

知的照明システムにおける WebAPI の構築と実用

水上 雅博

Masahiro MIZUKAMI

1 はじめに

近年、オフィスにおけるオフィスワーカーの快適性および知的生産性の向上に注目が集まっている。オフィス環境を改善することにより、知的生産性が向上すると報告されている¹⁾。我々は、オフィスにおける光環境に着目し、任意の場所に任意の照度を実現し、かつ省エネルギー性を実現できる照明（以後、知的照明）システムの開発を行っている。また、知的照明システムにおいて、ユーザが配線によらず照明の出力を自由に変更できるユーザインタフェース（User Interface：UI）の開発も盛んに行っている²⁾。その一方で、知的照明システムの UI 開発にはデータベース（Database：DB）の操作や照明機器への出力、センサからのデータ取得など様々な処理が必要とされ、知的照明システムの設計および処理について知識がなければ作ることができず、非常に高い学習、開発コストがかかっていることも問題となっている。本稿では、この知的照明システムの UI 開発における学習、開発コストの軽減を目的とした WebAPI（Application Programming Interface）を提案、構築し、低コストで知的照明システムの UI を構築方法について提案する。

2 WebAPI とは

WebAPI とは、Web サイトやプログラムなどの開発を効率的に行うための技術である。そもそも API とは”application programming interface”の略で、アプリケーションの開発者が、他のハードウェアやソフトウェアの提供している機能を利用するための手法である。これをインターネットのサーバ上に構築し、HTTP プロトコルでアクセス、利用できるようにしたものを WebAPI という。開発者にとって WebAPI は、高機能なコンテンツを、より短期間・低コストで開発できるという利点がある³⁾。

3 WebAPI の仕様

3.1 設計計画

WebAPI に実装するメソッドの内部処理は完全に隠蔽し、API に照明の出力を受け渡すだけで動作するようにする。これによって、DB 操作や照明への出力などの知的照明システムの内部設計に関する知識がなくとも、知的照明システムにおける照明操作を行うことが可能と

なり、学習、開発コストの軽減が見込まれる⁴⁾。また WebAPI の呼び出しはどのような開発環境においても可能にするために、一般的に普及した規格を利用する。

3.2 メソッドの隠蔽

内部処理を完全に隠蔽し、WebAPI へ任意のパラメータを受け渡すことで動作するようにする。その動作の流れを Fig. 1 に示す。

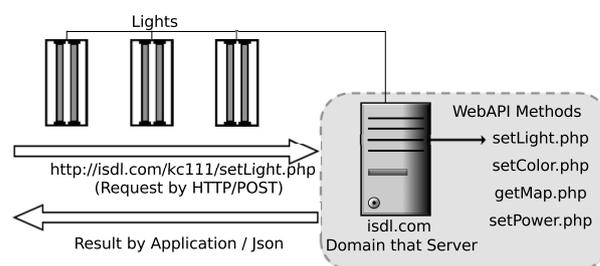


Fig. 1 処理の流れ

Fig. 1 の指定 URL におけるドメイン部分（Fig. 1 においては isdsl.com）が知的照明システムの導入されている部屋または建物を示しており、それ以降にメソッドを示すファイル名が続く。

3.3 通信プロトコルとデータ形式

WebAPI のメソッドは HTTP 通信の POST 形式でデータを引き渡すことで利用可能である。URL によって指定されたメソッドに POST 形式で引き渡されたデータはメソッド内で String 型から適切な変数型に還元され、それぞれ DB 操作や照明、照度センサを用いた処理をおこなう。処理された結果は Application/JSON (JavaScript Object Notation) 形式で返される。WebAPI を利用するプログラムは、WebAPI へのリクエストの作成、WebAPI へのリクエスト、WebAPI から返されるデータの取得、そして WebAPI から返されたデータの解析という流れで処理を行う。

3.4 サーバと API のマッピング

WebAPI は知的照明システムを操作するための API であるが、この知的照明システムは世界に 1 つのものではなく、同志社大学の知的照明システムの実験室をはじめとして、様々なオフィスに導入されている。そのため、一つの API で全ての知的照明システムを操作することはセキュリティ上の問題がある。そこで、1 つの知的照

明システムに対してそれを操作する固有の API を持つものとした。また、外部からの不正な利用を防止するため、WebAPI を設置する Web サーバは知的照明システムが導入されている部屋または建物の LAN 内のみから利用できる設定とした。これにより、外部から不正に操作をされることを防ぐことができる⁵⁾。

4 メソッドの実装

知的照明システムの開発および知的照明システムに関する周辺プログラムの開発者 5 名にアンケートを行い、WebAPI を用いて知的照明システムを利用する場合はどのような機能を実装すべきか調査し、特に希望の集まった機能について実装を行った。以下に実装した命令を列挙する。

1. 部屋に設置されている照明の間取りを出力
2. 光度と信号値の関係情報を出力
3. 照明出力値の一覧を出力
4. 照明出力を変更前に戻す
5. 照明出力の変更履歴を出力
6. 消費電力量の出力

なお、今回は制御する照明を蛍光灯のみとし、さらに照明の設置位置と光度と信号値、および色温度と信号値の関係情報は知的照明システムの DB に格納されていることを前提としている。

5 従来の UI と WebAPI による UI の比較

WebAPI を用いた UI の実装と従来のクライアント・サーバモデルの UI の実装とを比較して、WebAPI を利用する利点および欠点について以下に示す。

Table 1 WebAPI を用いる利点と欠点

利点	欠点
サーバ構築が不要	用意された命令セット以外利用不可
開発時間の短縮	
DB など技術の学習コスト軽減	
プログラム競合の回避	

また、開発コストとして WebAPI を利用して知的照明システムを操作するためのソースコード量とサーバ・クライアントを用いて知的照明システムを操作するためのソースコード量を比較する。なお、ソースコード量の比較のための単位として今回はステップ数を用いる。

Table 2 開発コストの比較

	WebAPI (ステップ)	サーバ・クライアント (ステップ)
制御 PC 側	なし	100~200
UI 側	5~20	25~30

WebAPI を用いた場合、開発は UI のみでよく、制御 PC (サーバ) 側にはプログラムを追加する必要は無い。そのため、WebAPI を利用するためにプログラムを書くだけで知的照明システムを操作可能である。それに対して、従来手法のサーバ・クライアントモデルによる UI の開発では制御 PC (サーバ) 側に受信プログラムを、UI (クライアント) 側には送信プログラムを書かなければならない。この、サーバ側プログラムの開発を省くことは、知的照明システムにおける UI の開発コストを大幅に削減することができると思われる。

これによって、知的照明システムの UI を様々な端末や開発環境で開発することが容易となり、さらにサーバプログラムの一元化による競合の減少から、UI の不具合による照明の点灯異常の発生も抑えることが可能であると考えられる。本稿で提案した WebAPI は十分に学習、開発コストを軽減し、UI を初めとする知的照明システムに関連するプログラム開発に貢献することが可能であると言える。

6 今後の課題

今後はフルカラー LED に関するメソッドの実装や、さらに多くの命令セットを WebAPI に用意していくことが必要となる。また、WebAPI を知的照明システムの利用されているオフィスに導入し、実際に様々な UI を開発することでデータを収集することが望ましいと言える。

参考文献

- 1) 大林文明ら. オフィスワークの生産性改善のための環境制御法の研究 - 照明制御法の開発と実験的評価, ヒューマンインタフェースシンポジウム 2006. 2006.
- 2) 三木光範, 廣安知之, 池田聡. 知的照明システムにおけるユーザインタフェースの構築 (音声認識およびタッチパネルを用いた照明コントロール), 情報処理学会第 67 回全国大会講演論文集. 2005.
- 3) 中野所武司, 津久井浩. 予約業務を例題とした web アプリケーション用フレームワークの再利用性と評価 (ソフトウェア学会), 電子情報通信学会論文誌, vol.j88-d-i, no. 5 (20050501) . pp. 930-939, 2005.
- 4) 北野優, 本間圭, 高橋佳嗣, 富樫敦. 中小規模のサーバ api 構築に適した web api framework の設計開発 (特集:web2.0 時代のビジネスモデル-検索エンジンと利用技術-), 電子情報通信学会論文誌, 電子情報通信学会技術研究報告. swim, ソフトウェアインタプライズモデリング 108(185) . pp. 37-40, 2008.
- 5) IPA 独立行政法人情報処理機構セキュリティセンター. 安全なウェブサイトの作り方. 2011.