

氏名	藤田 至一			
ヨミガナ	フジタ ヨシカズ			
学位の種類	博士（映像メディア学）			
学位記番号	映博第6号			
学位授与年月日	平成25年3月25日			
学位論文等題目	〈論文〉 創作における言語化プロセスと水平思考			
論文等審査委員				
（主査）	東京芸術大学	教授	（映像研究科）	桐山 孝司
（論文第1副査）	東京芸術大学	教授	（映像研究科）	桂 英史
（副査）	東京芸術大学	教授	（映像研究科）	佐藤 雅彦
（副査）	東京芸術大学	教授	（映像研究科）	藤幡 正樹
（副査）	東京工業大学	准教授		長谷川 晶一

（論文内容の要旨）

本論は、コンピューターを利用した創作において、ソフトウェア開発を含む創作プロセス全体をどの様に進めたら良いかを、筆者の創作事例を通して明らかにしようとするものである。

第一章「はじめに」では、筆者がビデオゲーム開発の現場で常日頃感じている問題意識と、それを踏まえた本論の目的について述べる。現状のビデオ・ゲーム開発においては、利便性の高いツールやライブラリで構成された抽象化レイヤーを用意する事で、異業種間のデータの流れをスムーズにしている。一方、本論が対象とする新規性のある創作には、関係者が議論を重ね、意見の衝突や誤解から新しい認識を獲得共有するといった言語化プロセスが必要である。抽象化レイヤーはプロセスを固定し、新しい認識を獲得共有する機会を減少させる。本論は、言語化プロセスをどの様に進めるべきかを明らかにする。

第二章「問題の明確化と理論的考察」では、第一章で述べた問題意識の背景を理論的に考察する。アラン・ケイのSmalltalkソフトウェア・エージェント、オブジェクト指向を軸としたGOFデザイン・パターン、アジャイル開発におけるDSL(Domain Specific Language)といったソフトウェア開発の歴史を俯瞰する。現状では、ユーザーの意図を明示化する手法が確立されていない事を示す。

第三章「創作プロセスモデル」では、芸術家のアトリエ取材、ビデオ・ゲーム開発や本学におけるメディア・アート作品制作への参加といった筆者自身の創作事例を題材に、新規性のある創作が生まれる状況のモデル化を試みる。このモデルでは、アイデアを徹底的に言語化した結果、ある種の「事故」が認識可能となり、それまでの認識の連鎖を切断し、水平思考によって新たな言語が創り出される。モデルを検証するために、機械の不具合をきっかけに抽象化レイヤーを細分化し、再構築する事によって新しい創作領域を創り出した事例、集団でアイデアを議論する過程で相互の認識の相違が明らかとなり、それが発想の転換点となった事例について考察する。

第四章「ケース・スタディ」では、言語化プロセスの実例として、筆者と芸術家によるアニメーション・ソフトウェア開発事例を分析する。このソフトウェアでは、既存の抽象化レイヤーが想定していなかった

ジェネラティブ・アニメーションを実現した。筆者と芸術家は、創作プロセスモデルに沿って言語化を進め、言語化の難しい概念をプログラム言語に翻訳した。この言語によって芸術家は、生成したグラフィックスを確認しながら生成ルールを編集できるようになり、新しい創作領域を探索する事が可能となった。芸術家から得られた「面白い、楽しい」といったフィードバックは、言語化プロセスとソフトウェア開発を正しい方向に導く事ができた事を示している。

第五章「まとめ」では、これまでの議論をまとめる。ソフトウェア開発者が、創作言語を生み出すという立場で創作プロセスに参加する事で、適切なDSLの設計が可能となる事を確認する。

(論文審査結果の要旨)

本論文は、コンピュータを使ったゲームや映像などの創作において、どのようなプロセスをとることで新規性のあるものが作り出される可能性があるかを、著者自らが参加した事例を通して研究したものである。全5章からなる論文全体を通じて、創作に参加する人々の間で意図を伝えるための言語が共有できることが、従来にないものを生み出す水平思考を可能にすることを明らかにしている。

第1章「はじめに」では、著者のゲーム開発者としての経験を踏まえて、創作の場でどのようなことが必要かという問題意識を述べている。現在のゲーム開発体制ではミドルウェアと呼ばれる開発ツールが用いられる。しかし効率的なソフトウェアを積み上げた抽象化レイヤーに頼った開発方法では開発者間のコミュニケーションが乏しくなり、新しい発想が生まれる余地がないことを指摘する。忍考のジャンプを可能にするには、意見の衝突や誤解の発見などの事故を積極的に起こす必要がある。そこでこの論文では、創作言語を共有することから始めて新しいソフトウェアを作りだした事例を研究し、必要なプロセスをあきらかにすることを目的と定めている。

第2章「問題の明確化と理論的考察」では、ソフトウェア開発からみた抽象化レイヤーの問題点を分析する。ソフトウェアを創作など思考のメディアとして使うことに積極的だったアラン・ケイはダイナブックというコンピュータを想定し、そのコンピュータで利用者自身が自在にプログラムを作って思考を深めるためにプログラミング環境Smalltalkを開発した。しかし創作に関していえば、非定型的な思考を記号化してプログラムにすることのハードルは高く、結局Smalltalkで利用者がそのままプログラマになって自分のツールを開発するという事は起きなかった。一方、世界を納象化してモデル化するオブジェクト指向プログラミングは、Smalltalkから派生してソフトウェア開発の標準的なパラダイムとなった。さらにオブジェクトの役割をさらに抽象化するデザインパターンが建築から借用され、再利用可能なソフトウェア開発の主要な方法になっている。現在では、特定分野での問題解決に対応する領域記述言語(DSL)を最もユーザに近いレイヤーに想定し、それ以下のレイヤーでは汎用化を進めて側発効率を上げるアジャイル開発が主流になっている。しかしアジャイル開発においても結局DSLをどのように構築するかは解決していない。

ここで著者は、創作のためのソフトウェアにおいては、DSLを作る以前に芸術家と開発者が経験的な知識を共有できるか創作言語が必要なことを指摘している。

第3章「創作プロセスモデル」では、創作言路がどのようにできるかを考察する。著者はまず一人の創作者がグラフィックスもソフトウェア開発も行っている場合として時計作家の浅岡肇氏からヒアリングを行った。その中で、グラフィック・ソフトウェアが扱う色空間を標準化すると、制約の多い側に合わせることで、本来可能な創作ができなくなることを指摘している。またCAD/CAMで設計した部品を加工しようとして工具が破損したことから自らカッターパスを作って加工したというエピソードを引用し、想定外の事故が新しい作り方につながることを指摘している。次に集団での創作においては、言語化の過程で思考の飛躍が生まれることを指摘している。言語を媒介にして意志疎通しようとするときに、問題点や矛盾が明らかになったり、無意識に前提としていた文脈に気付いたりして、創作上の事故が起きるのである。

著者はこの創作の過程を、創作言語の共有としてモデル化している。創作言語の背後にはさらに言語化

できない操作的知識や視覚的知識があるが、言語化できない領域でも創作言語によって共通の索引を持つ形で体験の共有ができることがある。このモデルを確認するために、著者は東京芸術大学で行ったメディアアートプロジェクトについても再考している。

第4章「ケース・スタディ」では、著者がゲーム開発者としてグラフィックデザイナーと意思疎通を図りながらDSLを構築し、コンピュータを使ってこれまでグラフィックデザイナーが試みたことのない方法で創作を行った例を分析する。この事例では、ゲームのエンディング・スタッフ・クレジットというゲーム全体を象徴する役割も持つCGアニメーションを作成している。グラフィックデザイナーは、ラピュタの械のように空中に浮かぶ立方体が龍のように飛び回る感じにしたいという考えを持っていた。著者はこれを現実的な時間で実現するために、個々のグラフィック要素を手で作成するのではなく、アルゴリズムに従って要素が生成されていく方法を提案した。ただすべての要素が自動的に生成されるとコントロールできないものになるので、デザイナーが結果を見ながら自分で生成ルールを構築できるようにした。この仕組みを実現するために、L-Systemという再帰的生成アルゴリズムをLISPで実装し、グラフィック・インタフェース上で生成の指示をデザイナーが行えるようにした。ここでは創作プロセスモデルにあるように、言語化の作業が重要な役割を果たしており、デザイナーが擬音語を駆使して伝えようとするのが反映できるように、生成ルールに工夫を要した。また言語化する中で、キーフレームアニメーションではできず、生成ルールをつかったジェネラティブなアニメーションが必要とされる点も明確になった。この方法によって作ったエンディングロールは従来にないものとなり、デザイナーからもここで開発したツールでグラフィックスを作ることが面白いという反応があった。

第5章「まとめ」では、創作言語を共有することによる芸術家とソフトウェア開発者との共同制作についてケース・スタディを行った結果を振り返り、本論文が目約としたように、創作言語を共有することで新しい開発方法が可能になったこと確認している。本論文の審査の過程で指摘されたこととして、初稿で記述されていた創作過程での身体知、身体性の定義があいまいで全体の論旨を分かりにくくしているという点があった。身体知、身体性とは、創作者が身をもって経験したこと、身体を使って表現しなければ伝えられないことを指すが、これがいかに4章の実践例につながるかについては直接説明がなかった。そこで3章は改めて創作プロセスのモデルを明確にし、そこでは創作者の中にある身体的な経験や知識が、創作言語によって参照され、伝達可能になるということを指摘した。身体知は経験的知識であり言語化はできないので言葉で伝えられるものではないが、共通の経験基盤を持つことで、DSLで表現されたグラフィックスなど実物を参照することで共有できる場合がある。このような理由から創作言語は創作に必要な全ての情報を表すものではなく、創作に参加する人の中で意思疎通を図るための言礎として位置づけるようにした。

また、創作のためのソフトウェアと、ソフトウェア開発手法の発展との関係を客観的に議論すべきであるという指摘もあった。そのため、オブジェクト指向、デザインパターン、アジャイル開発という現在に至るソフトウェア開発手法の流れを概観することとした。これによって、DSLを作る方法論の欠如という現在の問題点も明確になった。

さらに、論文全体の中で事例分析が重要な位置を占めるので、ソフトウェア開発者の級点から、デザイナーとどのように開発を進めたかを、発言などを再現しながら詳細に記述するべきであるという指摘があった。この点を補強することにより、文献としての価値を確かなものにすることができた。

以上のような議論を経て、本論文は映像研究科映像メディア学専攻修士後期課程の論文としてふさわしい水準にあることを確認し、合格と判定した