

## 1. システム名

「draw the world」

## 2. 外部仕様

### 2.1. システムの狙いとコンセプト

( ユーザにとって、社会にとって、開発チームにとって、と言った観点から )

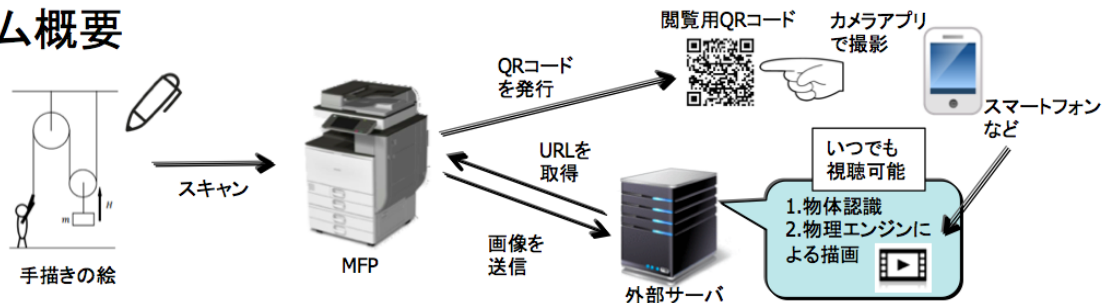
#### ◆ 複合機を使って、手描きの絵から物理シミュレーションを生成するシステム

システムの狙い：MFPを用いた新しい学習支援ツール

コンセプト：物理法則の定性的な理解をMFPとICTで簡単に実現

ピタゴラススイッチのような複雑なシステム（物理現象モデル）を紙面に手描きして、複合機に読み取らせます。スキャンした画像データは用意した外部サーバに送信します。得られた画像データに対して、OpenCVを用いた物体認識により特徴的な図形の検出を行います。検出した図形を物理エンジンでシミュレートさせ、WEB上に動画として描画します。動画のURLからブラウザで視聴でき、またMFPでURLのQRコードを発行し、スマートフォンのカメラアプリ等から簡単にアクセスすることも可能です。学習支援ツールとして様々な面で活用できることが期待されます。

## システム概要



### 2.2. 解決したい課題と提供する価値

#### ◆ 感覚的には理解しにくい物理法則の学習支援

-バネを取り付けたおもりが天井からつるすとどのような挙動を示すのか・・・

-教科書で物理法則を学んでも、実際にどんなふうに動くのか・・・

物理の学習において定性的な理解は培うには、実験による学習が効果的です。しかし自学自習において、気になる度にその実験を行うことはなかなか難しいのが現状であり、結局曖昧なイメージのまま捉えてしまいがちです。

今回提案するシステムでは、そのような「感覚的には理解しにくい物理法則を、実験を通して定性的に理解することは個人で難しい」という問題に対してICTを活用し解決します。

MFP（多機能プリンタ）は、手描きの絵と物理シミュレーションの橋渡し役としての機能を担っています。

## 2.3. ユーザと利用シナリオ

### ①幼稚園向けのお手軽ピタゴラ装置

お絵かきをより高度なものへ・・・

「手描きで物理実験シミュレーター（仮）」を使えば、自分の描いたお絵かきを物理法則に従った挙動で動かすことができます。手軽にアニメーションを作成できるとともに感覚的に物理の学習ができます。ピタゴラ装置のお手軽版として「考える力」を養うことのできる教材が、幼稚園にあるMFPから簡単に利用できます。



図 1.幼稚園での利用イメージ

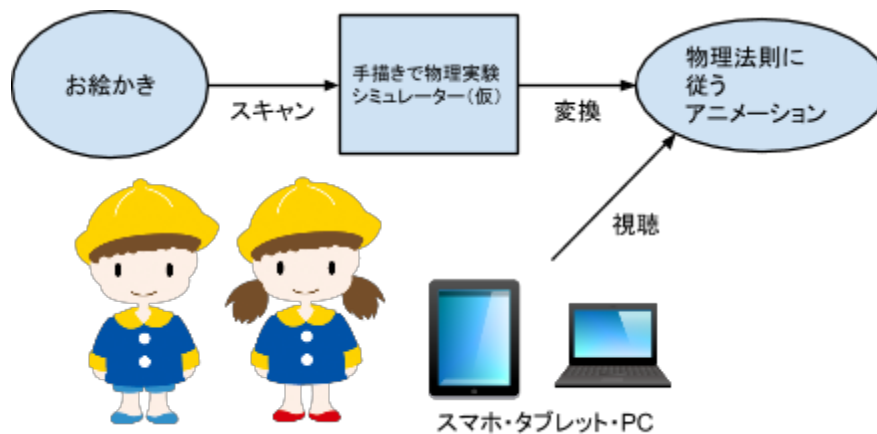


図 2.システム使用のフローチャート①

### ②高校生が物理法則を感覚的に理解する手助け

物理法則の定性的な理解のために・・・

「手描きで物理実験シミュレーター（仮）」を使えば、高校生が物理の問題の実際の挙動を動画で確認することができます。自由落下や剛体の衝突、ばねマスモデルなど、物理公式のイメージだけでは分かりにくいものの理解の手助けとなります。

物理現象モデルの閲覧サイトのURLのQRコードが発行されるので、ノートに添付し、スマホで簡単に確認することも可能です。



図3.物理の自習中での利用イメージ

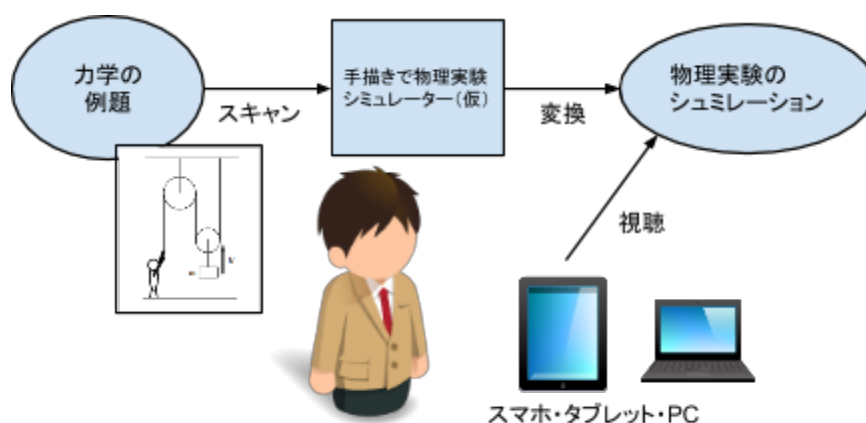


図4.システム使用のフローチャート②

## 2.4. シナリオの各場面でのシステム機能とユーザインタフェース

### ①幼稚園向けのお手軽ピタゴラ装置

#### システム機能

##### MFP側

- ・ スキャン機能 ( 画像の読み込み )
- ・ QRコードプリント機能
- ・ 視聴機能

##### ユーザインタフェース

### ②高校生が物理法則を感覚的に理解する手助け

#### システム機能

- ・
- ・

##### ユーザインタフェース

#### 画面遷移図をつける

## 2.5. ビジネスモデル

- 2.6. 非機能要件
- 2.7. 用語集
- 3. 内部仕様
  - 3.1. 技術的な要件と制約
    - ・ 手描きで記述された記号を認識し座標や大きさを得る
    - ・ 重力、弾性力、速度、摩擦などを考慮した物理演算
    - ・ ブラウザ上での描画処理

### 3.2. 利用技術および基本設計方針とその採用理由 サーバ

Apache + Tomcat

画像認識のためのライブラリ

OpenCV + JAVA

採用理由：画像認識のための機能が豊富に用意されていて、様々な言語で利用できるため本システムでは画像認識のためのライブラリとしてOpenCVを採用した。

また開発言語は過去に開発経験があるJAVAを選択した。

シュミレーションの描画

HTML5 + WebGL

採用理由:javaやFlashを再生できないブラウザからでも閲覧できることを考慮した

### 3.3. 開発環境と開発プロセスの特徴

開発言語

-java

-Android

アプリケーションサーバ

-Tomcat

データベース

-Mysql

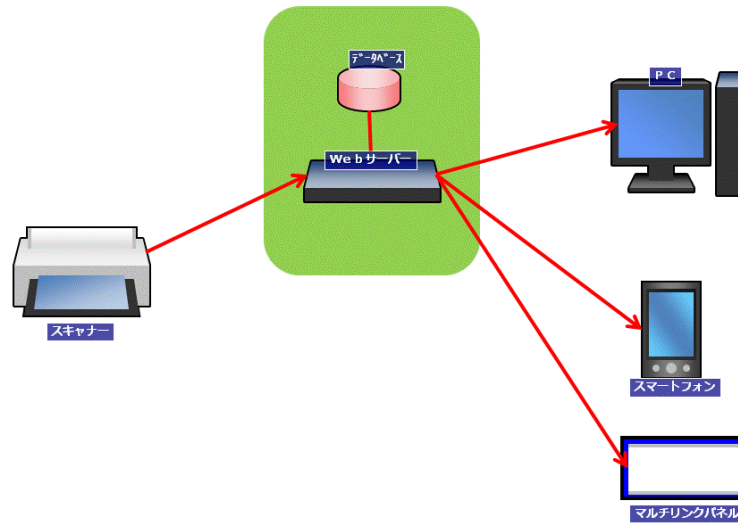
バージョン管理

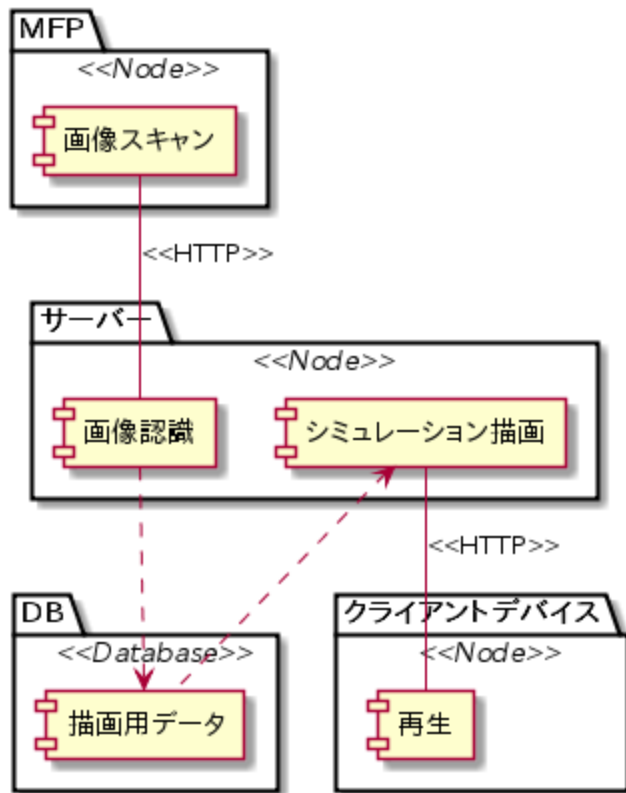
-GitHub

### 3.4. アーキテクチャの図と説明

下の図がシステム環境図である。

MFPから画像データを送信し、ブラウザを使用できるデバイスからシミュレーションを閲覧する

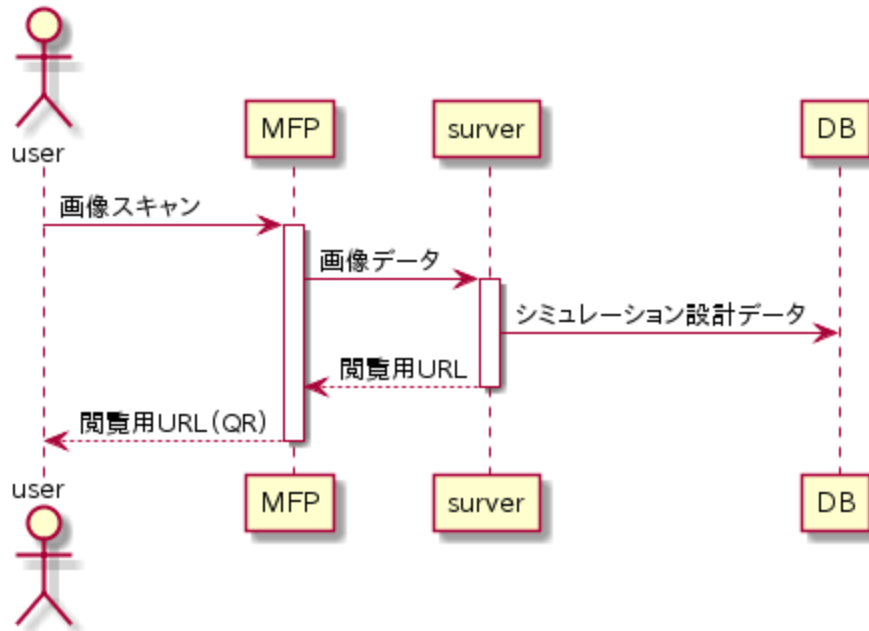




以下のシーケンス図は手描きの画像をスキャンしてから閲覧用のURLが生成されるまでの流れである。MFPは画像データの送信、閲覧用URLの受信、URLの表示、印刷を行う。

サーバでは画像データからオブジェクトの大きさや座標を取得し設計データとしてデータベースに保存する。

そして、保存された設計データと関連付けられた閲覧用URLを生成し返す。





下の図は閲覧用のURLにアクセスして物理シミュレーションが実行されるまでの流れである

サーバに閲覧用URLのリクエストが送られるとデータベースから設計データを取得し物理シミュレーションを生成し閲覧用ページをブラウザに返す。

