

論文番号 fm2016-01(仮)

LTIに準拠したネットワーク 自己学習機能の提案と実装

15RD093 菅原 良太, 15RD150 沼田 悠貴

指導: 藤本 衡 准教授

提出日: 2018 年 12 月 25 日

概 要

近年、様々な種類の LMS(学習管理システム:Learning Management System) が存在し、また多くの企業や学習機関が LMS を用いることで e ラーニングを活用している。しかし、LMS は学習者の成績や進捗の管理などを行うのが主な役割である。具体的には教材や資料の配信、共有情報技術者にとってネットワーク技術の理解は必要不可欠であるが、ネットワーク技術は知識として学習しても実際にネットワークを構築しなければ

目 次

概要	2
1 はじめに	4
2 LTI	5
3 システム概要	5
3.1 システム	5
3.2 Ruby on Rails	7
3.3 UI について	8
4 まとめと課題	11

1 はじめに

2 LTI

3 システム概要

3.1 システム

本研究では、プラグインとして LMS 上に新しい機能を提供するのではなく、LTI に準拠した Web アプリケーションを用いて、異なる LMS で同様の機能が提供でき、Web アプリケーション側での操作に対し LMS 側に特定の点数を返すことを目的とした。そこで、異なる仮想マシン上にそれぞれ LMS である Canvas、Moodle と、独立した Web アプリケーションとしてネットワークシミュレータを導入した。また、ネットワークシミュレータは Ruby on Rails を用いて実装した。これらは図 1 で表しているようにネットワークシミュレータは実際には独立した Web アプリケーションであるが、あたかも LMS 側にプラグインとして導入されているように機能を提供する。

また、Web アプリケーション側は LMS に呼び出された際、独立した Web アプリケーションとしてネットワークシミュレータとしての機能を提供する。この機能にはネットワークを自由に構成し、機器情報を設定することのできる自由描画モードと、予め問題として構成されたネットワークに正しい機器情報を追加することで正しいネットワークの作成を目指す問題演習モードが有る。問題演習モードでの正誤によって得られた得点を LMS 側に返すことで LMS での学習者の評価を行う。

魚本、大須賀、中村 (2018) らの制作したネットワークシミュレータは Moodle の独自プラグインとしてネットワークシミュレータを実装している。これらの構成

図 1: 仮想マシンの構成

を図 2 として示す。

クライアントサイドである独自プラグインとしてのシミュレータ部分は HTML と JavaScript で、Moodle のプラグインとしての設定の部分は PHP で、シミュレータで作成されたネットワークの構成の正誤の判定プログラムは Ruby でそれぞれ記述されている。これは、様々なシステムを使用しているため、複数のシステム間でデータのなどの連携を行わなければならない、安定性にかけていた。

そこで、本研究ではすべてのシステムを Ruby on Rails の中で実装した。MVC アーキテクチャに基づいて設計することにより、魚本、大須賀、中村 (2018) らの図 2 で構成されたシステムをすべて Ruby on Rails 内で実現した。これにより、複数のシステム間でのデータの送受信などを行う必要性がなくなり、システムとしての安定性を実現した。

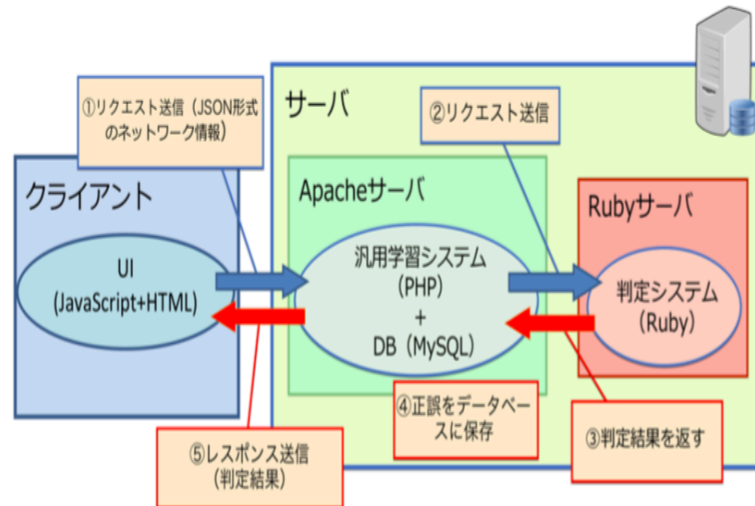


図 2: 魚本ら (2018) のシステム構成 (差し替え案件)

3.2 Ruby on Rails

本研究で提案したネットワークシミュレータは、Ruby on Rails を用いて実装されている。Ruby on Rails とは、Ruby で構築された、Web アプリケーションを開発するためのフレームワークである。特徴として MVC アーキテクチャの採用や設定より規約という設計哲学などが挙げられる。

MVC とは「Model」, 「View」, 「Controller」の頭文字であり、MVC アーキテクチャとはアプリケーションの構成が以下の図 3 のように分類することに由来している。

ここで「Model」とはデータベースに収めたデータやそのデータのルールなどを表す。「View」は、アプリケーションの UI の部分を指す。HTML や CSS を用いて配置やデザインを決定する。「Controller」は View と Model の間を取り持つ部

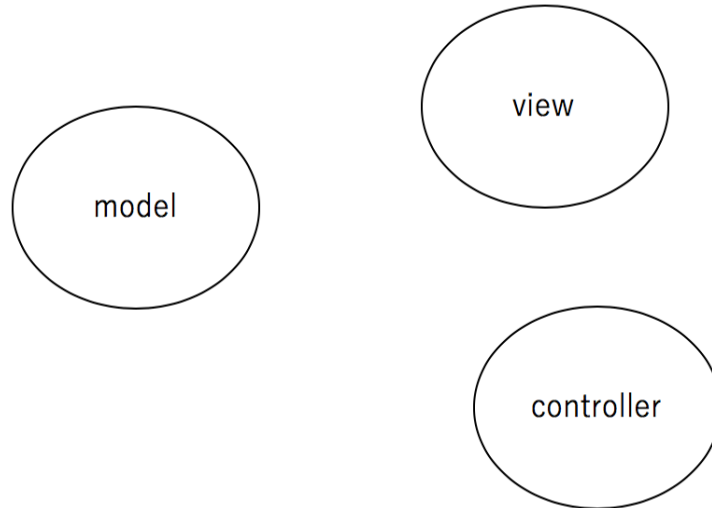


図 3: MVC アーキテクチャ

分である。Model と View の間でデータの受け渡しなどを行う。

Ruby on Rails ではこれら「Model」,「View」,「Controller」が機能として独立しているため、それぞれの部分の開発を効率良く行うことができ、仕様変更や新たな機能の追加が容易に行える。また、「View」として UI 部分が独立しているため UI の変更も容易である。

3.3 UI について

UI の基本的な部分は、魚本、大須賀、中村 (2018) らの制作したネットワークシミュレータを採用した。この概要を図 4 に示す。

図 4 のネットワークシミュレータは、実際にネットワークに関する学習を終えた学生に対しアンケートを行い、9 割以上の学生がデザインについて見やすいと答えていた。これにより図 4 のネットワークシミュレータの UI は変更する必要性が

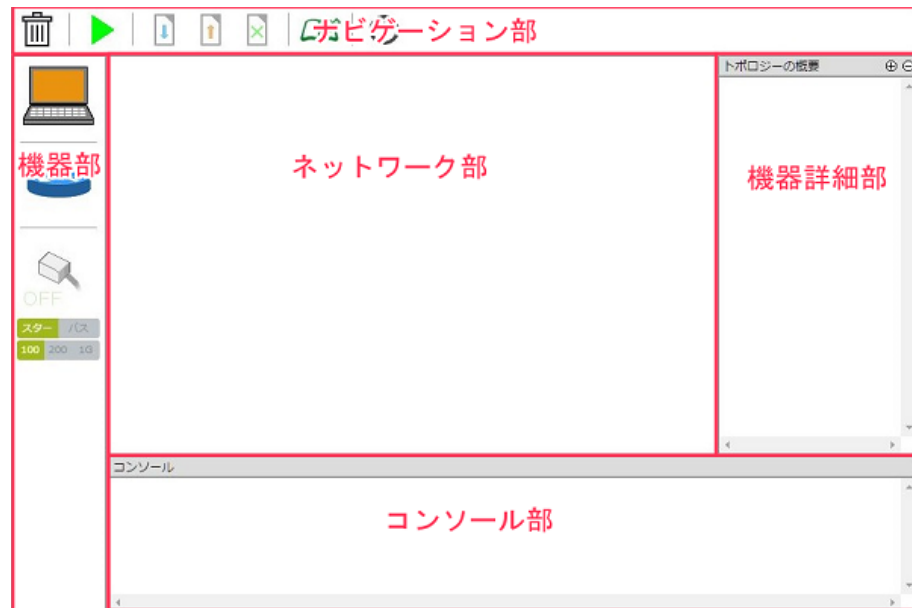


図 4: ネットワークシミュレータ UI(差し替えます)

ないと判断した。 図 4 は 5 つの部分に分けられており、機器部、ナビゲーション部、ネットワーク部、機器詳細部、コンソール部となっている。また、図 4 では自由描画モードと問題演習モードの 2 つのモードが用意されている。自由描画モードの際、ナビゲーション部ではそれぞれのアイコンをクリックすることでモードの変更、構築したネットワークの正誤の判定、それぞれの機器の詳細情報の確認、すべての要素の削除を行うことができる。問題演習モードの際は、これに加え練習問題一覧の表示、現在の状況のセーブ、セーブした状態のロード、問題演習モードの終了を行うことができる。機器部では自由描画モードの際に PC やルータをネットワーク部にドロップし、LAN をそれぞれつなげることで自由にネットワークを構築することができる。ネットワーク部では構築されているネットワークのそれぞれの機器に必要な情報を追加する事ができる。これによって正しいネット

ワークを構築していくことが本ネットワークシミュレータの目的である。機器詳細部はネットワーク部で追加されたそれぞれの機器の情報を確認する部分である。コンソール部は不可能な操作やエラーなどの不具合が起こった場合などにそれぞれの理由や結果などをコンソールとして入力される部分である。

これらの機能により、学習者はPCを複数用意し、実際にネットワークを構築することなくネットワークシミュレータ上で擬似的にネットワークの構築を行うことができる。これにより、知識として学習しただけでは分かりづらいネットワークの分野を、視覚的に構築することで実際のネットワークの構成などを理解する助けとなる。

4 まとめと課題

謝辞

本研究を引き継ぐ際に様々な情報を教えていただいた魚本悠太氏、大須賀旭氏、中村優氏に感謝したいと思います。また、本研究の御指導や実験への協力をして下さいました藤本准教授とシステム評価研究室の皆様に対し、ここに心より深く御礼申し上げます。

参考文献

- [1] 魚本裕太, 大須賀旭, 中村優, ”応答性を向上した IP ネットワーク個人学習システム”, 2018.
- [2] aa