいてもデータの爆発がある. 今現在共通して直面している課題に, 分析対象であるデータ規模の増加がある. 例えば楽天では, 商品は 5600 万種類, ユーザ数は 7500 万 (アクティブユーザ数は 3800 万) を超す [19], [20]. この商品数, ユーザ数において閲覧履歴, 購入履歴のデータを保持し, レコメンデーション, パーソナライズに活用する場合, 日次で億を超えるデータ件数を処理する必要がある. 更には, 楽天のレコメンド研究で対象としているデータ量もここ 1 年で約 4 倍に膨れ上がっている [18]. そのため, どうデータを解析し, 関係を見出すのかというアルゴリズム的な工夫だけでなく, そもそも大量のデータが幾何級数的に増大化していく中で今後も, より効率的に処理するにはどうしたらいいかという, コンピューティングパワーを高める研究も必要である.

これら情報の爆発に呼応し, データ処理の重要性はどの企業においても高まっている. 大量データを処理する基盤技術をもつことが, 企業の生命線を分ける.

情報爆発に対応するには, データ処理の大規模化・高速化の実現が必須である. しかも, 情報爆発は従来のスケーラビリティとは違うスケールになってきている. 単体ハードウェアの幾何級数的な性能向上はあまり期待できない. 情報爆発に対しては従来とは違う試みが必要である. ここにおいては, 今までのアーキテクチャ的発想ではない, 違う発想が必要であり, そのため, 後述する failure oblivious computing や eventually consistency という概念を導入した技術等が採用されている.

#### 4.3 処理の大規模化, 高速化の実現

処理の大規模化, 高速化の実現にあたっては, Rinard らが提唱した failure oblivious computing [21] のような概念は, インターネットサービス企業が特に重要視し, 活用している. failure oblivious computing では, 従来のエラーの厳格な制御を行う仕組みを積極的に回避し, エラーがあっても処理は中断することなく, そのまま継続することを原則とする. こうして, エラーを忘却することによって, 処理の大規模化・高速化を優先できる. インターネット上の大量のデータ

の中には、開発者の予想しないような無効なデータが混ざることが多く、それらに対応するためには、failure oblivious computing のような処理方式が適している。そのため、Google にて開発された大規模分散処理技術である GFS や MapReduce を基盤とするプログラミング言語 Sawzall 等においては、この failure oblivious computing の概念が反映されており、それにより効率的な処理を実現している [22]。

ほかにも、Amazon の Dynamo [23] や、ミクシィの TokyoTyrant、GREE の Flare、Microsoft の Azure、楽天の ROMA 等の分散 key-value ストアで採用されている、eventually consistency という概念も重要である。分散 key-value ストアは、オンメモリで高速に処理する簡易データストアであるが、トランザクション処理の不整合、それによるデータの不整合をある程度許容しながらも処理の高速化を実現する。例えば、一つのキーに対応するデータが（本来は一つであるべきなのにもかかわらず）2 個あったら、そのまま 2 個とも返してしまうような処理を許容する考え方をする。この例の場合は、それを受け取ったアプリケーション側が 2 個のデータの中身を評価し、どちらかを破棄して結果的に一貫性を担保することになる。まさに「結果一貫性 (eventually consistency)」というわけであるが、これにより、データ管理の大規模化・高速化を優先する。

情報爆発は保持されるデータの爆発であるが、それは、ユーザ数やトランザクション量の爆発によってもたらされている。そのため、サーバ側のアプリケーションや前述のコンテキスト基盤、インフラはこのような failure oblivious computing や eventually consistency のような考えを導入し、大規模なトランザクションに耐え得る技術を備えておく必要がある。

#### 4.4 分散並列コンピューティング

以上のように、サーバ側においては、情報爆発に対応した大規模化・高速化を実現することが必須であるが、それには、分散並列コンピューティングの技術も重要である。

大規模化する際、データ格納の手段や、処理の部分をいかに分散させていくか、複数のコンピュータやプロセッサ、ストレージを、ネットワークを通じて結び付けて並列で処理させていくか、ということが大事になる。プロセッシングやデータの I/O 処理をネットワーク経由で複数のコンピュータに分散させ、処理の並列度を高めることが必須になる。そこで分散並列コン

ピューティングの技術を用い、いかに複数のインフラリソースに分散させ、大規模化していくか。それが重要になってくる。

分散並列コンピューティングにおいては、実行すべき処理の工程の中で、どこを並列に扱えるか、どこは直列に扱わざるを得ないか、それらを全体としてどう制御しなければいけないかの、処理モデル上の設計がまず必要であり、またそれを実際にコントロールする管理フレームワークやシステムが必要になる。

そのようなことを踏まえて、サーバ側のミドルウェアやインフラは、大量のサーバ機器の上で稼働させることが可能な技術を有している必要がある。一部は、サーバの仮想化によって実現されるが、一部にはミドルウェアそのものが大量のノードの上でも稼働できる、技術を有している必要がある。一方、データベースやファイルシステムにおいてもスケラビリティを有するものが必要となり、大量のデータを格納することが必要となってくる。

もちろん、耐障害性も重要である。膨大なサーバ機器を運用すると、日々多数の機器が故障する。それゆえ、故障することを前提とした基盤システムを開発する必要があり、独自ハードウェアを開発する等、対応を行っている [24]。

今後、このような多数のコンピュータリソースを活用し、分散並列コンピューティングを実現していく状況の進展を考えると、例えば、グリッドコンピューティングを支援する基盤システム [25] ~ [29] 等は、再び重要性が増してくるだろう。これらの技術が基盤となってサーバ側の大規模化を促しており、各社のクラウドコンピューティングと呼ばれるサービスの提供へとつながっている。

### 5. プラットホームのエコシステム化

多様なデバイスへの対応、情報爆発による大量のトランザクションやデータへの対応、それに続いて、対応が必要となってきている、もう一つの変化がある。それは、インターネットにおいてユーザに利用される、サービスの提供のあり方そのものの変化である。

ユーザの嗜好は技術環境の変化に呼応して多様化している。画一的なサービスではなく、ユーザごとの嗜好に応じたサービスセットが必要となってくる。ユーザの多様化・細分化するニーズに対応するには、単一のサービス提供者が企画・開発し、提供することには限界があり、多くの参加者によってより多くのサービ

スを開発, 提供して行く必要がある. 一つの手法としては, 複数のサービス提供者の参加を可能にするプラットフォームをまず提供する手法がある. これは, プラットフォーム戦略と呼ばれ, プラットフォームを補完するようなサービスの増加により, プラットフォームそのものの価値を増大させていく手法である. この手法による効果は, 出口ら [30], [31] によってプラットフォーム外部性として提唱されている. また, 類似の概念を, Garwer ら [32] が提唱している.

現在更に複数のサービス提供者によってサービス開発が行われるだけでなく, サービス開発そのものにユーザがかかわるか, あるいはユーザ自身が求めるサービスを開発してしまうという形態が出現している.

Alvin Toffler が著書 *The Third Wave* [33] で, 生産者 (producer) と消費者 (consumer) のギャップは技術によって埋められ, prosumer (生産にかかわる消費者, トフラーの造語) という自身のニーズを満たすことができる存在が登場すると主張した. まさにこの prosumer が自らサービス開発に参画し, そして開発したサービスを利用していくためのプラットフォームを系統的に実現していくことになる.

### 5.1 PaaS

PaaS (Platform As a Service) という言葉がある. SaaS (Software As A Service) と同様に, Salesforce 社が提唱した概念で, サービスのあるサーバ側でユーザがプログラムを開発することで, 既存のサービスを拡張したり, あるいは, 自分が使用したいアプリケーションそのものを開発し動かすことができる形態のことを指す.

SaaS がインターネットを介してソフトウェアをサービスとして提供するモデルであるのに対して, PaaS はそこから更に一步踏み込んで, インターネットを介して開発環境から提供するモデルであるといえる.

PaaS は, 今のところ, 独自の開発言語の提供や, 開発者向けの環境であるミドルウェアの提供がメインであるが, これにより, 開発者やアプリを囲い込み, 独自のエコシステム, 経済圏を作ることを目的としている. これらはインターネット上のサービスによく見られる集合知の考え方を組み合わせた開発手法ともいえる.

PaaS の代表格は, やはり Salesforce である. Salesforce では, Force.com というブランド名で PaaS 型サービスを展開しており, Java をベースとした Apex というの専用のプログラミング言語を提供している.

開発者はこの上でアプリを構築するわけだが, 既に Salesforce では Apex で作られた 800 以上のアプリが稼動している.

ほかの PaaS の実現例としては, Google の GoogleAppEngine が挙げられる [34] ~ [36]. プログラミング言語としては, Python ベースと Java ベースの 2 種類の環境を提供しており, 基本は無料のサービスで, 料金を払えば CPU, ストレージ, 電子メール送信数など柔軟に増やせる. 高度にスケーラブルな自動運用を実現している.

また, 米 SNS 最大手の企業である Facebook でも, FBML, FQL という独自のプログラミング言語を提供し, PaaS 環境を実現している. Facebook 上では, 5 万を超えるアプリケーションが開発されており, 一番人気のあるアプリケーションは, 4500 万以上のユーザに活用されているという「世界一売れたゲーム」として, ギネスブックに登録されている「スーパーマリオブラザーズ」の販売数が全世界で, 約 4000 万本であり, それをも上回っている. このアプリケーション利用の盛況振りを受けて, 日本の SNS 最大手である Mixi も, Mixi アプリという Facebook と同種の PaaS 型のアプリケーション・プラットフォームを提供している.

### 5.2 多様なニーズに対応するための拡張性

このように企業各社は開発環境や拡張可能な環境を提供することで, ユーザがほしいサービスを自ら作ったり, あるいは, サービスを拡張していくプレーヤーが現れることを促している.

それにより, ユーザを多く集め, 一種のエコシステムと呼べる環境を作り上げていくということがビジネスで非常に重要になってくる. エコシステムとは, 本来の意味である生物界の生態系を指しているわけではもちろんなく, ビジネスにおける相互依存や価値連鎖のあり方を指した表現である. プラットフォーム上に関連するアプリケーションが構築され, またそれらによってプラットフォームの価値がますます高まっていくような現象を指す. 今後は, 開発環境や拡張可能な環境がプラットフォームとして提供されることにより, ユーザがほしいものをユーザが開発, 拡張し, それにより更にプラットフォームの価値が高められていくことが当たり前になるだろう [37].

このような動向にどう対応していくか. 基本としては, PaaS のようにプラットフォーム戦略を系統的に実現していくことが必要である. その際には, 各シ

システムにおいては、サービスそのものを拡張していく仕組みを有する必要がある。ある程度の機能以上は、拡張可能な仕組みによって、後から追加開発できるようにしていくことで、対応できるようにする。これにより、多様なユーザーニーズ、あるいは変化するビジネスニーズに柔軟に対応できるようになる。

例えば、Java モジュールの動的追加や実行を管理するための基盤システムである OSGi [38], [39] のように、運用中に変更・追加が可能であるような拡張ポイントを備えていくことがソフトウェアやミドルウェアにとって基本的要件となっていくだろう。

更には、そもそもの振舞いを変更できるような仕組みである、dynamic configuration management [40] や、それにおける、reflective middleware [41] の技術も再度重要性を増してくるだろうと考えられる。

## 6. む す び

インターネットを取り巻く技術環境は日々変化を遂げており、それに呼応する形でユーザの嗜好、ビジネスのニーズも多様化・細分化している。本論文では、このような変化の中、今後、どのような技術的対応がサービスやビジネスとして必要となるのかについて考察した。

特に、多様化するデバイス環境を支える基盤技術、大規模な情報爆発に対応する分散処理技術、プラットフォームでのエコシステムの開発を中心に、代表的研究事例にも言及しつつ、考察した。

モバイルエージェントやコンテキスト基盤、大規模分散処理技術、拡張性の実装やプラットフォーム戦略の実現が、技術的対応の代表的なものとなってくるが、このような技術の開発を通し、今後ますます拡大するインターネットを豊かにし、支えていく技術基盤を提供していくべきではないかと考えている。

謝辞 本論文において御教授頂きました東京大学、増原英彦准教授に深く感謝致します。また、楽天技術研究所の平野廣美氏、平手勇宇氏、西澤無我氏、益子宗氏、梅田卓志氏には本論文の議論にお付き合い頂きました。ここに感謝の意を表します。

## 文 献

- [1] Google Inc., "Conversation with Eric Schmidt hosted by Danny Sullivan," Search Engine Strategies Conference, <http://www.google.com/press/podium/ses2006.html>
- [2] 丸山不二夫, "クラウドの成立過程とその技術的特徴について," 情報処理, vol.50, no.11, pp.1055-1061, 2009.
- [3] D. Tapscott and A.D. Williams, "Wikinomics: How Mass Collaboration Changes Everything," Portfolio, 2006.
- [4] 総務省, "H20 年通信利用動向調査報告書(世帯編)," <http://www.soumu.go.jp/johotsusintokei/statistics/pdf/HR200800.001.pdf>, p.53, 2008.
- [5] M. Weiser, "The computer for the 21st century," in Scientific American, pp.94-104, 1991.
- [6] NTT DOCOMO, Inc., "i モードブラウザ 2.0 新機能一覧," [http://www.nttdocomo.co.jp/service/imode/make/content/browser/browser2/new\\_function/](http://www.nttdocomo.co.jp/service/imode/make/content/browser/browser2/new_function/)
- [7] Sun Developer NEWS, "携帯コンテンツビジネスの交差点," <http://jp.sun.com/developers/news/200104/danwa01.html>
- [8] 佐藤一郎, "モバイルエージェント," コンピュータソフトウェア, vol.17, no.2, pp.45-54, 2000.
- [9] 佐藤一郎, "モバイルエージェントの動向," 人工知能学会論文誌, vol.14, no.2, pp.598-605, 1999.
- [10] 阿蘇和人, "「セッション管理」のすべて," 日経 NET-WORK, pp.43-59, May 2008.
- [11] Yahoo!Japan, IR 関連情報, <http://ir.yahoo.co.jp/jp/archives/monthly/>
- [12] ニールセン・オンライン, インターネット利用動向調査「NetView」, 2010 年 3 月データ.
- [13] 長野雅広, "大規模システムの運用," SoftwareDesign, pp.45-51, 技術評論社, 2010.
- [14] 情報爆発プロジェクト, "情報爆発時代に向けた新しい IT 基盤技術の研究," <http://www.infoplosion.nii.ac.jp/infoplosion/>
- [15] EMC, "Digital universe," <http://www.emc.com/leadership/digital-universe/expanding-digital-universe.htm>
- [16] D. Goldberg, D. Nichols, B.M. Oki, and D. Terry, "Using collaborative filtering to weave an information tapestry," Commun. ACM, vol.35, no.12, pp.61-70, 1992.
- [17] P. Resnick, N. Lacovou, M. Suchack, P. Bergstrom, and J. Riedl, "GroupLens: An open architecture for Collaborative filtering of netnews," Proc. 1994 ACM Conference on Computer Supported Cooperative Work, pp.175-186, 1994.
- [18] 森 正弥, "第 9 章 大規模分散処理基盤の開発," クラウド大全 サービス詳細から基盤技術まで, 日経 BP 社出版局(編), pp.283-286, 日経 BP 社, 2009.
- [19] 楽天市場, <http://www.rakuten.co.jp/>
- [20] 楽天(株), 投資家向け情報, <http://corp.rakuten.co.jp/ir/>
- [21] M. Rinard, C. Cadar, D. Dumitran, D.M. Roy, T. Len, and W.S. Beebe, Jr., "Enhancing server availability and security through failure-oblivious computing," <http://people.csail.mit.edu/rinard/paper/osdi04.pdf>
- [22] R. Pike, S. Dorward, R. Grisemer, and S. Quinlan, "Interpreting the data: Parallel analysis with swazall," <http://static.googleusercontent.com/>

external\_content/untrusted\_dlcp/labs.google.com/ja/papers/sawzall-sciprogram.pdf

- [23] G. DeCandia, D. Hastorun, M. Jampani, G. Kakulapati, A. Lakshman, A. Pilchin, S. Sivasubramanian, P. Vosshall, and W. Vogels, "Dynamo: Amazon's highly available key-value store," SIGOPS Oper. Syst. Rev., vol.41, no.6, pp.205-220, ACM, New York, NY, USA, 2007.
- [24] S. Shankland, "Google uncloaks once-secret server," cnet news, [http://news.cnet.com/8301-1001\\_3-10209580-92.html](http://news.cnet.com/8301-1001_3-10209580-92.html), 2009.
- [25] Net Solve, <http://www.cs.utk.edu/~casanova/NetSolve/>
- [26] Globus, <http://www.globus.org/>
- [27] Legion, <http://www.cs.virginia.edu/~legion/>
- [28] RCS, <http://www.inf.ethz.ch/personal/arbenz/rcs.html>
- [29] Ninf, <http://ninf.etl.go.jp/>
- [30] 出口 弘, "ネットワークの利得と産業構造," 経営情報学会誌, vol.2, no.1, pp.41-61, 1993.
- [31] 植田弘和, 奥田 栄, 浜本光紹, 細田衛士, 石川雅紀, 出口 弘, 藤崎成昭, 新しい産業技術と社会システム, 日科技連出版社, 1996.
- [32] A. Gawer and M.A. Cusumano, Platform Leadership: how Intel, Microsoft, and Cisco drive industry innovation, Harvard Business School Press, 2002.
- [33] A. Toffler, The Third Wave, Bantam Books, 1980.
- [34] <http://code.google.com/intl/ja/appengine/>
- [35] <http://code.google.com/p/appscale/>
- [36] N. Chohan, C. Bunch, S. Pang, C. Krintz, N. Mostafa, S. Soman, and R. Wolski, "AppScale: Scalable and open AppEngine application development and deployment," <http://www.cs.ucsb.edu/~nagy/docs/cloudcomp09.pdf>
- [37] 森 正弥, ウェブ大変化 パワーシフトの始まり, pp.109-113, 近代セールス社, 2010.
- [38] <http://www.osgi.org/>
- [39] OSGi Alliance, "OSGi service platform service compendium," Release 4, Aug. 2005.
- [40] G.S. Blair, G. Coulson, A. Andersen, L. Blair, M. Clarke, F. Costa, H. Duran, N. Parlavantzias, and K.B. Saikoski, "A principled approach to supporting adaptation in distributed mobile environments," 5th International Symposium on Software Engineering for Parallel and Distributed Systems (PDSE-2000), Limerick, Ireland, 2000.
- [41] F. Kon, M. Rom'an, P. Liu, J. Mao, T. Yamane, L.C. Magalhães, and R.H. Campbell, "Monitoring, security, and dynamic configuration with the dynamicTAO reflective ORB," Proc. Middleware 2000, International Conference on Distributed Systems Platforms, vol.1795 of LNCS., pp.121-143, Springer-Verlag, New York, USA, 2000.

(平成 22 年 3 月 6 日受付, 5 月 17 日再受付)



森 正弥

1998 慶大・経済卒。2006 楽天(株)入社。現在, 同社執行役員兼楽天技術研究所長として, 研究開発組織のマネジメントに従事。情報処理学会会員。