

「D 情報の技術」における構想設計のポイント

情報技術の原理と課題の設定

21 世紀を豊かに生きるための「科学技術の智」¹⁾によれば、情報科学技術の基本原理は、

- ① 「情報を 0 と 1 の列で表わす (デジタル化)」
- ② 「それを単純な演算の組み合わせで処理する (計算化)」

の 2 点であるとされています。

この②について記述したものをプログラムと言います²⁾。現在のコンピュータは、いわゆるノイマン型コンピュータとして知られ、プログラムとしてメモリに格納された命令を解釈して自動的に実行するため、メモリの命令を書き直すことで汎用性を高めています。この原理を成立させるために、新しい学習指導要領では、情報の表現、記録、計算、通信の特性³⁾という、より科学に近い原理・法則が用いられるとしています。こうした情報科学技術の基本原理は、私たちの生活の中で様々な条件の中で最適化されて製品・サービスとなっています。最適化を検討する条件としては、社会からの要求、使用時の安全性、情報の倫理やセキュリティ、経済性、情報・情報技術の活用、情報をどのように管理するか、システム全体の条件等が挙げられます。これらの観点で、既存の情報の技術について調べたり確かめたりする活動を通して、これらの技術の仕組みや問題解決の工夫について考えさせ、最適化について気づかせることとなります。例えば、情報科学技術における基本原理の「デジタル化」は、「情報の表現」を変えました。画像であれば解像度と階調によって、音声であれば標本化周波数と量子化ビット数といういわゆるパラメータによってアナログの情報をデジタル化することが可能になりました。それらのパラメータの設定次第で、デジタル化の品質や情報の量がどのように変わるのか「調べ・確かめる」ことができます。その情報をどのように「記録」するか、サイズの変更やフォーマットの変更など「計算」によって行わせる処理を考えた場合に、画像を写真のように印刷するのか、web 用のサムネイルとして表示するのか、「活用」の目的によっても最適な設定は変わり、保存媒体の容量にも影響を与えることから「経済性」にも関わってきます。計算によって行わせる処理が CPU 等に与える「負荷」によっては、ハイスペックなコンピュータが必要になるかもしれません。記録媒体へデータを読み書きしているときや、計算処理を行っているときなどは、OS に付属のリソースモニターで、コンピュータを構成する要素に掛かる負荷の様子を確認させると、システムとしてデータを処理していることをイメージしやすいかもしれません。

こうした内容は従来も指導されて来たものですが、コンピュータの構成、デジタル変換の方法、歩留まり・トレードオフ (経済性) などのように、個別に扱われることが多かったのではないのでしょうか。新学習指導要領では、それぞれの情報の技術がどのように原理・法則と関連しているのかをより意識する必要があります。そうするからこそ、生活や社会で利用されている情報の技術について、問題を見いだして課題を設定することができ、解決策を構想し、製作図等に表現してから、試作 (プロトタイプ作成) を通じて、修正を行ってより良い結果が得られるように解決していく学習の流れを持たせることが求められます。

加えて、新たに「双方向性のあるコンテンツのプログラミング」という内容が追加されました。技術分野の改訂の方向性として示されていた時点では「動的コンテンツ」⁴⁾であったことを考えると、双方向性とは、いわゆる「インタラクティブ」と捉えるのが妥当でしょう。つまり、入力された情報によって動的にフィードバックが変わる内容をプログラミングすることになると思われます。それには UI (ユーザインタフェース) が必要になります。UI の設計は、認知科学的な側面を扱わずに中学生の範疇を越しますので、昔と今の UI を比較するなどして、使いやすくするための工夫、誤操作を防ぐための工夫およびユニバーサルデザインを知っておけば、自分が制作するプログラムの UI にも、その先人の知恵や工夫、ガイドラインを活かすことができるでしょう。

解決策の構想

従来の、計測・制御のプログラミングがそうであったのと同様に、双方向性のあるコンテンツのプログラムの制作もまた、必ずしも実用性のあるものを制作することだけでなく、実際のシステムのモデル化やシミュレーションを通した学習もあることでしょう。しかし、いきなりプログラミングすることは、「当てずっぽう」のものづくりであり、最も避けなければいけません。情報の技術におけるものづくりのプロセスの特徴は、これから制作しようとするプログラムのコンセプトや全体構想などの概念設計と、それに基づいてどのような画面や機能で構成する必要があるのかという個別の詳細設計を行い、プロトタイプを作りながら、その動作結果次第で、再度設計を見直すという、スパイラル型で制作が進めていくことです。

これまで、設計ではフローチャートが扱われてきましたが、これはノイマン型コンピュータが開発された 1940 年代頃から使われているのではないと言われる歴史ある図示の方法です。現在でも特定の処理の流れを示すのに利用されることがあります。しかし現在では、JIS および ISO で規格化されている UML (Unified Modeling Language) と呼ばれる、モデル化の図示方法群から、目的に応じて必要な図示方法を用いることが主流です。なぜならば、プログラムやシステムはフローチャートでは十分に表現できないような、多くの機能が相互に状態を変えながら関連していたり、利用者の操作の流れからも設計したりすること等が求められるようになってきたからです。そこで、単に 1 つの処理の流れを書くだけではなく、プログラムやシステムの「構造」や、プログラムの「振る舞い」を表現・設計する UML という統一的方法が作られました。中学校段階でのシステムやプログラムでは、「振る舞い」について図示するだけでも十分かもしれません。そこで「振る舞い」を図示する手法の 1 つである「アクティビティ図」を紹介します。この図は、プログラムの手順だけではなく、日常生活や計測・制御システムの全体の流れを示すことにも利用できる、大変汎用性の高い図示方法です。図 1 は、Astah というサイトで紹介されていた、銀行の ATM でお金を引き出す際のアクティビティ図です⁵⁾。フローチャートと類似していますが、処理がより分かりやすく可視化できていることが見て取れると思います。設計段階では、特定の言語に依存しないで、こうした図示によって対話的な学習活動を行い、より良い処理を検討し、どのような機能が必要になるか構想してから制作に入ると良いでしょう。特に、双方向性のあるコンテンツでは、従来の計測・制御のプログラムよりも、大規模で仕組みが複雑になることが予想されます。全体構造と必要な機能の関係を捉えてからプログラミングしないと、予期せぬ大幅な手戻りが発生することにもなりかねません。

文献

- 1) 21 世紀を豊かに生きるための「科学技術の智」情報学 <http://www.scj.go.jp/ja/info/kohyo/pdf/kohyo-20-h64-3.pdf>(参照日 2017/4/5)
- 2) 科学技術の智プロジェクト、総合報告書、p.63 (2008)
- 3) 学習指導要領、http://www.mext.go.jp/a_menu/shotou/new-cs/_icsFiles/afieldfile/2017/03/31/1383995_3_1.pdf(参照日 2017/4/5)
- 4) 教育課程部会家庭、技術・家庭ワーキンググループ資料 8、技術・家庭科(技術分野)の改訂の方向性(たたき台案) http://www.mext.go.jp/b_menu/shingi/chukyo/chukyo3/065/siryo/_icsFiles/afieldfile/2016/04/04/1369020-08.pdf(参照日 2017/4/5)
- 5) Astah, <https://ja.astahblog.com/2015/09/24/uml-activity-diagram/> (参照日 2017/4/5)

安藤明伸(宮城教育大学)

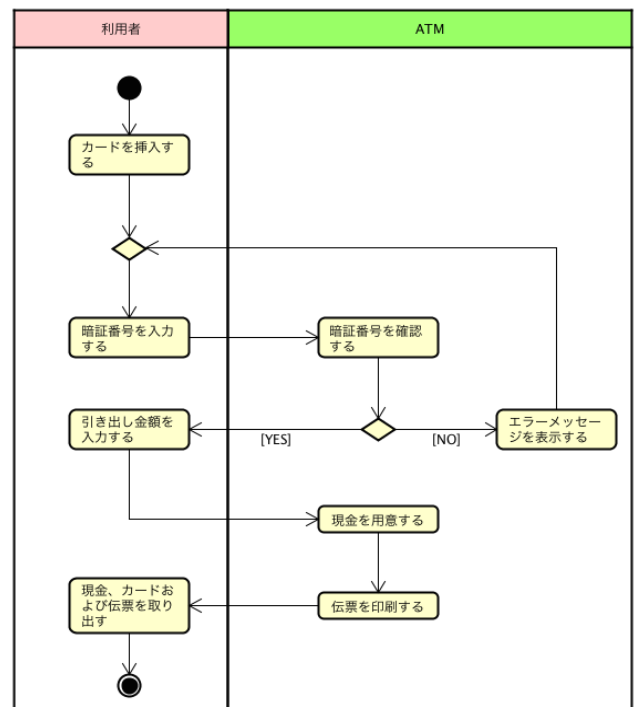


図 1 銀行の ATM でお金を引き出す処理の流れ⁵⁾

<https://ja.astahblog.com/2015/09/24/uml-activity-diagram/>より