

コラム第4回：地を這うソフトウェア、空飛ぶソフトウェア

私は、ソフトウェアについて話すときに、よく「地を這うソフトウェア・空飛ぶソフトウェア」という比喻を使う。「地を這う」と同じ意味で「湿った」あるいは「あたたかい」という言葉を使うときもある。この対比をまとめると以下のようになる。

(ソフトウェアの対比)

地を這う ↔ 空飛ぶ

湿った ↔ 乾いた

あたたかい ↔ 冷たい

もちろん、これは相対的な評価であって、ある条件を満たしていれば地を這っていると定義できるわけではない。

「地を這うソフトウェア」というのは、どのようなものだろうか。「地を這う」とは、「自然現象を直接扱う」ということである。自然現象には、人間の行動なども含まれる。たとえば、ロボットを利用したサッカーをする場合、実際のロボットを動作させる場合は「地を這って」いるが、物理シミュレーターを使って仮想空間でロボットを動作させる場合は「空を飛んで」いる。どちらが悪いということではなく、最終的に目標を達成するためにはどちらも必要なのだが、単純に性質が違うのである。性質が違うので、それを得意とする人も違ってくる。

現実社会で例えると、農業、建設、医療、教育など、自然現象を直接扱う産業は地を這っている。金融業、不動産業などは空を飛んでいると言える。運輸産業などはその中間と言えるかもしれない。

ソフトウェア開発では、よく「低レベル」という言葉を使う。これは、否定的な意味ではなく、単純に、「物理層に近い」ということを意味する。デバイスドライバのような低レベルソフトウェアは、Rubyのような高レベル処理系にくらべると、地を這っていると言える。最近、Namazuを開発した高林氏が「Binary 2.0カンファレンス」(*1)を開いていたが、これは、「地を這う」コンピューティングへの回帰という観点から、非常に頼もしい動きだといえる。私は、彼のメッセージを、軽量言語を操ってコンピューターを使いこなしたような気になっていたら、とても大きな問題を見逃すよ、という警鐘だと受け取った。ソフトウェアに関しては、一次データを扱うのが「地を這う」ソフトウェアで、メタデータを扱うのが、空飛ぶソフトウェアだ、と言えるかもしれない。

私が、ソフトウェアのエンジニアに対して、「地を這うコンピューティング」に向かうように促すのは、そこには解決を待つ無限の問題があるからだ。コンピューターにとって扱

しやすい（メタ）データばかりを扱うのではなく、一次情報を取得し、さらに自然に直接働きかけるソフトウェアを作るという仕事は、ソフトウェア産業にとっての農業であり、建設業である。つまりソフトウェア産業のインフラである。ソーシャルブックマークだ、RubyOnRailsでクイック開発だ、AJAXだといって扱いやすいデータばかりをいじりながら「空飛ぶ」ソフトウェアに時間を費やしてばかりいるエンジニアを尻目に、人間がかかえる問題にじっくり取り組み、問題を解決していく骨太の開発者が必要なのだ。AJAXで操作性を良くする前に、まずAJAXで操作する価値のある情報を自然から取り出さなければならぬのは明白だ。

実際に人間が抱える問題

都市、教育、交通、食料、衣服、医療、健康、エネルギーなど、人間は膨大な問題をかかえている。地球上に増え続ける人間が、より長期間生き延びていくには、これらの問題を解決していく必要がある。コンピュータの小型化と低価格化が爆発的に進んだ結果、これらの問題の中でも、ソフトウェア・エンジニアがいますぐにでも関わることができそうな分野が増えてきた。次の10年は、これらの問題に着目した新興企業が、続々と登場することになるだろう。それらの企業はみな、ソフトウェア・エンジニアを必要としているはずだ。

たとえば、コンピュータによって高度に制御された電気自動車を利用して、都市の環境汚染、交通渋滞、エネルギーの問題を解決する、まったく新しい自動車会社の登場が予想される。医師・看護師・技師など病院の専門家の人的資源配置をリアルタイムに最適化し、現在の医療機関における非効率と、人材不足の問題を解決する企業も出てくるだろう。モノを所有せず、共有することを徹底的に支援する、オンライン対応版トランクルーム。少子化の問題を解決するために、地域のシニアや主婦など、余剰労働力の発見を容易にするためのシステムも必要だろう。仮想空間において物理シミュレーションを行い、衣服のコーディネートをオンラインで体験できるようにするための試着システムも登場する。ロボットとLED照明を使って完全自動で計画的に野菜を生産する野菜工場。各家庭にある電気製品の利用状況を把握・調整し、電力需要のピークをずらすシステム。あらゆる電気製品を通信機能をもつ組み立て式のパーツで構成し、再利用性を一気に高めようとするプロジェクト、排泄物を情報処理し、健康管理を助けてくれるトイレ、森林の生態系をリアルタイムに監視できるセンサーネットワークの構築、コンピュータを内蔵して動的に機能が変わる建築部材などなど・・・

これから登場する企業は、業界にかかわらず、ほぼ例外なく企業活動の根本部分にソフトウェアを活用することになるだろう。そして、より深く自然を観察し、一次データを取り出し、うまく操ることが必要になってくる。コンピュータの端末を離れ、自分で歩き回り、観察をつづける必要がある。「空飛ぶ」ソフトウェアの開発は、確かに覚えやすく、短期的に結果が出るし、見た目にもわかりやすいし、何より、端末に張り付いていても仕事ができる。そのためソフトウェア開発への「とっかかり」としては、とてもうまく機能する。しかし、その段階にとどまっていたら、本当に重要な問題は解決できない。現在「空飛ぶ」ソフトウェアの開発に従事しているエンジニアは、自然現象を扱う仕事を、む

ずかしくて面倒な仕事だと考えてしまう傾向がある。特に、自然現象を扱うのは、物理学者や数学者の仕事だと考えがちである。もちろん、彼らの助言は、かならず必要になるだろう。しかし、彼らはソフトウェア・エンジニアではない。彼らの助言を活かしてソフトウェアを作るという仕事は、誰かがやらなければならない。本物のソフトウェアを作るという仕事は、地を這うソフトウェア開発の中に眠っているのである。

*1 Binary 2.0 <http://namazu.org/~satoru/blog/archives/000078.html>