

Lyee Internet Information

東京工科大学 穂鷹 良介教授 インタビュー（前編）



日付 : 10月25日(水)

場所 : 東京工科大学 メディア学部

穂鷹 良介氏（筑波大学名誉教授 東京工科大学教授） 略歴

- 1937年 3月20日小樽生まれ。 現在63歳。
- 1959年 小樽商科大学卒業。
- 1959年 北海道大学大学院へ進学。
この時デジタルコンピューターを初めて使用。コンピューターの仕組みに興味を持ちのめり込む。
- 1964年 小樽商科大学講師、その後助教授として計算機科学を教える。
- 1966年 経済学博士を取得。
- 1969年 どうしてもコンピューターの世界を極めたく上京。
日本ソフトウェア株式会社に入社。
- 1978年 筑波大学 社会工学系に教授として任官。
データベース、データモデルを研究。
- 2000年4月 筑波大学を退官し、東京工科大学教授として着任。
時を同じくして、筑波大学名誉教授を授与される。

主な著作

- 「データベース入門」(オーム社刊)
「データベースシステムとデータモデル」(オーム社刊)
「データベースの論理設計」(情報処理学会刊) など

<インタビュー内容>

- Q : 最初に穂鷹先生のバックグラウンドを教えてください。

- A : 大学院時代は数理経済学をやっており、学位もそのテーマで取りました。

しかし、数理経済学は世の中の為に実際に使われることは絶対無いと確信するに至ったので、徐々に計算機に研究テーマを移し始めました。小樽商科大学では常に最先端の計算機を使用するという環境には恵まれないと判断したので、日本ではじめてのソフトウェア会社である日本ソフトウェア株式会社に移ることを決断しました。昭和44年3月のことです。

日本ソフトウェア株式会社に入りたての4月頃、電気試験所（現在の電総研）との共同で行う情報検索（IR: Information Retrieval）の研究プロジェクトに入りました。情報検索は利用者に蓄積しているデータの変更はさせないのが建前なのですが、当然更新機能は必要なのでプロジェクトの中でデータの更新の問題を考えるようになりました。ちょうど電気試験所の西野室長が米国からデータベースという言葉を持ち帰ったものですから、われわれのプロジェクトもデータベースを取り入れるものとしてすこしずつ変わって行った訳です。

これがデータベースとの出会いですが、昭和44年ですから、日本でデータベースという名前を聞いた人間としてはかなり早い部類に入るといえます。

大学に居たときファイル言語という新しい言葉を一度聞いたことがありまして、プロシジャではなくてデータを記述するという言語が気になっていたこともあってデータベースは性に合うなって感じがしたのです。

これがきっかけでずっとデータベースを研究することになった訳です。データベースは具体的にデータの処理をする機能の話ですが、私としてはその背後にあるデータモデルに惹かれて、いろいろなデータモデルについて考えたり、提案したりなんて事をしてきました。

情報システム設計に関してはセンテンシャル・データベース・デザイン・メソッド（SDDM）というデータベース主体の方法を自分自身で考えてみたりしました。

何れにしてもデータベース・データモデル屋ですよ。

- Q : 解りました。では Lyee に巡り合った経緯について教えてください。

- A : 日経新聞の見開き広告ですよ。（注：7月26日刊）

- Q : ごく最近ですね。それまでは全然ご存知なかったのでしょうか？

- A : 全く知りませんでした。日経新聞を購読していて良かったって初めて思いましたよ。（笑）あれ、見てなかったら今でも知らなかったでしょう。

- Q : あの広告の何に惹かれましたか？
- A : なんか誇大広告っぽい事が書いてあったんですよ。でこれは偽者に違いない！
って思いましたね。(笑)
そこで、ひとつ出向いて“道場破り”してやろうって思ったのです。だって色んな方法論が出てきますけど、数年で皆つぶれていくでしょう。
主にビジネス系の情報システム開発に関してですが、ソフトウェアメーカー、コンサルタント、雑誌等はいつも3文字略語とか4文字略語を使ってユーザを脅かすでしょう？古くはMIS、ちょっと前まではBPRとかSISとか。でも言葉変えてるけど本質は一緒なんですよ。しかしながらそうした言葉を“バズワード”(BuzzWord=ぶんぶんと触れ回る)にして流行させる。新しい略語を知らない利用者は新しい概念に乗り遅れてはいけないと思って一所懸命に新略語のお勉強をする。でも新略語で答えが出たかという従来域を出ない。結局決定版ではない。こいうことがMIS以来ずっと続いているように思います。
プログラミング言語も同じようなことが起きています。毎年新しい言語が出てくるのですが、どれも決定版じゃない。これからも決定版でないものが出続けるでしょう。どれも本質でないものを毎年聴かされつづけるといい加減飽きちゃうんですね。
そんな気持ちの時に、Lyeにはそうではないような“何かひっかけ”を感じてましてね。早速序論セミナーに申し込んで参加したんです。
- Q : 聞いたところでは、Lyeの“道場破り”の為、3つの議論材料を用意してお出掛けになったそうですね。
- A : そうそう。セミナーを受講して、一晩考えてみて、セミナーでは説明のなかったデータベースに関する部分でね。非常に重要な問題があるんだけど、Lyeではその部分を扱ってないように思えたので。ビジネスで一番大切な問題をどうしてくれるんだ？っていう事で、再度乗り込んだんですよ。(笑)で、論破してやろうと思ったら、さっとかわされましたというか、きちんとした答えができたんですよ。で、“道場破り”は出来なかった。
- Q : 道場破りに用意された内容は「排他制御」・「ファイルの共有」・「順序の決定」だったそうですが？
- A : そうそう。その3つでしたね。

- Q : それぞれの質問の意図と得た回答、それに対する見解について教えて下さい。先ず「排他制御」についてはいかがでしょうか。
- A : Lyee ではデータを取得した時に、その時点で信用ならないデータは持ってこない訳で、結果的に排他制御の必要なデータは排他制御を行った後のデータを持ってくると説明されました。仕様を書く人がデータベースの排他制御をきちんと使いこなせればそれで問題ないこととなります。具体的には DBMS などの下のルーチンに任せる事になっている。私は当初 Lyee がその部分の責任を持つのかなあと考えていたのだけど、それは DBMS 側の既存能力を使うと言われたので、なるほど納得と。Lyee はその部分は責任範囲外な訳です。
- Q : 良くお客様にも言われますが、どこかで何にせよ、排他制御を規定する作業は残ると思いますか？
- A : 通常はユーザのインテンションをくんで DBMS 側で対応します。普通人は排他制御を意識してないですね。でもちゃんと排他制御はかかっている。そういうやり方をしていますが、微妙な制御をしようと思った場合、そこはユーザが制御を握って、排他制御をやるという場合もあるでしょう。応用によってはですね、DBMS 側に排他制御を任せておくとまずい場合があるでしょう。その場合には Lyee 側でも何か考える必要があるでしょう。DBMS に任せられないのですから。
- Lyee とデータベースに関しては私自身未だ充分理解していない部分がありますので、釈然としない部分も正直残っています。
- ただ、Lyee とデータベースは別の世界に住んでいるなあって気がするんですよ。つまりテリトリーが違うから矛盾がないと思うのです。
- Q : ちょっとここはくどくなりますが、別の世界で矛盾がないとしても、作業自体はなくなりません。この部分ユーザはどうするんだ？という疑問を持っているのですが。
- A : 自己生成という部分がありますね。ここでは簡単なロジックしか記述しないと睨んでいます。DBMS 中のロジックはもっと複雑ですから、あの部分つまり、自己生成では書ききれないし、公開されている例題でも取り扱っていません。その一連のものは、DBMS を信頼し、お任せして投げている訳でしょう。Lyee はそういう立場をとっている。それで正しい答えも出る。その様に明確に分業するならば、それはそれで納得できます。
- 但し、データベースの内側に深入り、エラーリカバリー等をやろうとするとどうするのか？この点私なりに Lyee をまだまだ研究しなければ、と思います。

- Q : 次に「ファイルの共有」について教えて下さい。
- A : 私が一番興味を持っているのは、複数個のファイルの中にある、同じデータを多数の人が同じ意味だと認識して、情報処理を誤り無くかつ効率的に行うことです。ところが Lyee はそう考えずに、認識（ある人がある対象について思った意味）は毎回全部違うものであると考えているということでした。

つまり、私は同じ意味のデータをいかにして普及させるか、そのために方法論はどのように役立つのかと考えていたのに対し、Lyee はそのようなことは方法論の解決できることではないと考えているということです。

どちらが正しいのかという私の考えは願望にすぎなく、事実は Lyee の方が正しいと言わざるを得ないということだと思えます。勿論 Lyee でも、ある会社 A の「商品」というデータと、別の会社 B の「商品」というデータが同じ意味であると仮定できるときには、喜んで同じものとして扱うのだと思えます。

しかし、一般的には A 社の「商品」というデータと、B 社の「商品」というデータが全く同じ時ではないときには、利用者の責任でそれらの変換を用意しなくてはならないという回答でした。つまり“同じ”という事は最初からは想定できないんだと。だから同じと睨むからには、利用者が同じものだという事を記述しなくてはならないと言われました。今にして思うと多分これが Lyee であろうところの同期構造の一部だと思えます。

言われてみれば確かにそりゃそうで、認めざるを得ないと納得したのです。グーの音も出せませんでした。なるほど大変良く考えられているなど感じさせられました。こちらが普段、無意識に期待している幻想を打ち破って「本当に良く考えるとこうなのです」という回答を出してくれたのです。

- Q : そうですか。少し話が飛んでしましますが、現在の DBMS の問題点に通ずるところがありますね。

- A : そう。コンセプト、つまり意味はテレパシーがありませんから、相手にきちんと（正確）に伝わらないんですよね。テレパシーがあると「こういう意味だ」と相手に理解してもらえらるんですけどね。

したがって通常は意味を表すものとしてシンボルを与え、そのシンボルに対しては、皆が同じ意味を抱いてくれると仮定するわけです。そのような仕組みが無ければデータの共有ができないし、効率も上がりません。

今流行りの ERP も少し勉強してみましたが、例えば A 社の ERP と B 社の ERP はそのままではくっつかないんですなあ。

つまり意味の表現しているシンボルが違う意味に捉えられていたり、同じコンセプトが一方にあるのに他方になかったりして。データベースの一番の問題点というのは、結局シンボルに共通の意味を与えることに尽きると思うのです。

一時、私はそれを標準化すべく、ISO などで統一したいなあという幻想を抱いて随分努力を重ねたのですが、結局意味を何の言葉で表すかを押さえるという事は、業界を仕切る事になるため、皆用心してしまって仕切らせないんですね。一つの団体によっては。

で、結局今は各社意味がバラバラ、各人バラバラであるというのが一番の問題点だと思います。

いわゆる“標準化”という手段では、共通の意味を持つシンボルを定めるのは著しく困難で、一社が勝って市場を制覇するデファクトスタンダードが意味を標準化するしか方法がないかなと思っています。

- Q : すいません話がそれました。「順序の決定」については？

- A : 順序の決定はですね、2つのデータのどちらかを先に計算しなければいけないという問題が実務では存在するのですが Lyee では計算順序を気にしなくて良いというようなことを言っている。どうしてそのようなことが可能なのか詳しく聞こうとしたのです。

Lyee では計算を行う単純な「プログラムのようなもの」が沢山実行されるのを待っています。その個々の「プログラムのようなもの」の実行順序は確かに指定しないのですが、肝心の論理的な順序を守らなくてはならない部分は、“自分は「これこれ」の条件が整わない限りは本当の計算は行いませんよ”と条件を書いておく訳です。したがってその「プログラムのようなもの」は順不同で実行されるのですが、条件が整わない間は門前払いを繰り返すのです。

この方式では確かに「プログラムのようなもの」の実行順序は論理的な計算をするのに関係が無い、ということが納得できたので得心したわけです。

そういう訳でエレメンタルな3つの事に関しては、私が Lyee 側の回答に対して反例を挙げる事ができなかったのも、納得と。

とりあえず道場破りできなかったんですよ。(笑)

- Q : それで Lyee をさらに掘り下げる事にしたわけですね？

- A : そうですね。まだ他にも例えば大変良いパフォーマンスが出るというのが本当だろうか、とか色々知りたい事があって、そういった事を私は噂とか何とかではなくて自分で確かめてみたいという気持ちなのです。

実は Lyee に関しては色々悪く言う人がいて、私に近づかない方がよいよと忠告してくれた人もいましたし、怪しそうだとか疑いの目で見ている人が大勢居るらしいということを最近知りました。私は自分で物事の判断をいささかながらでもできると思っているので、とことん自分で調べようと思っている訳です。

- Q : ではエレメンタルな疑問点とは別に良い印象はありましたか？
- A : ええ、先ほど少しだけ触れましたが、良さそうだなあと思ったのは、Lyee という端点と始点の部分をプログラムできちんと規定してくれているところです。その部分が、私が研究していた SDDM に似ていたのです。
- Q : どのあたりが Lyee と似ていますか？
- A : SDDM もユーザが「これこれのデータを欲しいと」言う訳です。そうするとそれに応じて、そのデータはどう作られるのかという事を展開していった最終的にはデータのインプットまでいく訳です。
この見方・考え方は Lyee と同じですね。ただ、私の場合は Lyee で言うところの単語ではなく、センテンスだったのです。だって「3」とか言ったって意味ないでしょう。「私には3人の子供がいる」という風にセンテンスにならないと意味がないので、意味の最小単位はセンテンスだと思った訳です。だから「3」とかいう単語とは思いつかなかったのです。私に言わせれば Lyee には「3」という単語の中に隠れた主語がいるんです。例えば「3」とは“この伝票の中の”「3」ですよ。単なる「3」ではありません。伝票と「3」がくっついてるんですよ。そういう意味ではセンテンスなんです。どうも要求ドリブンでブレイクダウンしていくところが良く似ているなあと。もっとも、私自身それ以上深める事が出来なかったのですがね。
- Q : では違いは。
- A : SDDM は、実はあるデータからあるデータが出るよという事を設計者が宣言する訳です。それをどの様の実現するかというと、入力から出力をつくるようにプログラムを書くことによって実現する。
で、プログラムはどう書くのか？というと、「適当に書いて下さい」という感じな訳です。つまりここから先はプログラマの能力とセンスに頼る事になります。Lyee はここがきちんとしたスペックになっているのです。ここが大変感心しているところです。
- Q : ほかに良い印象はもちましたか？
- A : ああ、これ真っ先に言わなければいけなかった。
デバックの方法が好きなんです。序論セミナーでもやってますが「ここにバグがありますね」と言って、見つけて、出来上がったプログラムを全部消してしまって、再度、自動生成するんですよ。これねえ、これ好きですよ。
チマチマ修正するより、捨ててもう一度作るって気分いいなあと思ひましてね。こんな事できるのってすごいなあって。

これはプログラムの動き方をちゃんとハーネスしていて、一つの基本原則で違ったパターンが出ない様にしているんですね。そこを見抜いて、ごく少ない形で全部をうまくやる様に型にはめた。あれは非常にミステリアスで私はどうして動くのか未だわかってないんですが、これは大変な話しですから、真っ先に調べて、究めて、私なりに「本物だ」と確認出来たならば学生にも説明しておいて上げたいと思います。教師として教える義務があるなあと。

後編につづく

* 当内容の無断転載を禁じます。

* Copyright (c)2000 CATENA CORPORATION