

システム構築ツール RTSytemBuilder
操作マニュアル

名城大学メカトロニクス工学科
ロボットシステムデザイン研究室
2023 年 12 月 14 日

目次

1. はじめに	2
1.1 目的	2
1.2 本書を読むにあたって	2
1.3 動作環境	2
 2. 本ツール利用のための準備	 2
2.1 wasanbon を利用するための環境構築	2
2.2 wasanbon binder の作成	2
2.3 RTSystemBuilder を利用するための環境構築	3
 3. RTSystemBuilder の操作方法	 5
3.1 ホーム画面	5
3.2 設定画面	5
3.3 コンポーネント情報管理画面	6
3.4 システム構築画面	8
3.5 wasanbon 連携画面	11

1. はじめに

1.1 目的

近年、サービスロボットの導入が加速している。複雑化するロボットシステム開発の効率向上に向けて、ロボット用ミドルウェアが広く用いられるようになってきている。しかしながら、ミドルウェアで動作するロボットシステムの管理・運用は開発者の手に委ねられ、簡便な方法は存在しなかった。この課題に対して、OpenRTM で動作する RT コンポーネントや、それを基盤として構築される RT システムを管理・運用する仕組みとして、wasanbon が提案されてきた。wasanbon は、登録されたりポジトリから RT システムや RT コンポーネント単体でのダウンロードや各 RTC のビルド、RT システムの運用が可能なフレームワークである。

しかしながら、wasanbon に対しての新規 RT システムの登録は手間がかかるため、エンドユーザが新規システムを登録しやすくする仕組みが必要である。そこで、wasanbon で利用可能な RT システムを登録するツールとして RTSysBuilder を開発した。

1.2 本書を読むにあたって

本書は wasanbon に関する基礎知識を有した利用者を対象としている。

1.3 動作環境

本ツールの動作環境を以下に示す。

OS	Windows
----	---------

2. 本ツール利用のための準備

2.1 wasanbon を利用するための環境構築

以下サイトに順して行う。

http://wasanbon.org/?page_id=1509

2.2 wasanbon binder の作成

wasanbon を使用するに当たって、wasanbon binder の作成が必要になる。wasanbon binder は、RTC や RTS のリポジトリ情報の集まりである。図 1 に示すようにリポジトリの名前や URL、使用可能プラットフォームなどが書かれている。

wasanbon binder の作成は以下サイトに順して行う。

http://wasanbon.org/?page_id=1727

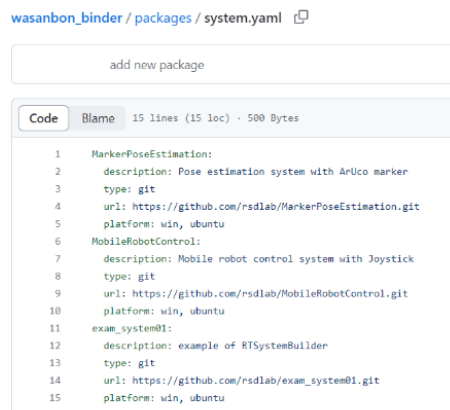


図 1 wasanbon binder

2.3 RTSystemBuider を利用するための環境構築

2.3.1 インストール

下記リポジトリのファイルをインストールする。

<https://github.com/RTSystemDesigner/RTSystemBuilder>

2.3.1 サーバー側の設定

・ データベースの準備

① データベースのインストール

リレーショナルデータベース管理システム（RDBMS）として、PostgreSQL を使用する。インストーラは、以下サイトなどからダウンロードすることができる。

<https://www.enterprisedb.com/downloads/postgres-postgresql-downloads>

（※Windows 版 Ver. 13.5 で動作確認を行っている。）

インストーラを使用してインストールを行う場合には、基本的には全てデフォルト設定のままでインストールすれば良い。途中で管理者パスワードを設定する部分があるが、これは任意のパスワードを設定する。

インストール後、コマンドプロンプトなどで、

```
<PostgreSQL インストール先ディレクトリ>/bin/psql.exe -version
```

を実行し、バージョン番号が表示されることを確認する。

② データベースの構築

「スタート」メニューから表示されるプログラム一覧から、「PostgreSQL X.X」→「SQL Shell (psql)」を選択して起動する。「Server」や「Database」など、幾つかの情報を入力されるように求められるが、全てデフォルト値のままでよい。パスワードについては、インストールした際に設定したパスワードを入力する。ユーザ名は任意のものでよい。

起動した SQL Shell で以下の SQL コマンドを順番に実行する。

ユーザ（ロール）の作成

```
CREATE ROLE <ユーザ名> LOGIN PASSWORD '<パスワード>';
```

データベースの作成

```
CREATE DATABASE componentdb;
```

権限の付与

```
GRANT ALL PRIVILEGES ON ALL TABLES IN SCHEMA public TO <ユーザ名>;
```

一度、SQL Shell を終了した後、再度、SQL Shell を起動する。以下を指定する。

Database : componentdb

ユーザ名 : <ユーザ名>

SQL Shell 内に Server/doc/TranTables.sql の中身を貼り付けて各テーブルを作成する。

また、テーブルを作成する際には、一度に全てのテーブル作成を行うのではなく、一つずつ作成し、エラーが発生していないか確認する。

・Python 実行用仮想環境の構築

まず Python 本体をインストールする。（※Windows 版 Ver3.6.8 を使用して動作確認を行っている。）

サーバ実行用 Python 仮想環境を保持するディレクトリを作成し、コマンドプロンプトなどで作成したディレクトリに移動する。

以下のコマンドを実行し、仮想環境を作成する。

```
<Python インストールパス>/python -m venv <仮想環境名>
```

コマンドプロンプトで、以下のコマンドを実行し、作成した仮想環境に入る。

```
<仮想環境構築ディレクトリ>/Scripts/activate.bat
```

作成した仮想環境ディレクトリに、Server/doc/requirement.txt をコピーした後、以下のコマンドで仮想環境の構築を行う。

```
pip install -r requirements.txt
```

・サーバ実行用パッチファイルの修正

DB サーバ起動用パッチファイル (Server/ComponentDatabaseService/start_DB_server.bat) の中身を構築した環境に応じて修正する。修正する箇所は、「APP_ROOT」と「APP_Python」の2カ所である。

```
set APP_ROOT=" <DB サーバのソースを配置したディレクトリ名>"
```

```
set APP_Python=<Python 仮想環境を構築したディレクトリ名>
```

上記を修正した後、DB サーバ起動用パッチファイルを実行する。図2に示すような画面が表示されればDBサーバが正常に起動している。

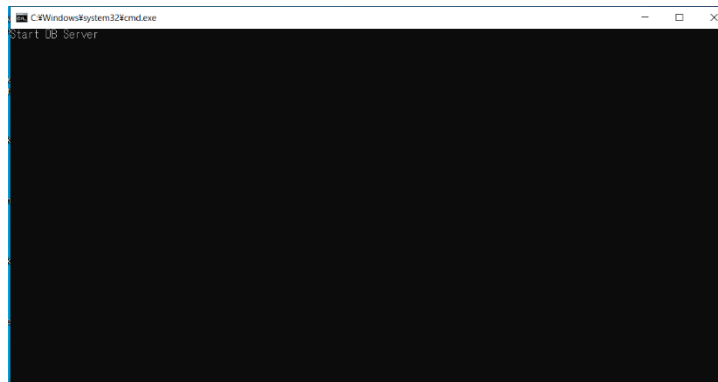


図 2 start_DB_server.bat

2.3.2 起動方法

① サーバの立ち上げ

RTSystembuilder > Server > CopmonentDatabaseService フォルダ内の「start_DB_server.bat」を起動する。

② クライアントの立ち上げ

RTSystemBuilder > Client > RTSystemBuilder フォルダ内の「RTSystemBuilder.sln」を開き、表示された Visual Studio の画面上で「デバッグなしで開始」(Ctrl+F5) をクリックする。(タスクバーに出てきたアイコンをピン留めしておくといよい。)

3. RTSystemBuilder (クライアント側) の操作方法

3.1 ホーム画面

図 3 に示すように設定、コンポーネント情報管理、システム構築、wasanbon 連携、終了のボタンで構成される。終了ボタンを押すと、RTSystemBuilder の画面が終了する。それ以外のボタンを押すと、それぞれの画面に遷移する。

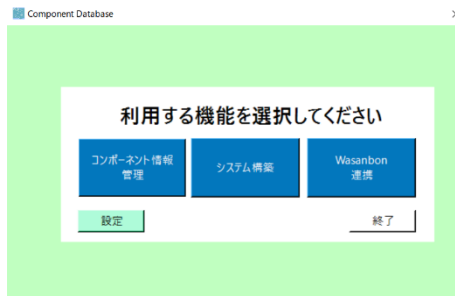


図 3 ホーム画面

3.2 設定画面

図 4 に設定画面を示す。

3.2.1 データベース設定

- IP アドレス

サーバー側ソフトを実行している PC の IP アドレスを設定する。クライアントアプリと同じ PC 上で起動している場合には、「localhost」と設定する。

- ポート番号

データベース管理システム（PostgreSQL）が使用しているポート番号を設定する。PostgreSQL のインストールを行う際に、ポート番号を変更していなければ、デフォルト設定の「50201」を設定する。

3.2.2 GitHub 設定

- ユーザ id

クライアントアプリを使用するユーザの GitHub アカウント名を設定する。

- アクセストークン

「ユーザ id」で設定した GitHub アカウントで取得したアクセストークンを設定する。

- システム構築時ベースリポジトリ URI

構築したシステムを登録するユーザ（組織）のリポジトリの URI を設定する。

- wasanbon binder リポジトリ URI

構築したシステムを wasanbon で使えるようにするために、使用する wasanbon binder リポジトリ URI を設定する。

Component Database

設定情報

データベース設定

IPアドレス: localhost ポート番号: 50201

GitHub設定

ユーザid:

アクセストークン:

システム構築時ベースリポジトリURI

追加 更新 削除

URI

https://github.com/rsdlab

Wasanbon binderリポジトリURI

追加 更新 削除

URI

https://github.com/rsdlab/wasanbon_binder

OK キャンセル

図 4 設定画面

3.3 コンポーネント情報管理画面

この画面では、RTSystemBuilder でシステムを構築するために必要な RTC を追加し、その RTC の情報を確認することができる。図 5 にコンポーネント情報管理画面を示す。



図 5 コンポーネント情報管理画面

3.3.1 RTC の追加

- ① 「追加」ボタンを押す。
- ② 図 6 に示すような画面が表示される。この画面で登録名、プロフィール、リポジトリ URI、コメントを入力する。最低限必要な最低限必要な入力情報としては、登録名とリポジトリ URI である。登録名は、RTSystemBuilder で使用される RTC の名前を入力する。リポジトリ URI は、追加する RTC の GitHub 上のリポジトリ URI を入力する。注意として、ここで入力する RTC は設定画面で設定したベースリポジトリ URI 上に格納されている必要がある。

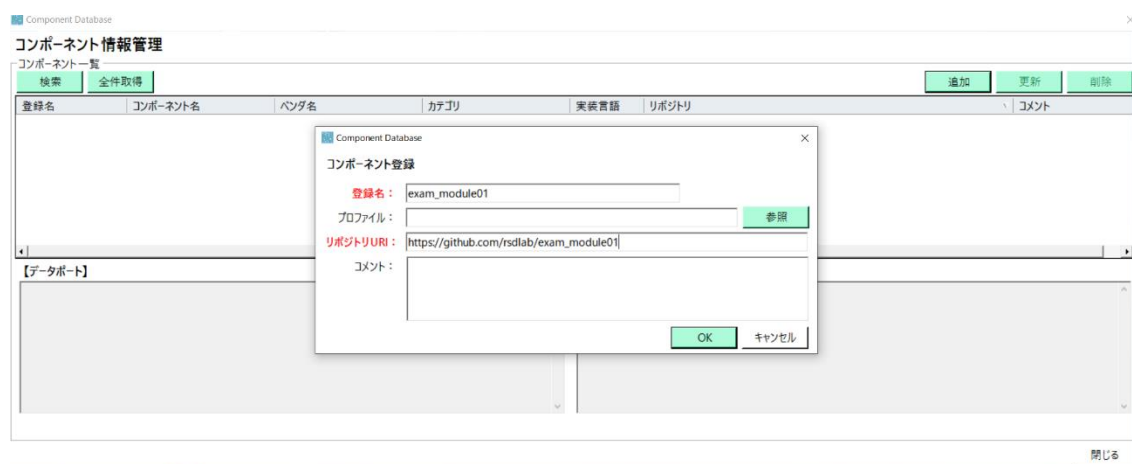


図 6 コンポーネント登録画面

- ③ 「OK」ボタンを押すと、RTSystemBuilder に RTC が登録される。

3.3.2 RTC 情報の確認

追加された RTC は、図 7 に示すように RTC のコンポーネント名やデータポートなど、様々な情報を確認できる。また、追加した全ての RTC を確認したい場合は、全件取得ボタンを押

すことで確認できる。更に検索ボタンでは、図 8 に示すように検索したい RTC の情報を入力することで、特定条件の RTC を検索できる。

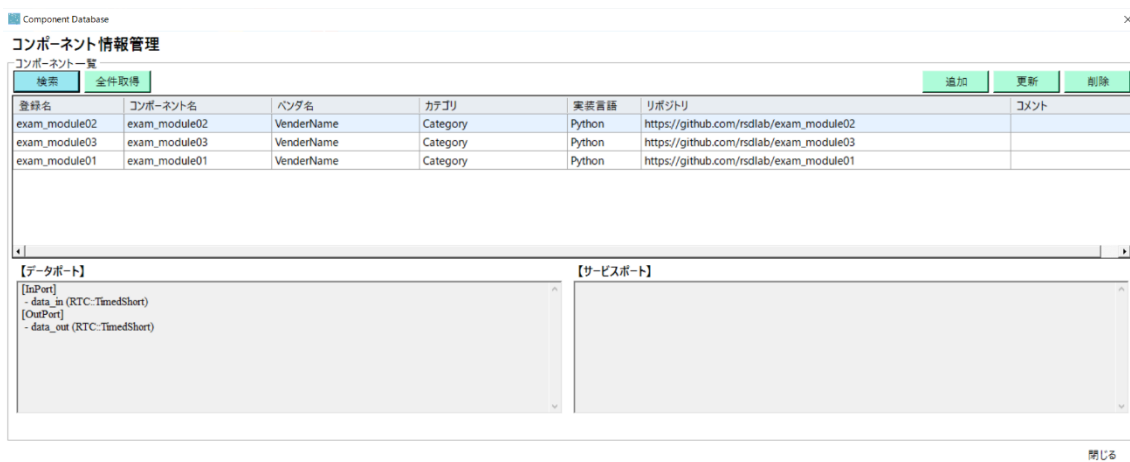


図 7 RTC の情報確認

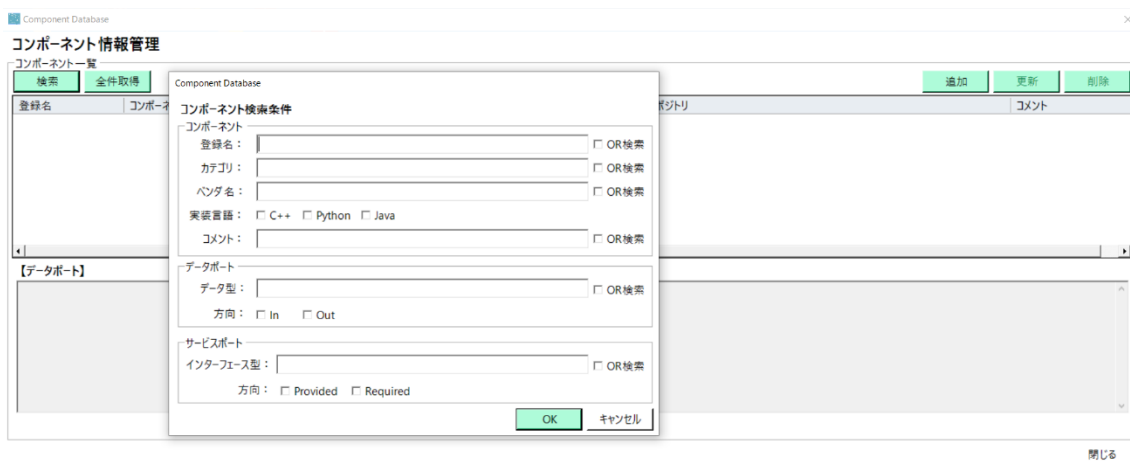


図 8 コンポーネント検索

3.3.3 その他

更新ボタンでは、選択されている RTC の登録名とコメントを変更できる。削除ボタンでは、選択されている RTC を RTSystemBuilder から削除できる。

3.4 システム構築画面

この画面では、3.3 で追加した RTC を複数選択して一つの RT システムのパッケージとして GitHub にアップロードすることができる。図 9 にシステム構築画面を示す。

3.4.1 新しいシステムの構築

① 「ベースリポジトリ URI」, 「システム名」に新しい RT システムのリポジトリの位置と名前を設定する。ベースリポジトリ URI は設定画面で設定した URI が使用できる。そして、「新規作成」ボタンを押す。

Component Database

システム構築

システム構成

ベースリポジトリ URI: システム名:

コンポーネント名	概要	URL

コンポーネント一覧

登録名	コンポーネント名	ベンダ名	カテゴリ	実装言語	リポジトリ	コメント

【データポート】

【サービスポート】

終了

図 9 システム構築画面

② 「コンポーネント一覧」で 3.3.2 と同様に、「検索」か「全件取得」ボタンを押して RTC を表示する。図 10 に示すように RTC を指定し、「対象システムに登録」ボタンを押すことで、システムに RTC を追加する。（「システム構成」に表示される）

Component Database

システム構築

システム構成

ベースリポジトリ URI: システム名:

コンポーネント名	概要	URL
exam_module01		https://github.com/rsdlab/exam_module01.git

コンポーネント一覧

登録名	コンポーネント名	ベンダ名	カテゴリ	実装言語	リポジトリ	コメント
exam_module02	exam_module02	VenderName	Category	Python	https://github.com/rsdlab/exam_module02	
exam_module03	exam_module03	VenderName	Category	Python	https://github.com/rsdlab/exam_module03	
exam_module01	exam_module01	VenderName	Category	Python	https://github.com/rsdlab/exam_module01	

【データポート】

【OutPort】
- data_out (RTC:TimedShort)

【サービスポート】

終了

図 10 システムに RTC が追加された画面

③ 必要な RTC を全て選択した後、「情報設定」ボタンを押す。（「情報クリア」ボタンを押すと、選択した RTC が「システム構成」から消え、3.4.1 ①の状態に戻る）図 11 に示すような画面が表示される。この画面で、「システム説明」（README.md）、「ネームサーバー」、「コミットメッセージ」を設定することでリポジトリに反映される。そして、「OK」ボタンを押すことで、システムがリポジトリに登録される。

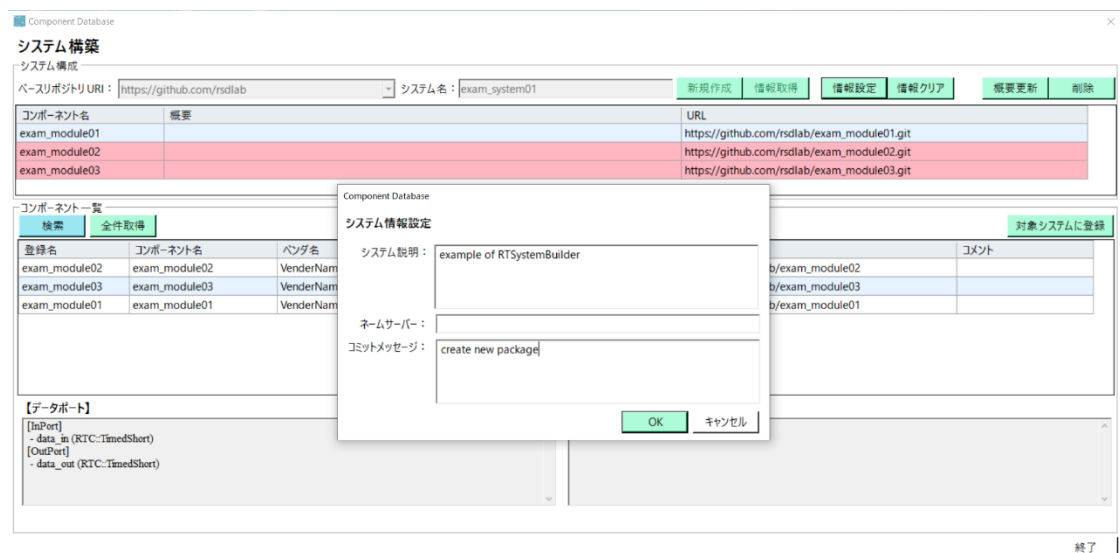


図 11 システム情報設定画面

3.4.2 元のシステムの RTC の編集

- ① 「ベースリポジトリ URI」, 「システム名」に編集したい RT システムのリポジトリの位置と名前を設定する. そして, 「情報取得」 ボタンを押す.
- ② 図 12 に示すように, 「システム構成」に対象システムに含まれる RTC が表示される. 対象システムに RTC を追加したい場合, 3.4.1 ②と同様に行う. RTC を削除する場合, RTC を指定し, 「削除」 ボタンを押すことで削除できる. また, 「概要更新」 ボタンは, 選択した RTC の概要を書き換えることができる.

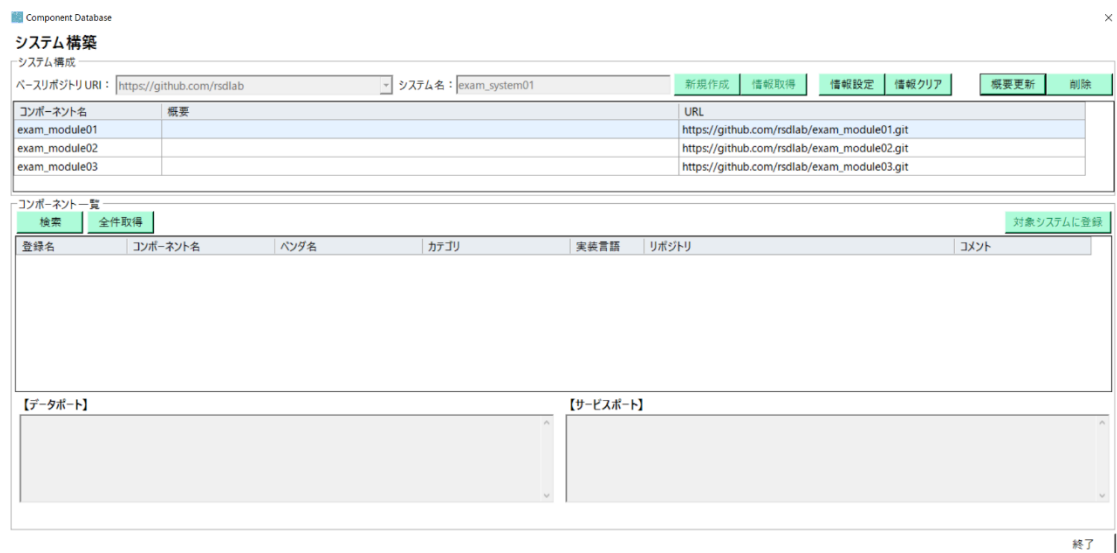


図 12 システムの RTC が表示された画面

- ③ 編集が完了した後, 3.4.1 ③と同様に行うことで, 編集されたシステムがリポジトリに登録される.

3.5 wasanbon 連携画面

この画面では、リポジトリ上の RT システムを wasanbon binder に追加することができ、wasanbon でそのシステムを使うことができるようになる。図 13 に wasanbon 連携画面を示す。

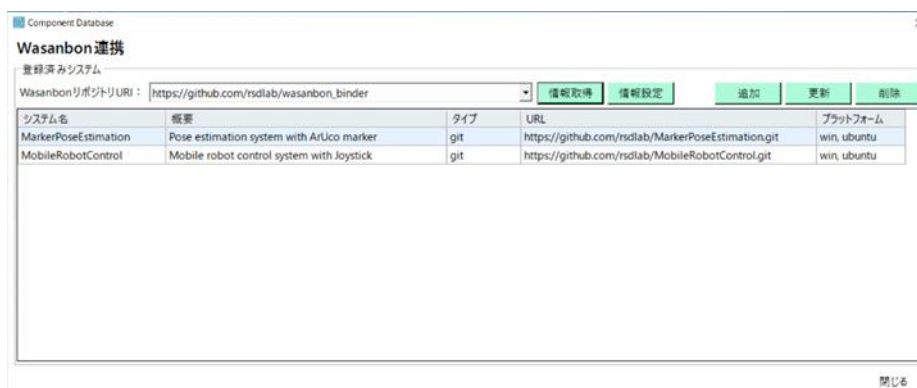


図 13 wasanbon 連携画面

3.5.1 wasanbon binder への追加

- ① 「wasanbon リポジトリ URI」で、システムを追加する binder のリポジトリ URI を選択する。設定画面で、設定したリポジトリ URI を選択できる。「情報取得」ボタンを押した後、「追加」ボタンを押す。
- ② 図 14 に示すような画面が表示される。この画面でシステムの名前、概要、リポジトリ URI、使用可能プラットフォームを設定する。これらの情報は wasanbon binder に反映される。そして、「OK」ボタンを押す。

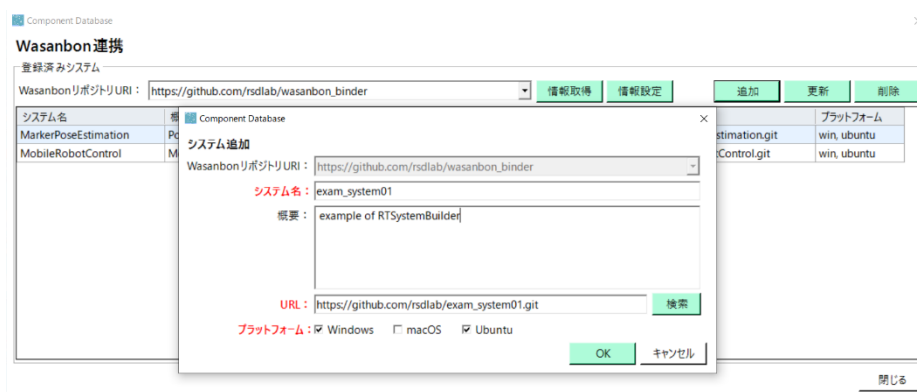


図 14 システム追加画面

- ③ 図 15 に示すように「登録済みシステム」に設定したシステムの情報が表示される。（この時点では、RTSystemBuilder に追加されただけで、wasanbon binder には追加されていない） binder に追加するシステムを指定し、「情報設定」ボタンを押す。

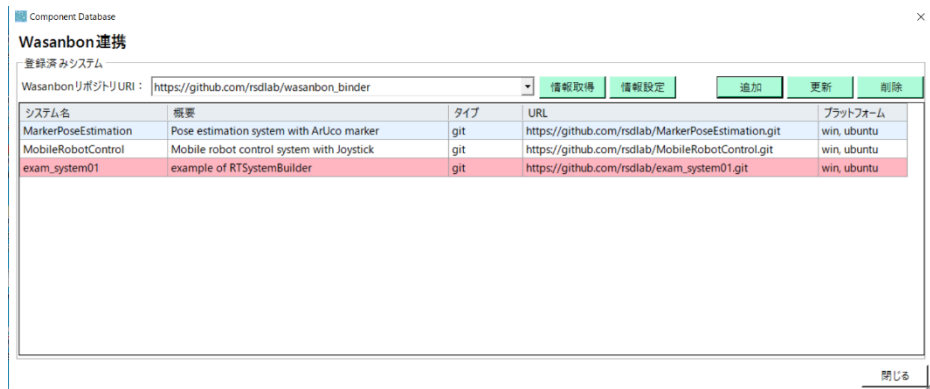


図 15 システム情報の表示

④ 図 16 に示すような画面が表示される。「コミットメッセージ」を入力し、「OK」ボタンを押すことで、図 17 に示すように binder に追加される。

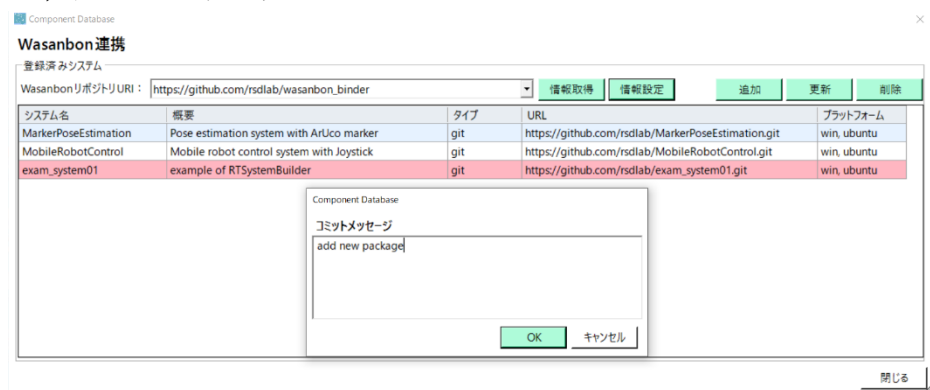


図 16 コミットメッセージ入力画面

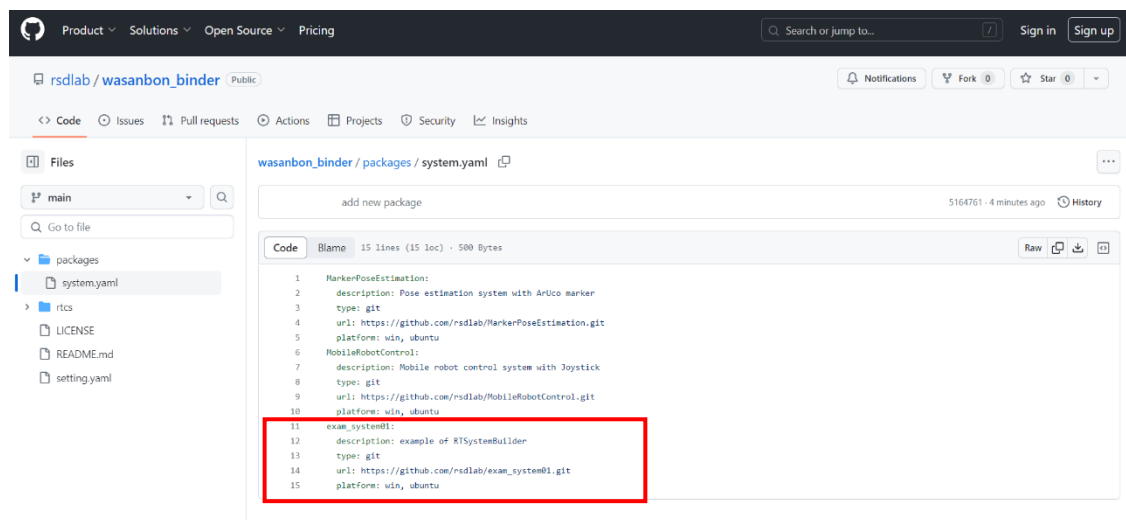


図 17 新しいシステムが追加 wasanbon binder

3.5.2 wasanbon binder の編集

- ① 「wasanbon リポジトリ URI」で、編集したい binder のリポジトリ URI を選択する。設定画面で、設定したリポジトリ URI を選択できる。「情報取得」ボタンを押すと、binder に記述されているシステムの情報が表示される。
- ② システムを削除する場合、対象のシステムが選択された状態で「削除」ボタンを押すことで、削除される。システムの情報を編集する場合、対象のシステムが選択された状態で「更新」ボタンを押すことで、システム更新画面で概要とプラットフォームの情報を変更できる。
- ③ 編集が完了した後、3.5.1 ③、④と同様に行うことで、binder の編集が反映される。