

Lyee Internet Information

対談

「リクワイヤメント・エンジニアリングについて」

フランス パリ第一大学 コレット・ローランド教授

セルミン・ヌルカン準教授

ソフトウェア生産技術研究所株式会社 根来 文生社長



ヌルカン準教授

ローランド教授

根来社長

日時 : 2001年5月25日(金)

場所 : ソフトウェア生産技術研究所株式会社 本社

聞き手 : 藤田ハミド教授(岩手県立大学)

【コレット・ローランド(Colette Rolland)教授 略歴】

- 1943年 フランスのタルン・エ・ガロンヌ県ディユパンタル生まれ。
現在58歳。
- 1966年 ナンシー大学応用数学科で博士号を取得。
- 1973年 ナンシー大学コンピュータ・サイエンス学科の教授に就任。
- 1979年 パリ第一大学コンピュータ・サイエンス学科の教授に就任。
現在に至る。

<主な研究分野>

- ・情報システム分析とデザイン・リクワイヤメント・エンジニアリング
- ・データベース・デザイン、ケースツールとケース環境、メタ・ケース
- ・メソッド・エンジニアリングと CAME 環境

<主な著作>

- ・「From Conceptual Modeling to Requirements Engineering」
(リクワイヤメント・エンジニアリングへの概念モデリング)
- ・「Comparative Studies of Engineering Approaches
for Software Engineering」
(ソフトウェア工学の為のエンジニアリング・アプローチの比較研究)
- ・「Information Systems and Web Information Systems
: A Methodological Perspective」
(情報システムと Web 情報システム：方法論展望)
- ・「Requirements Engineering for COTS based Systems」
(COTS ベース・システムの為のリクワイヤメント・エンジニアリング)

【セルミン・ヌルカン(Selmin Nurcan)準教授 略歴】

- 1961年 イスタンブール生まれ。現在40歳。
- 1991年 リヨン INSA (国立応用科学研究所) のコンピュータ学部で
博士号を取得。
リヨン大学の準教授に就任。
- 1998年 パリ第一大学ビジネス・アドミニストレーション研究所の
準教授に就任。現在に至る。

<主な研究分野>

- ・共同情報システム
- ・プロセスモデリング
- ・CSCW

<主な著作>

- ・「Analysis and design of co-operative work processes: a framework」
(協力ワーク・プロセスの分析および設計：フレームワーク)
- ・「Main concepts for cooperative work place analysis」
(共同ワーク・プレイス分析の為のメイン・コンセプト)
- ・「Contributions of Workflow to quality requirements」
(クオリティ・リクワイヤメントへのワークフローの貢献)

【 Lyee に対する評価 】

Lyee の優れた点はプログラムの構築および生成を助けるソフトウェアの一般的な構造を発見したということです。(ローランド教授)

- ハミド教授 今回、国際学術共同研究プロジェクトへの参加の可能性を判断して頂く為に、来日して頂いた訳ですが、数日間 Lyee について学んで頂いた率直な感想をお聞かせ下さい。
- ローランド教授 Lyee をもっとよく理解する事を目的に来日した訳ですが、現在はまだ Lyee のほんの一部分しか理解できていないと思います。
今日多摩のソフトウェア工場で見せて頂いたプレゼンテーションは大変有意義でした。と言いますのは、根来さんが考えられた哲学と開発現場の技術者がやっている実際の作業との関連性について具体的理解する事ができたからです。
根来さんがやられていることを理解して話しをしますと、Lyee という方法の優れた点はプログラムの構築および生成を助けるソフトウェアの一般的な構造を発見したということです。
この構造によってどの様なプログラムでも作り出す事ができる。オーガナイズする事が出来るという事を見つけたのだと理解しています。そのプログラムは色々なプログラムの集合でも、いくつかのコードでも構わない訳ですが、それらがソフトウェアの実行を制御する1つの一般的な構造の中に組み込まれています。コードをその構造にプラグ・インします。そしてその構造を繰り返し使うことによって、プログラムが出来上がるということが分かりました。
構造は一般的ではありますが、コードの方は扱っているアプリケーション固有のものです。
- 根来社長 従来法の世界の人達はロジックに対して評価という作業をしなくてはならないのですが、その作業を止めようというのが Lyee の特徴です。ほとんど人間の思考に頼らないという方法を目指しています。
リクワイアメントを決めるのに、可能な限り人間の能力を使わない。少し極端かも知れませんが Lyee はそういう立場をとっています。
Lyee はリエンジニアリングという局面を全て捨て去りまして、本当にフィロソフィカルな概念だけでソフトウェアを捉えています。
そういう意味では、シナリオ関数という構造は、自然言語と同じようにリクワイアメントを定義するための物として見て頂きたいと思いません。

シナリオ関数の構造をそのままコンピューターの中にセットして実行させることが出来るものですから、エンジニアリングされたプログラムだと取り違いをされるかもしれませんが、シナリオ関数はエンジニアリングされたプログラムと考えない方が宜しいかと思います。言い方を変えれば、シナリオ関数はリクワイヤメントを取り扱う言語構造だと思って下さい。全く新しい言語認識構造です。

リエンジニアリングという概念は一時期、ソフトウェアの世界では盛んに議論され、日本でも流行になりました。しかし今では影をひそめています。日本ではリエンジニアリングという概念は、もはや大きな改善策としては認められていません。コンベンショナルな世界でリエンジニアリングという概念が成立しなかったという1つの証明だと思っています。

ローランド教授 根来さんは否定されるかもしれませんが、シナリオ関数は、現在西欧のソフトウェア業界で使われている用語でいうなら、フレームワークとかパターンに相当すると思います。このフレームワークを使えば、いかなるソフトウェアも作成できるのだと私は理解しています。フレームワークというものは、繰り返し使えるもので、シナリオ関数もそのようなものだと思っています。またシナリオ関数はアウトプット・ドリブンのようなものだと思っています。

ハミド教授 Lye の場合はアウトプットが中心となっています。この点は珍しいユニークな特徴だと思っています。その特徴がシナリオ関数の機能となって表現され、実際はフレームワークのパターンになっているのです。

ローランド教授 Lye で面白いと思う点は、ソフトウェアの全体構造があるので、大規模なソフトウェアの効率的な生成が可能であるということです。これは一般的な構造を扱っているアプリケーションに当てはめ、コードの全てをプラグ・インするとプログラムが生成されるというものです。

これは非常に面白いと思います。モジュール性がソフトウェア開発の要だということは分かっていますし、ある意味で Lye はモジュール・プログラムの生成を助けるものです。

根来社長 私は昨日ローランド先生からクルーズ・レクイトワというシステムをご紹介頂き、大変興味を持ちました。どういうところに興味を持ったかと言いますと、従来法の延長線上にあるこの発想と、Lye を上手く組み合わせることが出来る。そういう可能性があると思ったからです。一昨日、先生と茶室へ参りましたが、茶室は概念的に宇宙の全てを

現しています。あの空間に全宇宙があるという発想に立っています。この話をなぜしているかと申しますと、ヨーロッパのカルチャーは物事を分析していくというのが思考の基本になっていると思います。そしてその分割したものに対して概念を与える訳です。ですから例えば難しいとか難しくないとか、小さなシステムだとか大きなシステムだとか、簡単なシステムだとか複雑なシステムだとかというのは、東洋的思想ではなく、ヨーロッパ的概念のように思います。ヨーロッパの概念は物理的社会においては成功を収めています。しかし非物理的なものを対象としているソフトウェアの世界にこの概念がそのまま適用できるかどうかという問題を提起したいのです。このようなヨーロッパ的思想の概念は、全てシナリオ関数の中に取りこまれているのです。このことはベクトルの構造をご覧頂ければご理解頂けると思います。Lyeの構造は茶室のような性質を持っておりまして、非常にシンプルで小さいですが、実は宇宙全部を捉えているのです。従来法のソフトウェアのコンセプトというのは物理的なものでありますので、Lyeと概念上の比較をすることは難しいのです。例えば現在我々が作成した一番大きなシステムは規模が1,400程度（新規および修正の画面数、帳票数、ファイル間処理数の合計）あります。それでもLyeの方法ではシステムが完成するまで、わずか10ヶ月でした。同様のシステムを日本では他に2つの企業グループで開発しましたが、多分費用も期間もLyeと比べ約3倍以上かかっていると思います。

【 リクワイヤメントに関する見解 】

経験の中で私が導き出した結論は、リクワイヤメントというのは本来的に決定出来ないということでした。（根来社長）

ハミド教授

根来社長とローランド先生のおっしゃっていることは少し違うと思います。なぜならリクワイヤメントというものの理解を測定することはできないからです。ユーザーというのはシステムを思い浮かべるときにユーザー自身のリクワイヤメントが非常に大きく、それを上手く表現できないのです。しかもモデルがないので自然言語で表現しきれない訳です。

それは1、2時間話をしたからといって、また1、2ヶ月してシステムという成果となって出てきたからといって、それではリクワイヤメントを表現しきれない訳です。リクワイヤメントからは色々な物が出てきます。ですから、そういう意味の複雑さや大きさがあります。軍のシステムであるとか管制塔のようなもの、あるいは宇宙の管制官、ああいったものはやるべき事が多いので複雑なリクワイヤメントになります。

こういうことから根来社長は先ほどデザインという事から Lyee と茶室を比較していましたが、私からみますとシナリオ関数と言うのは、いわばエンジンに相当すると思います。ですからある意味ではコンパクトで小さなリクワイヤメントを通してエンジン稼働しますし、大きなリクワイヤメントを通してエンジン稼働します。しかしそれとは別にリクワイヤメントそのものが非常に複雑だという事があって、それ自体はやはり問題だと思えます。もちろんリクワイヤメント自体が、1つ1つ公式性のあるもので構成されていて、非常にスムーズに噛み合ったものもあると思えます。

しかし結局シナリオ関数というものはどのように機能していくのかと言った事が、私も重要に思っておりまして、ローランド先生もその点を聞きたいのだと思えます。ともかくリクワイヤメントにある種の複雑さはあると思えます。

ローランド教授

おっしゃる通り、私たちフランスの文化はデカルト以来、問題に対応するには分解して考えるように教えられてきました。ですから私たちは、問題をより小さなところに持っていくために分解する方法を採用することに慣れています。もちろん伝統的にそうしてきた訳ですから、私もこの方法を否定しません。しかし何より私が興味を持っているのは抽象化ということです。抽象化というのはデコンポジション（分解）や分析と同じではありません。抽象化というのは、例えば統合論や分類を用いていく事です。私は職業人としては、抽象化ということに大半の時間を費やしてきた訳で、問題解決の手法としてそれがいかに有効であるかということも認知しており、学生にもそれを教えてきました。ヌルカン先生はデコンポジション（分解）に興味を持っていません。

ヌルカン準教授

私はローランド先生のおっしゃる通り抽象化の理念は尊重しておりますが、私自身はそれよりもデコンポジションの方が向いているかも知れません。同じようなレベルで抽象化するのは得意ではありません。話しは変わりますが、大規模なソフトウェアの問題点は、単に分解し

根来社長

ていくという事ではなく、分解したものをもう一度再構築するという事が困難であるという点です。

例えばビジネス・アプリケーションは大企業が使い、その子会社も使います。その間のコミュニケーションを取りたいとか様々な企業の統廃合がある中で、それらに関係をつけていく必要があります。

こういう状況で、顧客に対して大局的にはどのように対応するかを考えていく場合に、例えば Lye がこれまである現存のシステムをどう活用していくか、アプリケーションのリユースといった現実的な問題にどう取り組むのかについて疑問があります。

対象を分解すると、概念がどんどん膨らみ大きくなるわけです。そこで分解しただけでは意味が無いので合成する必要があります。分解と合成の場合、そこで概念が生じない世界であればこの方法は上手くいきます。しかしソフトウェアの場合、合成が上手くいきません。そういった意味で、ソフトウェアの世界で上手くいく抽象論とは何かを生み出す必要があります。私がずっと研究をしてきたのはそういったテーマです。

私も若いときは大学に籍をおいておまして、ソフトウェアと数学を教えていましたが、当時の日本の大学では私が考えたようなテーマは受け入れられませんでした。そこで私は大学を辞めまして独自に研究を始めた訳です。そういう意味では、私はアカデミックな世界にも属していませんし、正しくは産業界にも属していませんでした。

たった1人でこの問題について考え続けてきました。

多分私がアカデミックな世界で研究を続けていたならば、先生方と同じようにペーパーを書き、同じ見解の上に立っていると思います。

私は研究の途中でいろいろなシステム開発の指導に当たりました。

その経験の中で私が導き出した結論は、リクワイヤメントというのは本来的に決定出来ないということでした。

他方、リクワイヤメントが難しいというのはそれに携わる人達が難しいと言っているだけで、リクワイヤメントそのものが複雑かどうかは客観的に吟味することはできません。こういう考え方に立つと、エンジニアリングをどう使っていくかについては、まだ考えが整理されていません。

【 リクワイヤメントの捉え方 】

実際のプロジェクトにおける失敗の第一の原因は、
リクワイヤメントの理解がまずかったという点にあるのです。
(ローランド教授)

ハミド教授

ある意味で、根来社長のおっしゃることはもっともだと思います。ですが、ある意味で同意しかねます。リクワイヤメントが大きい小さいという言い方をしていましたが、大小のカテゴリーがあるわけではなく、小さくもあれば、大きくもあるわけです。と言いますのは、多くの場合、リクワイヤメントには相互に、様々なレベルで複雑な絡み合いがあるのです。結局どうやってリクワイヤメントを捉えるかという時に、ユーザーと話をして捉えるということがいかに難しいかは明らかな事です。根来社長はそれを全て単語単位でやっていらっしゃるんですが、それでもリクワイヤメントは絡み合っているので捉えることが出来ないわけです。

ローランド先生はそれを自然言語で捉えて表現しようとしています。そして自然言語には曖昧さがあるので、一定のルールを講じる事でそのあいまいさを取り除くという方法を考えられました。

そこからユーザーがいったい何をしたかったのかという事を探るという形をとられています。

多くの場合ゴール同士が複雑に絡んでいます。それがシステムを複雑にします。よくあるのは、ふたつの目標を達成しようとするのですが、その2つの目標が矛盾した関係にあり、両方同時には達成できないという場合があります。あるいはAという目標に対して、関係付けられているようなシステムが色々あって、どこか1箇所変更するとシステム全体が影響を受けてしまうような場合があります。そうしますと、単純に計算式だけでこうした相互関係に対応するのは、容易ではありません。当然、多くの理由づけが必要になります。ですから、リクワイアメントは大きくも小さくもあるのです。

ローランド教授

根来さんは学界に大歓迎の人物です。抽象化を理論の根拠に置く人は歓迎なのです。またソフトウェアのデザイン論を発明するというのは、私たち皆が望んでいることです。しかしソフトウェアはまだ成熟していないので、デザイン論を創設するのは時期尚早だと思います。物理学の場合も、そういった基本的発想が生まれてから実際に様々な

理論が生まれるまで150年くらいの年月がかかっています。ですからソフトウェアの世界でも今世紀の終わりくらいには、理論が打ち建てられるのではないかと思います。

私がソフトウェアの抽象化ということを考えた時に、ソフトウェアのインテンション（意図）を通して行うことを考えました。ですから、根来さん達がバりにいらしてインテンションを中心としてソフトウェアの生成についてご自身の考えを語られた時に興味を持ったのです。私のソフトウェアに対する考え方とフィットしたからです。

しかしながら私は女性ですから、生活上も仕事上も男性よりも具体的な問題に関わりながらやってきました。ですからインテンションを中心とした観点と具体的な実装の観点を関係付けることに大変興味があります。

実際にソフトウェアを開発する時には、私たちの位置付けは、現実的には、顧客が目標を実現するのを手伝うという場合が多いのです。これは多くのソフトウェアが企業の目標達成に貢献しているからです。ですから私たちが開発したデザイン・アプローチでも、ソフトウェアと企業の目標や戦略を関連づけようとしています。こういう意味で、ソフトウェアの抽象的な側面は、そのソフトウェアによって達成される企業の目標や関連した戦略の集合なのです。ソフトウェア・システムを有効で役に立つものしかなかったら、企業の目標達成に貢献しなくてはなりません。

こうした考え方にのっとって、企業の目標からソフトウェアの要件を引き出していきます。システムの機能性はこうした目標を満たす手段とも考えられます。

Lyee では、リクワイヤメントは所与のものと考えています。いずれにしてもある時点で顧客もシステム開発者もリクワイヤメントに合意しなくてはなりません。一連のリクワイヤメントが一つの安定した状態のものだという仮定に立たなくては、システム開発はできません。

しかしながら、Lyee は私達が一番重要視しているリクワイヤメントの引き出しに関して、Lyee の範囲外にしているように思えます。

なぜこの問題がそれほど重要なのかといいますと、経験的に言って、実際のプロジェクトにおける失敗の第一の原因が、リクワイヤメントの理解がまずかったという点にあるからです。

次に今日のコンピュータ・システムはビジネス・プロセスに組み込まれる傾向が強くなっており、プロセスに付加価値をつけられるような情報システムが求められています。ですからシステムがビジネスに

付加価値をもたらすことを示すことはきわめて重要になっています。だからこそ、リクワイヤメントつまりシステムの仕様書と企業の目標を結びつけることが重要になっています。

第三の理由としては、企業の目標とするところは日々刻々と変化しています。目標とリクワイヤメントの間に概念的な関係を維持できなければ、目標の変更がリクワイヤメントに及ぼす影響およびそれに続くシステムの仕様書に及ぼす影響を評価することがきわめて難しくなります。

最後に1つ付け加えたいのですが、最近の例をあげますと、いかにユーザーが欲しいと思っているものが実際に必要なものと違うかという例ですが、フランスのあるソフトハウスで小・中学校などで使う教育ソフトを売っている企業からある依頼を受けました。ご本人たちは自分が何を欲しているか非常にはっきりしていると言い切っていました。クリックして使えるあらゆるソフトウェアを含むスーパー・オペレーティング・システムが求めるソフトウェアだということです。私はその方々と2日間話をして、リクワイヤメントを明らかにしました。その結果、実際に欲しいものはオペレーティング・システムではなく、知識基盤型システムだということがはっきりと分かりました。生徒1人1人の学習状況や成績、先生と生徒の関係、生徒同士の関係、生徒の家族との関係などを追跡できるようなリポジトリを中心として構築するシステムが必要なものだったのです。つまりこの顧客は、スーパー・オペレーティング・システムの最初のアイデアが有効であることを確認するためにプロトタイプを作るのに6ヶ月かけたにも関わらず、結果的には実際に必要なソフトウェアは最初にほしいと思ったソフトウェアとは違うということが明らかになりました。リクワイヤメントはローランド先生に相談するとローランド先生のインテンションで、他の先生に相談すれば他の先生のインテンションで表現されるわけです。だから相談するというのは、大変複雑な話になるのです。どれだけ間違っていようが最初に考えた人のインスピレーションに近いものを受け入れるという形が大切です。Lyee が最も素直にその方の用件を実現できると思います。

根来社長

ヌルカン教授

そうは言いますが、最初にローランド先生があげた統計のように、実際に作ったソフトウェアの50%が使われないと言うことが現実にあるわけで、ユーザーが何をしたいかを表現しきれないことが、大きな原因となっていることは変わりません。作成したものが使われないと言うのは、非常に残念です。これが現実です。

ローランド教授 根来さんのアプローチは、「コードとフィックス」によってユーザーのニーズにソフトウェアを適応させることができる方法として、よく理解できます。RAD（迅速なアプリケーション開発）タイプの開発では、プロトタイプを構築して、ユーザーのリクワイヤメントが満たされるまで、それを見直して作成していきませんが、この方法を支援できる効率のよいメカニズムがあれば問題ないでしょう。ですがお金持ちの日本だから出来るのでは？という気がします。

根来社長 私の考えなのですが、ソフトウェアの開発に成功はなく、全て失敗しているのだと思います。地球が存続する限り失敗は続くのです。ヨーロッパの方で使われずに捨てられたというのは、カルチャーの違いから生じている問題で、リクワイヤメントを捉えたとか、捉えなかったという問題が原因ではないように思われます。

【 Lyee 国際学術共同研究プロジェクトについて 】

クルーズ・レクイトワという私たちのツールを、

Lyee を使ってリエンジニアリングする事を考えています。

（ローランド教授）

ハミド教授 最後になりましたが、国際学術共同研究プロジェクトに関してコメントやビジョンがありましたらお聞きしたいのですが。

ローランド教授 国際学術共同研究プロジェクトについて私が興味を持ったのは Lyee がどのように機能するかという事をより深く理解することで、具体的には現在 Prolog と VB を使用して作成されているクルーズ・レクイトワという私たちのツールを、Lyee を使ってリエンジニアリングする事を考えています。リエンジニアリングは、Lyee の機能がよくわかるという点で私たちにとってプラスになりますし、クルーズ・レクイトワと同様のソフトウェアが手に入るという点では、根来さんの側にとってもプラスになるのではないかと思います。それを使ってデモをしたり、Lyee を適用する前のリクワイヤメント・エンジニアリングの段階で使ったりできるでしょう。

根来社長 素晴らしいお考えです。とても嬉しく思います。

ローランド教授 次に考えているのが、こうしたプロセスを通じて Lyee の開発工程を私たちのやり方で評価してみたいということです。私たちはメソッドモデリングやメタモデリングに関して経験が豊富ですので、Lyee を

フォーマライズするためにこういったエンジニアリングの手法を用いた
たいと思います。

先ほどクルーズ・レクイトワを Lyee でリエンジニアリングすると申し
上げましたが、これを通して Lyee 全体のプロセスに対する理解が得ら
れ、指針を出すことで Lyee を支援できるのではないかと思います。

と同時に、まだはっきりしませんが、今日見た多摩のソフトウェア
工場で働く人たちの持っている知識を定式化したり、学習支援方法や
指針を示したりすることで、そこで働く人たちを支援できるのでは
ないかとも思いました。開発プロセスをさらに簡素化できるような
ものを提案できたらと思います。

根来社長

別のプロジェクトで進めて頂いてもいいような、素晴らしい内容です
ね。是非ともよろしく願い致します。

以上

* 当内容の無断転載を禁じます。

Copyright (c)2002 CATENA CORPORATION