

Konohascript のソースコードから UML 準拠の構造図(クラス図、コンポーネント図、オブジェクト図、パッケージ図)の生成

導入：

konohascript で開発を行う時、事前に UML に則ったモデリングを行わずに開発する場合がある。それには

- ・ モデリングをする時間と手間を省いて素早く開発したい場合
- ・ 上流工程の段階で要求と設計が定まっていない状況下でコーディングしなくてはならない場合
- ・ モデリング方法を知らない場合

等といったケースがある。

チームで開発を行う場合等、その成果物を再利用したりリファクタリングする必要がある時、(構造図がなければ)他のプログラマーがその構造を理解するにはソースコードを読み、解析する必要がある。また、クラスやメソッド、生成されたオブジェクト間の関係等、再利用時に知っておかなければならない内容も多い。その際、

- ・ その解析に時間が掛かってしまい、成果物の再利用による開発サイクルの短縮の恩恵が受けにくい
- ・ 成果物が要求、設計と見合ったものなのかが測り辛い

等の問題点が生じる。

本論文では、このような問題点を解決するため、UML 等によるモデリングを行わずに開発した場合でも構造図を自動で生成出来るようなツールについて提案する。

(また、

- ・ 素早く簡単にコーディングが出来る konohascript の良さを伸ばし、初心者の方でも、作成したソースコードの振り返りを行えるような用途としても使えるものを目指す。
- ・ チームで開発する際に、全体のシステムの要求・設計と見合った実装なのかを確認する際にも使えるものを目指す。

)

背景：

konohascript のコーディング

→konohascript は静的言語と動的言語のハイブリッドなオブジェクト指向スクリプト言語であり、静的言語が持つ型安全性や高速性を保持したまま、動的言語のもつ簡単なコーディングを行うことが出来る。

UML

→省略

アジャイル開発についても

→省略

問題：

konohascript はオブジェクト指向言語であるため、UML のモデリング系とは相性が良いものの、UML 等のモデリング言語・記述法と併せてコーディングを行うことを支援するツールはまだ持ち合わせていない。

結果として、UML を用いないことによって、プログラムのソースコード、仕組みを他のプログラマーが理解することが難しいといった問題が生じ、

- ・再利用、リファクタリングの難化
- ・要求、設計との確認の難化

目的：

konohascript のソースコードから UML に準拠した構図（クラス図・コンポーネント図・オブジェクト図・パッケージ図）を生成できるツールを作成し、問題解決を行う。

提案：