B-020

Python を用いたインテリジェントパッドの開発と応用
Development and Application of IntelligentPad with Python

野口 孝文† **布施 泉**†
Takafumi Noguchi Izumi Fuse

1. はじめに

物理シミュレーションやエンジニアリングシミュレーションをはじめ数値計算やデータ解析といったシステムの機能は，コンピュータの高性能化に伴い高度化している．さらに近年では，生成 AI の利用によりシステム同士を組み合わせた高度な連携も可能になってきている．一方で，各システムは独立しているため，従来の連携方法では，あらかじめ共通に用意されたファイル連携やデータベース連携を用いることができなければ，その都度何らかのデータ変換が必要となる．本論文では，PC 上で複数のシステムを組み合わせる試行錯誤を重ねながら新しい解法を発見するという活動を支援するための，システムの組み合わせ操作と入出力をダイナミックなオブジェクト部品の組み合わせ操作で実現する IntelligentPad の開発について報告する．IntelligentPadは， MVCからなる可視化されたオブジェクトをダイナミックに組み合わせプログラムを作ることができるシステムである．

本論文の 2 章では，これまでに開発されてきた IntelligentPad とオブジェクト部品間の連携について述べる．3 章では，Python を用いて IntelligentPad を開発することについて，4 章では，これを用いた応用として，学習支援システムについて紹介し，5 章でまとめと今後の予定について述べる．

2. IntelligentPad システム

2.1 IntelligentPad システムの種類

IntelligentPad は，1987 年に北海道大学の田中によって考案されたシステムである[1]．IntelligentPad ではパッドと呼ばれるプログラム部品を紙のイメージで貼り合わせたり複写したりして，目的のプログラムを実現する．

これまでにいくつかのシステムが開発されている．初期に開発されたシステムは，Smalltalk を用いて開発され，その後 C++や C#を用いて開発されている[2][3]（図 1）．本論では Python を用いて開発した Python 版 IntelligentPad を紹介する．このことは，3 章で述べる．

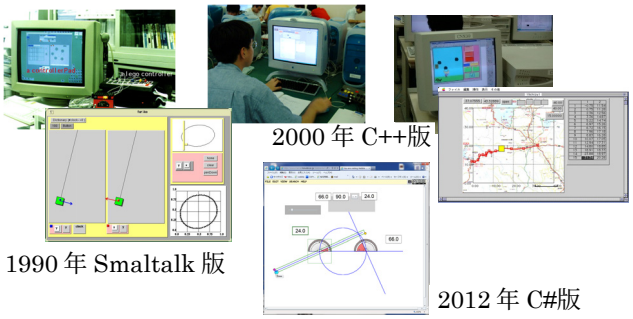

図 1　これまでに開発された IntelligentPad

† Information Initiative Center, Hokkaido University

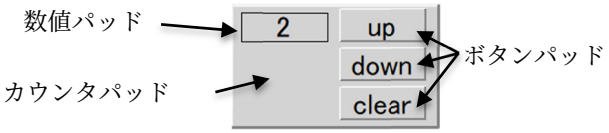

図 2　パッドの貼り合わせによる機能合成（カウンタ）

2.2 システムの操作性

パッドは，M(Model) V(View) C(Controller)からなる複合オブジェクトである．V（ビュー）同士を貼り合わせることができる．貼り合わせたパッドもパッドと呼び，下のパッドを親パッド，上に貼付したパッドを子パッドと呼ぶ．利用者は，ディスプレイ上のパッドをマウスの直接操作で，貼り合わせたり複写したり削除したりできる．

M（モデル）には，データを保持するためのスロットが定義され，子パッドのスロットの 1 つを親パッドのスロットに結合することによって，それぞれのパッドが持つ機能を併せ持った複合パッドを作ることができる．図 2 は，計数機能を持つカウンタパッドにボタンや数値表示パッドを貼り合わせたカウンタの例である．カウンタパッドは，スロット{<value>, <up>, <down>, <clear>} を，数値パッドは，スロット<value>を，ボタンパッドは，スロット<command>をそれぞれが持っている．

カウンタパッドのスロット<value>は計数値を保存し，数値パッドの<value>と結合することで計数値を数値パッドに表示することができるようになる．カウンタパッドの<up>は，ボタンパッドの<command>と結合することで，ボタンパッドからの set メッセージ（親パッドから子パッドへ送るメッセージ）でカウンタ値を 1 増す．カウンタパッドの<down>や<clear>も同様にして，カウンタ値の 1 減や 0 設定を行う．

上述の基本機能を持ったパッドを基本パッド呼び，これまでに様々なパッドが作られ，基本パッドを組み合わせた複合パッド（パッド）が作られてきた．

3. Python を用いた IntelligentPad システム

3.1 Python 版 IntelligentPad

2019 年から Python を用いた IntelligentPad の開発を開始した．Python には多くのライブラリがあり，IntelligentPad 開発に Python を用いることで，多様なシステムをダイナミックに試行錯誤しながら構築することが可能になる．

図 3 は，Python 版 IntelligentPad で作成した滑車の教材である．これと同様のシステムは，30 年以上前から実現している[2]．各滑車やばねを定義したパッドを貼り合わせることによって動作させることができる．図の中央は左の滑車機構の一部を複写したもので，これを部品として，元の滑車の一部に組み込んで，図の右の滑車機構を作ることができる．この時の部品 (パッド)間のデータ交換において，ばね定数や力のほか部品の取り付け位置情報を標準化した方

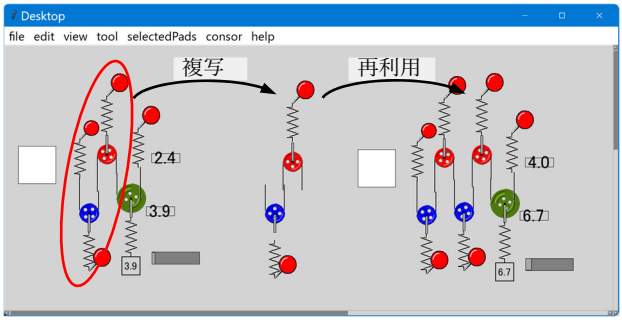

図 3　機構の一部を部品として再利用

法で授受することによって，任意の部分（パッドの貼り合わせ）を複写して利用することができる．Python 版 IntelligentPad は，インタプリタにもかかわらず，過去に作られた C++（コンパイラ）版 IntelligentPad 以上に滑らかに動作させることができている．

3.2 Python 版 IntelligentPad のクラス構成

Python 版 IntelligentPad は，GUI に Tkinter を用いて開発を行っている．システムは， Desktop クラス，Model クラス，View クラス，Controller クラスで構成される．Desktop オブジェクトは，Tkinter を用いたデスクトップウインドウとパッドの管理を定義している．基本的な Pad の操作性は，Model，View，Controller の基本クラスで定義している．また様々な機能を定義したパッドは，これらの派生クラスで実現している．Model，View，Controller の基本クラスのオブジェクトの合成からなるパッドを基本パッド（SimplePad）と呼ぶ．また派生クラスの合成からなるパッドも基本パッド（xxx.Pad）と呼ぶ．派生クラスの基本パッド操作性は，SimplePad から継承する．

図 4 に，デスクトップとその上に基本パッドの基本的な機能として，サイズの変更，貼り合わせ，複写，色付け，透明化の様子を示す．各クラスの機能は，次節以降で述べる．

3.3 Desktop class

Desktop クラスには，デスクトップウインドウの表示とパッドの管理機能を定義している．パッドの管理では，パッドの生成やファイルへの保存のほか，マウスやキーボードイベントを各パッドへ振り分けることも行っている．

デスクトップ上部に 7 つのメニュー項目を表示し，さらにその項目からプルダウンメニュー表示して各種操作を選択することができる．file 項目には，基本パッドの生成機能や，基本パッドを組み合わせて作成した複合パッドの保存機能や読み込み機能がある．

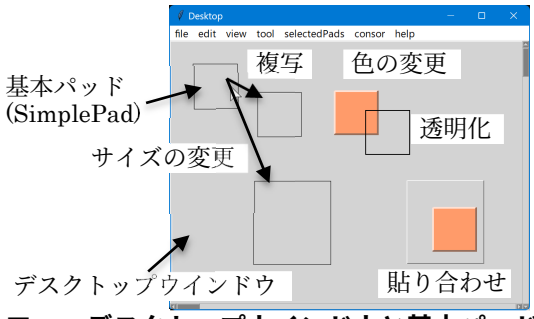

図 4　デスクトップウインドウと基本パッドの操作

デスクトップでは，ウインドウに表示されないが，デスクトップにあるパッド（Controller オブジェクト）を登録し，マウスカーソルが View の表示領域内にあるパッド（View オブジェクト）を抽出して，これにマウスイベントやキーボードイベント送るというイベントの振り分けを行っている．イベントは初めに，View オブジェクトと連携した Controller オブジェクトに送られる．

3.4 Model class

Model クラスには，データを保持するためのスロットやスロットへの書き込み，読み出しや書き込みがあったときの処理を行うメソッドを定義している．また，オブジェクトの複写や保存，読み出しの処理も定義している．

スロットへの書き込み，読み出しは，Viewから行われるため，Model には View を登録する機能も定義している．

２章の図２に示したカウンタの Model および View クラス定義を図５に示す．カウンタの Model クラスでは，Model クラスを継承して CounterM として定義している．スロットは，２章で述べた{<value>, <up>, <down>, <clear>}の 4 つである．それぞれのスロットに結合したパッドからのメッセージは，View（CounterV）を介して CounterM の setValue(self, key, value)メソッドが呼び出され，スロット名に応じた処理が行われる．

loadSlot(self, key, value)メソッドは，ファイルに保存されたパッドを読み込んだときに，保存されたスロット値を再生する機能を定義している．そのほかの Model としての機能は，上位クラスで定義しているものを継承している．

```
class CounterM(Model):
    def __init__(self):
        super().__init__()

    def initSlot(self): #スロットの定義
        super().initSlot()
        self.slots.update({"value" : 0.0,"up": "", "down": "", "clear":"" })

    def setValue(self,key,value): #子パッドからのメッセージ処理
        if key in self.slots:
            if key == "up":
                self.setValue1("value", self.getValue("value") + 1.0) #1 加算
            elif  key == "down":
                self.setValue1("value", self.getValue("value") - 1.0) #1 減算
            elif  key == "clear":
                self.setValue1("value", 0.0) #0 にクリア
            elif  key == "value":
                if isinstance(value, int) or isinstance(value, float):
                    self.setValue1(key, value)
                else: return
            for v in self.views:
                v.updateView()

    def loadSlot(self,key,value):
        super().loadSlot(key,value)
        if key == "value":
            self.setValue1(key,float(value))
        elif key == "up":
            self.setValue1(key,str(""))
        elif key == "down":
            self.setValue1(key,str(""))
        elif key == "clear":
            self.setValue1(key,str(""))

    def getClassName(self):
        return "CounterM"

class CounterV(View):
    def __init__(self,desktop):
        super().__init__(desktop)
        self.output = "value"

    def initProperties(self):
        super().initProperties()
        self.setPropertyValue( "label"  , "CounterPad")
        self.setPropertyValue( "samplePad"  , "CounterPad.ip")

    def getClassName(self):
        return "CounterV"
```

図 5 カウンターパッドの Model と View の定義

3.5 Controller class

Controller クラスは，デスクトップから送られてきたイベントを View に送る機能を定義している．そのため，Controller には View を登録する機能を定義している．また，View の表示の変更や，貼り合わせやスロット結合を行うためのメニュー表示の機能も定義している．

２章で述べたカウンタパッドでは，Controller クラスを継承したクラスは定義せずに，基本クラスの Controller を利用している．

3.6 View class

View クラスは，デスクトップ上の表示機能を定義している．パッドのサイズや色など View のプロパティ変数に基づいて表示するときと，文字列や数値など Model のスロットにあるデータに応じて表示を変えるときがある．そのため， View には Model を登録する機能を定義している．

パッド同士のスロット結合は，View を介して行う．View には Model を介してスロットからデータを読み出すばかりでなく，書き込むこともできる．

２章で述べたカウンタパッドの View は，View クラスを継承して CounterV として定義している．しかし，図５のプログラムに示すように，上書きによる変更は，結合に使用するスロット名とパッドの名称のみの定義にとどまっている．したがって，ほとんどの機能は上位クラスで定義されているものを継承している．

4. IntelligentPad を用いた学習支援システム

4.1 プログラミング授業での利用

北海道大学で 2018 年からロボット[4]を用いた演習の授業を継続して行っている[5][6][7][8]．主に 1 年生が履修する一般教育としての授業（選択）であり，人数は 1 クラス最大で 23 名である．文理や男女に大きく偏らず履修され，学習者のプログラミングに関する知識レベルも多様である．本授業では，ロボットのプログラム作成と作成したプログラムをロボットに転送する等の支援に IntelligentPad を使用している．PC とロボットの接続の様子を図 6 に示す．システムは，USB メモリを PC に挿入することで利用することができる．作成したプログラムをロボットに転送すると，USB メモリにもこのプログラムが時刻名のファイルとして，日付名のフォルダに保存する．このことは，4.4 節で述べる．

4.2 プログラム作成支援システム

IntelligentPad を用いて作成したプログラム作成支援システムとそれを構成するパッドの貼り合わせの様子を図７に示す．システムは 43 枚の基本パッドから構成されている．図の上方は，操作画面のスナップショットである．下方の

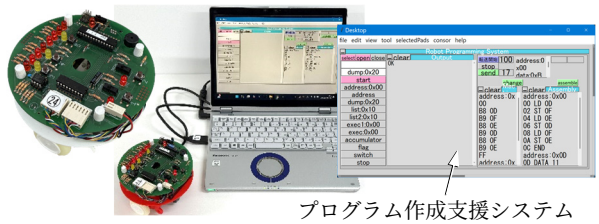

図 6 ロボットとプログラム作成支援システム

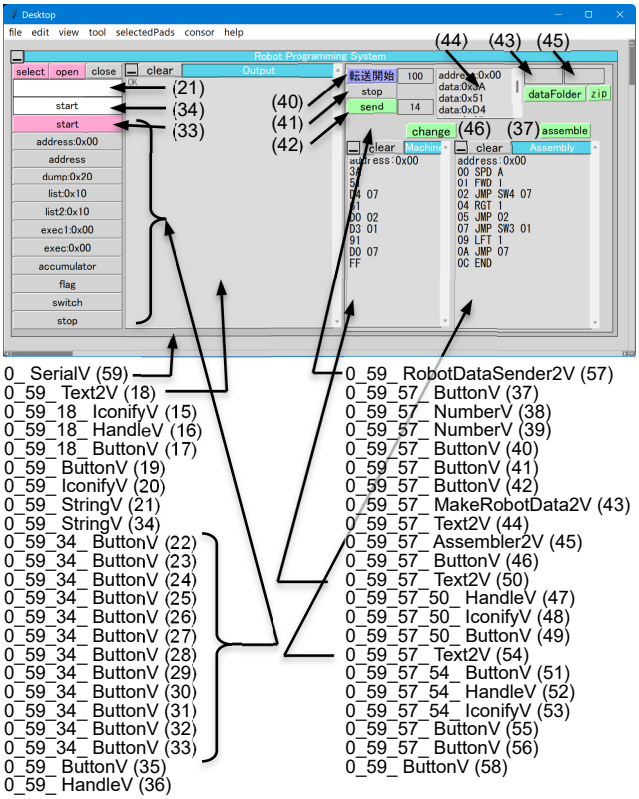

図 7 プログラム作成支援システムとパッドの貼り合わせ

番号付き名前のリストは View オブジェクトの名前と基本パッドに付された id である．

id 0 は，デスクトップを表している．リストの左上にある「0_ SerialV (59)」は，シリアル通信機能を持つ SerialPad の View が id 59 の SerialV で，デスクトップに貼付されていることを表している．リストから，そのほかのパッドはすべてこの SerialPad 上にあることが分かる．

このリストには，スロット結合の関係は示していない．ロボットとこのプログラム作成支援システムの間は，シリアル通信ケーブルで接続され，コマンドやプログラムを文字列で授受している．id 22 から id 33 のボタンパッドには，start や address といった文字列が保存表示されているが，ボタンを押すことで，SerialPad からロボットにその文字列が送られる．

id 54 の TextPad では，アセンブリ言語によるプログラムを作成する．id 37 の assemble と記されたボタンを押すと直下の id 57 の RobotDataSenderPad が id 45 の AssemblerPad に作成されたプログラムを送り，代わりに機械語を受け取る．このデータ授受の仕組みは，図 3 に示した滑車のパッドと同じ方法を用いている．機械語は id 50 の TextPad に表示される．次に，id 46 の change と記されたボタンを押すと直下の id 57 の RobotDataSenderPad が id 43 の MakeRobotDataPad に機械語プログラムを送り，代わりに送信形式に変換したデータを受け取る．この送信形式データは id 44 の TextPad に表示される．この仕組みも AssemblerPad に送った方法と同様である．

さらに送信形式データは，id 42 の send ボタンパッドを押すことで，順番にロボットに送られる．

4.3 機能の追加と削除を容易にした構造

アセンブリ言語の機械語変換機能と機械語の送信形式データ変換機能は，RobotDataSenderPad の上に貼付されていることを前節で述べた．また，データ授受の方法に滑車パッドで用いている方法を使用していることも述べた．この方法を採用することで，これらのパッドは容易に交換することができる．

本プログラム作成支援システムは，43 枚のパッドから構成されているが，このシステムに新たに開発したパッドは，RobotDataSenderPad，MakeRobotDataPad，AssemblerPad の 3 枚である．そのほかは汎用部品として用意していたパッドと他のプロジェクトで使用していたパッドの再利用である．

機能の追加や変更が，多くの場合，部品の追加や交換で対応することができる．特に，システムの開発時は，パッドの再利用により，必要な機能作成に集中して試行錯誤できることから，効率的にシステムを構築することができる．本プログラム作成支援システムの開発においても，上述の 3 つのパッドは，個別に試行錯誤しながら作成している．

4.4 ログの保存と分析機能

図 7 に示したプログラム作成支援システムでは，ロボットにプログラムを書き込む度に，アセンブリ言語と機械語のプログラムをセットとして，日付時刻とアセンブリ言語 A と機械語 M の別をつけたファイル名（アセンブリ言語でのファイル名の例：11_09_45A.txt，機械語は A を M に変更したもの）とし，書き込んだ日付のフォルダ（2024-10-10）内に，保存している．

図 8 に保存したプログラムファイルを基にした分析システムを示す．図の丸で囲んだ部分に，上述の方法で命名した日付のフォルダ名と時刻のファイル名が表示されている．

図の右側では，学生名を選択して，授業日ごとの活動を表示している．また，図の左側では，２つの隣り合う時刻のファイルを並べ変更部分と差分をその右に並べ表示している．この図を見るとやや雑然とした部品配置になっているが，システムの作成を試行錯誤的に行ったため，このような結果になっている．

この図には，表示されていないが，学生が使用している命令の使い方等も出力できるようにしている．その結果，2024 年度における学生が作成したプログラムと 2023 年度に作成したプログラムの作り方の差を簡便に比較することができるようになった[8].

5. おわりに

本論文では，システムの組み合わせ操作と入出力をダイナミックなオブジェクト部品の組み合わせ操作で実現する IntelligentPad について，Python を用いて開発したことを報告した．システムを構成するクラス間の関係を説明し，各クラスの機能を説明した．また，IntelligentPad を用いたプログラム作成支援システムを授業で用いていることと，授業の分析にも用いていることを紹介した．

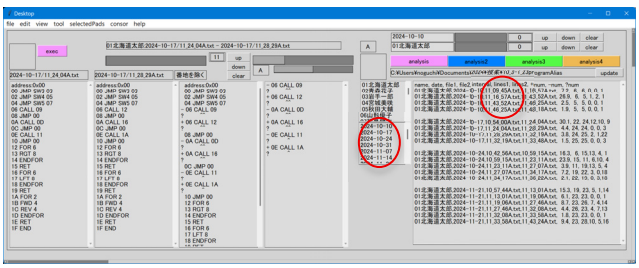

図 8 プログラム分析システム

Python を用いた IntelligentPad をシステム開発に利用することで，Python の豊富なライブラリが利用でき，また IntelligentPad の試行錯誤が容易なことから，システム構築が効率的にできることも紹介した．今後は，生成 AI をはじめ Python で利用できるライブラリをパッド化することで，様々なシミュレーションや分析に対応した試行錯誤環境の整備を進めたいと考えている．

参考文献

[1] Y. Tanaka, A. Nagasaki, M. Akaishi, and T. Noguchi, “ A Synthetic Media Architecture for an Object-Oriented Open Platform”, Proc. of the IFIP 12th Computer Congress, pp.104-110 (1992)
[2] 野口孝文, 田中譲, “ コンストラクションセットを持つマイクロワールド", 情報処理論文誌, vol.11, No.1, pp.152-166 (1995)
[3] M. Kuwahara, Micke, Y. Tanaka, Yuzuru "Webble World 3.0: In the Borderland Between Being a User or a Developer", ISIP 2014, CCIS, vol. 497, pp. 85-96, Springer, Heidelberg (2016)
[4] T. Noguchi, H. Kajiwara, K. Chida and S. Inamori, “Development of a Programming Teaching1-Aid Robot with Intuitive Motion Instruction Set”, Journal of Robotics and Mechatronics, Vol.29 No.6, pp.980-991 (2017)
[5] 布施泉，野口孝文，“ プログラミングロボットを用いた協調学習の実践と展開可能性 ”，教育システム情報学会研究会報告, vol.37, no. 2, pp.1-5 (2022)
[6] 野口孝文，布施泉：「思考を支援するユーザインタフェースの研究」，教育システム情報学会全国大会, pp.7-8 (2024)
[7] 布施泉，野口孝文：「プログラミングロボットを用いたペア学習と協調学習の連携の効果」，教育システム情報学会全国大会, pp.267-268 (2024)
[8] 野口孝文，布施泉，梶原秀一，千田和範，稲守栄：「プログラミングロボットを用いたプログラミング思考育成型協調学習」，ロボティックス・メカトロニクス講演会報告，2A2-I09，p4 (2025)