

Title	動作の伴う野菜の育成情報を円滑に共有可能にする野菜育成情報自動共有システムの提案と評価
Sub Title	A proposal and an evaluation of the system that enables sharing kinetic information of growing vegetables automatically
Author	芝原, 隼人(Shibahara, Hayato) 砂原, 秀樹(Sunahara, Hideki)
Publisher	慶應義塾大学大学院メディアデザイン研究科
Publication year	2011
Jtitle	
JaLC DOI	
Abstract	<p>本論文は、家庭菜園に満足していない初心者の割合が高いという問題に着目する。その問題の主な要因として、家庭菜園を行うための環境が十分に整っていないことと、野菜の育成情報を十分に理解していないことが挙げられる。</p> <p>これらの課題を解決するために、家庭菜園を行うための環境が整っていない初心者でも、野菜の育成情報を容易に理解し、実践することで長く楽しめるような環境の実現が必要であると考えた。初心者が野菜の育成情報を十分に理解していない要因として、インターネットから取得する野菜の育成情報の多くは文書と写真のみで表現されており、動作の伴う野菜の育成情報の共有が難しいという負担が考えられる。そこで、本論文ではその負担を低減するために、野菜育成情報自動共有システムの提案を行う。</p> <p>野菜育成情報自動共有システムは、動作の伴う野菜の育成情報の共有を円滑にし、理解を促すという指針に基づいて設計する。動作の伴う野菜の育成情報の理解を促すためには、家庭菜園の工程を施すタイミングと、その工程を施す過程における複数の動作の理解を促すことが重要である。一方、動作の伴う野菜の育成情報の共有を円滑にするためには、それらの情報をシステムに保存するまでの負担と、システムに蓄積された複数の情報から必要な情報を取得するまでの負担を低減することが重要である。</p> <p>本論文では、これらの洞察により実装したシステムの提供する、野菜育成動画自動共有機能、環境情報自動共有機能、野菜育成ダイジェスト動画自動共有機能、ウェブアプリケーションを設計指針に沿って評価した。その結果、野菜育成動画自動共有機能は、ユーザの施した全ての工程のうち78%が安定して自動的にシステムに保存されていたことが明らかになった。環境情報自動共有機能は、家庭菜園の環境情報をシステムに安定して自動的に保存していたことが明らかになった。野菜育成ダイジェスト動画自動共有機能は、家庭菜園の環境と野菜の育成状況の変化を野菜育成ダイジェスト動画として、システムに安定して自動的に保存していなかったことが明らかになった。ウェブアプリケーションは、蓄積された野菜育成動画と環境情報と野菜育成ダイジェスト動画をユーザに体系的に提供したが、ユーザがシステムに蓄積された複数の情報から必要な情報を取得するまでの負担を十分に低減していなかったことが明らかになった。一方、野菜育成動画と環境情報グラフにより、家庭菜園の工程を施すタイミングと、その工程を施す過程における複数の動作の理解を容易にしたことが明らかになった。</p> <p>これらの結果から明らかになった課題を解決することで野菜育成情報自動共有システムは、動作の伴う野菜育成情報の共有を円滑にし、理解を促すことが期待できる。</p>
Notes	修士学位論文. 2011年度メディアデザイン学 第166号
Genre	Thesis or Dissertation
URL	https://koara.lib.keio.ac.jp/xoonips/modules/xoonips/detail.php?koara_id=KO40001001-00002011-0166

慶應義塾大学学術情報リポジトリ(KOARA)に掲載されているコンテンツの著作権は、それぞれの著作者、学会または出版社/発行者に帰属し、その権利は著作権法によって保護されています。引用にあたっては、著作権法を遵守してご利用ください。

The copyrights of content available on the KeiO Associated Repository of Academic resources (KOARA) belong to the respective authors, academic societies, or publishers/issuers, and these rights are protected by the Japanese Copyright Act. When quoting the content, please follow the Japanese copyright act.

2011 年度 修士論文

動作の伴う野菜の育成情報を円滑に共有可能にする
野菜育成情報自動共有システムの提案と評価



KEIO MEDIA DESIGN

慶應義塾大学大学院
メディアデザイン研究科

芝原 隼人

本論文は慶應義塾大学大学院メディアデザイン研究科に
修士(メディアデザイン学) 授与の要件として提出した修士論文である。

芝原 隼人

指導教員：

砂原 秀樹 教授 (主指導教員)

稲見 昌彦 教授 (副指導教員)

審査委員：

砂原 秀樹 教授 (主査)

稲見 昌彦 教授 (副査)

石倉 洋子 教授 (副査)

動作の伴う野菜の育成情報を円滑に共有可能にする 野菜育成情報自動共有システムの提案と評価

内容梗概

本論文は、家庭菜園に満足していない初心者の割合が高いという問題に着目する。その問題の主な要因として、家庭菜園を行うための環境が十分に整っていないことと、野菜の育成情報を十分に理解していないことが挙げられる。

これらの課題を解決するために、家庭菜園を行うための環境が整っていない初心者でも、野菜の育成情報を容易に理解し、実践することで長く楽しめるような環境の実現が必要であると考えた。初心者が野菜の育成情報を十分に理解していない要因として、インターネットから取得する野菜の育成情報の多くは文書と写真のみで表現されており、動作の伴う野菜の育成情報の共有が難しいという負担が考えられる。そこで、本論文ではその負担を低減するために、野菜育成情報自動共有システムの提案を行う。

野菜育成情報自動共有システムは、動作の伴う野菜の育成情報の共有を円滑にし、理解を促すという指針に基づいて設計する。動作の伴う野菜の育成情報の理解を促すためには、家庭菜園の工程を施すタイミングと、その工程を施す過程における複数の動作の理解を促すことが重要である。一方、動作の伴う野菜の育成情報の共有を円滑にするためには、それらの情報をシステムに保存するまでの負担と、システムに蓄積された複数の情報から必要な情報を取得するまでの負担を低減することが重要である。

本論文では、これらの洞察により実装したシステムの提供する、野菜育成動画自動共有機能、環境情報自動共有機能、野菜育成ダイジェスト動画自動共有機能、ウェブアプリケーションを設計指針に沿って評価した。その結果、野菜育成動画自動共有機能は、ユーザの施した全ての工程のうち78%が安定して自動的にシステムに保存されていたことが明らかになった。環境情報自動共有機能は、家庭菜園の環境情報をシステムに安定して自動的に保存していたことが明らかになった。野菜育成ダイジェスト動画自動共有機能は、家庭菜園の環境と野菜の育成状況の

変化を野菜育成ダイジェスト動画として、システムに安定して自動的に保存していなかったことが明らかになった。ウェブアプリケーションは、蓄積された野菜育成動画と環境情報と野菜育成ダイジェスト動画をユーザに体系的に提供したが、ユーザがシステムに蓄積された複数の情報から必要な情報を取得するまでの負担を十分に低減していなかったことが明らかになった。一方、野菜育成動画と環境情報グラフにより、家庭菜園の工程を施すタイミングと、その工程を施す過程における複数の動作の理解を容易にしたことが明らかになった。

これらの結果から明らかになった課題を解決することで野菜育成情報自動共有システムは、動作の伴う野菜育成情報の共有を円滑にし、理解を促すことが期待できる。

キーワード

家庭菜園、知識共有、暗黙知、ノウハウ、センサネットワーク、動画共有

慶應義塾大学大学院 メディアデザイン研究科

芝原 隼人

A proposal and an evaluation of the system that enables sharing kinetic information of growing vegetables automatically

Abstract

In this paper, we focus on the percentage of home gardening beginners who are unsatisfied is high. There are two reasons for this problem. One of them is that conditions of individual environments for home gardening are not satisfied. The other is that users do not understand entire situation of growing vegetables.

In order to solve these problems, the environment that enables beginners to enjoy home gardening sustainably even the conditions of individual environments for home gardening who are not satisfied is required . There are two factors which beginners do not understand information of growing vegetables. One of the factors is due to difficulty of sharing the information that deals with kinetic information of growing vegetables described in pictures and writing via the Internet.

Therefore, we propose a new system that enables sharing kinetic information of growing vegetables automatically. The system is designed to enhance sharing kinetic information of growing vegetables and promote beginners to understand them. To promote beginners to understand the information, it is essentially important to ease comprehension of the timings of operations for a vegetable and dynamic motions during the procedures. On the other hands, to enhance sharing kinetic information, it is critically important to reduce the burden of storing the information on the system and finding out the needed information from much accumulated information.

To evaluate three functions and the application by comparing the design policy, we set three points: the function that enables sharing movies of operations automatically, the function that enables sharing environmental information auto-

matically, the function that enables sharing digest movies, and the web application to serve accumulated information systematically. As the result, the function that enables sharing movies of operations automatically stored 78% of whole operations. The function that enables sharing environmental information automatically stored them stably. The function that enables sharing digest movies automatically did not stored them stably. The web application served accumulated information systematically.

Keywords:

home gardening, knowledge sharing, implicit knowledge, knowhow, sensor network, sharing movies

Graduate School of Media Design, Keio University

Hayato Shibahara

目 次

第1章 背景	1
1.1. 概要	1
1.2. 家庭菜園の人気の高まり	1
1.3. 家庭菜園における課題	2
1.4. 目指す環境	4
1.5. 家庭菜園の情報媒体における課題	5
1.6. 家庭菜園における野菜の育成情報の工程	7
第2章 関連サービス	12
第3章 野菜育成情報自動共有システムの提案	14
3.1. 概要	14
3.2. 野菜育成情報自動共有システムの提案	14
3.3. 野菜育成情報自動共有システムの役割	15
3.4. 野菜育成動画の役割	16
3.5. 野菜育成情報自動共有システムの機能	16
3.5.1 野菜育成動画自動共有機能	16
3.5.2 環境情報グラフ自動共有機能	17
3.5.3 投稿情報共有機能	17
3.5.4 野菜育成ダイジェスト動画自動共有機能	18
第4章 設計	19
4.1. 概要	19
4.2. 野菜育成情報自動共有システムで扱う情報	19
4.3. システムアーキテクチャ	20

4.4.	クライアントサイドシステムの設計	21
4.4.1	クライアントサイドシステムの構成	21
4.4.2	環境情報取得用マイクロコントローラの役割	22
4.4.3	野菜育成情報送信用マイクロコントローラの役割	24
4.5.	サーバサイドシステムの設計	24
4.5.1	サーバサイドシステムの構成	24
4.5.2	アプリケーションサーバの役割	25
4.5.3	データベースサーバの役割	29
4.5.4	動画サーバの役割	29
第5章	実装	32
5.1.	概要	32
5.2.	クライアントサイドシステム	32
5.2.1	環境情報取得用マイクロコントローラ	33
5.2.2	野菜育成情報送信用マイクロコントローラ	36
5.3.	サーバサイドシステム	39
5.3.1	データベースサーバ	40
5.3.2	アプリケーションサーバ	40
5.3.3	動画サーバ	53
第6章	評価	58
6.1.	概要	58
6.2.	野菜育成動画自動共有機能の評価	59
6.3.	環境情報自動共有機能の評価	63
6.4.	野菜育成ダイジェスト動画自動共有機能の評価	67
6.5.	ウェブアプリケーションの評価	70
第7章	考察	75
7.1.	概要	75
7.2.	野菜育成動画自動共有機能の考察	75
7.3.	環境情報自動共有機能の考察	80

7.4. 野菜育成ダイジェスト動画自動共有機能の考察	81
7.5. ウェブアプリケーションの考察	82
第8章 結論	86
第9章 展望	89
付録 A 野菜育成動画自動共有システムの主なプログラム	94
A.1. アプリケーションサーバーの主なプログラム	94
A.1.1 sensors_controller.php の armadillo ファンクション	94
A.1.2 sensors_controller.php の demo ファンクション	96
A.1.3 sensors_controller.php の video ファンクション	100
A.1.4 sensors_controller.php の pictures ファンクション	103
A.2. 動画サーバーの主なプログラム	103
A.2.1 autoMovs.php	103
A.2.2 movies.php	105
A.2.3 autoPics.php	105
A.2.4 pictures.php	106
A.3. 野菜育成情報送信用マイクロコントローラーの主なプログラム	108
A.3.1 post.php	108

目 次

1.1	家庭菜園向け野菜苗、果樹苗の市場規模推移	2
1.2	家庭菜園を始める理由	3
1.3	家庭菜園の経験がある人にとっての魅力	4
1.4	家庭菜園に対する家庭菜園歴ごとの満足度	4
1.5	家庭菜園をやめた理由	5
1.6	家庭菜園の参考にしている媒体 [6]	7
1.7	家庭菜園におけるコミュニケーション [4]	8
3.1	野菜育成情報自動共有システムの概要	15
4.1	システムアーキテクチャ	22
4.2	クライアントサイドシステム構成	23
4.3	サーバサイドシステム構成	25
4.4	ストリーミング映像保存プロセスのダイアグラム	30
4.5	ウェブアプリケーションの構造	31
5.1	実装したクライアントサイドシステム	33
5.2	クライアントサイドシステムの構成要素	34
5.3	クライアントサイドシステムの回路図	35
5.4	NaPiOn(微動検出タイプの検出範囲)	36
5.5	ビュー	44
5.6	インデックスページ	45
5.7	環境情報グラフ共有ページ	46
5.8	環境情報グラフ	47
5.9	野菜育成動画共有ページ	48

5.10	野菜育成動画と投稿情報の例	49
5.11	ユーザ登録/ログインページ	50
5.12	ウェブサーバのフォルダ構造	57
6.1	ステップごとの野菜育成動画が保存されるまでの経過時間 [秒]	64
6.2	ステップごとの autoMovs.php によるプロセスの経過時間 [秒]	65
6.3	改良したクライアントサイドシステム	67
6.4	ストリーミング画像の保存枚数と時刻の関係	68
6.5	定期的に行う工程の野菜育成動画のスクリーンショット	72
6.6	蓄積した野菜育成動画のスクリーンショット	74
7.1	11/28 の環境情報グラフ	77
7.2	12/09 の環境情報グラフ	78
7.3	モーションセンサ検出範囲	79

目 次

5.1	armadillo データベースの構成	40
5.2	sensor テーブルのフィールド	41
5.3	movies テーブルのフィールド	41
5.4	pictures テーブルのフィールド	42
5.5	armadillos テーブルのフィールド	42
5.6	comments テーブルのフィールド	42
6.1	動画サーバに保存された野菜育成動画	61
6.2	追加して記録した野菜育成動画の情報	66
6.3	蓄積した環境情報	66
6.4	データベースに保存されたダイジェスト動画	69

第1章 背 景

1.1. 概要

本章では本研究を行うにあたっての背景について述べる。またその背景から導出される課題を明確にする。そして、本研究がその課題に対し、どのように取り組むべきかについて述べる。

1.2. 家庭菜園の人気の高まり

近年、家庭菜園向け野菜苗、果樹苗の市場規模が拡大している [1]。図 1.1 に家庭菜園向け野菜苗、果樹苗の市場規模推移を示す。この統計が示すように 2009 年まで家庭菜園向け野菜苗、果樹苗の市場規模推移の増加の割合は毎年 10%前後で高成長であり、家庭菜園の人気の高まりが継続的に高まっている。

家庭菜園の人気の高まる中、家庭菜園の未経験者の多くは娯楽を求めて家庭菜園を始める傾向にある [5]。図 1.2 は家庭菜園を始めた理由についての調査結果である。この結果によると、未経験者が家庭菜園を始める理由として最も多かったのは、自分で食べる野菜を作る面白さであった。また、家計の節約や、安全安心のために野菜を作りたいという理由をきっかけにして家庭菜園を始める人の割合も高い。つまり、多くの人は自分で育成した野菜の収穫に期待を抱いて家庭菜園を始めている。

未経験者が野菜の収穫に期待を抱いて家庭菜園を始める傾向にあるのに対し、経験者は野菜を育成する過程に家庭菜園を行う理由を特に見出している傾向にある。図 1.3 は、経験者にとっての家庭菜園の魅力についての調査結果である [3]。

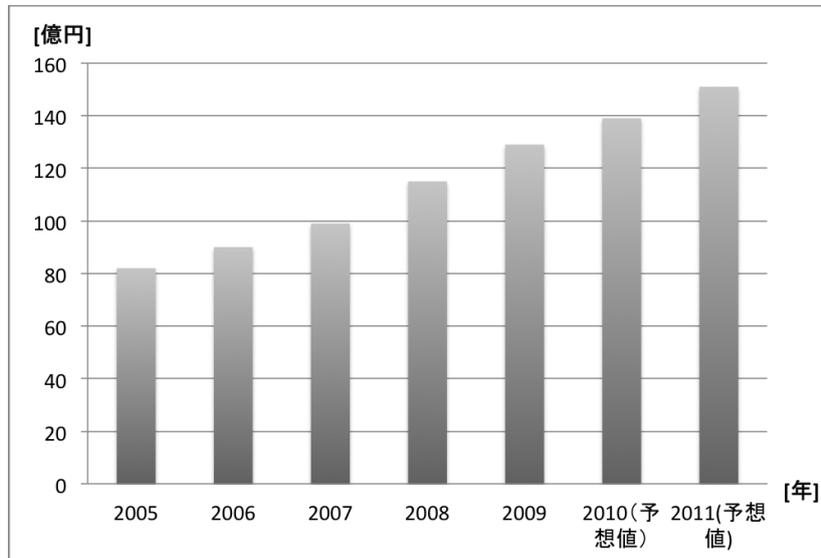


図 1.1 家庭菜園向け野菜苗、果樹苗の市場規模推移

この結果によると、経験者にとっての家庭菜園の主な魅力は、野菜の成長を日々実感できることであった。また、未経験者が野菜の収穫に期待を抱いているのと同様に、経験者も収穫した新鮮な野菜を食べられることやその達成感にも魅力として感じている傾向が見られた。

以上より、家庭菜園の人気の高まりを支えているのは、野菜の収穫に対する期待だけではなく、収穫に至るまでの過程で得られる継続的な楽しみが含まれるといえる。

1.3. 家庭菜園における課題

家庭菜園の人気の高まる一方、家庭菜園に満足していない初心者の方の割合の高さが課題として懸念される。フマキラー株式会社の調査によると、家庭菜園を始めて1年以内に中断する人の割合は4割に達するという。図 1.4 に家庭菜園歴ごとの人の家庭菜園に対する満足度に関するアンケートの結果を示す [5]。このアンケートの結果によると、家庭菜園歴1年以上の人が家庭菜園に満足していない割合は19.4%であったのに対し、家庭菜園歴1年未満の人が家庭菜園に満足してい

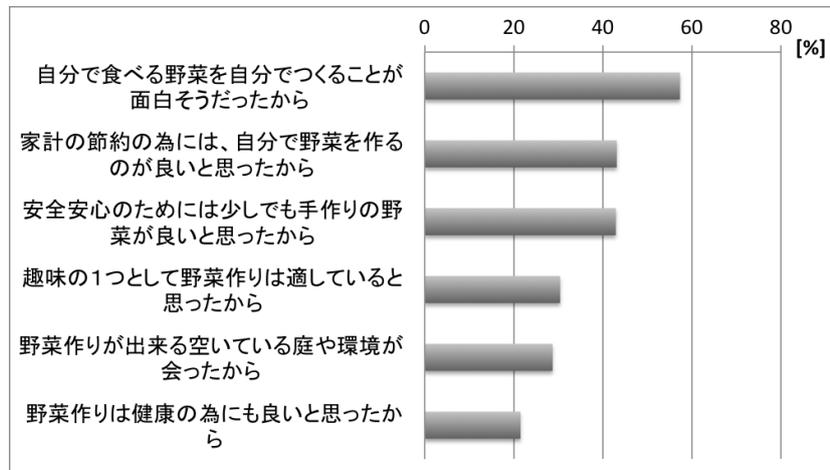


図 1.2 家庭菜園を始める理由

ない割合は 33.4%であった。つまり、家庭菜園に満足していない初心者は比較的多い。

一部の初心者が家庭菜園に満足していない要因として、家庭菜園を行うための環境が十分に整っていないことと野菜の育成情報を十分に理解していないことが挙げられる。図 1.5 は家庭菜園を辞めた理由に関する調査結果である [5]。この結果によると 37.0%の人が、野菜育成を行うための環境が整っていないために家庭菜園を辞めていた。その理由として一人暮らしで場所が無いことや日当たりの良いスペースが確保できないことなどが挙げられた。また、28.5%の人が、野菜育成を行うための時間に余裕が無いために家庭菜園を辞めていた。その理由として、仕事や家事に忙しく、野菜の世話をする暇が限られていることなどが挙げられた。従って、家庭菜園を行うための環境（時間や空間）が十分に整っていないことが要因で、初心者は家庭菜園に満足しにくいと推測できる。一方で、19.6%の人が害虫や病気の対策に困るため、14.7%の人が野菜の育成情報を理解していないために家庭菜園を辞めていた。このことから、野菜の育成情報を十分に理解していないことも、初心者が家庭菜園に対して満足を得られにくい要因だと考えられる。

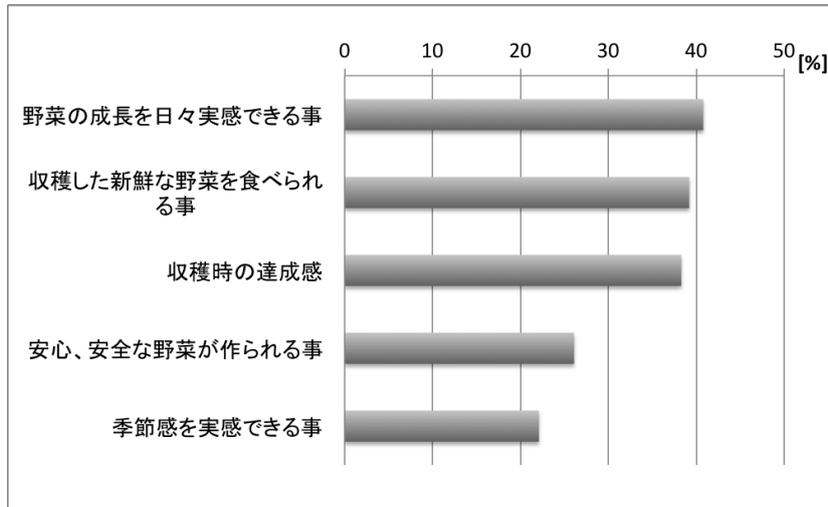


図 1.3 家庭菜園の経験がある人にとっての魅力

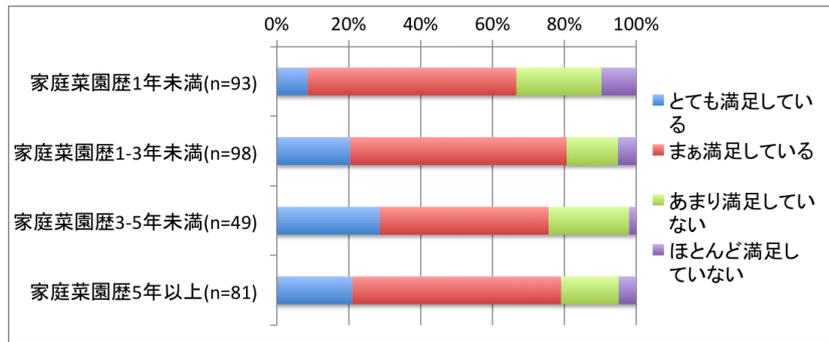


図 1.4 家庭菜園に対する家庭菜園歴ごとの満足度

1.4. 目指す環境

一部の初心者が家庭菜園に満足していない要因として、家庭菜園を行うための環境が十分に整っていないことと野菜の育成情報を十分に理解していないことを挙げた。また、初心者の家庭菜園の継続を促すためには、野菜を収穫するに至るまでの過程をより楽しくすることも大切である。これらの課題を初心者にとって負担の少ない方法で解決するために、家庭菜園を行うための環境が十分に整っていない初心者でも容易に野菜の育成情報を理解し、実践することで家庭菜園を長く楽しめるような環境の実現を本研究では目指す。

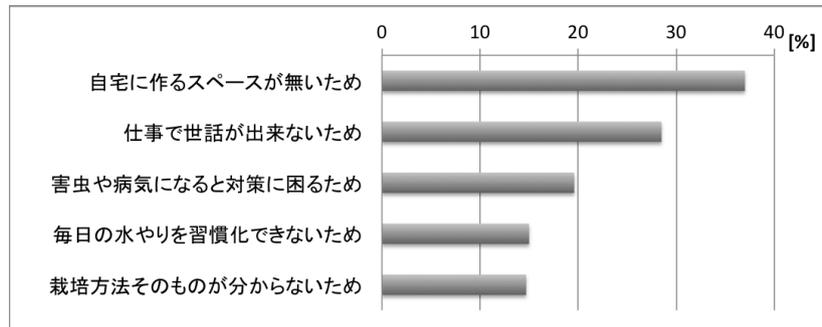


図 1.5 家庭菜園をやめた理由

1.5. 家庭菜園の情報媒体における課題

初心者が野菜の育成情報を十分に理解していない要因として、インターネットから取得する家庭菜園の情報を理解し実践する際の負担が大きいことが考えられる。多くの家庭菜園を行う人はインターネットから得られる情報を参考にしている。図 1.6 は、家庭菜園を行う上で参考にしている媒体についての調査結果である [6]。この結果によると、友人や知人から得られる情報を参考にしている人の割合が 15%、書籍から得られる情報を参考にしている人の割合が 15%であるのに対して、インターネットから得られる情報を参考にしている人の割合は 36%と比較的高い。しかし、インターネットには家庭菜園に関する情報を提供するサイトが複数あり、情報が豊富にあるのにも関わらず、一部の初心者は野菜の育成情報を十分に理解していないという問題が生じている。その要因として、インターネットから取得する野菜の育成情報を理解し、実践する際に以下に挙げる負担があるためだと考えられる。

- A インターネットから取得する野菜の育成情報は特定の家庭菜園の環境を前提にしている場合が多く、環境に制約のある初心者にとって実践が難しいこと
- B インターネットから取得する野菜の育成情報の多くは文書と写真で表現されており、動作の伴う野菜の育成情報の共有が難しいこと
- C 経験者とのコミュニケーションの機会が限られており、適切な助言を受けられにくいこと

項目 A については、家庭菜園の環境が十分に整っておらず、インターネットから取得する野菜の育成情報が前提とした環境を満たしにくく、初心者にとって野菜育成で実践することが難しいという負担である。初心者が家庭菜園に満足していない要因として挙げたように、生活の中で家庭菜園を行うための空間や時間の余裕は十分に整っていない傾向にある。従って、育成情報を個人の野菜育成に実践する際に難しさが生じる。

項目 B については、動作の伴う野菜の育成情報は文書や写真で表現することは難しく、初心者にとって理解が難しいという負担である。動作の伴う野菜の育成情報として、例えば、水やりや追肥が挙げられる。水やりの場合、水の与え方やその分量だけではなく水を与えるタイミングが重要である。しかし、そのタイミングは個人の家庭菜園の環境の湿度や土壌の水はけの度合いに依存するため、これらの環境に依存する情報を文書や写真で理解しやすく表現することは難しい。また同様に追肥の場合も、その与え方や分量は野菜の成長の度合いや土壌の性質に依存するため、初心者にとって直感的に理解しやすく描写することは難しい。

項目 C については、経験者とのコミュニケーションを行う機会が限られており、個人の環境や野菜の育成状況に応じた適切な助言を受けられにくく、適切な野菜の育成情報を理解しにくいという負担である。これは、家庭菜園を行う人の多くが野菜育成を個人で楽しむ傾向によるためである。図 1.7 に家庭菜園におけるコミュニケーションについての調査結果を示す [4]。この結果によると、野菜の育成情報を教え合う人の割合が 20%であるのに対し、家庭菜園を個人で楽しんでいる人の割合は 48%で比較的高い。また、収穫物の写真を撮ってインターネットで公開する人の割合も 4%と少なく、オンラインでのコミュニケーションの機会も限られていることが推測できる。従って、初心者は経験者と密に接する機会が少なく、個人の家庭菜園の環境や野菜の育成状況に応じた適切な助言を受けられにくいと考えられる。

これらの負担を解消するために、(1) 様々な環境における野菜の育成情報を参照でき、(2) 動作の伴う野菜の育成情報の共有を円滑にし、理解を促し、(3) 個人の家庭菜園の環境や野菜の育成状況を基にコミュニケーションを図れるようなシステムが必要だと考えられる。(1) の機能によって、家庭菜園に費やす時間や空

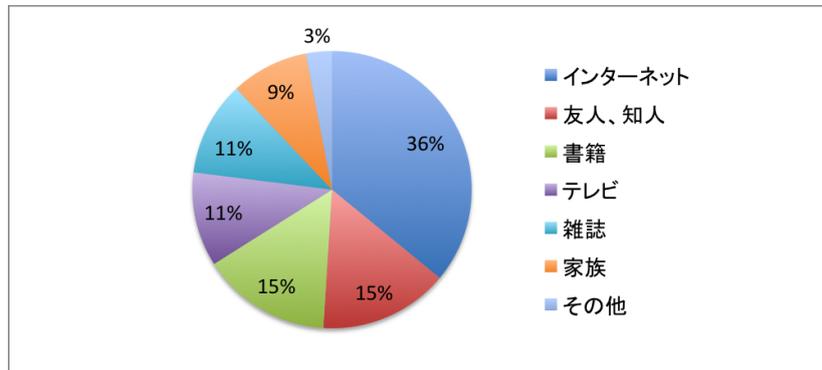


図 1.6 家庭菜園の参考に使っている媒体

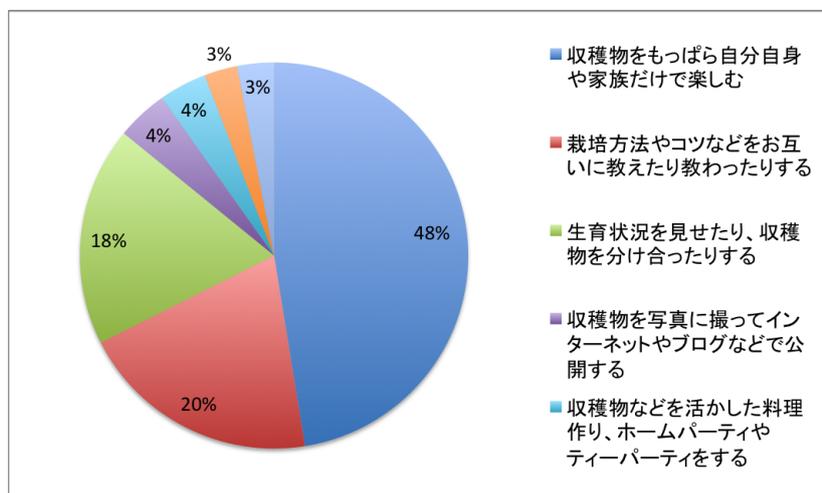


図 1.7 家庭菜園におけるコミュニケーション

間の限られている初心者でも、類似した環境を前提にした野菜の育成情報を参照することができる。そして、その方法を実践する際の負担を抑えることが可能になる。(2)の機能によって、文書や写真により表現の難しかった動作の伴う野菜の育成情報の共有ができる。これにより初心者は、経験者の野菜の育成情報をより効果的に理解することが可能になる。(3)の機能によって、経験者は初心者の家庭菜園の環境や野菜の育成過程を把握することができる。これにより初心者は経験者から適切な助言を受けることができ、野菜育成の継続を促すことが可能になる。

本論文では、これらの機能のうち特に(2)の機能に着目して提案を行う。この機能が実現されることで、従来インターネットを利用して難しかった家庭菜園での動作の伴う野菜の育成情報を効果的に共有することが可能になる。その結果、それらの情報を蓄積することで(1)の機能と(3)の機能の十分条件が満たすことができるためである。

1.6. 家庭菜園における野菜の育成情報の工程

家庭菜園における野菜の育成情報は、水やり、土作り、種まき、植え付け、間引き、芽かき、摘心、追肥、中耕、土寄せ、支柱立て、誘引、を含む複数の工程から構成される。ここでは各工程の概要について説明した後、それらの工程を定期的に行う工程と非定期的に行う工程に分類する。

- 水やり

プランタでの野菜作りにおいて、水やりは非常に重要である。適切な水やりを行うためには、そのタイミングと回数を環境によって調整することが重要である。例えば水を与えるタイミングについては土の表面が乾いた場合、夏場は早朝に、冬場は土の温度が上昇し始める午前中に水やりを行うのが良いとされる。また、水やりの仕方も育成する野菜によって異なる。葉菜類、根菜類、株の高くない果菜類の水やりの場合は葉の上から水を与え、一方、株の高い果菜類の場合、根元から水を与えて夕方には土の表面が乾く程度に水やりをするのが良いとされている。

- 土作り

土作りは、野菜の根をはらせて、生育に必要な水分や養分を根に供給するために必要である。プランタでの菜園では、市販の野菜用の栽培用土を使用し、水はけを良くするために用土の下に大粒の土を入れることが一般的である。また直播き用や植えつけ用など用途や育成する野菜によって土作りを変えることも重要である。

- 種まき

種まきは、種を発芽させ、株間の間隔を調整するために重要な工程となっている。種は野菜によって発芽に適した温度がある。適温より温度が低すぎたり、高すぎたりすると発芽がしにくくなる。主な種まきの方法は、すじまき、点まき、ばらまきが存在し育成する野菜によって方法を変えることが一般的である。

- 植え付け

植え付けはプランタの大きさを考えながら、植える株の数や間隔を考える必要がある。また、根や茎の位置に気をつけて植え付けの深さを気をつけると良いとされる。プランタで野菜を育てる場合、育苗に温度が必要な果菜類は種から育てるより、苗を購入した方が失敗は少なくすむとされる。

- 間引き

間引きは発芽後、生育の悪い苗などを抜き取りながら最終的に必要な株間を作るために必要である。種をまいて苗を育てる場合、普通一カ所に数粒の種をまくが、発芽後生育の良いものを残して他を抜き取るか切り取って最終的な株間をとる。一般的に間引きは数回に分けて行われ、タイミングは育成する野菜に合わせて行われる。間引きの際には残す苗を痛めないように、間引き苗の株元を2本の指先で抑え引き抜く。そして葉の傷んだもの、生育の悪いものなどを間引き、生育の良い苗を残す。間引きの方法は種まきの手法、野菜の種類、株の成長の度合いに合わせて行われる。

- 芽かき、摘心

多くの野菜で、株が生長してくると枝と葉の付け根の間からわき芽と呼ばれる芽が出てくる。このわき芽をそのままにしておくと、株全体の生育が遅れたり、よい果実がならない野菜がある。そのため、芽かきによってわき芽を手で摘み取る必要がある。一方、主枝が大きく生長しての届かないところまでのびてしまう場合は主枝の先端の生長点を手で摘み取って生長

を止める摘心を行う。また、茎の先端や葉を収穫する野菜に関しては、あえて摘心をし、わき芽をのばして収穫量を増やす。

- 追肥、中耕、土寄せ

市販の野菜用の栽培用土など、土中の肥料分は野菜が育つ過程の中で根からの吸収や水やりなどで流れ出てしまうため、肥料分は少なくなる。そのため、育成途中に肥料を追加で施す必要があり、その工程を追肥と呼ぶ。追肥では短時間で効果が現れる即効性の肥料を使うことが一般的である。肥料分は根の先端から吸収されるため、根の広がりに沿って行う。生長によって、根は外に広がっていくため生育の初期では株の周囲に、次第に根から離れた位置の株間などに施すように行う。追肥後は、水やりなどで固くなった土の表面を指先でほぐす中耕を行う。中耕を行うことで、根の通気性が良くなり、酸素が供給されやすくなるためである。中耕の後は、株元に土を寄せる。この工程を土寄せという土寄せを行うことで株が倒れにくくなり、また芋生類などでは、芋が土から出てしまうことを防ぐ。

追肥で与える肥料の養分としては、生育に必要なものとして特に窒素、リン酸、カリウムが重要となっている。各栄養素の役割として主に、窒素は葉と茎に働き、リン酸は花と実働き、カリウムは実と根によく働く。肥料には多くの種類がありいくつかに分類することができる。原料で大まかに分類すると、化学的に合成された無機質肥料、天然由来のものを原料とした有機質肥料とに分けられる。化学肥料には、緩効性肥料と速効性肥料が存在する。速効性肥料は、効果がすぐに現れる反面それが長続きしない点が特徴である。一方、緩効性肥料は、じわじわと長い間ゆっくりと効果を現すタイムの肥料である。特に有機質肥料は微生物などによって分解されてから吸収されるため、効果が長く続きやすいという特徴を持つものが多くある。植え付け前に、土にあらかじめ施しておく肥料を元肥、生育の途中で足らなくなった肥料分を補う目的で施すものを追肥という。元肥には感光性の肥料、追肥には速効性の肥料が向いているとされる。

- 支柱立て、誘引

支柱は、竹が高く育ったり、株が倒れやすい野菜を支える役割をする。誘引とはつる性の植物を立体的に育てて、区部に日を良く充てたり、風通しを良くするためにつるなどを支柱に結んだり、ひもや支柱に絡ませることである。支柱の立て方にも数種類があり、育成する野菜の種類によって変えることが良いとされる。

本論文では、以上の育成情報を定期的に行う工程と非定期的に行う工程に分類する。定期的に行う工程として水やりが分類される。それらの工程において、初心者にとって野菜育成の継続を促すためには、工程を行うタイミングの理解が重要である。例えば水やりの場合、初心者は水を与えるタイミングを理解していないことが要因で野菜の根を腐らせてしまう場合が多い。一方、非定期的に行う工程として土作り、種まき、植え付け、追肥、中耕、土寄せ、芽かき、摘心、土寄せ、支柱立て、誘引が分類される。それらの工程においては、工程を施す過程における複数の動作を理解することが重要である。例えば摘心の場合、摘心を行う箇所や分量、その手法などの、複数の動作を理解し、野菜育成に実践することで、野菜の収穫量を高めることができる。

第2章 関連サービス

- BIGLOBE ファーム NEC

BIGLOBE ファームは管理人付きの貸し農園である [9]。この貸し農園は、都会に在住していて、家庭菜園に十分に時間を充てられないが、気軽に楽しみたいという人を主なターゲットとしている。この貸し農園の特徴として3つのサービスとコミュニティがある。3つのサービスは、管理人によるサポート、ライブカメラによる遠隔からの野菜の育成状況の把握、農具の無料レンタルである。ユーザは管理人のサポートを受けることにより専門的なアドバイスを受けることができる。また、ライブカメラを閲覧することにより、野菜の育成状況を把握できる。そして、農具無料レンタルによって、気軽に野菜育成を楽しめる。コミュニティは、農園コミュニティと呼ばれる会員制 SNS である。この SNS によって、管理人とユーザを繋げる事ができ、ユーザは害虫の対策方法や野菜の調理方法など細かいアドバイスを受ける事ができる。

- Click&Grow

Click&Grow は、水や肥料の適切料を把握するためのセンサを内蔵した水耕栽培キットである [10]。このキットは、水耕システム本体とパソコンを USB ケーブルで接続し、各植物の育成プログラムをウェブサイトからインストールできる事が特徴である。各植物の育成プログラムをキットにインストールする事によって自動的に製品が、溶液、水量を適当な値を保つように調整する。これにより、ユーザは植物の育成に失敗しにくくなる。

- EasyBloom Plant Sensor

EasyBloom Plant Sensor は、特定の環境で育成に適した植物を提示する [11]。この製品にはは照度、気温、湿度、水はけ具合などのセンサが実装されている。この製品を特定の環境の土壌に差し込み、24 時間測定し、USB を通してパソコンに接続しウェブサイトから情報を取得する事で、ユーザはその環境で生育した植物を把握できる。このシステムには、6000 以上の植物のライブラリが蓄積されていて、ユーザはそれらの内から育成に適した植物を見つける事ができる。

- farmbox

farmbox は、Android OS とクラウドシステムに対応した LED 植物工場である。[12] 植物工場は高度な環境制御を行う事により、野菜の周年、計画生産を可能にする施設園芸農業の一形態であるが、近年はその小型化が進み個人に対するニーズも高まって来ている。この製品では、ユーザがスマートフォンから遠隔で環境を制御でき、植物の育成状況がクラウドシステムに自動的に保存される点が特徴である。そして、マーケットに保存された大量の栽培記録をデータマイニングし、栽培環境と生育状況に関する統計情報から植物に関する新たな知見を導き出すことが特徴である。

第3章

野菜育成情報自動共有システムの 提案

3.1. 概要

本章では、まず野菜育成情報自動共有システムの提案を行う。続いて、野菜育成情報自動共有システムと野菜育成動画の役割について説明する。そして、野菜育成情報自動共有システムの機能である、野菜育成動画自動共有機能、環境情報グラフ自動共有機能、投稿情報共有機能、野菜育成ダイジェスト動画自動共有機能について述べる。

3.2. 野菜育成情報自動共有システムの提案

本研究では、家庭菜園に満足していない初心者の割合が高いという問題に着目した。その問題の要因として、家庭菜園を行うための環境が十分に整っていないことと初心者が野菜の育成情報を十分に理解していないことを挙げた。これらの課題を初心者にとって負担の少ない方法で解決するために、家庭菜園を行うための環境が十分に整っていない場合でも野菜の育成情報を容易に理解し、実践することで家庭菜園を長く楽しめるような環境が必要であると考えた。初心者が野菜の育成情報を十分に理解していない要因として、インターネットから取得する野菜の育成情報を理解し実践する際の負担が大きいことを挙げた。その負担を解消するために、(1) 様々な環境における野菜の育成情報を参照でき、(2) 動作の伴う野菜の育成情報の共有を円滑にし、理解を促し、(3) 個人の家庭菜園の環境や野

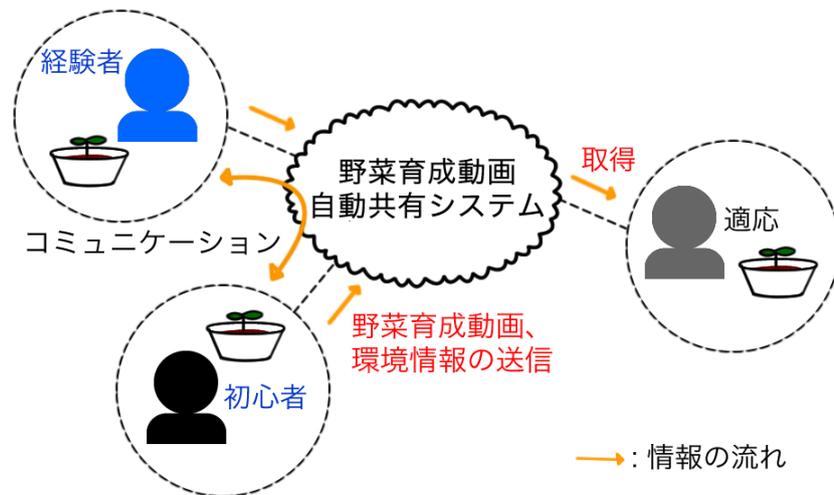


図 3.1 野菜育成情報自動共有システムの概要

菜の育成状況をもとにコミュニケーションを図れる様なシステムが必要であると考えた。本論文では、そのシステムを実現するために (2) の機能に着目し、野菜育成情報自動共有システムの提案を行う。

3.3. 野菜育成情報自動共有システムの役割

図 3.1 に野菜育成情報自動共有システムの概要を示す。野菜育成情報自動共有システムは、ユーザが野菜に施した工程を記録した野菜育成動画と家庭菜園の環境情報をサーバに蓄積し、ウェブアプリケーションを通して複数のユーザで共有することを可能にする。野菜育成動画と環境情報を共有することによって、従来文書や写真により表現の難しかった動作の伴う野菜の育成情報を効果的に共有できる。また、システムにウェブアプリケーションを実装することによって、インターネットを通して家庭菜園の情報を取得している多くの人の負担を抑える効果が期待できる。このシステムを初心者が利用することにより、例えば、経験者の野菜育成動画と環境情報を参照し、共有された情報をもとにコミュニケーションを図ることで、経験者の野菜の育成情報を深く理解し実践することが可能になる。

3.4. 野菜育成動画の役割

野菜育成動画とは、ユーザが野菜に施した工程を記録した動画である。野菜育成動画の役割は、動作の伴う野菜の育成情報の理解を容易にすることである。現状の家庭菜園において、多くの人はインターネットから得られる家庭菜園の情報を取得し、野菜育成の参考にしている。しかし、それらの情報の多くは文書と写真によって表現されており、動作の伴う野菜の育成情報は効果的に共有されていないと考えられる。野菜の育成情報は、工程を施すタイミングとその過程における連続した複数の動作の理解が重要である。しかし、それらの動作の伴う野菜の育成情報を文書や写真で理解しやすく表現することは難しい。そこで、環境情報とともに野菜育成動画を共有することによって、動作の伴う野菜の育成情報の理解をより効果的に促すことを期待する。

3.5. 野菜育成情報自動共有システムの機能

野菜育成情報自動共有システムの機能として、野菜育成動画自動共有機能、環境情報グラフ自動共有機能、投稿情報共有機能、野菜育成ダイジェスト動画自動共有機能を提案する。これらの機能によって、動作の伴う野菜の育成情報の共有を円滑にし、野菜の育成情報の理解を促す効果を期待する。また、動作の伴う野菜の育成情報の共有を円滑にするために、(1) 野菜育成動画と環境情報をサーバにアップロードするまでの負担と(2) サーバに蓄積された野菜育成動画と環境情報から必要な情報を見つけ出すまでの負担を低減する。

3.5.1 野菜育成動画自動共有機能

野菜育成動画自動共有機能は、家庭菜園においてユーザが野菜に施した工程を野菜育成動画としてサーバに自動的に蓄積し、ウェブアプリケーションを通して複数のユーザで共有を可能にする機能である。この機能によってユーザは、野菜育成動画をサーバにアップロードするまでの負担を大幅に低減できる。そして、従来、文書や写真での共有の難しかった動作の伴う野菜の育成情報を円滑に共有

することが可能になる。例えば支柱立ての場合、複雑に入り組んだ支柱の立て方をインターネットで文書や写真を通して理解し実践することは難しい。また、支柱立てを行うタイミングについても野菜の育成状況によって異なるため、初心者が適切なタイミングで施すことも難しい。この機能によって初心者が経験者の支柱立てを行う野菜育成動画を参照し、工程を施すタイミングと工程を施す過程で重要となる複数の動作についての理解を容易に深め、実践することができる。

3.5.2 環境情報グラフ自動共有機能

環境情報グラフ自動共有機能は、家庭菜園の環境情報をサーバに自動的に蓄積し、蓄積した情報をウェブアプリケーションで環境情報グラフとして円滑に共有可能にする機能である。この機能によってユーザは、野菜育成動画と合わせて環境情報グラフを共有することにより、環境の変化や野菜に加えられた工程を時間軸で連続的に把握することができ、必要な野菜育成情報を容易に見つけ出すことができる。そして、従来、文書や写真での共有が難しかった動作の伴う野菜の育成情報を円滑に共有することが可能になる。この機能を通して定期的に行う工程のタイミングについての理解を促す効果が期待できる。例えば追肥の場合、初心者は肥料を与える間隔が短かすぎて野菜の根を傷めてしまう傾向がある。このシステムによって初心者が経験者の追肥のタイミングを理解を深めることで、未然に失敗を防ぐことができる。

3.5.3 投稿情報共有機能

投稿情報共有機能は、野菜育成動画と環境情報グラフに対して、不特定多数のユーザが投稿を加えることを可能にする機能である。投稿された情報は、ウェブアプリケーションで野菜育成動画と環境情報グラフと共にユーザに共有される。この機能によって、ユーザは野菜育成動画に対する投稿情報を読むことで動画を閲覧する手間を省いて内容を把握することが可能になり、必要な野菜育成動画を見つけ出すまでの負担を抑えられる。また、野菜育成動画と環境情報グラフの内

容を補うことで、動作の伴う野菜の育成情報の理解をより効果的に促す効果が期待できる。

3.5.4 野菜育成ダイジェスト動画自動共有機能

野菜育成ダイジェスト動画自動共有機能は、1日ごとの野菜の状態と家庭菜園の環境の変化を高速で表示する動画を自動的に共有可能にする機能である。野菜の状態の変化は緩やかで、野菜の生長や野菜に施した工程がおよぼす影響を育成者以外のユーザが短い時間で把握することは難しい。この機能によって、ユーザは野菜と家庭菜園の環境の変化を短い時間で把握することができ、必要な野菜育成動画を見つけ出すことが容易になる。また、育成の過程を連続して提示することにより、動作の伴う野菜の育成情報の理解をより効果的に促す効果が期待できる。例えば、水やりの場合、ユーザはこの機能によって野菜の茎や葉の状態の変化を短い時間で把握することができ、水を与える適切なタイミングについての理解を深めることができる。

第4章 設 計

4.1. 概要

本章では野菜育成情報自動共有システムの設計について説明する。まず、野菜育成情報自動共有システムで扱う情報について説明した後、その情報を共有するためのシステムアーキテクチャを説明する。続いてシステムアーキテクチャを構成するクライアントサイドシステムとサーバサイドシステムの設計について述べる。

4.2. 野菜育成情報自動共有システムで扱う情報

野菜育成情報自動共有システムの機能として、野菜育成動画自動共有機能、環境情報グラフ自動共有機能、投稿情報共有機能、野菜育成ダイジェスト動画自動共有機能を提案した。

野菜育成情報自動共有機能は、ユーザが家庭菜園において野菜に加えた工程を野菜育成動画としてサーバに自動的に蓄積し、ウェブアプリケーションを通して複数のユーザに共有可能にする機能である。この機能では、複数のユーザから家庭菜園の様子を映したストリーミング映像を取得する。取得したストリーミング映像から、ユーザが野菜に加えた工程の映像を保存し、野菜育成動画として蓄積する。そして、蓄積した野菜育成動画をウェブアプリケーションで複数のユーザに提供する。

環境情報グラフ自動共有機能は、家庭菜園の環境情報をサーバに自動的に蓄積し、蓄積した情報をウェブアプリケーションで環境情報グラフとして共有可能に

する機能である。この機能では、複数のユーザから家庭菜園の環境情報を自動的に取得し、蓄積する。そして、蓄積した環境情報を環境情報グラフとしてウェブアプリケーションで複数のユーザに提供する。

投稿情報共有機能は野菜育成動画と環境情報グラフに対する投稿情報を共有可能にする機能である。この機能では、ウェブアプリケーションに訪れる複数のユーザが投稿する情報を取得し、蓄積する。そして、蓄積した投稿情報をウェブアプリケーションで複数のユーザに提供する。

野菜育成ダイジェスト動画自動共有機能は、一日ごとの野菜の状態と家庭菜園の環境の変化を高速で表示する動画を自動的に共有可能にする機能である。この機能では、複数のユーザから家庭菜園の様子を映したストリーミング画像を取得し、蓄積する。一日ごとに蓄積したストリーミング画像を合成し、野菜育成ダイジェスト動画として保存する。そして、蓄積した野菜育成ダイジェスト動画をウェブアプリケーションを通して複数のユーザに提供する。

野菜育成情報自動共有システムは、ユーザ側に設置されたシステムから取得する情報を識別するために、システム情報を環境情報と同時に取得し、関連づけて保存する。ウェブアプリケーションでは、システム情報にもとづいて、野菜育成動画、環境情報グラフ、投稿情報、野菜育成ダイジェスト動画をユーザごとに体系的に閲覧することを可能にする。

4.3. システムアーキテクチャ

野菜育成情報自動共有システムは、複数のユーザからストリーミング映像、環境情報、投稿情報、ストリーミング画像、システム情報を取得する。そして複数のユーザにシステムに蓄積された、野菜育成動画、環境情報グラフ、投稿情報、野菜育成ダイジェスト動画を提供する。これらの情報のやり取りを満たすためにシステムアーキテクチャを図4.1の様に設計する。野菜育成情報自動共有システムは複数のクライアントサイドシステムとサーバサイドシステムで構成される。

クライアントサイドシステムは家庭菜園を行うユーザの環境に設置されるシステムで、ストリーミング映像、ストリーミング画像、環境情報、システム情報を

サーバサイドシステムに自動的に送信する。このシステムによって、野菜育成動画と環境情報をサーバにアップロードするまでの負担を大幅に低減することができる。

サーバサイドシステムは、複数のクライアントサイドシステムからストリーミング映像、環境情報、投稿情報、ストリーミング画像、システム情報を取得し、蓄積する。また、ウェブアプリケーションを通して、野菜育成動画、環境情報グラフ、投稿情報、野菜育成ダイジェスト動画を提供する。ユーザはウェブアプリケーションで、それらの情報をユーザごとに体系的に取得することができる。そして、このシステムによって、サーバに蓄積された野菜育成動画と環境情報から必要な情報を見つけ出すまでの負担を低減することができる。

このシステムアーキテクチャを採用することにより、野菜育成動画自動共有機能、環境情報グラフ自動共有機能、投稿情報共有機能、野菜育成ダイジェスト動画自動共有機能を実現できる。その結果、動作の伴う野菜の育成情報の共有を円滑にし、野菜の育成方法の理解を促す効果が期待できる。また、動作の伴う野菜育成情報の共有を円滑にする際に課題であった、野菜育成動画と環境情報をサーバにアップロードするまでの負担と、サーバに蓄積した野菜育成動画と環境情報から必要な情報を見つけ出すまでの負担を解消することができる。

4.4. クライアントサイドシステムの設計

クライアントサイドシステムは家庭菜園を行うユーザの環境に設置されるシステムで、ストリーミング映像、環境情報、システム情報、ストリーミング画像をサーバサイドシステムに自動的に送信する。

4.4.1 クライアントサイドシステムの構成

図 4.2 にクライアントサイドシステムの構成を示す。クライアントサイドシステムは環境情報取得用マイクロコントローラと、野菜育成情報送信用マイクロコントローラによって構成される。環境情報取得用マイクロコントローラは、実装されている複数の環境センサから家庭菜園の環境情報を取得し、野菜育成情報送

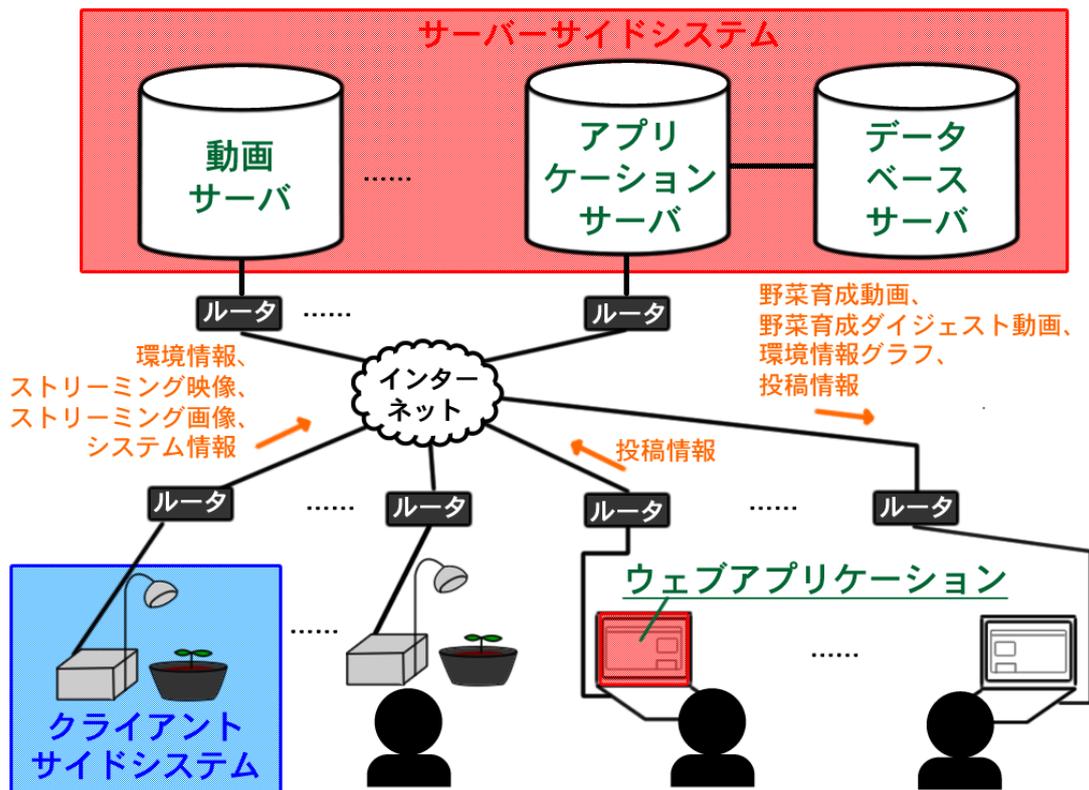


図 4.1 システムアーキテクチャ

信用マイクロコントローラに送信する。野菜育成情報送信信用マイクロコントローラは、接続されているカメラからストリーミング映像とストリーミング画像を取得し、環境情報取得用マイクロコントローラから環境情報を取得する。そして、野菜育成情報送信信用マイクロコントローラ内部のシステム情報を付加してサーバサイドシステムに送信する。

4.4.2 環境情報取得用マイクロコントローラの役割

環境情報取得用マイクロコントローラは実装されている複数の環境センサから環境情報を毎秒ごとに取得し、野菜育成情報送信信用マイクロコントローラに同時に送信する。実装されている複数の環境センサは、2つのモーションセンサ、温

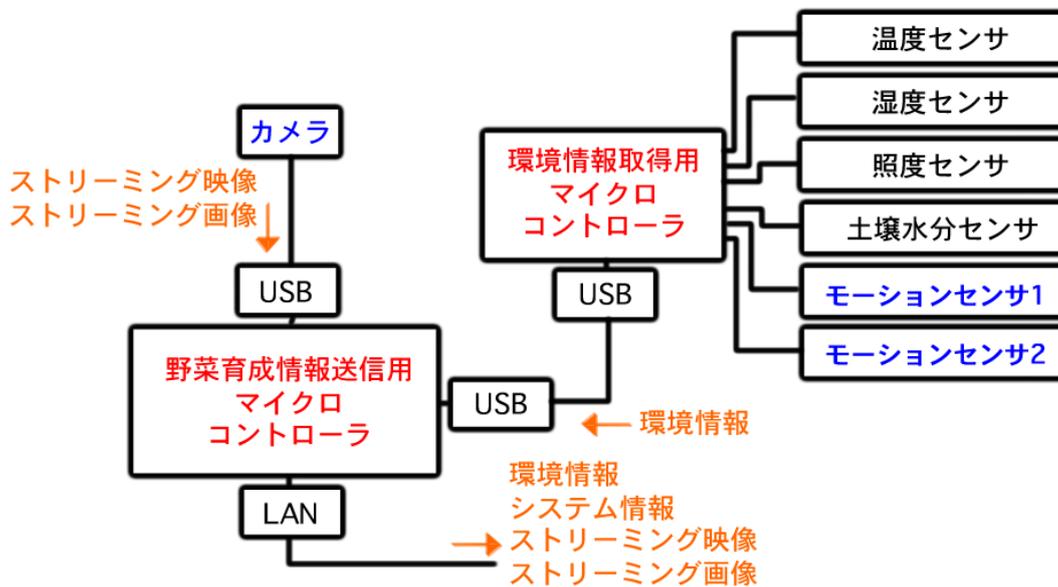


図 4.2 クライアントサイドシステム構成

度センサ、湿度センサ、照度センサ、土壌水分センサ、である。

モーションセンサは、野菜に加える工程の動作を検出する役割を果たす。サーバサイドシステムは、モーションセンサの検出値を含む環境情報を毎秒ごとに受信することによって、ユーザが野菜に工程を施している時間をリアルタイムに知ることができる。これにより、サーバサイドシステムは、クライアントサイドシステムから受信するストリーミング映像からユーザが野菜に工程を施している時間の映像を保存して、野菜育成動画を蓄積できる。また、2つのモーションセンサを実装することによって、動作検出の誤差を低減できる。

モーションセンサ以外の環境センサは、家庭菜園の環境の変化を測定する役割を果たす。例えば、土壌水分センサは土壌中の水分量を相対値として検出する。その値を含む環境情報をサーバサイドシステムが蓄積し、ウェブアプリケーションで環境情報グラフとして土壌の水分量の変化をわかりやすく提示することによって、ユーザは水やりを行うタイミングを容易に把握することができる。

4.4.3 野菜育成情報送信用マイクロコントローラの役割

野菜育成情報送信用マイクロコントローラは、ストリーミング映像、ストリーミング画像をコンスタントにサーバサイドシステムに送信する。また、環境情報とシステム情報を毎秒ごとに送信する。このマイクロコントローラには、ストリーミング映像とストリーミング画像を取得するためにカメラが接続されている。そして、環境情報を取得するために環境情報取得用マイクロコントローラが接続されている。

ストリーミング映像とストリーミング画像は、マイクロコントローラの内部で動作するストリーミングサーバによって配信される。サーバサイドシステムはこれらの情報を受信し保存することで、野菜育成動画と野菜育成ダイジェスト動画の蓄積が可能になる。

環境情報とシステム情報は、マイクロコントローラの内部のプログラムによって毎秒ごとに送信される。システム情報は、マイクロコントローラのイーサネットの物理アドレスである。サーバサイドシステムは、環境情報を毎秒ごとに受信し、蓄積することが可能になる。

4.5. サーバサイドシステムの設計

サーバサイドシステムは、複数のクライアントサイドシステムから情報を取得し、野菜育成動画、環境情報、野菜育成ダイジェスト動画、を蓄積する。また、蓄積した情報の共有を可能にするウェブアプリケーションを提供する。

4.5.1 サーバサイドシステムの構成

図 4.3 にサーバサイドシステムの構成を示す。サーバサイドシステムはアプリケーションサーバ、データベースサーバ、動画サーバによって構成されている。アプリケーションサーバは、クライアントサイドシステムから受信する情報の管理、動作サーバの管理、ウェブアプリケーションの提供を行う。データベースサーバは、アプリケーションサーバを通して情報の蓄積を行う。

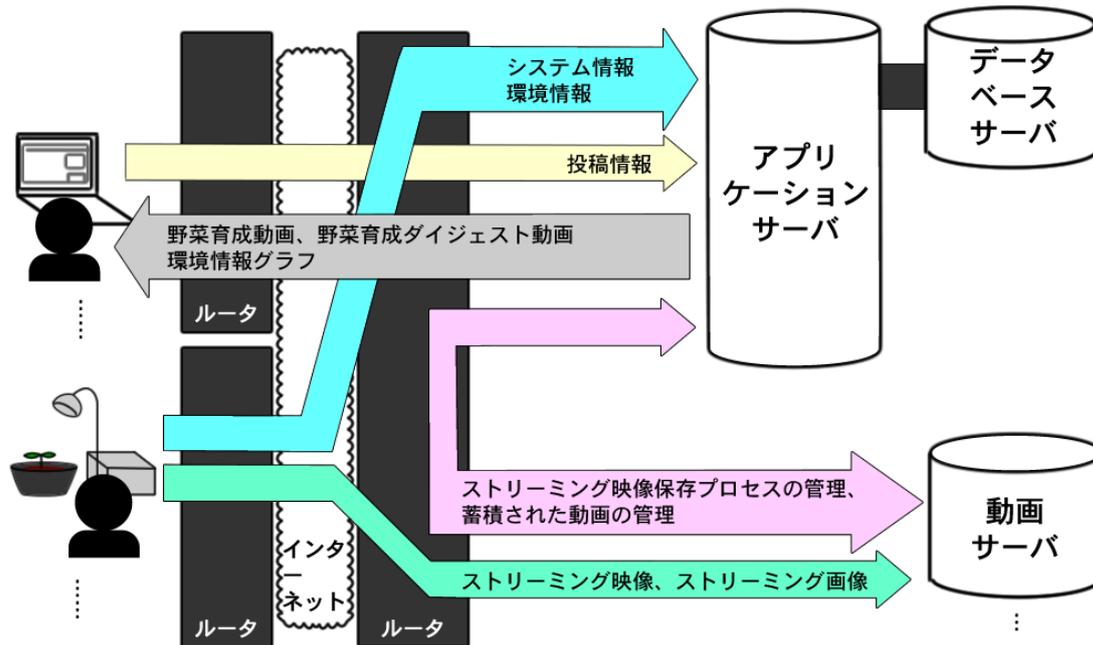


図 4.3 サーバサイドシステム構成

4.5.2 アプリケーションサーバの役割

アプリケーションサーバは (1) クライアントサイドシステムから受信する情報の管理、(2) 動画サーバの管理、(3) ウェブアプリケーションの提供、を行う。

(1) クライアントサイドシステムから受信する情報の管理

アプリケーションサーバは、クライアントサイドシステムから環境情報とシステム情報を毎秒ごとに受信する。環境情報は家庭菜園の環境の、温度、湿度、土壌水分量、照度、2つのモーションセンサの検出値、である。システム情報は、野菜育成情報送信用マイクロコントローラの物理アドレスである。

環境情報とシステム情報を受信後、アプリケーションサーバは 30 秒間隔でこれらの情報に加えて、情報を保存する日時、過去 30 秒間に情報を受信した回数、過去 30 秒間に 2つのモーションセンサが動作を検出した回数、情報送信元の IP アドレス、をデータベースサーバに保存する。これらの情報を保存することで、環境情報グラフに動作検出の回数を割合で表示することができる。これにより例

例えば、ユーザが環境情報グラフを閲覧するだけで、工程中の動きの量が分かり、野菜に加えた工程の内容を容易に推測可能になる。また情報送信元の IP アドレスをデータベースサーバに保存することで、動画サーバは情報送信元のストリーミング映像やストリーミング画像を取得可能になる。

(2) 動画サーバの管理

アプリケーションサーバは、動画サーバのストリーミング映像保存プロセスの管理と動画サーバに蓄積されている動画の管理を行う。

動画サーバのストリーミング映像保存プロセスを管理するために、アプリケーションサーバは特定のクライアントサイドシステムのストリーミング映像保存開始命令と、保存停止命令を動画サーバに送信する。そのダイアグラムを図 4.4 に示す。アプリケーションサーバは毎秒ごとにクライアントサイドシステムから環境情報を受信する。環境情報には 2 つのモーションセンサの検出値が含まれており、これによりアプリケーションサーバはユーザが野菜に工程を施している時間を知ることができる。そして、2 つのモーションセンサが動作を検出した場合、アプリケーションサーバは動画サーバにストリーミング映像保存開始命令を送信する。保存開始命令には、ターゲットとなるクライアントサイドシステムの IP アドレス、物理アドレス、命令を送信した日時が含まれる。動画サーバはそれらの情報を知ることにより、ストリーミング映像の保存と野菜育成動画の蓄積をクライアントサイドシステムごとに体系的に行うことができる。保存開始命令の送信と同時に、クライアントサイドシステムはデータベースサーバにターゲットの IP アドレス、物理アドレス、命令を送信した時刻をプロセスの最終確認日時として一時的に保存する。アプリケーションサーバはその六十秒後に、モーションセンサの動作検出履歴を確認する。過去六十秒間に動作が検出されていた場合、動画サーバはプロセスの最終確認日時を更新する。その結果、ストリーミング映像保存プロセスは継続する。一方、過去六十秒間に動作が検出されなかった場合、アプリケーションサーバは動画サーバにストリーミング映像保存停止命令を送信する。その結果、ストリーミング映像保存プロセスが中断され、野菜育成動画は保存される。この仕組みにより、ユーザが野菜に施した工程の動画を保存するこ

とができ、野菜育成動画を蓄積することが可能になる。

動画サーバに蓄積されている動画の管理は、動画サーバが出力する蓄積された動画のリストをアプリケーションサーバが参照することにより行われる。アプリケーションサーバは、そのリストから蓄積された野菜育成動画の情報をデータベースサーバに送信し、保存する。これにより、ユーザにウェブアプリケーションを通して動画サーバに保存された特定の野菜育成動画を提供することができる。

(3) ウェブアプリケーションの提供

ウェブアプリケーションは、サーバサイドシステムに蓄積された情報をユーザに提供し、動作の伴う野菜の育成情報の円滑な共有を可能にする。その役割を果たすためにウェブアプリケーションを、インデックスページ、環境グラフ共有ページ、野菜育成動画共有ページ、ユーザ登録/ログインページの3層で構成する。図4.5にウェブアプリケーションの構成を示す。

インデックスページの役割 インデックスページは、ユーザが全てのユーザの野菜の育成状況を短い時間で把握することを可能にする。このページでは全てのユーザの野菜育成ダイジェスト動画とストリーミング映像を閲覧することができる。野菜育成ダイジェスト動画によって、野菜の育成状況と家庭菜園の環境の変化を短い時間で把握することができる。また、ストリーミング映像によって、現在の野菜の育成状況と家庭菜園の環境を把握することが可能になる。そして、インデックスページで複数のユーザの野菜の育成状況を把握した後、その中から特定のユーザを選択し、環境情報グラフ共有ページにアクセスする。

環境情報グラフ共有ページの役割 環境情報グラフ共有ページは、特定のユーザの特定の日付の野菜の育成状況と家庭菜園の環境の変化を短い時間で把握することを可能にする。このページでは、ユーザは特定のユーザの環境情報グラフ、投稿情報、野菜育成ダイジェスト動画を日付ごとに閲覧することができる。環境情報グラフは家庭菜園の環境の、温度、湿度、土壌水分量、照度、モーションセンサの検出値、の相対的な変化を時間軸で表示する。投稿情報は、環境情報グラフと

野菜育成動画に対して投稿された情報である。また、ユーザはウェブアプリケーションにログインすることで投稿を行うことができる。ユーザは、特定のユーザの特定の日付の野菜の育成状況と家庭菜園の環境の変化を把握した後、別の日付にアクセスしたり、野菜育成動画共有ページにアクセスすることができる。環境情報グラフ共有ページによって、定期的に行う工程の理解を特に深めることができる。例えば、水やりのタイミングを環境情報グラフで表示される関数の変化や投稿情報から容易に把握でき、その工程による影響を野菜育成ダイジェスト動画で把握することができる。

野菜育成動画共有ページの役割 野菜育成動画共有ページは、特定のユーザが野菜に施した特定の工程の内容を容易に把握することを可能にする。このページでは、ユーザは環境情報グラフ、野菜育成動画、投稿情報を特定の時間帯に絞って閲覧することができる。また、ウェブアプリケーションにログインすることで投稿を行うことができる。野菜育成動画共有ページによって2つの効果が期待できる。第一に、ユーザは工程を施す過程における複数の動作の理解を容易に深めることができる。例えば水やりの場合、その水の与え方や分量を野菜の育成状況に応じて調整する必要があるが、野菜育成動画と時間帯で絞り込んだ環境情報グラフによってその工程の理解を深めることが可能になる。第二に、サーバに蓄積された複数の野菜育成動画と環境情報から必要な情報を見つけ出すまでの負担を低減することができる。ユーザはこのページで環境情報グラフや野菜育成動画を時間帯で再帰的に絞り込むことが可能である。また、同時に野菜育成動画に対して投稿された情報や環境情報グラフを参照することで、野菜育成動画を閲覧する手間を省いてその動画の内容を把握することができる。

ユーザ登録/ログインページの役割 ユーザ登録/ログインページは、ウェブアプリケーションへのユーザ認証とクライアントサイドシステムの登録を行うことを目的にしている。ユーザは、新規ユーザ名と新規パスワードとクライアントサイドシステムの野菜育成情報送信用マイクロコントローラの物理アドレスを入力する事でユーザ登録ができる。またクライアントサイドシステムを持たないユーザは、新規ユーザ名と新規パスワードを入力することでユーザ登録ができる。一度、

登録された情報はユーザ名とその該当するパスワードを指定する事でウェブアプリケーションにログインできる。ログインする事でユーザは環境情報グラフや野菜育成動画に対してコメントを加えることが可能になる。

4.5.3 データベースサーバの役割

データベースサーバの役割は、アプリケーションサーバから受信する情報の蓄積である。データベースサーバが保存する情報は主に、ユーザの情報、環境情報、動画サーバに蓄積した動画の情報、動画サーバに蓄積した画像の情報、投稿情報である。データベースの情報の入出力はアプリケーションサーバを介して行われる。これにより、外部からのアクセスに対する隠ぺい性を高めることができる。

4.5.4 動画サーバの役割

動画サーバの役割は2つある。第一に、野菜育成動画の蓄積と管理である。動画サーバは、アプリケーションサーバの命令に応じて特定のクライアントサイドシステムのストリーミング映像の保存を行い、ユーザがブラウザで閲覧できるファイル形式に変換して野菜育成動画を保存する。野菜育成動画は、ユーザや日付ごとに分類され管理される。第二に、野菜育成ダイジェスト動画の蓄積と管理である。動画サーバは、毎分ごとに全てのクライアントサイドシステムからストリーミング画像を取得し、ユーザと日付ごとに分類して蓄積する。蓄積された画像は、ストリーミング画像の保存と同時に動画に合成される。合成された動画は、ユーザと日付ごとに分類され保存され、ユーザがブラウザで閲覧できるファイル形式に変換されて、野菜育成ダイジェスト動画として蓄積される。これにより、ウェブアプリケーションを訪れるユーザは、常に更新された野菜育成ダイジェスト動画を閲覧することができる。また、動画サーバはアプリケーションサーバとデータベースサーバから独立して設計されており、状況に応じてスケラブルに増設することが可能となっている。

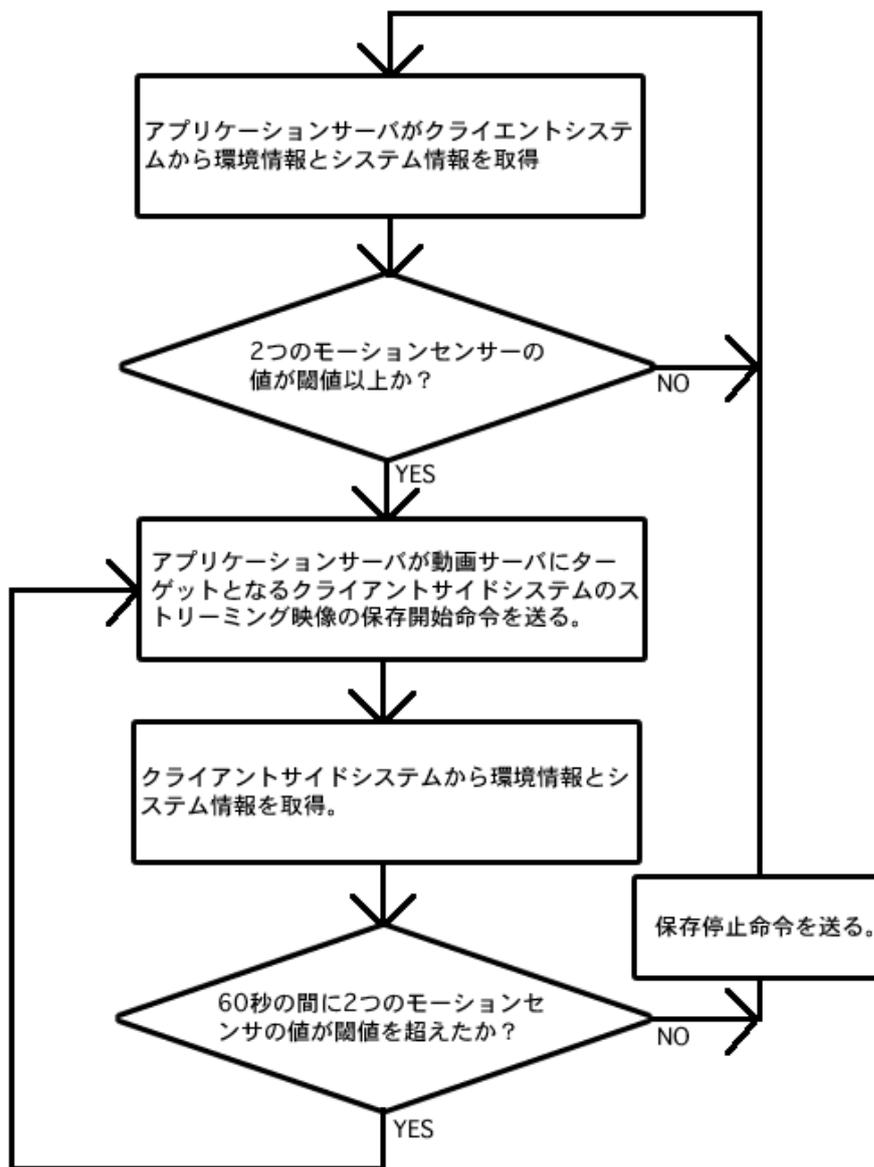


図 4.4 ストリーミング映像保存プロセスのダイアグラム

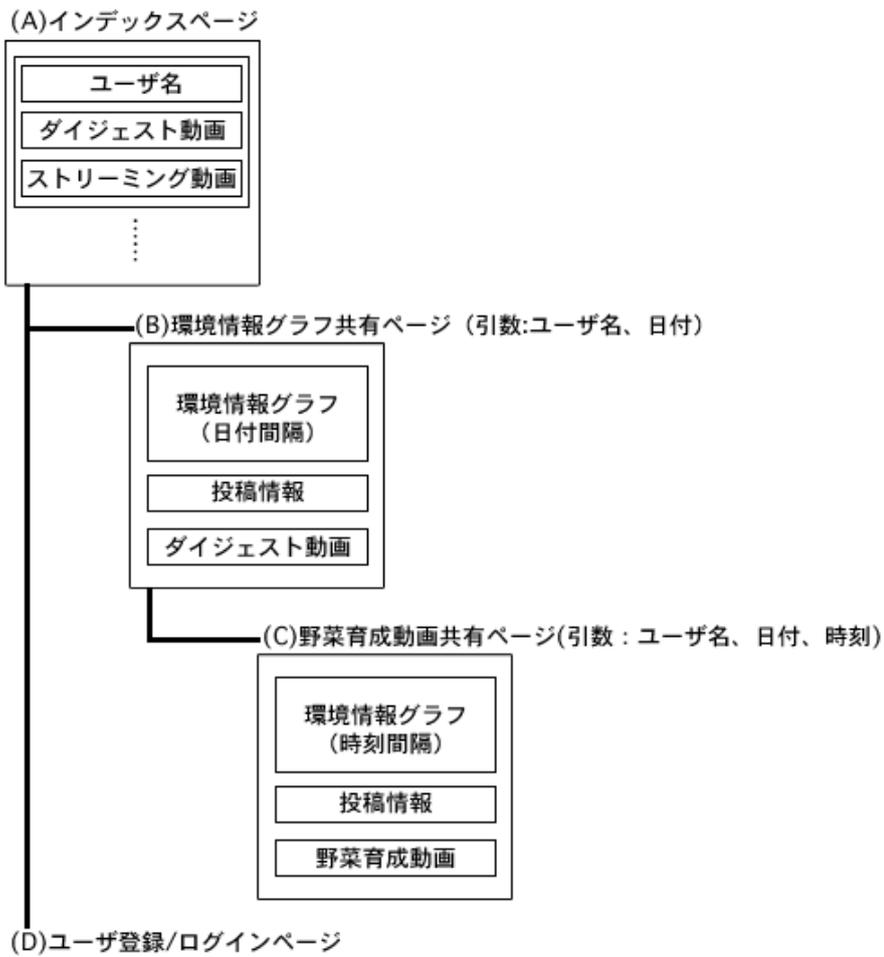


図 4.5 ウェブアプリケーションの構造

第5章 実装

5.1. 概要

本章では、野菜育成情報自動共有システムの実装について説明する。野菜育成情報自動共有システムはクライアントサイドシステムとサーバサイドシステムから構成される。

クライアントサイドシステムは、環境情報取得用マイクロコントローラと野菜育成情報送信用マイクロコントローラによって構成される。ここでまず、それぞれのマイクロコントローラのハードウェアとソフトウェアの実装について述べる。

一方、サーバサイドシステムは、データベースサーバ、アプリケーションサーバ、動画サーバによって構成されている。それぞれ、データベースの実装、ウェブアプリケーションの実装、野菜育成動画及び野菜育成ダイジェスト動画を保存する機能の実装について述べる。

5.2. クライアントサイドシステム

クライアントサイドシステムは環境情報取得用マイクロコントローラ、野菜育成情報送信用マイクロコントローラ、カメラ、環境センサによって構成される。図5.1にクライアントサイドシステムの構成を写真で示す。また、図5.3にクライアントサイドシステムの回路図を示す。環境情報取得用マイクロコントローラは、各環境センサから電圧値を取得し、USBを介して外部に毎秒ごとにシリアル送信する。野菜育成情報送信用マイクロコントローラは、環境情報取得用マイクロコントローラから環境情報を取得し、カメラからストリーミング映像とストリーミ

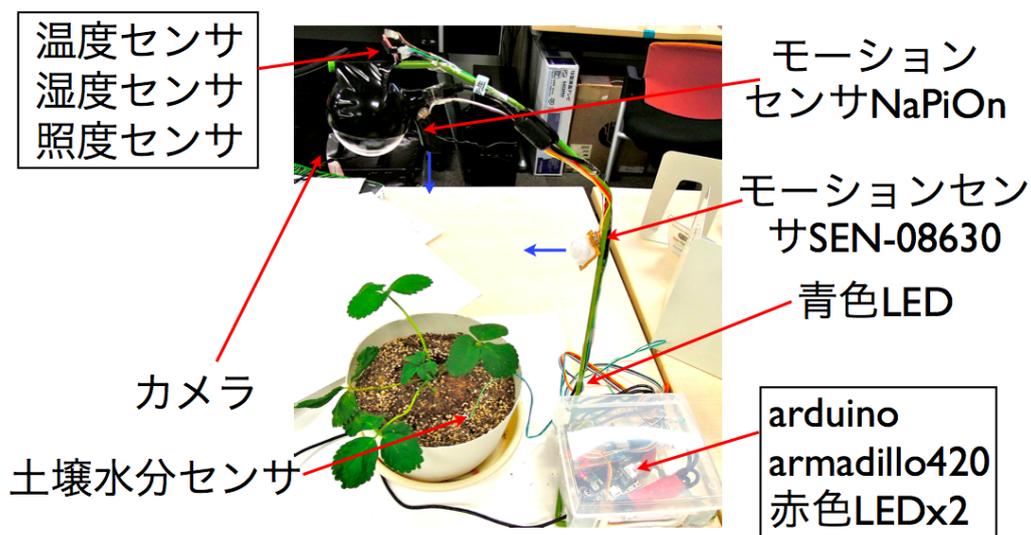


図 5.1 実装したクライアントサイドシステム

ング画像を取得する。そして、内部のシステム情報と環境情報を毎秒ごとにサーバサイドシステムへ送信する。また、内部で動作するストリーミングサーバによって、ストリーミング映像とストリーミング画像をコンスタントに配信する。

5.2.1 環境情報取得用マイクロコントローラ

環境情報取得用マイクロコントローラは Arduino uno で実装した。Arduino uno は、AVR マイクロコントローラの種類であり、複数の入出力ポートを備えている。また、Arduino 言語とその統合開発環境から構成されるシステムである。Arduino uno には、複数の環境センサが実装されている。環境センサは、2つのモーションセンサ、温度センサ、湿度センサ、土壌水分センサ、照度センサ、である。Arduino uno は、環境センサから毎秒ごとに電圧値を取得し、USB を介してシリアル送信する。

ハードウェア

- Arduino uno

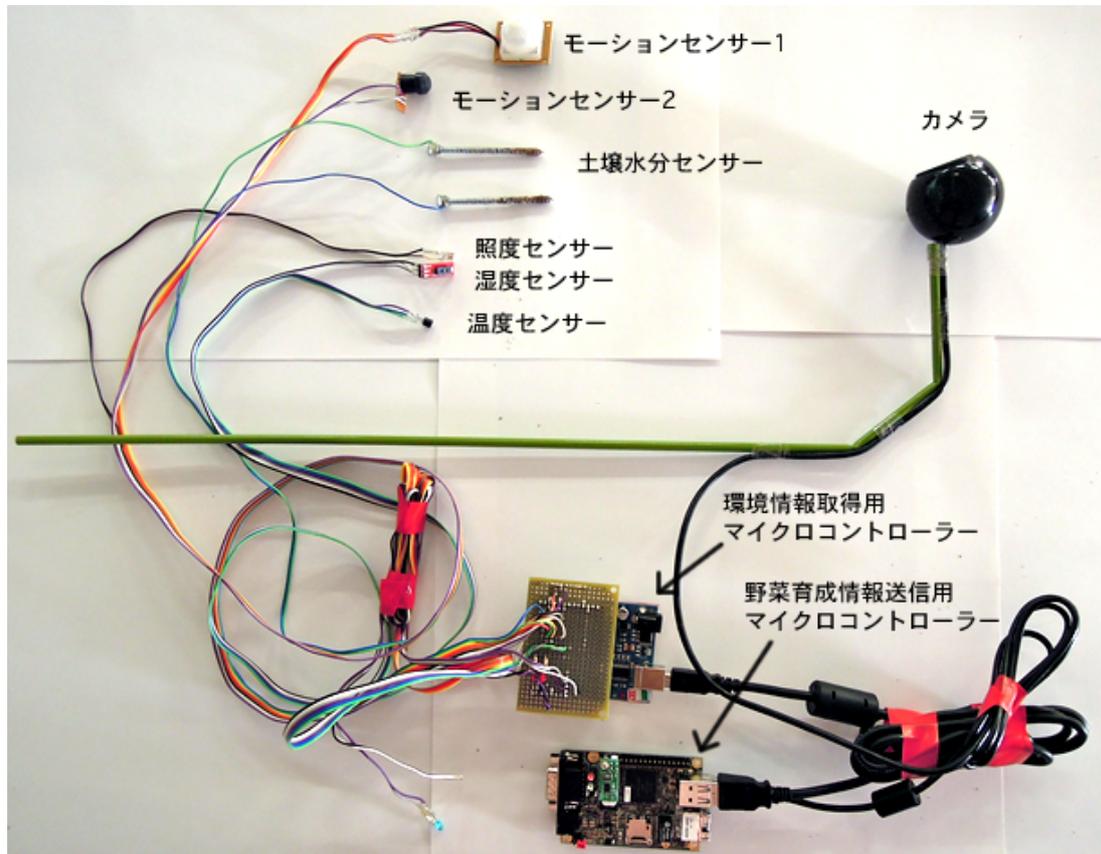


図 5.2 クライアントサイドシステムの構成要素

Arduino uno は ATmega328 を搭載したマイクロコントローラであり、14 のデジタル入出力ピン、6つのアナログ入出力ピンを持つ。また FTDI USB-to-Serial ドライバチップが実装されており、USB を介して容易にシリアル送信が可能である。従って、扱いが容易であるため本システムでは Arduino uno を使用した。

- 環境センサ

環境センサは、2つのモーションセンサ、温度センサ、湿度センサ、土壤水分センサ、照度センサである。

モーションセンサは、システムの利用者が野菜に作業を施す動作を検出するために実装した。本システムでは検出の精度を上げるために SEN-08630

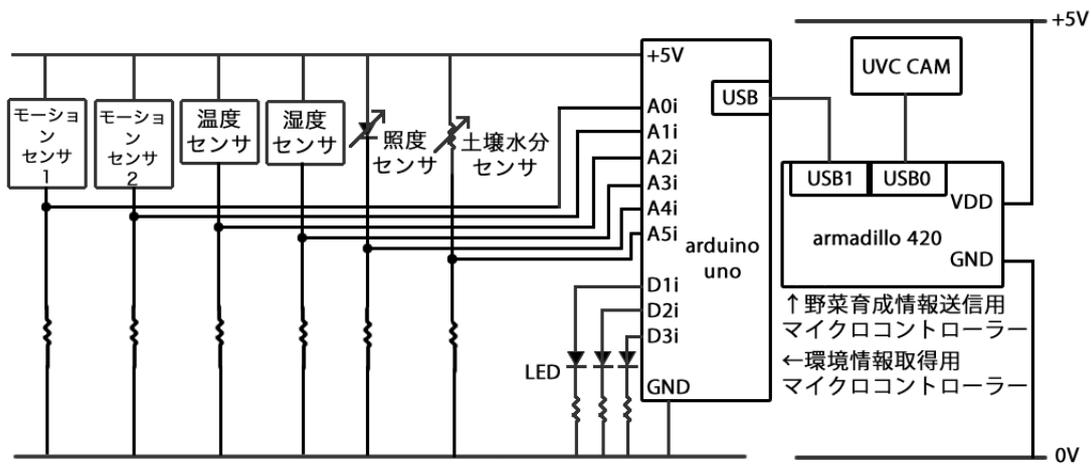


図 5.3 クライアントサイドシステムの回路図

と NaPiOn(微動検出タイプ) の 2 つのモーションセンサを使用した。

SEN-08630 は物体の位置を記憶し、その範囲の中での物体の動きを検知する。電源を与えると周囲の物体の位置を 1 から 2 秒かけて記録する。記録した後、センサの検出範囲内で動きがあった場合、電圧の変化を外部に出力する。検出角度は 120 度である。また、比較的感度が高くこれにより物体の動きを検出できる。

NaPiOn は検知対象と周囲の温度差を検知する。背景との最低必要温度差は 4 度である。また検出距離は 2m となっており、検出範囲は図 5.4 に示すように水平方向に 91 度、垂直方向に 91 度となっている。比較的、検出範囲は狭いが、これにより人の手の動きを検出できる。

温度センサは LM61VIZ を使用した。LM61BIZ は高精度 I C 温度センサであり出力電圧は摂氏温度に対して良好な線形性 (+10mV/度) を持つ。湿度センサは HIH-4030 を使用した。HIH-4030 は動作させると、相対湿度に比例する電圧を出力する。照度センサはフォト I C ダイオード S9066-111 を使用した。S9066-111 は視感度特性に近い分光感度特性であり、照度に対して良好な線形性を持つ。土壌水分センサは、土壌中の水分の変化を検出するために 2 本の釘を使用した。釘の間隔は 5cm とした。これらのセンサを、

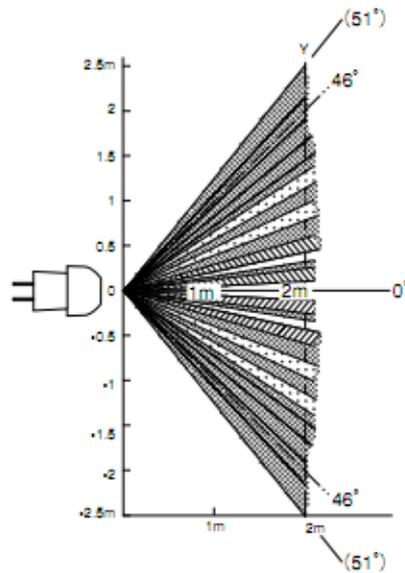


図 5.4 NaPiOn(微動検出タイプの検出範囲)

arduino uno の電源電圧と電圧の検出精度に注意しながらキャリブレーションを行い、相対的な環境の変化を効果的に検出できる様に抵抗を調整し、実装した。

ソフトウェア

Arduino uno のソフトウェアは Arduino IDE 22 を利用して実装した。環境情報取得用マイクロコントローラは毎秒ごとに環境センサの電圧値を取得し、外部にシリアル出力する。またデバッグのために、それぞれのモーションセンサが反応した場合、相当する赤色の LED が点滅し、2つのモーションセンサが反応した場合、青色の LED が点滅するように実装した。

5.2.2 野菜育成情報送信用マイクロコントローラ

野菜育成情報送信用マイクロコントローラは Armadillo-420 で実装した。このマイクロコントローラは、組み込みシステムに求められる機能を搭載した小型汎

用 CPU ボードである。外形サイズは、75mmx50mm で、消費電力は約 1.2W となっている。従って、小型で、省電力で動作し、量産が容易になるように設計されているため、これにより家庭菜園に費やす空間が限られているユーザの負担と開発の負担を抑えることが出来る。

ハードウェア

野菜育成情報送信用マイクロコントローラはカメラと環境情報取得用マイクロコントローラが USB を介して接続されている。

- Armadillo-420

Armadillo-420 はインターフェイスとして、RJ45 (100BASE-TX/10BASE-T, AUTO-MDIX 対応)、USB2.0(High Speed, Full Speed)、RS232C が実装されている。RJ45 には、LAN ケーブルと LAN が接続されている。これによりサーバサイドシステムとの接続が可能になる。USB2.0(High Speed) には、情報の転送速度を考慮して UVC カメラが接続されている。一方、USB2.0(Full Speed) には環境情報取得用マイクロコントローラが接続されている。USB2.0 の 2 つのモードには情報の転送速度に差がある。理論上では Full Speed モードでは 12Mbps、High Speed モードでは 480Mbps であり、UVC カメラによりストリーミング情報を配信するためには High Speed モードでの通信が必要である。RS232C はシリアルケーブルでパソコンに接続することで、フラッシュメモリの書き換えを可能にする。

- UVC 対応カメラ

UVC(USB Video Device Class) カメラは Linux 2.6.26 に組み込まれている UVC ドライバーに対応する規格である。Linux カーネルで安定した動作を実現するため UVC 対応カメラを使用した。

ソフトウェア

Armadillo-420 のルートファイルシステムとカーネルのイメージファイルは、Atmark Dist によって作成した。Atmark Dist は、アットマークテクノ独自のソースコードベースの開発ディストリビューションである。開発環境は、ATDE3で行った。ATDE3 は Armadillo-400 シリーズの標準開発環境として配布されており、Debian GNU/Linux 5.0 をベースに作られている。Armadillo-420 のソフトウェアは、ブートローダ、カーネル、ユーザランドによって構成される。

- ブートローダ

ブートローダは、メーカーのサイトで予め用意されているものを使用した。ブートローダは電源投入後、最初に動作するソフトウェアである。電源投入後、最低限のハードウェア初期化を行い、カーネルイメージとルートファイルシステムイメージをメモリに展開する。

- カーネル

カーネルは、Linux カーネル 2.6.26 を使用した。このカーネルは、プロセス管理（スケジューリング）、時間管理、メモリ管理、デバイスドライバ、プロトコルスタック、ファイルシステムなどの OS としてのコア機能を提供する。電源投入後ブートローダが処理を終えた後、様々なコアやデバイスドライバの初期化処理を行い、ユーザランドの `init` というアプリケーションプログラムを実行する。また、UVC ドライバとそれに依存するアプリケーションが実装されており、安定してストリーミング映像を配信できる。

- ユーザランド

ユーザランドは、Atmark Dist を利用してイメージファイルの作成を行った。ユーザランドは、アプリケーションプログラムやライブラリ、設定ファイル、データファイル、デバイスファイルなど Linux システムが動作する上で必要なカーネル以外のものをさす。ユーザランドをビルドする際に、ATDE3 で標準のユーザランドをカスタマイズし、`cdc-acm` ドライバーと `PHP5.2.6 (cli)` を組み込んだ。`cdc-acm` ドライバーは環境情報取得用マイクロコント

ローラが出力するシリアル情報を取得するために実装した。PHP はサーバサイドシステムが提供するアプリケーションサーバに情報を送信するプログラムを動作させるために実装した。

Linux カーネルは `init` を起動した後、ユーザランドでコンソールの初期化を行い、初期化スクリプトを実行する。初期化スクリプトでは特にストリーミングサーバ MJPG-streamer の起動、環境情報取得用マイクロコントローラから取得する環境情報のファイルへの書き出し、システム情報の書き出し、サーバサイドシステムへ取得した値を送信するプログラム `post.php` の実行を開始する。MJPG-streamer は、低負荷で動作する JPEG ファイルの配信を行うアプリケーションである。MJPG-streamer の動作はインプットプラグインとアウトプットプラグインによって決定される。本システムではインプットプラグインとして `input_ uvc.so`、アウトプットプラグインとして `output_ http.so` を実装した。`input_ uvc.so` は V4L2 が提供する API により UVC 対応のカメラから画像情報を取得する。`output_ http.so` はインプットプラグインから取得した画像をストリーミング配信する。安定して動作させるため調整した結果、配信する Motion jpeg のストリーミングの画質は QVGA、フレームレートは 10fps とした。Arduino uno は環境情報を毎秒ごとに転送速度 11520bps でシリアル出力する。Armadillo は、その情報を受信し、`sensor.dat` に書き出す。`post.php` は、環境情報とシステム情報を毎秒ごとにサーバサイドシステムへ送信する。環境情報は `sensor.dat` より、システム情報は `system.dat` より取得する。

5.3. サーバサイドシステム

サーバサイドシステムはアプリケーションサーバ、データベースサーバ、動画サーバによって構成される。アプリケーションサーバは、環境情報とシステム情報の管理、動画サーバの管理、ウェブアプリケーションの提供を行う。データベースサーバはアプリケーションサーバから受信する情報の保存を行う。動画サーバは、ストリーミング映像とストリーミング画像を受信し、野菜育成動画と野菜育

成ダイジェスト動画の保存を行う。ここでは便宜上、データベースサーバ、アプリケーションサーバ、動画サーバの順に実装の説明をする。

5.3.1 データベースサーバ

データベースサーバは、アプリケーションサーバから受信する情報の保存を行う。この機能を実現するために、MySQL(5.1.52)で実装した。情報を保存するデータベースは armadillo という名前で作成した。armadillo の内部には、armadillos、comments、movies、pictures、sensors というテーブルを作成した。それぞれのテーブルの役割を表 5.1 に示す。

表 5.1 armadillo データベースの構成

テーブル名	役割
armadillos	ユーザの情報を保持する。
sensors	各クライアントサイドシステムが送信する環境情報を保持する。
movies	動画サーバに保存した動画の情報を保持する。
pictures	動画サーバに保存した画像の情報を保持する。
comments	ユーザが投稿した情報を保持する。

続いて、各テーブルのフィールドごとの役割を表 5.2 から表 5.6 に示す。これらのデータベース設計により、アプリケーションサーバから受信する情報の保存を行うことが可能になる。

5.3.2 アプリケーションサーバ

アプリケーションサーバは、環境情報とシステム情報の管理、動画サーバの管理、ウェブアプリケーションの提供を行う。これらの機能を実現するために、アプリケーションサーバは Apache 2.0 で構築した。ウェブアプリケーションは PHP 5.3.2(cli)、フレームワークとして CakePHP 1.3.9 を利用して作成した。CakePHP は素早く安定したウェブアプリケーションを作成するための PHP フレームワー

表 5.2 sensor テーブルのフィールド

フィールド名	データ型	役割
id	int	ユニーク ID を保持する。
ir1	int	モーションセンサ 1 の値を保持する。
ir2	int	モーションセンサ 2 の値を保持する。
humid	int	湿度センサの値を保持する。
illumi	int	照度センサの値を保持する。
temp	int	温度センサの値を保持する。
water	int	土壌水分センサの値を保持する。
date	int	環境情報を加えた時間を保持する。
mac	varchar	送信元の物理アドレスを保持する。
irsum	int	60 秒間の間に 2 つのモーションセンサが同時に 反応した回数を保持する
num	int	60 秒間の間に特定のクライアントサイドシステムが 情報を送信した回数
day	int	環境情報を加えた日付を保持する。
diff	int	時刻の情報 (date-day) を保持する。

表 5.3 movies テーブルのフィールド

フィールド名	データ型	役割
id	int	ユニーク ID を保持する。
date	int	動画を保存した日時を保持する。
day	int	動画を保存した日付を保持する。
diff	int	動画を保存した時刻 (date-day) を保持する。
mac	varchar	動画の保存先のクライアントサイドシステムの 物理アドレスを保持する。

表 5.4 pictures テーブルのフィールド

フィールド名	データ型	役割
id	int	ユニーク ID を保持する。
date	int	画像を保存した日時を保持する。
day	int	画像を保存した日付を保持する。
mac	varchar	画像の保存先のクライアントサイドシステムの物理アドレスを保持する。

表 5.5 armadillos テーブルのフィールド

フィールド名	データ型	役割
id	int	ユニーク ID を保持する。
name	int	ユーザ名を保持する。
mac	int	クライアントサイドシステムの物理アドレスを保持する。
ip	int	クライアントサイドシステムの IP アドレスを保持する。
date	int	日時を保持する。

表 5.6 comments テーブルのフィールド

フィールド名	データ型	役割
id	int	ユニーク ID を保存する。
name	int	ユーザ名を保存する。
to_movie	int	投稿した対象の野菜育成動画名を保存する。
date	int	投稿した日時を保存する。
to_day	int	投稿した日付を保存する。
content	varchar	投稿の内容を保存する。
to_name	int	投稿したユーザ ID を保存する。

クである。MVCソフトウェアパターンに従っており、ウェブアプリケーションをモデル、ビュー、コントローラの3つに分けて構成する。モデルは、データベースサーバと密に連携しており、ウェブアプリケーションのデータの入出力を管理する。ビューは、ウェブアプリケーションのページと関係しており、モデルデータの表示方法を管理する。コントローラはユーザによるリクエストを受信して、ウェブアプリケーションの振る舞いを管理する。これにより実装することにより、素早く安定したウェブアプリケーションを実現できる。

ウェブアプリケーションはインデックスページ、環境情報グラフ共有ページ、野菜育成動画共有ページ、の3層とユーザ登録/ログインページで構成した。ウェブアプリケーションの実装を、MVCソフトウェアパターンに沿って説明する。

ビュー

実装したビューの構成を図5.5に示す。ビューは、index.ctp、demo.ctp、video.ctp、login.ctpの4つである。index.ctpはインデックスページ、demo.ctpは環境情報グラフ共有ページ、video.ctpは野菜育成動画共有ページ、login.ctpはユーザ登録/ログインページに相当する。

- index.ctp

index.ctpはインデックスページを表示する。インデックスページを図5.6に示す。インデックスページは、ユーザが全てのユーザの野菜の育成状況や家庭菜園の環境を短い時間で把握可能にし、必要な野菜の育成動画の参照を容易にする。このページでは、全てのユーザの野菜育成ダイジェスト動画と現在のストリーミング映像を閲覧できる。野菜育成ダイジェスト動画は動画サーバによって提供される。ウェブアプリケーションで閲覧可能にするために、動画の音声圧縮フォーマットはogg、動画圧縮ファイルフォーマットはtheoraとした。このページからユーザは各ユーザの環境情報グラフ共有ページとユーザ登録/ログインページにアクセスできる。

- demo.ctp

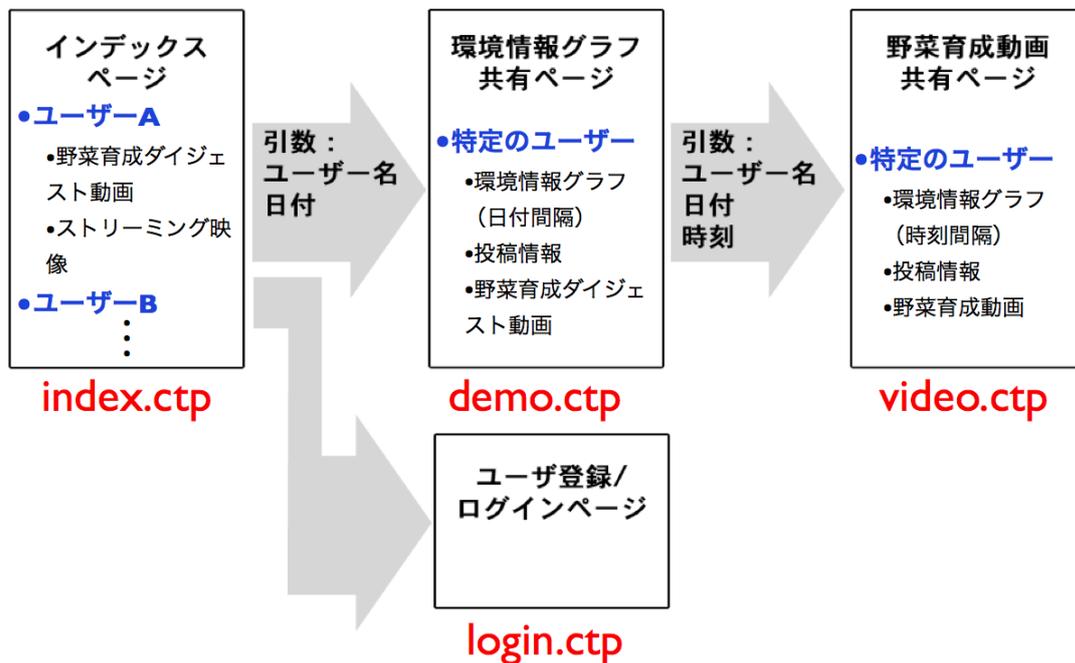


図 5.5 ビュー

demo.ctpは環境情報グラフ共有ページを表示する。環境情報グラフ共有ページを図5.7に示す。環境情報グラフ共有ページは、ユーザが特定のユーザの日付ごとの野菜育成状況と家庭菜園の環境の変化、動画に対する投稿を閲覧することで、野菜育成動画の参照を容易にすることを目的にしている。環境情報グラフ共有ページはユーザ名と日付を引数にする。このページで共有される情報は、引数で指定したユーザの環境情報グラフ、投稿情報、野菜育成ダイジェスト動画である。

図5.8に環境情報グラフの例を示す。ユーザは、環境情報グラフを閲覧することで、家庭菜園の環境に加わった変化を時系列で把握することができる。環境情報グラフは、javascriptのフレームワークjQueryによって実装した。環境情報グラフは、30秒間ごとのモーションセンサの検出値、温度、湿度、照度、土壌水分量の相対的な変化を表示する。横軸は時刻で、データは30秒ごとにプロットされている。縦軸は0から100の相対値である。全ての環境情報を100に正規化することで、ユーザは直感的に変化を把握するこ

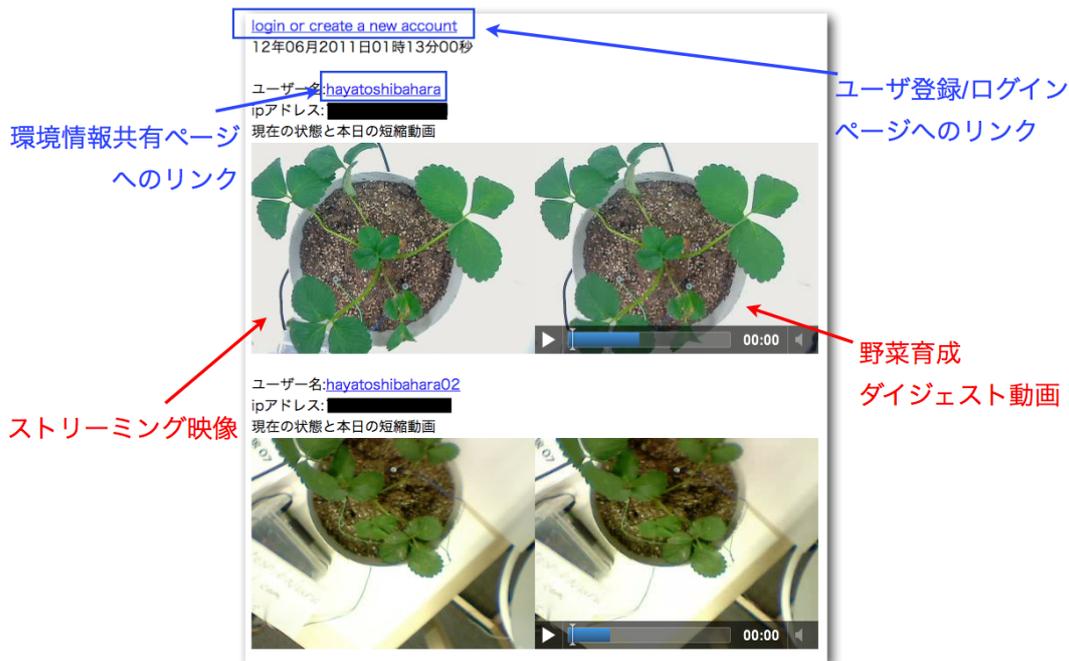


図 5.6 インデックスページ

とができる。ユーザは図中の番号1のリンクを参照することで、別の日付の環境情報グラフにアクセスすることができる。そして、図中の番号2の様に、非連続なグラフを見ることで環境の変化を容易に見つけ出すことが可能になる。例えばこのグラフの場合、土壌水分量とモーションセンサの値が急激に変化している。これにより、ユーザは15時前に水やりの工程が行われたことを推測することができる。これにより、環境情報グラフの下リンクを見るとその時間帯に動画が保存されていることがわかり、そこから野菜育成動画共有ページにアクセスすることができる。この仕組みにより、野菜に施した工程の時間とその変化を把握できるだけではなく、必要な野菜育成動画の取得を容易にすることが可能になる。

野菜育成動画共有ページへのリンクの下には、野菜育成ダイジェスト動画が配置されている。ユーザは環境情報グラフと野菜育成ダイジェスト動画をあわせて見ることにより、家庭菜園の環境の変化に加えて野菜の育成状況の変化を把握することができる。その結果、定期的に行われる工程のタ

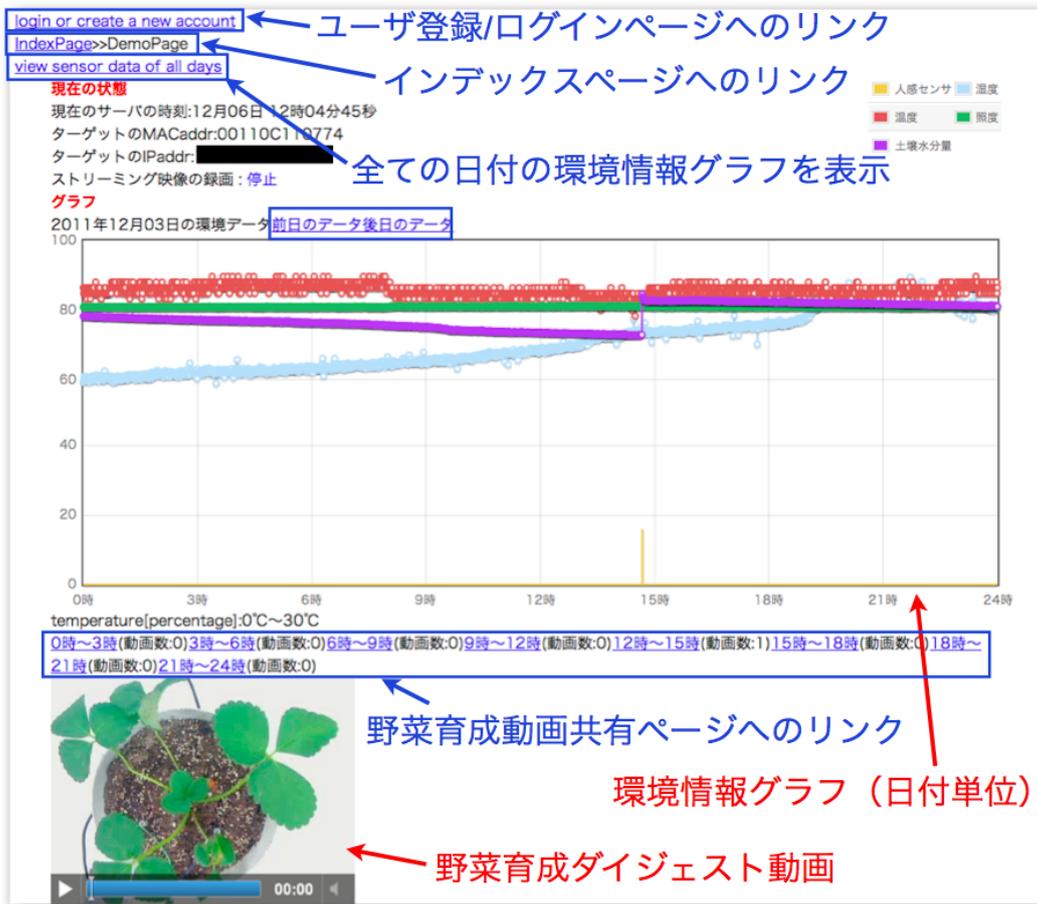


図 5.7 環境情報グラフ共有ページ

イメージを直感的に理解することができ、動作の伴う野菜の育成情報を効果的に理解することが可能になる。

- video.ctp

video.ctp は野菜育成動画共有ページを表示する。野菜育成動画共有ページを図 5.9 に表す。このページは、ユーザが必要な野菜育成動画を見つけ出すまでの負担を低減する。また、動作の伴う野菜育成情報の理解を促進する。野菜育成動画共有ページは、環境情報グラフ、投稿情報、野菜育成動画を表示する。環境情報グラフは、環境情報グラフ共有ページのグラフとは横

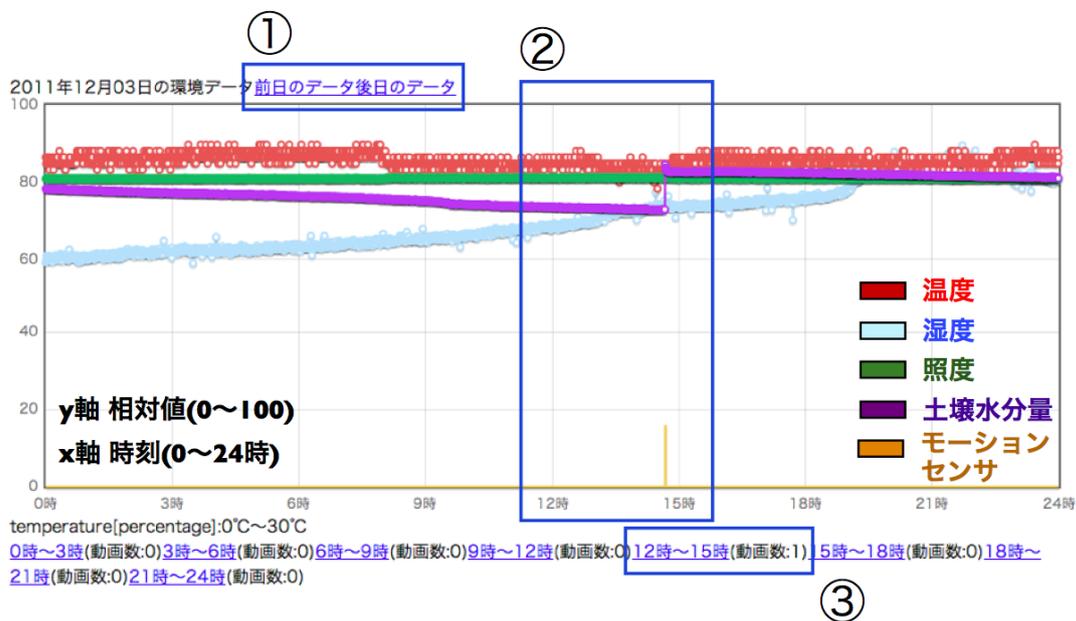


図 5.8 環境情報グラフ

軸のスケールが異なり、時刻単位で表示されている。図中の番号1にある時刻を繰り返すリンクによってユーザは環境情報グラフと野菜育成動画を再帰的に繰り返すことができる。時刻の繰り返すは、横軸のスケールの3分の1ごとに行われる。例えば横軸のスケールが1時間であった場合、リンクをたどっていくことで20分、9分、3分、1分と繰り返すことができる。この仕組みにより、ユーザは環境情報グラフと野菜育成動画を時間帯で絞り込むことが可能になり、必要な野菜育成動画の取得が容易になる。また、ユーザが野菜に施した工程による影響を細かく把握することができる。そして、投稿情報をあわせて閲覧することにより、効果的に動作の伴う野菜育成情報の理解を深めることができる。

続いて、野菜育成動画共有ページで共有される野菜育成動画と投稿情報の例について図5.10に示す。この例では、追肥の工程が記録されている。ユーザは投稿情報を見ることによって、野菜育成動画を見る負担を省いて野菜育成動画の内容を把握することができる。これにより、必要な野菜育成動画を見つけ出すまでの負担を低減することが可能になる。また、投稿情報

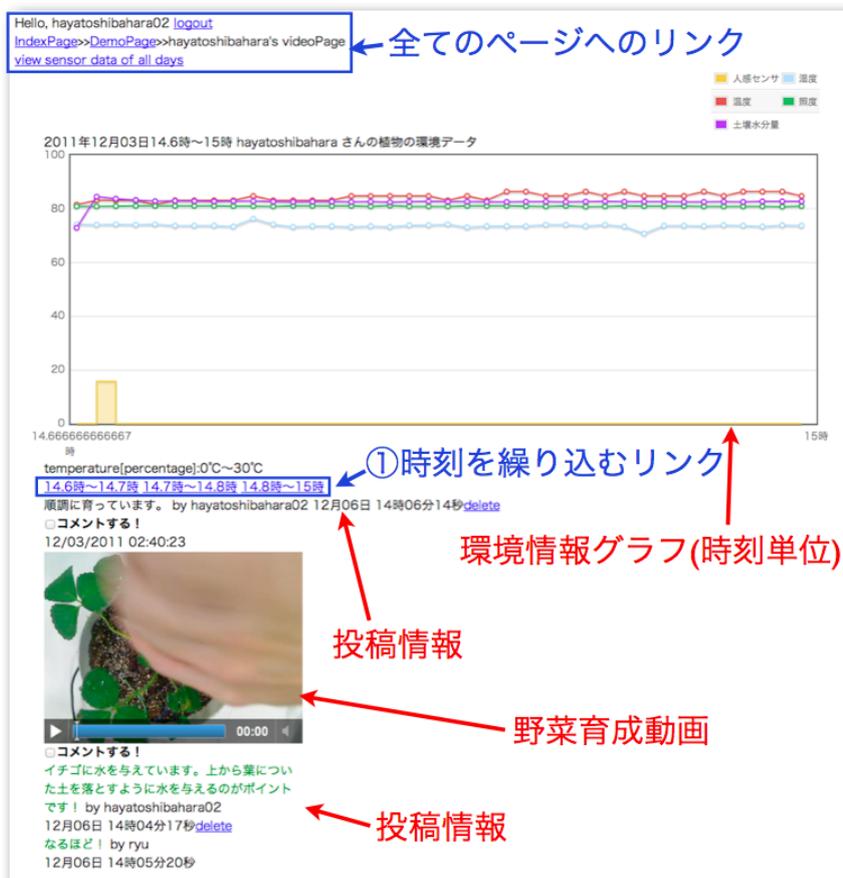


図 5.9 野菜育成動画共有ページ

を見ることによって野菜育成動画の理解を深めることもできる。この例の場合、追肥は根元だけではなく、根元の周囲に肥料を与えることが重要であることが補足されている。野菜育成動画を閲覧すると、苗の周囲に肥料が与えられていることを直感的に理解できる。野菜育成動画によって更に、文書や写真で共有の難しかった野菜育成の情報を共有することが可能になる。この場合、ユーザは動画で工程を注意して観察することにより、肥料の種類、肥料のまき方の速度や分量、苗のよけ方、工程後水を大量に与えていることなどの情報を連続して理解することができる。これにより、容易に動作の伴う野菜の育成情報の理解を効果的に深めることができる。



図 5.10 野菜育成動画と投稿情報の例

- login.ctp

login.ctp はユーザ登録/ログインページを表示する。図5.11に、ユーザ登録/ログインページを示す。このページではウェブアプリケーションでのユーザ認証とクライアントサイドシステムの登録を行うことができる。

モデル

実装したモデルは Sensor モデル、Picture モデル、Movie モデル、Comment モデル、Armadillo モデルである。各モデルはデータベースサーバの armadillo データベース内のテーブルに対応している。それぞれの主な役割として、Sensor モデルは環境情報を管理する。Picture モデルはストリーミング画像を管理する。Movie モデルは野菜育成動画を管理する。Armadillo モデルはユーザ情報を管理する。Comment モデルは投稿情報を管理する。

Please enter password
 create a new user
Name:
Password:
If you have a product and you haven't registered, please fill the below.
MAC address:
[return](#)

図 5.11 ユーザ登録/ログインページ

コントローラ

実装したコントローラは `sensors_controller.php` である。このコントローラは、対応するビューを持つファンクション `index`、`demo`、`video`、`login` と、対応するビューを持たないファンクション `armadillo`、`pictures`、`logout` によって構成されている。

- index ファンクション

index ファンクションは、インデックスページを表示するビュー `index.ctp` に全てのユーザ情報を提供する。ユーザ情報を取得するために、index ファンクションは Armadillo モデルを参照する。これにより、`index.ctp` は全てのユーザの家庭菜園のストリーミング映像と野菜育成ダイジェスト動画をユーザに提供できる。

- demo ファンクション

demo ファンクションは、環境情報共有ページを表示するビュー `demo.ctp` に環境情報、投稿情報を提供する。また、動画サーバが蓄積した野菜育成動画のリストを読み込み、保存する。

環境情報を提供するために demo ファンクションは、URL から引数としてユーザ名と日付を取得する。続いて、Armadillo モデルを参照し、ユーザ名からクライアントサイドシステムの物理アドレスを取得する。そして Sensor モデルを参照し、特定のユーザの環境情報を物理アドレスと日付から取得し、`demo.ctp` に提供する。

一方、投稿情報を提供するために demo ファンクションは、Comment モデルを参照し、ユーザ名と日付から投稿情報を取得する。これにより、demo.ctp は環境情報と野菜育成動画とユーザに提供できる。

動画サーバが蓄積した野菜育成動画のリスト video.dat は、ウェブアプリケーションに独立して動作するプログラムが動画サーバから取得する。demo ファンクションは、video.dat を読み込み、Movie モデルを通して情報を保存する。

- video ファンクション

video ファンクションは、野菜育成動画共有ページを表示するビュー video.ctp に環境情報と投稿情報を提供する。また、動画サーバが蓄積した野菜育成動画のリストを読み込み、保存する。

環境情報を提供するために video ファンクションは、URL から引数としてユーザ名と時刻を取得する。Armadillo モデルを参照し、ユーザ名からクライアントサイドシステムの物理アドレスを取得する。そして Sensor モデルを参照し、特定のユーザの環境情報を物理アドレスと時刻から取得し、video.ctp に提供する。

一方、投稿情報を提供するために video ファンクションは、Comment モデルを参照し、ユーザ名と時刻から投稿情報を取得する。これにより、video.ctp は環境情報と野菜育成動画とユーザに提供できる。

また、野菜育成動画のリスト video.dat を読み込み、Movie モデルを通して情報を保存する。

- login ファンクション

login ファンクションは、ユーザ登録/ログインページを表示するビュー login.ctp にサービスを提供する。

サービスを提供するために、login ファンクションは、ユーザがユーザ登録/ログインページで送信した情報を管理する。受信する情報は、ユーザ登録の場合とログインの場合で異なる。

ユーザ登録の場合、login ファンクションはユーザが入力した、名前、パスワード、クライアントサイドシステムの物理アドレスを取得する。そして、Armadillo モデルを参照して、ユーザが存在していなく、予め保存された物理アドレスと照合がとれ、パスワードが適切に入力されている場合、それらの情報を Armadillo モデルを通して保存する。

ログインの場合、login ファンクションはユーザが入力した、名前、パスワードを取得する。そして、Armadillo モデルを参照して、ユーザが存在し、パスワードと照合する。照合がとれた場合、ユーザ名とユーザの物理アドレスはセッションデータに保持され、ユーザはログインが完了する。

- logout ファンクション

logout ファンクションはセッションデータの破棄を行う。そして、アクセス元のアドレスにリダイレクトさせる。

- armadillo ファンクション

armadillo ファンクションは、クライアントサイドシステムから環境情報とシステム情報を受信し、Sensor モデルと Armadillo モデルを通してデータベースに保存する。また、環境情報に含まれるモーションセンサの検出値をもとに動画サーバのストリーミング映像保存プロセスの管理を行う。

クライアントサイドシステムは毎秒ごとにアプリケーションサーバへ環境情報とシステム情報を armadillo ファンクションに送信する。armadillo ファンクションは、クライアントサイドシステムの環境情報、物理アドレス、IP アドレスを取得する。

環境情報は 30 秒ごとに Sensor モデルを通してデータベースに保存される。この機能を実現するために、armadillo ファンクションは Sensor モデルを参照し、物理アドレスから最後に環境情報を保存した日時を取得する。現在時刻と、その日時の差が 30 秒以上である場合、Sensor モデルを通して環境情報を保存する。

システム情報は毎秒ごとに Armadillo モデルを通してデータベースに保存

される。これにより、ネットワークの状態をアプリケーションサーバが常に知ることができる。

動画サーバのストリーミング映像保存プロセスの管理は、毎秒ごとに受信される環境情報に含まれるモーションセンサの検出値によって行われる。ストリーミング映像保存プロセスのダイアグラムは図 4.4 に示した通りである。図中の、ストリーミング映像保存開始命令と保存停止命令は、動画サーバの `command.php` を通して行われる。ストリーミング映像保存開始命令の場合、`command.php` に `create` というコマンド、日時、ターゲットの物理アドレス、を送信する。また、それと同時に `armadillo` モデルに、保存開始命令を送信した日時を一時的に保存する。一方、ストリーミング映像保存停止命令の場合、`command.php` に `delete` というコマンド、物理アドレス、一時的に保存された保存開始命令を送信した日時を送信する。これにより、動画サーバのストリーミング映像の保存を自動的に行い、野菜育成動画を蓄積することが可能になる。

- `pictures` ファンクション

`pictures` ファンクションは、動画サーバが蓄積したストリーミング画像を管理する。動画サーバは、ストリーミング画像の保存と同時に、`pictures` ファンクションにターゲットの物理アドレスを送信する。`pictures` ファンクションは、情報を取得した日時と物理アドレスを `Picture` モデルを通して保存する。

アプリケーションサーバは、`getVideoList.php` を常に動作させている。これにより、動画サーバが蓄積した野菜育成動画のリスト `video.dat` を毎秒ごとにローカルに保存できる。

5.3.3 動画サーバ

動画サーバの役割の 2 つある。第一に、アプリケーションサーバの命令に応じてストリーミング映像の保存を動的に行い、ユーザがブラウザで閲覧できるファ

イル形式に変換して野菜育成動画を蓄積することである。第二に、ストリーミング画像の保存を毎分ごとに行い、蓄積した画像を合成して動画を出力し、ユーザがブラウザで閲覧できるファイル形式に変換して野菜育成ダイジェスト動画を蓄積することである。また、蓄積した動画をウェブアプリケーションで提供するために、動画サーバではウェブサーバ稼働している。

ここでは便宜上、ウェブサーバのフォルダ構成をまず説明する。続いて、野菜育成動画保存プロセスと野菜育成ダイジェスト動画保存プロセスについて述べる。

ウェブサーバのフォルダ構成

ウェブアプリケーションは、動画サーバのウェブサーバに保存された動画を参照してユーザに提供する。蓄積される動画を体系的に管理するために、ウェブサーバのフォルダ構成は図 5.12 の様に構成した。

まず、html フォルダの内部には armadillo フォルダ、video.dat、command フォルダ、tmp フォルダがある。armadillo フォルダは、全ての、野菜育成動画と野菜育成ダイジェスト動画を保存する。video.dat は、autoMov.php によって出力される armadillo フォルダ内に蓄積された動画のリストである。command フォルダは、アプリケーションサーバからの命令を受信するためのプログラム command.php を保持する。tmp フォルダは、ストリーミング映像保存プロセスを管理するコマンドファイルを保持する。このうち、command フォルダと tmp フォルダは、アプリケーションサーバの IP アドレスからのみアクセスできる様に設定されている。

armadillo フォルダの内部には、クライアントサイドシステムの物理アドレスごとのフォルダがある。これにより、複数のクライアントサイドシステム別に情報を管理できる。物理アドレスごとのフォルダの中には画像フォルダ pics と動画フォルダ movs がある。画像フォルダ pics は、ストリーミング画像と野菜育成ダイジェスト動画を保存する。一方、動画フォルダ movs は野菜育成動画を保存する。画像フォルダ pics には日付ごとのフォルダと複数の野菜育成ダイジェスト動画がある。それぞれの日付ごとのフォルダの内部には、保存した複数のストリーミング画像が保存されている。動画フォルダの内部には、保存した野菜育成動画が保存されている。

アプリケーションサーバが送信するストリーミング映像保存開始命令は command フォルダ内の command.php が受信する。command.php は、受信した情報に基づき、tmp フォルダにコマンドファイルを作成する。

野菜育成動画保存プロセス

ストリーミング映像保存プロセスは、autoMovs.php、movies.php によって管理される。autoMovs.php は、tmp フォルダ内のコマンドファイルに従って movies.php の管理を行う。movies.php は、ターゲットのストリーミング映像を保存し、野菜育成動画を出力する。

autoMovs.php は、tmp フォルダ内のコマンドファイルを監視する。コマンドファイルは、アプリケーションサーバから命令を受信する command.php によって作成される。コマンドファイルのタイトルは以下の書式で保存される。

{ 物理アドレス }-{ 日時 }-{ コマンド }-{ IP アドレス }

物理アドレスと IP アドレスは、ストリーミング映像を保存するターゲットの持つ値である。日時は、コマンドを受信した日時である。コマンドは、0 と 1 と 9 の 3 種類がある。0 はストリーミング映像保存開始命令を受信したことを示す。1 はストリーミング映像の保存を開始したことを示す。9 はストリーミング映像保存停止命令を受信したことを示す。autoMovs.php は、その数字によってストリーミング映像保存プロセスの管理を行う。コマンドが 0 の場合、ファイルに示された物理アドレス、日時、コマンドを引数にして movies.php を実行する。movies.php は、VLC media player(0.9.9a Grishenko) によりストリーミング映像の保存を開始する。動画の保存先は、物理アドレスごとのフォルダ内である。動画の保存名は、日付である。動画のファイル形式は ogg で、ビデオコーデックは theo である。ストリーミング映像を保存開始後、autoMovs.php はコマンドを 0 から 1 に書き換える。そして、アプリケーションサーバから保存停止命令を受信し、コマンド 9 のファイルが作成された場合、autoMovs.php は物理アドレス、日付を引数にして、起動している VLC のプロセスを停止する。また、停止が完了した後に

コマンドファイルを削除する。これにより、動画サーバはアプリケーションサーバの命令にもとづいて野菜育成動画を保存することができる。

野菜育成ダイジェスト動画保存プロセス

野菜育成ダイジェスト動画保存プロセスは、autoPics.php と picture.php によって管理される。autoPics.php は、picture.php のプロセスの管理を行う。picture.php は、物理アドレスと IP アドレスを引数にしてストリーミング画像を保存し、野菜育成ダイジェスト動画を出力する。autoPics.php は、毎分ごとに全てのクライアントサイドシステムの物理アドレスと IP アドレスを引数にして picture.php を実行する。picture.php は、物理アドレスと IP アドレスをもとにターゲットのストリーミング画像を特定のフォルダへ保存する。ストリーミング画像の形式は JPEG である。そして、蓄積されたストリーミング画像の名前を連番にした後、FFmpeg(version0.6.1)により動画に合成する。動画のファイル形式は avi、ビデオコーデックは mpeg4、フレームレートは 10fps である。更に VLC によって、ファイル形式を ogg、ビデオコーデックを theo、フレームレートを 10fps として、動画をエンコードし、野菜育成ダイジェスト動画として保存する。

VLC と FFmpeg のタイムアウト処理を行うため、全てのプロセスは autokill.php によって監視されている。autoMovs.php と autoPics.php は、VLC と FFmpeg のプロセスを開始する際に、プロセス名とタイムアウトと見なす時間を kill.list に追記する。autokill.php はその情報をもとに、タイムアウトしたと見なされるプロセスを自動的に中断する。

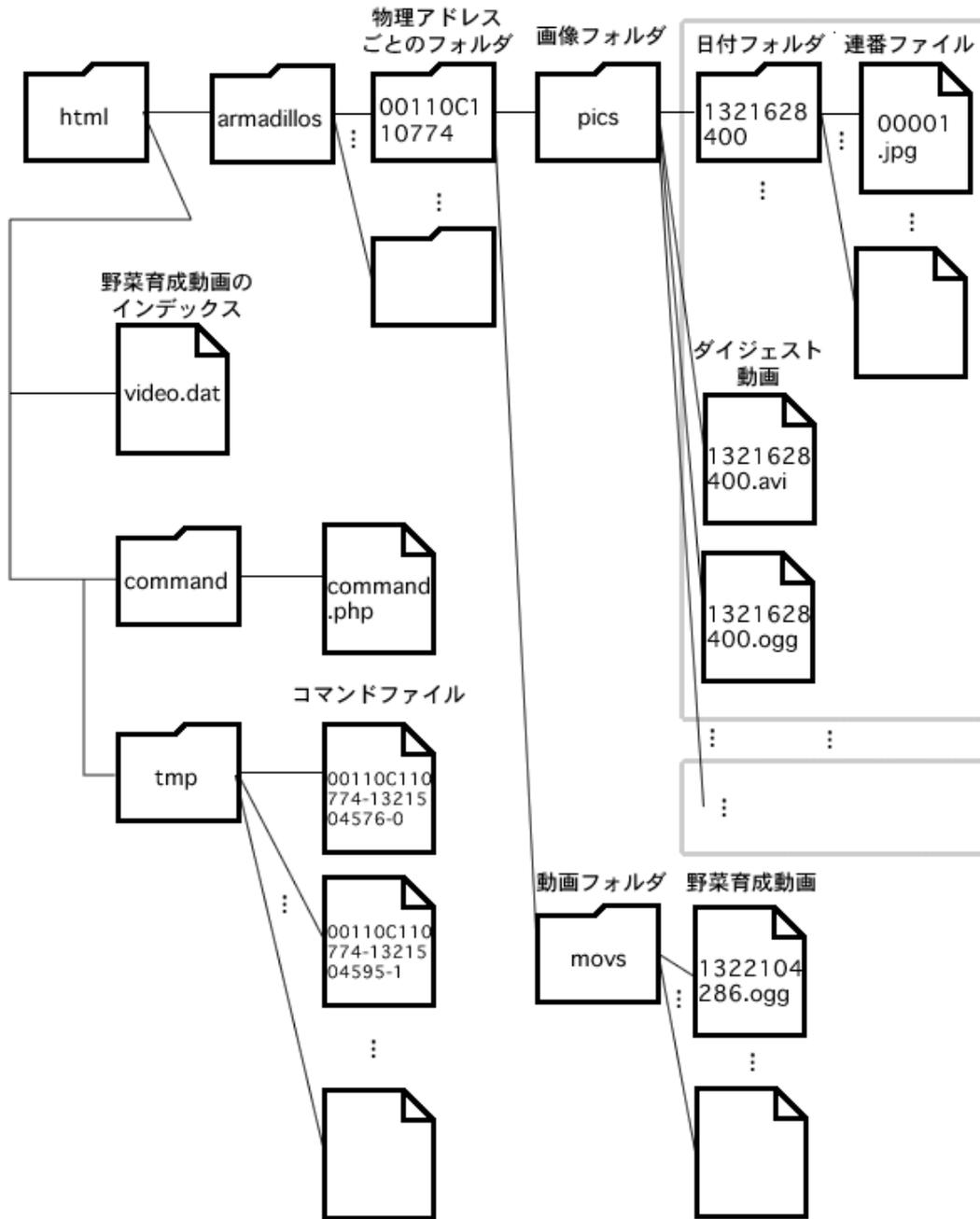


図 5.12 ウェブサーバのフォルダ構造

第6章 評 価

6.1. 概要

本章では、実装した野菜育成情報自動共有システムの評価について述べる。野菜育成情報自動共有システムは、動作の伴う野菜育成情報の共有を円滑にし、理解を促すという設計指針に基づいて設計した。

動作の伴う野菜育成情報の共有を円滑にするためには、野菜育成動画、環境情報、野菜育成ダイジェスト動画をシステムに蓄積するまでの負担と、システムに蓄積された複数の情報から必要な情報を取得するまでの負担を低減することが重要である。一方、動作の伴う野菜の育成情報の理解を促すためには、家庭菜園の工程を施すタイミングと、その工程を施す過程における複数の動作の理解をウェブアプリケーションを通して容易にすることが重要である。そこで本章では、野菜育成情報自動共有システムの評価を行うために、評価項目を次の様に設定した。

1. 野菜育成動画自動共有機能の評価

野菜育成動画自動共有機能が、ユーザの施した全ての家庭菜園の工程を野菜育成動画として、システムに安定して自動的に蓄積することを検証する。

2. 環境情報自動共有機能の評価

環境情報自動共有機能が、家庭菜園の環境情報を 30 秒間隔で、システムに安定して自動的に蓄積することを検証する。

3. 野菜育成ダイジェスト動画自動共有機能の評価

野菜育成ダイジェスト動画自動共有機能が、家庭菜園の環境と野菜の育成状況の変化を野菜育成ダイジェスト動画として、システムに安定して自動的に蓄積することを検証する。

4. ウェブアプリケーションの評価

本システムのウェブアプリケーションが、野菜育成動画と環境情報と野菜育成ダイジェスト動画を体系的に提供することで、ユーザがシステムに蓄積された複数の情報から必要な情報を取得するまでの負担を低減することを検証する。また、家庭菜園の工程を施すタイミングと、その工程を施す過程における複数の動作の理解を容易にすることを検証する。

評価項目1から3によって、野菜育成情報自動共有システムの3つの機能が野菜育成動画、環境情報、野菜育成ダイジェスト動画をシステムに蓄積するまでの負担を低減することを検証する。評価項目4によってウェブアプリケーションがシステムに蓄積された複数の情報から必要な情報を取得するまでの負担を低減することを検証する。そして、野菜育成情報自動共有システムが動作の伴う野菜育成情報の共有を円滑にすることを評価する。

続いて評価項目4によって、ウェブアプリケーションが家庭菜園の工程を施すタイミングと、その工程を施す過程における複数の動作の理解を促すことを検証する。そして野菜育成情報自動共有システムが、動作の伴う野菜育成情報の理解を促すことを評価する。

これらの評価項目によって野菜育成情報自動共有システムが、動作の伴う野菜育成情報の共有を円滑にし、理解を促すことを示す。

6.2. 野菜育成動画自動共有機能の評価

- 目的

野菜育成動画自動共有機能が、ユーザの施した全ての家庭菜園の工程を野菜育成動画として、安定してシステムに自動的に蓄積することを検証する。

- 方法

1. 本システムを15日間連続稼働し、ユーザがイチゴの苗を育成する過程を記録した。そして、それに施した全ての工程が野菜育成動画として、システムに保存されていることを確認した。
2. 実験1の結果、ユーザの施した工程の一部は保存されていなかったことが明らかになった。その要因が、モーションセンサの設置位置による動作検出の不具合と、モーションセンサが動作を検出してから動画サーバがストリーミング映像を保存開始するまでの遅延にあると仮説を立てた。そこでまず、モーションセンサが動作を検出してから動画サーバがストリーミング映像を保存開始するまでの遅延を計測した。これによりその遅延が、ユーザの施した工程の一部が記録されていなかった要因の1つであるという仮説を検証した。
3. 実験2に続いて、遅延の主な要因を解消し、モーションセンサの設置位置を改良した後、家庭菜園の工程を記録した。これによりモーションセンサの設置位置による動作検出の不具合が、ユーザの施した工程の一部が記録されていなかった要因の1つであるという仮説を検証した。

- 結果

1. 野菜育成情報自動共有機能は、ユーザの施した家庭菜園の全ての工程のうち78%を野菜育成動画としてシステムに安定して自動的に保存した。ユーザが期間中に施した工程は全て水やりであった。蓄積された環境情報とサーバに蓄積された野菜育成動画を比較した結果を表6.1に示す。表中の土壤水分上昇開始時間は、環境情報で土壤水分の値の上昇が開始した時刻である。ただし、環境情報は30秒間隔で保存されるため、土壤水分上昇時間とユーザが水を与えた時間は必ずしも一致していない。野菜育成動画作成日時は、動画サーバがストリーミング映像の保存を完了した日時である。動画の記録状態は○と×の2つ要素で表されている。○は工程が記録され、ユーザがその内容を理解できたことを意味する。×は工程の記録がされていなかったことを意味する。この結果によると、11/28 15:52:05と12/09 08:28:13に施された

工程は記録されていなかった。前者の工程の場合、工程を施したのにも関わらず野菜育成動画は保存されなかった。後者の工程の場合、工程を施した後のストリーミング映像が保存されており、工程自体は記録されていなかった。この2つの動画以外にも、工程を途中から記録していた野菜育成動画がいくつか見られた。

表 6.1 動画サーバに保存された野菜育成動画

日付	土壌水分上昇 開始時間	野菜育成動画 作成日時	動画の長さ [秒]	動画の記録状態
11/25	15:24:50	15:25:00	30	○
11/26	12:54:28	12:54:48	31	○
11/28	15:52:05	(動画無し)	×	×
11/30	23:28:31	23:28:59	27	○
12/01	11:05:28	11:06:27	33	○
12/02	21:02:05	21:02:38	30	○
12/03	14:40:10	14:40:54	19	○
12/09	01:56:47	01:57:40	30	○
12/09	08:28:13	08:28:56	32	×

2. モーションセンサが動作を検出してから動画サーバがストリーミング映像を保存開始するまでの遅延は平均 5.735 秒であり、ユーザの施した工程の一部が記録されていなかった要因の1つであったことが明らかになった。クライアントサイドシステムがユーザの動きを検出してから、動画サーバが野菜育成動画を保存するまでのステップは以下の通りである。便宜上、特定のステップをステップ 1~7 と定義した。

- (a) ユーザが野菜に手を加える (ステップ 1)
- (b) 環境情報取得用マイクロコントローラのモーションセンサが動作を検出する
- (c) 環境情報取得用マイクロコントローラが野菜育成情報送信用マイクロコントローラに環境情報を送信する

- (d) 野菜育成情報送信用マイクロコントローラが環境情報を取得し、sensor.dat に書き出す
- (e) post.php が sensor.dat を読み込み、ウェブアプリケーションの sensor コントローラの armadillo ファンクションに環境情報を送信する (ステップ 2)
- (f) ウェブアプリケーションが環境情報を受信し、動画サーバーの command.php にストリーミング映像保存開始命令を送信する (ステップ 3)
- (g) command.php が、tmp フォルダにストリーミング映像開始コマンドファイルを作成する (ステップ 4)
- (h) 動画サーバの autoMovs.php が、コマンドファイルを読み込み movies.php にストリーミング映像保存のターゲットの情報を送信する (ステップ 5)
- (i) movies.php がストリーミング映像の保存を開始する (ステップ 6)
- (j) アプリケーションサーバがストリーミング映像保存停止プロセスの命令を送信する
- (k) command.php が、tmp フォルダにストリーミング映像停止のコマンドファイルを作成する
- (l) 動画サーバの autoMovs.php が、ストリーミング映像停止のコマンドファイルを読み込み、該当するプロセスを終了させる (ステップ 7)
- (m) 野菜育成動画が保存される

図 6.1 にステップごとの野菜育成動画が保存されるまでの経過時間を示す。この結果によると、ユーザが野菜に手を加えてから (ステップ 1) ストリーミング動画保存プロセスが開始されるまで (ステップ 6) 平均して 5.735 秒かかることが明らかになった。もっとも遅延が大きかったステップは 5 であった。ステップ 5 の遅延時間は 4.666 秒であった。ステップ 5 は autoMovs.php によるプロセスである。autoMovs.php では主に以下のステップでプロセスを行う。便宜上、各ステップをステッ

プ1~4と定義した。この実験の際にサーバサイドシステムが蓄積していた総動画数は330であった。

- (a) autoMov.phpが実行される(ステップ1)
- (b) サーバサイドシステムに保存されている全ての野菜育成動画の情報を取得し、video.datに出力する(ステップ2)
- (c) tmpフォルダに保存されたコマンドファイルの情報を取得し、一時ファイルに出力する(ステップ3)
- (d) 出力された一時ファイルを読み込み、コマンド処理を行う(ステップ4)
- (e) autoMov.phpが終了する

autoMovs.phpの遅延を評価した結果を図6.2に示す。autoMovs.phpのプロセスが開始してから終了するまでにかかる時間は6.017秒であり、そのうちステップ2のプロセスにかかる時間は5.527秒であった。

3. 実験2で特定した遅延の要因を解消し、モーションセンサの設置位置を改良した後、ユーザの施す工程を記録した。その結果、野菜育成動画はシステムに安定して自動的に蓄積された。これにより、モーションセンサの設置位置による動作検出の不具合が、ユーザの施した工程の一部が記録されていなかった要因の1つであったことが明らかになった。図6.4に、モーションセンサの設置位置を改良したクライアントサイドシステムを示す。また、改良したクライアントサイドシステムで蓄積した野菜育成動画の情報を表6.2に示す。

6.3. 環境情報自動共有機能の評価

- 目的

環境情報自動共有機能が、家庭菜園の環境情報を30秒間隔で、システムに安定して自動的に蓄積することを検証する。

- 方法

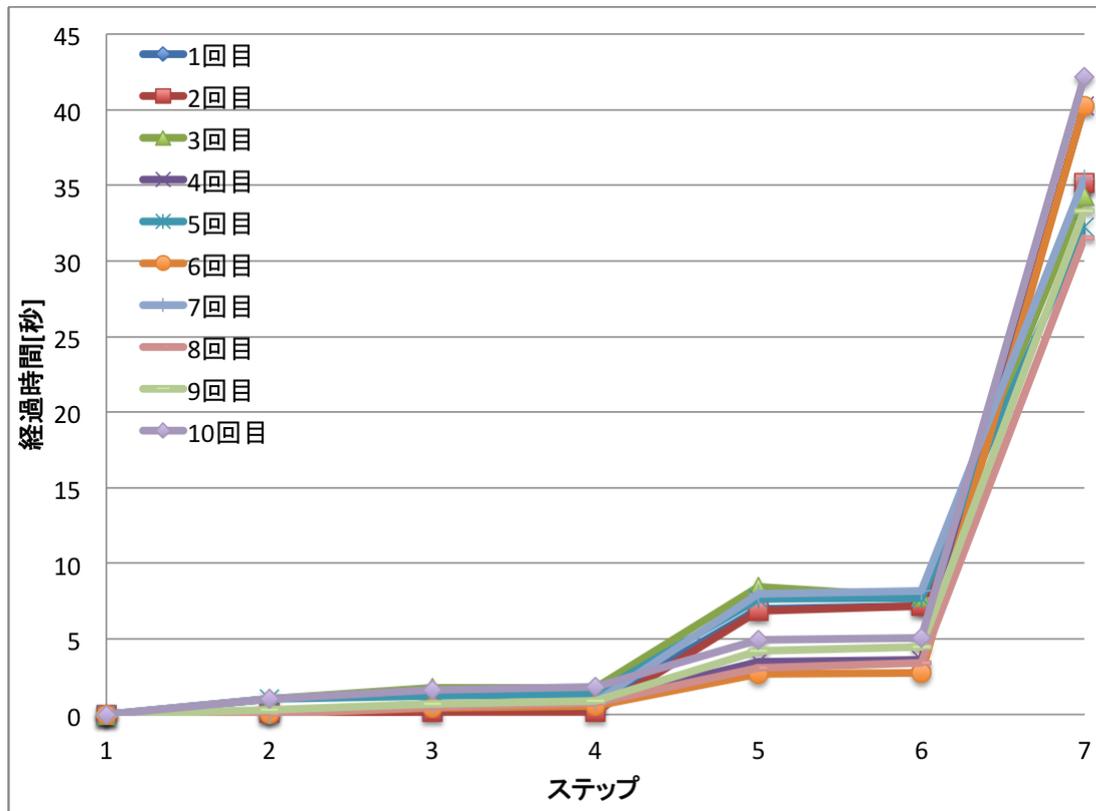


図 6.1 ステップごとの野菜育成動画が保存されるまでの経過時間 [秒]

1. 本システムを15日間連続稼働し、ユーザがイチゴの苗を育成する過程を記録した。そして、家庭菜園の全ての環境情報が、30秒間隔で蓄積されていることを確認した。
2. 実験1の結果、アプリケーションサーバが情報を受信する回数と保存する回数は、想定よりも少なかったことが明らかになった。その要因が、環境情報取得用マイクロコントローラの情報送信頻度と、野菜育成情報送信用マイクロコントローラの情報送信頻度にあると仮説を立てた。そこで、それらの情報送信頻度を測定することで仮説の検証を行った。

- 結果

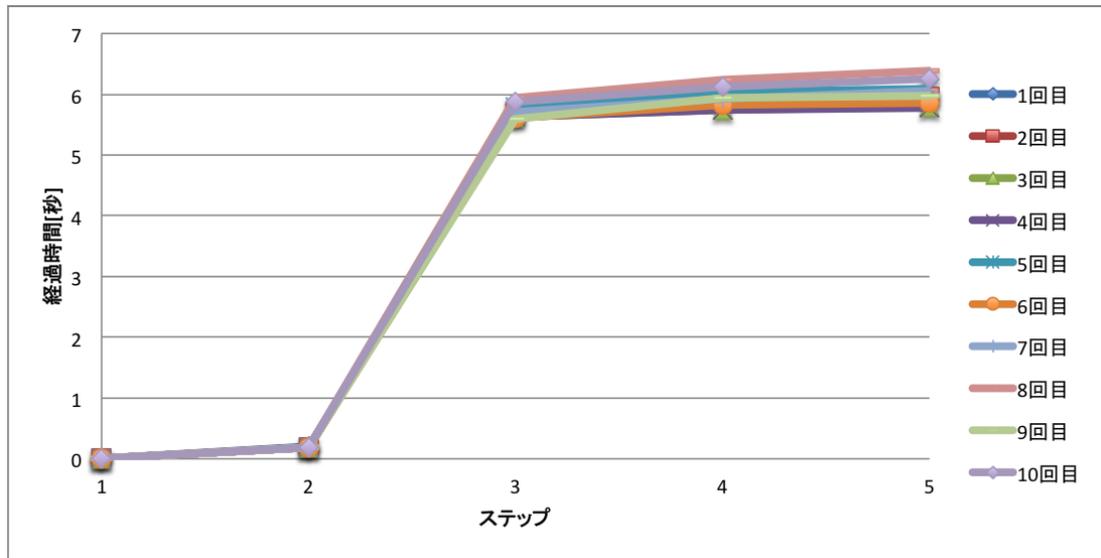


図 6.2 ステップごとの autoMovs.php によるプロセスの経過時間 [秒]

1. 環境情報自動共有機能は、家庭菜園の環境情報を 31.6 秒間隔で、システムに安定して自動的に蓄積した。蓄積した環境情報についての情報を表 6.3 に示す。この結果によるとアプリケーションサーバは、1.68 秒に 1 回のクライアントサイドシステムから情報を受信していた。設計の段階では 1 秒に 1 回情報を受信し、30 秒に 1 回情報を保存することを想定していた。従って、アプリケーションサーバが 1 分間あたりに情報を受信する回数は想定していたよりも 61% 少なく、保存する回数は 5% 少ないことが明らかになった。
2. アプリケーションサーバが情報を受信する頻度と保存する頻度が、想定よりも少なかった要因は、野菜育成情報送信用マイクロコントローラの情報送信頻度が低かったことによることが明らかになった。環境情報取得用マイクロコントローラが 1 分間あたりに情報を送信する回数は 60.00 回であった。一方、野菜育成情報送信用マイクロコントローラが情報を送信する回数は 1 分間あたり 35.10 回であった。

表 6.2 追加して記録した野菜育成動画の情報

保存された野菜育成動画	動画の長さ [秒]	項目	容量 [MB]
1323609754.ogg	184	イチゴの苗の植え付け、土寄せ、水やり	6.0
1323610070.ogg	122	イチゴの追肥	4.9
1323611202.ogg	155	イチゴの苗の植え付け、土寄せ、水やり、追肥	5.7
1323610702.ogg	185	ハウレンソウの種の条播き	5.4
1323610987.ogg	66	条播き後の水やり	2.0

表 6.3 蓄積した環境情報

日付	情報保存回数 [回]	1 分間に保存する回数 [回]	情報受信回数 [回]	1 分間に受信する回数 [回]
11/25	2753	1.912	52134	36.204
11/26	2752	1.911	53380	37.069
11/27	2745	1.906	53348	37.047
11/28	2719	1.888	52565	36.503
11/29	2743	1.905	52017	36.123
11/30	2746	1.907	51597	35.831
12/01	2745	1.906	51002	35.418
12/02	2753	1.912	51720	35.917
12/03	2749	1.909	51845	36.003
12/04	2752	1.911	51968	36.089
12/05	2752	1.911	51758	35.943
12/06	2751	1.910	50509	35.076
12/07	2748	1.908	50777	35.262
12/08	2743	1.905	49156	34.136
12/09	2733	1.898	46978	32.624
平均	2747.8	1.9082	51383.6	35.6830



図 6.3 改良したクライアントサイドシステム

6.4. 野菜育成ダイジェスト動画自動共有機能の評価

- 目的

野菜育成ダイジェスト動画自動共有機能が、家庭菜園の環境と野菜の育成状況の変化を野菜育成ダイジェスト動画として、システムに安定して自動的に蓄積することを検証する。

- 方法

本システムを15日間連続稼働し、ユーザにイチゴの苗を育成する過程を記録した。そして、家庭菜園の環境と野菜の育成状況の変化が野菜育成ダイジェスト動画として、1日ごとに保存されていることを確認した。

- 結果

野菜育成ダイジェスト動画自動共有機能は、家庭菜園の環境と野菜の育成

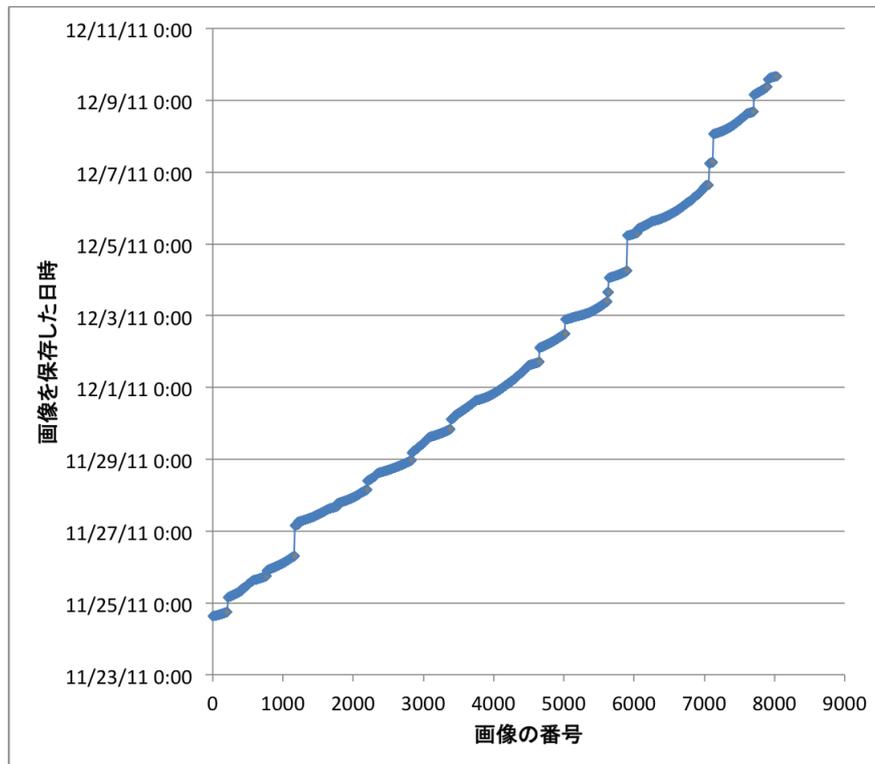


図 6.4 ストリーミング画像の保存枚数と時刻の関係

状況の変化を野菜育成ダイジェスト動画として、システムに安定して自動的に蓄積していなかった。野菜育成ダイジェスト動画は、動画サーバに保存される複数のストリーミング画像を組み合わせることで出力される。ストリーミング画像の保存枚数と時刻を表したグラフを図6.4に示す。この結果によると、ストリーミング画像の保存は断続的に行われていた。続いて、保存された野菜育成ダイジェスト動画についての情報を表6.4に示す。ここで動画の状態は、動画を再生できる事を意味する。この結果によると、野菜育成ダイジェスト動画の蓄積は行われたが、動画の長さや容量はばらつきが大きく、野菜の状態や環境の変化を十分に記録できていなかった。

表 6.4 データベースに保存されたダイジェスト動画

日付	動画の長さ [秒]	動画の容量 [KB]	画像の枚数 [枚]	動画の状態
11/25	28	371	707	○
11/26	3	72	82	○
11/27	22	234	572	○
11/28	13	147	488	○
11/29	28	270	705	○
11/30	25	236	646	○
12/1	30	192	769	○
12/2	20	127	511	○
12/3	24	166	601	○
12/4	11	106	282	○
12/5	14	114	353	○
12/6	30	192	762	○
12/7	4	80	104	○
12/8	19	134	479	○
12/9	13	135	333	○

6.5. ウェブアプリケーションの評価

- 目的

ウェブアプリケーションの評価は、二つの目的で行った。第一に、ウェブアプリケーションが野菜育成動画と環境情報と野菜育成ダイジェスト動画を体系的に提供することで、ユーザがシステムに蓄積された複数の情報から必要な情報を取得するまでの負担を低減することを検証する。第二に、家庭菜園の工程を施すタイミングと、その工程を施す過程における複数の動作の理解を、既存のブログや情報共有サイトなどの情報リソースと比較して容易にすることを検証する。

- 方法

1年以内に家庭菜園を中断した2人のユーザに、本システムのウェブアプリケーションを利用させた。そしてインタビューを行い、既存の情報リソースと比較して評価を行った。また、本システムに蓄積した野菜育成動画の評価を同様に行った。インタビューを行った2人のユーザは20代社会人男性、20代大学生男性であった。前者のユーザは、エンドウとネギの育成していたが、野菜を美味しく作ることに失敗したために家庭菜園を中断していた。一方、後者のユーザは、ミニトマトを育成していたが、途中で世話に負担を感じ放置した結果、枯らしてしまったために家庭菜園を中断していた。

- 結果

1. ウェブアプリケーションの評価

本システムのウェブアプリケーションは、野菜育成動画と環境情報と野菜育成ダイジェスト動画を体系的に提供した。ユーザは、ウェブアプリケーションを通してシステムに蓄積された、野菜育成動画、環境情報グラフ、野菜育成ダイジェスト動画をユーザ名と時間ごとに体系的に閲覧することができた。しかしウェブアプリケーションは、システムに蓄積された複数の情報から必要な情報を取得するまでの負担を

十分に低減することはできなかった。一方で、家庭菜園の工程を施すタイミングと、その工程を施す過程における複数の動作の理解を、既存の情報リソースと比較して容易にすることができた。

インタビューを行いウェブアプリケーションを評価した結果、既存の情報リソースと比較して次に示す特徴に優位性があることが明らかになった。

- － 野菜育成の経過と結果が明確に分かるため、要領を理解しやすいこと
- － 育成環境が似ているユーザの育成工程を参照でき、未然に失敗を防ぐのに役立つこと
- － 野菜の生長や施した工程の過程を容易に理解でき、野菜を育成している実感がわくこと

ユーザは、野菜育成動画の内容よりも環境情報グラフや野菜育成ダイジェスト動画により野菜の育成過程が連続して記録されることに対して優位性を見出す傾向にあった。また、既存の情報リソースであるブログや情報共有サイトと比較した場合、一貫した野菜の育成情報が共有されることや、野菜育成動画により情報の理解を容易にできることに良い評価を下していた。しかしその一方で、次に示す課題が指摘された。

- － コンテンツやシステムを利用するユーザの数が少ないこと
- － 動画の質が低いこと
- － 必要な野菜の育成情報を見つけ出すまでの負担が大きいこと

既存の情報リソースと比べ、現状のシステムではユーザの数が限られているため、コンテンツが少ない。動画の質については、画質の低さや、葉や手で育成工程の記録が隠れることで、育成工程の内容を理解しにくい場合のあることが指摘された。必要な野菜の育成情報を見つけ出すまでの負担を大きいことは、野菜育成動画を見つけ出すまでのステップが長いことなどが挙げられた。



図 6.5 定期的に行う工程の野菜育成動画のスクリーンショット

2. 野菜育成動画の評価

野菜育成動画を評価した結果、ユーザは家庭菜園の工程を施す過程における複数の動作と、工程を施すタイミングの理解を容易にすることが明らかになった。ここでは、定期的に行う工程と非定期的に行う工程に分けて、評価の結果を述べる。

－ 定期的に行う工程の野菜育成動画

定期的に行う工程を記録した野菜育成動画を評価した結果、家庭菜園の工程を施すタイミングとその工程を施す過程における複数の動作の理解を容易にすることができた。図6.5に、定期的に行う工程を記録した野菜育成動画のスクリーンショットを示す。ユーザは、これらの動画を閲覧することで、野菜の葉や茎のしなり加減や土の色から、水を与えるタイミングの理解が可能になった。また、動画を通して水の与え方や分量の理解を、比較的容易にすることが可能になった。

しかし、一方で土の表面の状態や水を与える分量の把握が難しい場合が想定される。今回育成したイチゴの苗の場合は、土の表面の乾燥度合いを動画から確認できたが、葉菜類の様に葉の面積の大きい野菜の場合、土の表面を確認できない場合が考えられる。また、野菜に与えた水の分量についても野菜の上方から記録した野菜育成動画の場合、把握が難しかった。

－ 非定期的に行う工程の野菜育成動画

非定期的に行う工程を記録した野菜育成動画を評価した結果、家庭菜園の工程を施す過程における複数の動作の理解を容易にすることができた。保存した非定期的に行う工程の野菜育成動画のスクリーンショットを図6.6に示す。ユーザは、例えば、イチゴの苗の植え付け、土寄せ、水やり、追肥の複数の工程が連続した動画(1323611202.ogg)を閲覧することで、それぞれの工程の野菜育成情報を動作の流れの中で理解することができた。この動画で保存した工程の野菜育成情報として、植え付けと土寄せではイチゴのランナーとクラウンを土中に埋めないこと、ハウレンソウの種まきの場合、種の間隔を広くして条播きをおこなうこと、追肥ではイチゴの根の張り方を意識して周囲に与えること、などが挙げられる。ユーザはそれらの複数の動作が伴う連続した工程の野菜育成情報を、既存の情報リソースと比較して容易に理解する事ができた。

しかし一方で、記録した野菜育成動画には次に挙げる課題があることが明らかになった。植え付けと土寄せの場合は上方からカメラで撮影し工程を記録していたため、保存した野菜育成動画でイチゴのランナーやクラウンにユーザの注意を促すことは難しかった。追肥の場合も同様に、イチゴが生長して葉が広がった際に、上方からの土壌の様子を記録することが難しくなることが懸念される。種まきの場合、カメラをプランターに近づけるによって対象を見やすくすることはできた。しかし、ハウレンソウの種は茶色であり、土の色と同系色であったため、種の間隔や深さなどの情報を把握することは難しかった。

以上より、本システムのウェブアプリケーションは野菜育成動画と環境情報グラフと野菜育成ダイジェスト動画を体系的に共有したが、システムに蓄積された複数の情報から必要な情報を取得するまでの負担を十分に低減することはできなかったことが明らかになった。一方で、家庭菜園の工程を

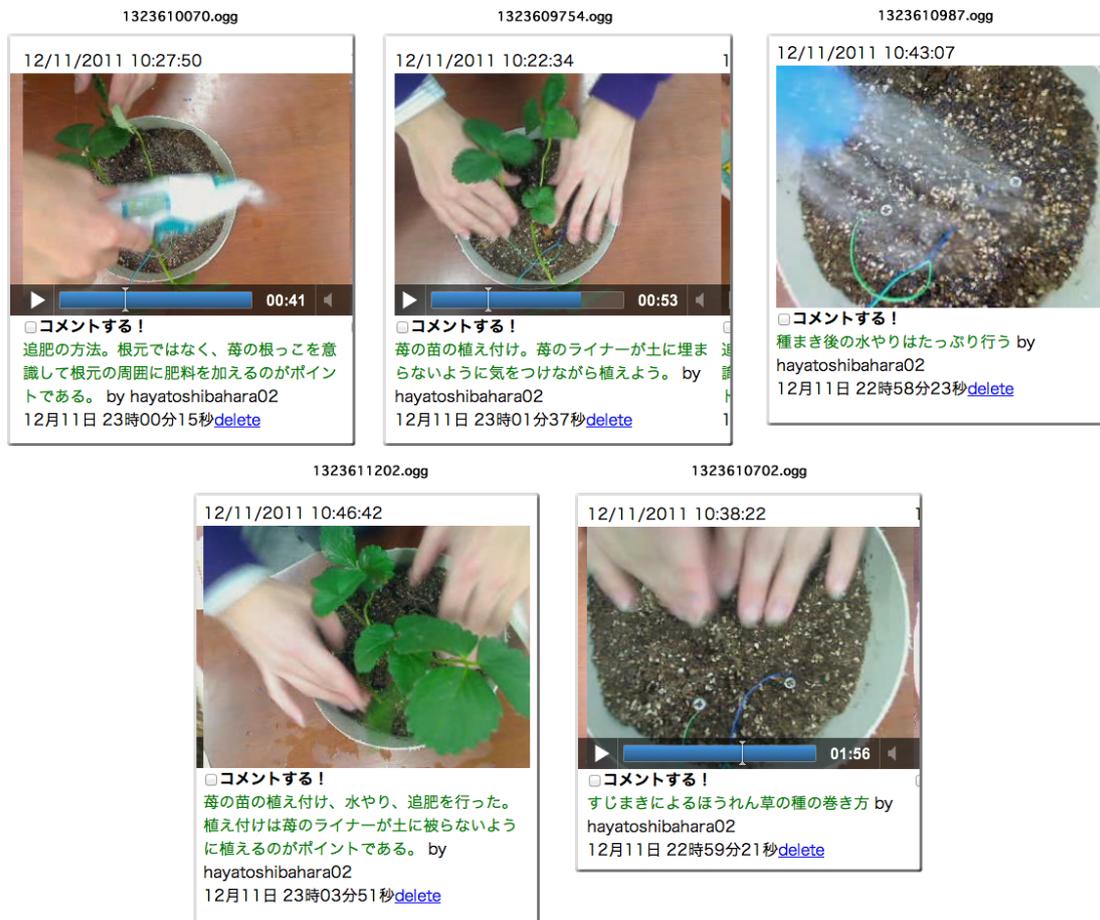


図 6.6 蓄積した野菜育成動画のスクリーンショット

施すタイミングと、その工程を施す過程における複数の動作の理解を、既存の情報リソースと比較して容易にできたことが明らかになった。

第7章

考 察

7.1. 概要

本章では野菜育成情報自動共有システムの考察について述べる。そこで、野菜育成動画自動共有機能、環境情報自動共有機能、野菜育成ダイジェスト動画自動共有機能、ウェブアプリケーションの考察を行う。そして、野菜育成情報自動共有システムが動作の伴う野菜の育成情報の共有を円滑にし、理解を促す為に必要な課題を明らかにする。

7.2. 野菜育成動画自動共有機能の考察

野菜育成動画自動共有機能の評価は、その機能がユーザの施した全ての工程を野菜育成動画として、安定してシステムに自動的に蓄積することを検証する目的で行った。

1. 結果1では、野菜育成情報自動共有機能が、ユーザの施した全ての工程のうち78%を野菜育成動画としてシステムに安定して自動的に保存したことを示した。しかし、11/28 15:52:05と12/09 08:28:13に施された工程は記録されていなかった。そこでここでは、その要因について考察する。

11/28の工程を記録できなかった要因は、モーションセンサが動作を検出しなかったためだと考えられる。11/28の環境情報グラフを図7.1

に示す。このグラフによると、土壌水分量が増加するタイミングに近い時間にモーションセンサが反応していなかったことが分かる、ストリーミング映像の保存が開始されていなかったと推測できる。

12/09の工程を記録できなかった要因は、モーションセンサの動作検出が遅れたためだと考えられる。12/09の環境情報グラフを図7.2に示す。このグラフによると、土壌水分の上昇と動作の検出の時間がほぼ同時になっていることが分かる。従って、動作検出が遅れたために、ストリーミング映像の保存開始に遅延が生じたと推測できる。

モーションセンサの動作の検出ができなかった、もしくは遅れた要因は、実装したモーションセンサの性質にあったと考えられる。本システムでは2つのモーションセンサを実装した。そのうちモーションセンサ NaPiOn は背景との温度差を利用して動作を検出する。しかし、実験では背景との温度差の少ないじょうろで水を与えていたため検出範囲内でユーザの動作を検出できなかったと考えられる。図7.3に、実験で用いたモーションセンサの検出範囲を示す。NaPiOnの検出角度は51度である。また、実験においてNaPiOnは高さ35cmの位置に垂直に下向きで設置されており、プランターの高さは13cm、イチゴの苗の最高長は16cmであった。そして、イチゴの苗は、上方からじょうろで水が与えられていた。しかし、苗の上方の動作検出範囲は半径7cm程度であり、温度差の大きいユーザの手の動きをNaPiOnが検出するまでに至らなかったと考えられる。従って、ユーザが野菜に手を加えたことを検出するためには、NaPiOnがじょうろだけではなく人の手の動きまで検出できる高さや角度に設置する必要がある。

2. 結果2では、モーションセンサが動作を検出してから動画サーバがストリーミング映像を保存開始するまでの遅延が平均5.735秒であり、ユーザの施した工程の一部が記録されていなかった要因の一つであることを示した。そこでここでは、その遅延の要因について考察する。モーションセンサが動作を検出してから動画サーバがストリーミング映像の保存を開始するまでの遅延の要因は、主に autoMov.php のプロ

グラフ

2011年11月28日の環境データ 前日のデータ 後日のデータ

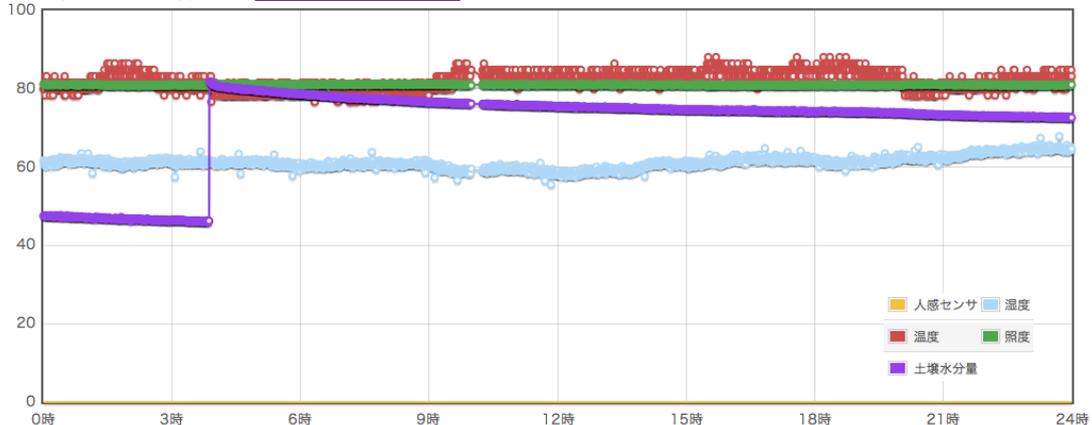


図 7.1 11/28 の環境情報グラフ

セスにあることが示された。評価結果より、動作検出からストリーミング映像保存開始までの遅延は 5.735 秒かかることが明らかになった。しかし、そのうち autoMov.php のプロセスによる遅延は 4.666 秒であり、81%を占めていた。autoMov.php の処理を分析すると、動画サーバに保存されている全ての野菜育成動画の情報を取得し、video.dat に書き出すステップの遅延が、autoMov.php の処理全体の遅延の 91%を占めていた。従って、そのステップを別のプログラムによるプロセスで行わせることにより、動作検出からストリーミング映像保存開始までの遅延を大幅に低減できると考えられる。

3. 結果3では、実験2で特定した遅延の要因を解消し、モーションセンサの設置位置を改良し、ユーザの施す工程を記録したところ、野菜育成動画はシステムに安定して自動的に蓄積され、モーションセンサの設置位置による動作検出の不具合がユーザの施した工程の一部が記録されていなかった要因の一つであったことを示した。そこでここでは、モーションセンサの設置位置を変えるために、新たに実装したクライアントサイドシステムの設計と、それに付随した効果について考察する。

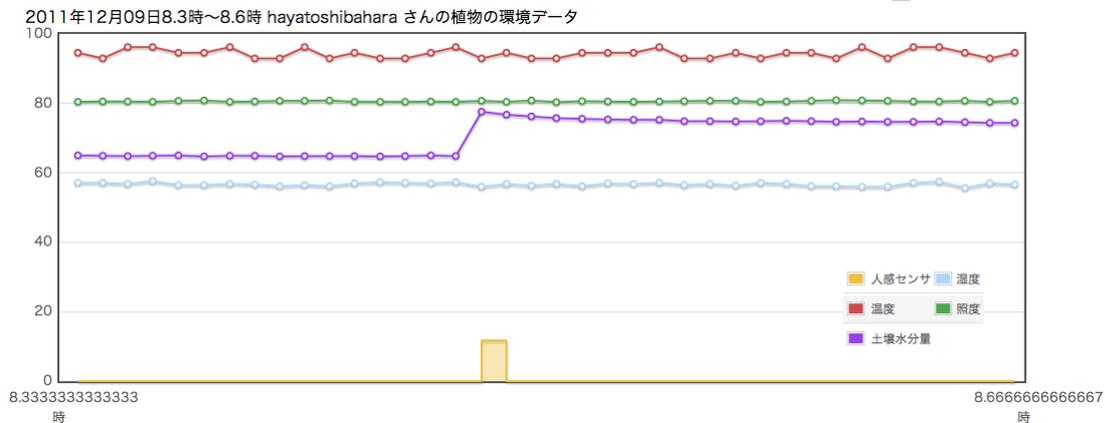


図 7.2 12/09 の環境情報グラフ

実験 1 を行った結果、ユーザが野菜に手を加えたことを検出するためには、モーションセンサ NaPiOn を適切な高さと角度に設置する必要があることがわかった。そこで、クライアントサイドシステムのアームを図 6.7 に示すように改良した。新しいアームは、可動部と非可動部に分けて構成した。可動部の先端にはモーションセンサ NaPiOn を設置した。これによりユーザは NaPiOn の検出範囲を任意に調整できる。そして、問題であったモーションセンサの設置位置による動作検出の不具合を解消できた。一方、非可動部の上方にはモーションセンサ SEN-08630 を設置した。SEN-08630 は電源を入れるのと同時に周囲の位置を記憶し、その変化を検出するため設置位置を固定する必要がある。従って、非可動部に設置した。

続いて、クライアントサイドシステムの設計の改良に付随した効果について考察する。クライアントサイドシステムのアームの改善とともに、アームに取り付けられるカメラもユーザが任意に位置や角度を調整できるように設計した。この設計により、映像を保存する対象を絞って工程を記録することが可能になった。例えば、種まきの工程の場合、ユーザがカメラをプランタに近づけ角度を調整することで、より種の間隔や深さが把握しやすくなり、効果的に野菜育成動画を保存することができた。

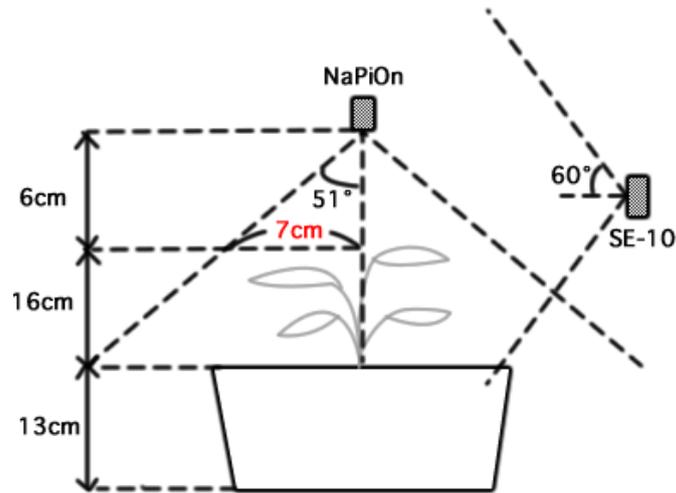


図 7.3 モーションセンサ検出範囲

以上の考察より、野菜育成動画自動共有機能がユーザの施した全ての家庭菜園の工程を野菜育成動画として、安定してシステムに自動的に蓄積するためには次の二つの改善が必要であることが明らかになった。第一に、autoMov.php のプログラムの改善である。autoMov.php の、動画サーバに保存されている全ての野菜育成情報を取得し video.dat に書き出すステップを別のプログラムによるプロセスで行わせることにより、遅延を抑えることができる。その結果、動作検出とほぼ同時にストリーミング映像の保存を開始することが可能になり、工程の全てを記録することができると考えられる。第二に、モーションセンサ NaPiOn の設置位置の改善である。NaPiOn の位置をユーザの手の動きまで検出できる十分な高さに設置することで、動作検出を改善することができる。その結果、NaPiOn は確実にユーザの動作を検出することが可能になり、全ての工程を記録することができると考えられる。

7.3. 環境情報自動共有機能の考察

環境情報自動共有機能の評価は、その機能が家庭菜園の環境情報を 30 秒間隔で、システムに安定して自動的に蓄積することを検証する目的で行った。

1. 結果1では、環境情報自動共有機能が、家庭菜園の環境情報を 31.6 秒間隔でシステムに安定して自動的に蓄積したことを示した。しかし設計の段階では、30 秒間隔で環境情報を保存することを想定しており、一分間あたりに情報を保存する回数は 5% 少なかった。その要因は、一分間あたりに環境情報を受信する回数が想定より 61% 少なかったことによると考えられる。その結果、環境情報を保存する時間の間隔にロスが生じたと推測できる。
2. 結果2では、アプリケーションサーバが情報を受信する頻度と保存する頻度が想定よりも少なかった要因が、野菜育成情報送信用マイクロコントローラの情報送信頻度が少なかったことによることを示した。そこで、ここではその情報送信頻度が少なかった要因について考察する。サーバサイドシステムの情報受信頻度が少なかった要因はクライアントサイドシステムの `post.php` の `sleep` 関数によることが示された。`post.php` では `sleep` 関数を利用して秒単位でタイミングの制御を行っていた。しかし、`post.php` がプロセスを完了するまでの時間は想定していたよりも長く、情報送信頻度を少なくする要因となっていた。これより `post.php` が一分間あたりに情報を送信する回数を設計で想定した回数にするために、`usleep` 関数を利用することで、厳密にタイミングを制御する必要があると考えられる。

以上の考察より、環境情報自動共有機能が家庭菜園の環境情報を 30 秒間隔で、システムに安定して自動的に蓄積するためには、`post.php` が情報を送信するタイミングの制御を厳密に行う必要があることが明らかになった。`post.php` の情報を送信するタイミングの制御を厳密に設計することにより、野菜育成情報送信用マイクロコントローラの情報送信頻度を改善すること

ができる。その結果、システムの環境情報受信頻度を改善することができ、保存する頻度を設計しそうで想定していた回数ににあわせることができる。

7.4. 野菜育成ダイジェスト動画自動共有機能の考察

野菜育成ダイジェスト動画自動共有機能の評価は、その機能が家庭菜園の環境と野菜の育成状況の変化を野菜育成ダイジェスト動画として、システムに安定して自動的に蓄積することを検証する目的で行った。

その結果、野菜育成ダイジェスト動画自動共有機能は、家庭菜園の環境と野菜の育成状況の変化を野菜育成ダイジェスト動画として、システムに安定して自動的に蓄積していなかったことが示された。そこでここでは、野菜育成ダイジェスト動画自動共有機能が不安定であった要因について考察する。

野菜育成ダイジェスト動画自動共有機能が不安定であった要因は、VLCの動作が完了せず、pictures.phpのプロセスが途中で停止しまったことによる。野菜育成ダイジェスト動画自動共有機能は、蓄積したストリーミング画像を、FFmpegにより動画に合成し、VLCで動画のエンコードを行う。そこでFFmpegとVLCのログを調べた結果、エンコードの途中でエラーが生じ、プロセスが完了していなかったことが明らかになった。従って、プロセスのタイムアウト処理を確実にを行うことで動作を安定にできると考えられる。

以上の考察より、野菜育成ダイジェスト動画自動共有機能が家庭菜園の環境と野菜の育成状況の変化を野菜育成ダイジェスト動画として、システムに安定して自動的に蓄積するためには、VLCのプロセスの確実なタイムアウト処理が必要であることが明らかになった。VLCのプロセスのタイムアウト処理を確実にを行うことによって、動作を安定にすることができると考えられる。

7.5. ウェブアプリケーションの考察

ウェブアプリケーションの評価は、二つの目的で行った。第一に、ウェブアプリケーションが野菜育成動画と環境情報と野菜育成ダイジェスト動画を体系的に提供することで、ユーザがシステムに蓄積された複数の情報から必要な情報を取得するまでの負担を低減することを検証した。第二に、家庭菜園の工程を施すタイミングと、その工程を施す過程における複数の動作の理解を容易にすることを検証することである。

1. 結果1では、本システムのウェブアプリケーションが、システムに蓄積された野菜育成動画と環境情報と野菜育成ダイジェスト動画を体系的に提供したが、システムに蓄積された複数の情報から必要な情報を取得するまでの負担を十分に低減できなかつたことを示した。また、家庭菜園の工程を施すタイミングと、その工程を施す過程における複数の動作の理解を、既存の情報リソースと比較して容易にしたことを示した。しかし一方で、コンテンツやシステムを利用するユーザの数が少ないこと、動画の質が低いこと、必要な野菜の育成情報を見つけ出すまでの負担が大きいことが課題として指摘された。そこでここではまず、ウェブアプリケーションがシステムに蓄積された複数の情報から必要な情報を取得するまでの負担を十分に低減できなかつた要因について考察する。続いて、ユーザが指摘した課題について考察する。ウェブアプリケーションがシステムに蓄積された複数の情報から必要な情報を取得するまでの負担を十分に低減できなかつた要因として、必要な野菜の育成情報を見つけ出すまでのステップが長いことが挙げられた。その負担を低減するためにウェブアプリケーションの構成をより単純化する必要があると考えられる。現状のウェブアプリケーションは、システムに蓄積された複数の野菜育成動画から必要な動画を見つけ出しやすくするために、インデックスページと環境情報グラフ共有ページと野菜育成動画共有ページの3層で構成されている。しかし、野菜育成動画自動共有機能の実験において15日間ユーザがシステムを

使用したところ、工程を施した回数は9回であり想定していた回数よりもはるかに少なかった。野菜育成動画共有ページの機能の一つとして、野菜育成動画を時間帯で再帰的に絞り込むことを可能にする機能がある。しかし、動画の数が少なかったために十分に役割を果たしていなかった。そこで、野菜育成動画共有ページと環境情報グラフ共有ページを統合することで、必要な野菜育成情報をユーザが見つけ出すまでのステップを減らすことができると考えられる。

ユーザに指摘された課題は、(a) コンテンツやシステムを利用するユーザの数が少ないこと、(b) 動画の質が低いこと、(c) 必要な野菜の育成情報を見つけ出すまでの負担が大きいことであった。(c) については、前述した通りである。(a) については、本システムのウェブアプリケーションは、保有するコンテンツやユーザの数が少なく、既存の情報リソースと比べ優位性に劣るという指摘である。しかし、本論文では、動作の伴う野菜の育成情報の共有を円滑にし、理解を促すことを目的に野菜育成情報自動共有システムの提案を行っていたため、議論の範疇ではない。ただ、様々な環境における野菜の育成情報を参照できる機能は今後必要になるため、展望の章にて言及する。(b) については、画質の低さや、葉や手で育成工程の記録が途切れることで、工程の内容を理解しにくい場合があるという指摘である。この指摘については、考察2で言及する。

2. 結果2では、野菜育成動画が、家庭菜園の工程を施す過程における複数の動作と、工程を施すタイミングの理解を容易にしたことを示した。しかしその一方で、野菜育成動画により工程の情報を共有する際に、いくつかの課題が明らかになった。そこでここでは、定期的に行う工程と非定期的に行う工程に分けて、指摘された課題を示し、それぞれの課題について考察する。

- 定期的に行う工程の野菜育成動画

動画の内容を評価した結果、工程を施すタイミングとその工程を施す過程における複数の動作の理解を容易にすることができた。

ユーザはこれらの動作を閲覧することで、野菜の葉や茎のしなり加減や土の色から水を与えるタイミングの理解を比較的容易にすることが可能になった。

一方で、土の表面の状態や水を与える分量の把握が場合によって難しく、動作の伴う野菜の育成情報を効果的に共有するためには改善の必要があることが明らかになった。例えば、葉菜類の様に葉の面積の大きい野菜の場合、上方からカメラで記録する際に、土の表面を確認できない場合が考えられる。また、野菜に与えた水の分量についても野菜の情報から記録した野菜育成動画の場合、把握が難しかった。

土の表面の状態や水を与える分量の把握を容易にするためには、野菜育成の工程を記録するカメラの角度と位置が重要となってくる。この課題を解決するために、カメラの角度と一を任意に帰られる様に改良したアームをユーザが使用することで、効果的に動作の伴う野菜の育成情報を共有することができると考えられる。

－ 非定期的に行う工程の野菜育成動画

動画の内容を評価した結果、工程を施す過程における複数の動画の理解を容易にすることはできたが、いくつかの課題のあることが明らかになった。

植え付けと土寄せの場合、イチゴのランナーとクラウンを土中に埋めないことが重要であった。しかし、カメラは上方から工程を記録していたため、保存した野菜育成動画でユーザの注意を促すことは難しかった。追肥の場合は、イチゴの根の張り方を意識して周囲に与えることが重要であった。しかし、カメラは上方から工程を映していたため、イチゴが生長して葉が広がる場合、土壌の様子を確認することが難しくなることが懸念された。ハウレンソウの条播きの場合、間引きを簡易にするために種の間隔を狭くしすぎない方法が良いとされる。しかし、ハウレンソウの種は茶色であり、土の色と同系色であったため、動画を通して種の間隔

や深さなどの情報を把握することは難しかった。

アームを改良したクライアントサイドシステムを使用した結果、工程ごとにカメラの配置を変えて野菜育成動画の保存を行うことができた。しかし、操作性が悪く、容易にカメラの位置を変えられなかったため、アームをより扱いやすい機構に改良する必要があると考えられる。これにより、カメラを工程中に近づけることが可能になり、細かい野菜の育成情報の共有を効果的に行えると期待できる。また、野菜育成動画の画質も重要な要素である。今回使用したマイクロコントローラよりも処理性能の高い機器を利用することで、高い画質で動画を共有可能になる。これにより、例えば条播きの種の配置や深さなどの情報を共有しやすくなると考えられる。

以上の考察より、システムに蓄積された複数の情報から必要な情報を取得するまでの負担を十分に低減するためには、野菜育成動画共有ページと環境情報グラフ共有ページを統合することで、必要な野菜育成情報を見つけ出すまでのステップを減らすことが必要であることが明らかになった。また、家庭菜園の工程を施すタイミングと、その工程を施す過程における複数の動作の理解をより容易にするためには、クライアントサイドシステムのアームを改良し、容易にカメラの位置を変えられる機構にすることと、処理性能の高い機器を利用し、高い画質で動画を共有可能にすることで、野菜育成動画の質を上げる必要のあることが明らかになった。

第8章

結 論

本論文は、家庭菜園に満足していない初心者の割合が高いという問題に着目した。その問題の要因として、初心者の家庭菜園を行うための環境が十分に整っていないことと、初心者が野菜の育成情報を十分に理解していないことを挙げた。これらの課題を初心者にとって負担の少ない方法で解決するために、家庭菜園を行うための環境が十分に整っていない場合でも野菜の育成情報を容易に理解し、実践することで家庭菜園を長く楽しむような環境が必要であると考えた。

初心者が野菜の育成情報を十分に理解していない要因として、インターネットから取得する野菜の育成情報を理解し実践する際の3つの負担を挙げた。第一に、インターネットから取得する野菜の育成情報は特定の家庭菜園の環境を前提にしている場合が多く、環境に制約のある初心者にとって実践が難しい負担である。第二に、インターネットから取得する野菜の育成情報の多くは文書と写真で表現されており、動作の伴う野菜の育成情報の共有が難しいという負担である。第三に、経験者とのコミュニケーションの機会が限られており、適切な助言を受けられにくいという負担である。

それらの負担を解消するために、(1) 様々な環境における野菜の育成情報を参照でき、(2) 動作の伴う野菜の育成情報の共有を円滑にし、理解を促し、(3) 個人の家庭菜園の環境や野菜の育成状況をもとにコミュニケーションを図れるようなシステムが必要であると考えた。本論文では、(1) と (3) の機能の十分条件を満たしシステムを実現するために、(2) の機能に着目し、野菜育成情報自動共有システムの提案を行った。

野菜育成情報自動共有システムは、動作の伴う野菜育成情報の共有を円滑にし、理解を促すという設計指針に基づいて設計した。その結果、機能として野菜育成動画自動共有機能、環境情報グラフ共有機能、投稿情報共有機能、野菜育成ダイジェスト動画自動共有機能を実装した。野菜育成動画自動共有共有機能は、家庭菜園においてユーザが野菜に施した工程を野菜育成動画としてサーバに自動的に蓄積し、ウェブアプリケーションを通して複数のユーザで共有可能にする機能である。環境情報グラフ自動共有機能は、家庭菜園の環境情報をサーバに自動的に蓄積し、蓄積した情報をウェブアプリケーションで環境情報グラフとして円滑に共有可能にする機能である。投稿情報共有機能は、野菜育成動画と環境情報グラフに対して、不特定多数のユーザが投稿を加える事を可能にする機能である。野菜育成ダイジェスト動画自動共有機能は、一日ごとの野菜の状態と家庭菜園の環境の変化を高速で表示する動画を自動的に共有可能にする機能である。

動作の伴う野菜育成情報の共有を円滑にするためには、野菜育成動画、環境情報、野菜育成ダイジェスト動画をシステムに蓄積するまでの負担と、システムに蓄積された複数の情報から必要な情報を取得するまでの負担を低減することが重要である。一方、動作の伴う野菜の育成情報の理解を促すためには、家庭菜園の工程を施すタイミングと、その工程を施す過程における複数の動作の理解をウェブアプリケーションを通して容易にすることが重要である。そこで本論文では、実装したシステムの、野菜育成動画自動共有機能、環境情報自動共有機能、野菜育成ダイジェスト動画自動共有機能、ウェブアプリケーションの評価と考察を行った。

その結果、野菜育成動画自動共有機能は、ユーザの施した全ての工程のうち78%をシステムに安定して自動的に保存していた。しかし、全ての工程を安定して保存するためには、autoMov.phpのプログラムの改善とモーションセンサ NaPiOn の設置位置の改善が必要であることが明らかになった。

環境情報自動共有機能は、家庭菜園の環境情報を31.6秒間隔でシステムに安定して自動的に保存していた。しかし、環境情報を設計で想定していた時間間隔で保存するためには、post.phpが情報を送信するタイミングの制

御を厳密に設計することが必要であることが明らかになった。

野菜育成ダイジェスト動画自動共有機能は、家庭菜園の環境と野菜の育成状況の変化を野菜育成ダイジェスト動画として、システムに安定して自動的に保存していなかった。そこで、安定して野菜育成ダイジェスト動画を保存するためには、VLCのプロセスの確実なタイムアウト処理が必要であることが明らかになった。

ウェブアプリケーションは、蓄積された野菜育成動画と環境情報と野菜育成ダイジェスト動画を体系的に提供したが、システムに蓄積された複数の情報から必要な情報を取得するまでの負担を十分に低減していなかった。一方、野菜育成動画と環境情報グラフにより、家庭菜園の工程を施すタイミングと、その工程を施す過程における複数の動作の理解を容易にしていた。必要な情報を取得するまでの負担を低減するためには、野菜育成動画共有ページと環境情報グラフ共有ページを統合することで、必要な野菜育成情報を見つけ出すまでのステップを減らす必要があることが明らかになった。また、家庭菜園の工程を施すタイミングと、その工程を施す過程における複数の動作の理解をより容易にするためには、クライアントサイドシステムのアームを改良し、容易にカメラの位置を変えられる機構にすることと、処理性能の高い機器を利用し、高い画質で動画を共有可能にすることで、野菜育成動画の質を上げる必要のあることが明らかになった。

これらの結果から明らかになった課題を解決することで、野菜育成動画自動共有機能、環境情報自動共有機能、野菜育成ダイジェスト動画自動共有機能により、野菜育成動画、環境情報、野菜育成ダイジェスト動画をシステムに蓄積するまでの負担を低減できる。また、ウェブアプリケーションによって、システムに蓄積された複数の情報から必要な情報を取得するまでの負担を低減できる。そして、それにより、家庭菜園の工程を施すタイミングと、その工程を施す過程における複数の動作の理解を容易にすることができる。その結果、野菜育成情報自動共有システムが、動作の伴う野菜育成情報の共有を円滑にし、理解を促すことが可能になると期待できる。

第9章

展

望

本論文は、家庭菜園に満足していない初心者の割合が高いという問題に着目した。その問題の要因として、家庭菜園を行うための環境が十分に整っていないことと、初心者が野菜の育成情報を十分に理解していないことを挙げた。これらの課題を初心者にとって負担の少ない方法で解決するために、家庭菜園を行うための環境が十分に整っていない場合でも野菜の育成情報を容易に理解し、実践することで家庭菜園を長く楽しめるような環境が必要であると考えた。初心者が野菜の育成情報を十分に理解していない要因として、インターネットから取得する野菜の育成情報を理解し実践する際の負担が大きいことを挙げた。その負担を解消するために、(1) 様々な環境における野菜の育成情報を参照でき、(2) 動作の伴う野菜の育成情報の共有を円滑にし、理解を促し、(3) 個人の家庭菜園の環境や野菜の育成状況をもとにコミュニケーションを図れる様なシステムが必要であると考えた。本論文では、そのシステムを実現するために(2)の機能に着目し、野菜育成情報自動共有システムの提案を行った。

野菜育成情報自動共有システムは、動作の伴う野菜の育成情報の共有を円滑にし、理解を促すという指針に基づいて設計された。その設計指針に沿って評価した結果、動作の伴う野菜育成情報の共有を円滑にするためには、以下の項目の改善が必要であることが明らかになった。

- autoMov.php とモーションセンサ NaPiOn の設置位置の改善
- post.php の情報を送信するタイミングの厳密な設計

- VLC のプロセスの確実なタイムアウト処理
- 野菜育成動画共有ページと環境情報グラフ共有ページを統合することで、必要な野菜育成情報を見つけ出すまでのステップの低減

また、動作の伴う野菜育成情報の理解を促すためには、以下の項目の改善が必要であることが明らかになった。

- 容易にカメラの位置を変えられるための、クライアントサイドシステムのアームの改善
- 処理性能の高い機器を利用し、高い画質で動画を共有する事による、野菜育成動画の質の向上

これらの課題を解決することで、野菜育成情報自動共有システムが動作の伴う野菜育成情報の共有を円滑にし、理解を促すことが可能になると期待できる。一方、今後の展望として(2)による情報の蓄積とともに、(1)と(3)の機能の実装も行う。

(1)の機能は、様々な環境における野菜の育成情報を参照可能にする。従来の情報リソースの場合、ユーザの家庭菜園の環境が十分に整っておらず、それらの情報が前提とした家庭菜園の環境を満たしにくく、初心者にとって実践することが難しかった。そこで、蓄積されたデータからその環境でも実践可能な育成情報を取得できる仕組みを提供する。この機能を実現するためには、蓄積した情報を分析し、その中から条件を満たす野菜育成情報を探す仕組みが必要である。

(3)の機能は、個人の家庭菜園の環境や野菜の育成状況を基にコミュニケーションを行うことを可能にする。現状のシステムには、投稿情報自動共有機能が実装されているが、よりユーザが効率的にコミュニケーションを図るためには改善が必要である。例えば、外部のソーシャルメディアサービスと連携や、コミュニケーションベースのユーザインターフェイスの設計の改変が考えられる。

これらの機能を実装する事で、本研究の目指す環境の実現を目指す。

謝 辞

本研究の指導教員であり、幅広い知見からの的確な指導と暖かい励ましやご指摘をしていただきました慶應義塾大学大学院メディアデザイン研究科の砂原秀樹教授に心から感謝いたします。

研究の方向性について様々な助言や指導をいただきました慶應義塾大学大学院メディアデザイン研究科の稲見昌彦教授に心から感謝いたします。

研究指導や論文執筆など数多くの助言を賜りました慶應義塾大学大学院メディアデザイン研究科の石倉洋子教授に心から感謝いたします。

研究を行う上で多くの助言や助力をいただいた山内正人さん、廣井慧さん、松井加奈絵さんに心から感謝いたします。

さまざまな面から研究活動を支えていただき、時に苦楽を共にした慶應義塾大学大学院メディアデザイン研究科 NetworkMediaProject の皆様に心から感謝いたします。

最後に、研究活動に関するご理解とともに、経済面や生活面において支援していただきました家族に心から感謝いたします。

参 考 文 献

- [1] 矢野経済研究所 ガーデニング市場に関する調査結果 2011
- [2] アイリスオーヤマ ベランダ菜園ニストが急増 (2010/08/27 アイリス家庭菜園ドットコム会員 2010/07/01-08/01 回答者数 1879 人)
<https://www.irisohyama.co.jp/news/2010/0827.pdf>
- [3] アサヒビール 旬の話題データバンク 家庭菜園をしていますか?(全国の 20 歳以上の男子、有効回答数 1510 人、インターネット調査 2010/03/03-03/09)
<https://snet.asahibeer.co.jp/contents/research/weekly/?W=324>
- [4] 東急沿線アンケート調査データ 家庭菜園に関するアンケート (東急沿線コミュニティ TOQ 楽会員 WEB アンケート調査 2008/07/25-2008/08/03 回答総数 3207 名)
<http://www.selun.ne.jp/business/marketing/areavoice/20080033/index.php>
- [5] タキイ種苗 2010 年度 野菜と家庭菜園に関する調査 関連資料
http://www.takii.co.jp/info/gif/news_100831.pdf
- [6] アイリスオーヤマ 家庭菜園に関するアンケート概要 (アイリス家庭菜園ドットコム会員 2006/05/25-6/25 回答者数 505 人 2007/05/24-06/25 回答者数 753 人 2008/06/02-07/07 回答者数 1462 人) <http://www.irisohyama.co.jp/news/2009/0928.pdf>
- [7] 北条雅章 ベランダでできるコンテナ野菜作り 池田書店
- [8] 藤田智 野菜づくり大図鑑 講談社
- [9] BIGLOBE ファーム NEC <http://farm.biglobe.ne.jp/>

- [10] Click&Grow <http://www.clickandgrow.com/>
- [11] EasyBloom Plant Sensor <http://www.easybloom.com/>
- [12] farmbox <https://sites.google.com/site/bsfarmbox/>

付録 A

野菜育成動画自動共有システムの主なプログラム

A.1. アプリケーションサーバーの主なプログラム

A.1.1 sensors_controller.php の armadillo ファンクション

```
public function armadillo(){
    $mac=$this->params['pass'][0];
    $ir1=$this->params['pass'][1];
    $ir2=$this->params['pass'][2];
    $armadillos=$this->Armadillo->find('all',
    array('conditions'=>array('Armadillo.mac'=>$mac)));
    if(!empty($armadillos)){
        $armadillos[0]['Armadillo']['ip']=$_SERVER['REMOTE_ADDR'];
        $armadillos[0]['Armadillo']['date']=time();
        $armadillos[0]['Armadillo']['ir1']=$ir1;
        $armadillos[0]['Armadillo']['ir2']=$ir2;
        $armadillos[0]['Armadillo']['num']=
        $armadillos[0]['Armadillo']['num']+1;
        if($armadillos[0]['Armadillo']['sensor']==0
        ||(time()-$armadillos[0]['Armadillo']['sensor'])>30){
            #sampling per second is 30
        }
    }
}
```

```

$save['Sensor']['date']=time();
$save['Sensor']['ir1']=$ir1;
$save['Sensor']['ir2']=$ir2;
$save['Sensor']['humid']=$this->params['pass'][6];
$save['Sensor']['temp']=$this->params['pass'][5];
$save['Sensor']['illumi']=$this->params['pass'][3];
$save['Sensor']['water']=$this->params['pass'][4];
$save['Sensor']['mac']=$armadillos[0]['Armadillo']['mac'];
$save['Sensor']['irsum']=$armadillos[0]['Armadillo']['irsum'];
$save['Sensor']['num']=$armadillos[0]['Armadillo']['num'];
$this->Sensor->create();
$out=$this->Sensor->save($save);
$armadillos[0]['Armadillo']['num']=0;
$armadillos[0]['Armadillo']['irsum']=0;
$armadillos[0]['Armadillo']['sensor']=time();
}

if($ir1>1000&&$ir2<20){
if($armadillos[0]['Armadillo']['streaming']==0){
$armadillos[0]['Armadillo']['streaming']=time();
if($armadillos[0]['Armadillo']['mov']==0){
$armadillos[0]['Armadillo']['mov']=time();
}
system("wget --spider http://{IP Address of Movie Server}/
command/command.php?create=".$armadillos[0]['Armadillo']['streaming']."_".
.$armadillos[0]['Armadillo']['mac']."_".$armadillos[0]['Armadillo']['ip']);
}
$armadillos[0]['Armadillo']['count']=$armadillos[0]['Armadillo']['count']+1;
$armadillos[0]['Armadillo']['irsum']=$armadillos[0]['Armadillo']['irsum']+1;

```

```

}

if($armadillos[0]['Armadillo']['streaming']
!=0&&(time()-$armadillos[0]['Armadillo']['streaming'])>30){
if($armadillos[0]['Armadillo']['count']>5){
$armadillos[0]['Armadillo']['streaming']=time();
}else{
#stop the process of the vlc
system("wget --spider http://{IP Address of Movie Server}/command/
command.php?delete=".$armadillos[0]['Armadillo']['mov']."_".
$armadillos[0]['Armadillo']['mac']);
$armadillos[0]['Armadillo']['streaming']=0;
$armadillos[0]['Armadillo']['mov']=0;
}
$armadillos[0]['Armadillo']['count']=0;
}
$this->Armadillo->save($armadillos[0]);
}
}

```

A.1.2 sensors_controller.php の demo ファンクション

```

public function demo(){
if(!empty($this->data["Comment"])){
$this->Comment->save($this->data);
}
$this->session["mac"]=$this->Session->read("mac");
$this->session["name"]=$this->Session->read("name");
}

```

```

$this->set("session",$session);
if(!empty($this->params['pass'][0])){
$name=$this->params['pass'][0];
$this->set("name",$name);
}
if(!empty($this->params['pass'][1])){
$day=$this->params['pass'][1];
$this->set("day",$day);
}
if(!empty($this->params['pass'][2])){
$hour=$this->params['pass'][2];
$this->set("hour",$hour);
}
if(!empty($day)){
if($day=="delete"&&!empty($hour)){
$check=$this->Comment->find('first',
array("conditions"=>array("Comment.id"=>$hour)));
if($check["Comment"]["name"]==session["name"])
$this->Comment->delete($hour);
}
}
$armadillos=$this->Armadillo->find('all');
$this->set("armadillos",$armadillos);
$armadillo=$this->Armadillo->find('first',
array('conditions'=>array("Armadillo.name"=>$name)));
if(empty($day)){
$sensors=$this->Sensor->find('all',
array('conditions'=>array("Sensor.mac"=>$armadillo["Armadillo"]["mac"])));
}else{

```

```

$sensors=$this->Sensor->find('all',
array('conditions'=>array("Sensor.mac"=>$armadillo["Armadillo"]["mac"],
'Sensor.date BETWEEN ? AND ?' =>array($day,$day+24*60*60)));
}
$date0 = date("m/d/Y 00:00 ",$sensors[0]["Sensor"]["date"]);
$date1 = date("m/d/Y 23:59 ",$sensors[0]["Sensor"]["date"]);
$count=count($sensors);
$date0=$date1=0;
for($i=0;$i<$count;$i++){
if(empty($sensors[$i]["Sensor"]["day"])){
$sensors[$i]["Sensor"]["day"]=
strtotime(date("m/d/Y 00:00 ",$sensors[$i]["Sensor"]["date"]));
$this->Sensor->save($sensors[$i]);
}
if(empty($sensors[$i]["Sensor"]["diff"])){
$sensors[$i]["Sensor"]["diff"]=
$sensors[$i]["Sensor"]["date"]-$sensors[$i]["Sensor"]["day"];
$this->Sensor->save($sensors[$i]);
}
}
}
$fp=fopen("/var/www/html/video.dat","r");
$i=0;
$j=0;
while($line[$i]=fgets($fp)){
if(!empty($line[$i])){
$movies_[$j]=$line[$i];
$j++;
}
}
$i++;

```

```

}
$j=0;
for($i=0;$i<count($movies_);$i=$i+2){
$data=explode(".", $movies_[$i]);
$args=explode("_", $data[0]);
$num=$this->Movie->find('count', array('conditions'=>
array('Movie.date'=>$args[1])));
if($num>0)continue;
$save[$j]["Movie"]["date"]=$args[1];
$day=strtotime(date("m/d/Y 00:00 ", $save[$j]["Movie"]["date"]));
$save[$j]["Movie"]["day"]=$day;
$save[$j]["Movie"]["diff"]=$save[$j]["Movie"]["date"]-$save[$j]["Movie"]["day"];
$save[$j]["Movie"]["mac"]=$args[0];
$this->Movie->save($save[$j]);
$j++;
}
$comments=$this->Comment->find('all',
array('conditions'=>array('Comment.to_name'=>$name), 'order'=>
array('Comment.date'=>'desc')));
$themovies=$this->Movie->find("all", array("conditions"=>
array("Movie.mac"=>$armadillo["Armadillo"]["mac"])));
$j=0;
for($i=0;$i<count($themovies);$i++){
$newmovies[$j]=$themovies[$i]["Movie"]["date"].".ogg";
$j=$j+2;
}
$this->set("movies", $newmovies);
$this->set("sensors", $sensors);
$this->set("comments", $comments);

```

```
}
```

A.1.3 sensors_controller.php の video ファンクション

```
public function video(){
if(!empty($this->data["Comment"])){
$this->Comment->save($this->data);
}
$session["mac"]=$this->Session->read("mac");
$session["name"]=$this->Session->read("name");
$this->set("session",$session);
$name=$this->params['pass'][0];
$day=$this->params['pass'][1];
$hour0=$this->params['pass'][2];
$hour1=$this->params['pass'][3];
if(!empty($this->params['pass'][4]))
$command=$this->params['pass'][4];
if(!empty($this->params['pass'][5]))
$comment_id=$this->params['pass'][5];
$armadillo=$this->Armadillo->find('first',
array('conditions'=>array("Armadillo.name"=>$name)));
if(!empty($command)){
if($command=="delete"&&!empty($comment_id)){
$check=$this->Comment->find('first',
array("conditions"=>array("Comment.id"=>$comment_id)));
if($check["Comment"]["name"]===$session["name"])
$this->Comment->delete($comment_id);
}
}
}
```

```

$fp=fopen("/var/www/html/video.dat","r");
$i=0;
$j=0;
while($line[$i]=fgets($fp)){
if(!empty($line[$i])){
$movies_[$j]=$line[$i];
$j++;
}
$i++;
}
$j=0;
for($i=0;$i<count($movies_);$i=$i+2){
$data=explode(".", $movies_[$i]);
$args=explode("_", $data[0]);
$num=$this->Movie->find('count',
array('conditions'=>array('Movie.date'=>$args[1])));
if($num>0)continue;
$save[$j]["Movie"]["date"]=$args[1];
$day=strtotime(date("m/d/Y 00:00 ", $save[$j]["Movie"]["date"]));
$save[$j]["Movie"]["day"]=$day;
$save[$j]["Movie"]["diff"]=
$save[$j]["Movie"]["date"]-$save[$j]["Movie"]["day"];
$save[$j]["Movie"]["mac"]=$args[0];
$this->Movie->save($save[$j]);
$j++;
}
$themovies=$this->Movie->find("all",
array("conditions"=>array("Movie.mac"=>$armadillo["Armadillo"]["mac"])));
$j=0;

```

```

for($i=0;$i<count($themovies);$i++){
$newmovies[$j]=$themovies[$i]["Movie"]["date"].".ogg";
$j=$j+2;
}
$this->set("movies_",$newmovies);
$armadillo=$this->Armadillo->find('first',
array('conditions'=>array("Armadillo.name"=>$name)));
$movs=$this->Movie->find('all',
array('conditions'=>
array("Movie.mac"=>$armadillo["Armadillo"]["mac"],
"Movie.day"=>$day,'Movie.diff BETWEEN ? AND ?' =>
array($hour0*60*60, $hour1*60*60))));
$sensors=$this->Sensor->find('all',
array('conditions'=>
array('Sensor.mac'=>$armadillo["Armadillo"]["mac"],
'Sensor.day'=>$day,'Sensor.diff BETWEEN ? AND ?'=>
array($hour0*60*60,$hour1*60*60))));
$comments=$this->Comment->find('all',
array('conditions'=>
array('Comment.to_name'=>$name,'Comment.to_day'=>$day),
'order'=>array('Comment.date'=>'desc'))));
$this->set("name",$name);
$this->set("mac",$armadillo["Armadillo"]["mac"]);
$this->set("movies",$movs);
$this->set("day",$day);
$this->set("hour0",$hour0);
$this->set("hour1",$hour1);
$this->set("sensors",$sensors);
$this->set("comments",$comments);

```

```
}
```

A.1.4 sensors_controller.php の pictures フังก์ション

```
public function pictures(){
if(!empty($this->params['pass'][0])){
$data['Picture']['date']=time();
$data['Picture']['day']=strtotime(date("m/d/Y 00:00 ",time()));
$data['Picture']['mac']=$this->params['pass'][0];
$this->Picture->save($data);
$armadillo=$this->Armadillo->find('first',
array('conditions'=>array('Armadillo.mac'=>$data['Picture']['mac'])));
echo $armadillo["Armadillo"]["ip"];
}else{
echo "404";
}
}
```

A.2. 動画サーバーの主なプログラム

A.2.1 autoMovs.php

```
<?php
while(1){
system("ntpdate {NTPserver}");
system("php /var/www/html/video.php");
system("mv /var/www/html/video.dat.tmp /var/www/html/video.dat");
$out=system("ls /var/www/html/tmp/ > /var/www/html/command/list.dat");
```

```

$fp=fopen("/var/www/html/command/list.dat","r");

$i=0;
while($line[$i]=fgets($fp)){
$out[$i]=split("-", $line[$i]);
if($out[$i][2]==0){
$formArray = explode( "\n", $out[$i][3] );
system("sudo -u {GeneralUserName}
php /var/www/html/command/movies.php ".
$out[$i][1]." ".$out[$i][0]." ".$out[$i][3]);
$command="echo '".time()." ".$out[$i][0]." ".$out[$i][1]
.'" ".$out[$i][2]." \n'
>> /var/www/html/command/streaming.log";
system($command);
$command="mv /var/www/html/tmp/".
$out[$i][0]."-"$out[$i][1]."0-"$out[$i][3]."-"
"/var/www/html/tmp/"$out[$i][0]."-"$out[$i][1]."1";
echo "start getting streaming".$command."\n";
system($command);
}
if($out[$i][2]==1){
echo "now Streaming...["$out[$i][0]."-"$out[$i][1]."1\n";
}
if($out[$i][2]==9){
system("rm -f /var/www/html/tmp/"$out[$i][0]."-"$out[$i][1]."9");
system("rm -f /var/www/html/tmp/"$out[$i][0]."-"$out[$i][1]."1");
system("php /php/process.php ".$out[$i][1]);
echo "stop getting streaming".$command."\n";
}
}

```

```
$i++;  
}  
}  
?>
```

A.2.2 movies.php

```
<?php  
$date=time();  
$command="nohup cvlc -vvv http://".$argv[3].  
":8080/?action=stream --sout='#transcode{vcodec=theo,vb=1024}:  
standard{mux=ogg,dst=/var/www/html/armadillos/".$argv[2]."/movs/" .  
$argv[1].".ogg,access=file}'> /var/www/html/command/process.log &";  
?>  
system($command);
```

A.2.3 autoPics.php

```
<?php  
while(1){  
  
$fp=fopen("/tmp/mac.dat","r");  
$i=0;  
while($mac=fgets($fp)){  
$macAddresses[$i]=$mac;  
}  
fclose($fp);  
for($i=0;$i<count($macAddresses);$i++){  
system("nohup php /php/pictures.php ".$macAddresses[$i]);
```

```
}  
}
```

A.2.4 pictures.php

```
<?  
$mac=$argv[1];  
$date=time();  
$day=strtotime(date("m/d/Y 00:00 ",$date));  
if(file_exists("/var/www/html/armadillos/".$mac."/pics/".$day)){  
}else{  
system("mkdir /var/www/html/armadillos/".$mac."/pics/".$day);  
}  
system("wget http://{ApplicationServer}/armadillo/sensors/pictures/" .  
$mac." -O /tmp/".$mac);  
$fp=fopen("/tmp/".$mac,"r");  
$ip=fgets($fp);  
fclose($fp);  
$ip=rtrim($ip);  
system("wget http://".$ip.":8080/?action=snapshot -O  
/var/www/html/armadillos/".$mac."/pics/".$day."/".$date.".jpg");  
$check=getimagesize("/var/www/html/armadillos/".$mac .  
"/pics/".$day."/".$date.".jpg");  
if(empty($check[0])||$check[0]==0)exit();  
  
system("ls -t /var/www/html/armadillos/".$mac."/pics/" .  
$day."/> /tmp/".$mac.".list");  
$fp=fopen("/tmp/".$mac.".list","r");  
$i=0;
```

```

while($list[$i]=fgets($fp)){
    $i++;
}
fclose($fp);
sort($list);

for($i=0;$i<count($list);$i++){
    if(empty($list[$i])){
        echo $i." is passed";
        continue;
    }
    $list[$i]=rtrim($list[$i]);
    $num=sprintf("%05d",$i);
    $size=getimagesize("/var/www/html/armadillos/"
        .$mac."/pics/" . $day . "/" . $list[$i]);
    if(empty($size[0])){
        system("rm -f /var/www/html/armadillos/"
            .$mac."/pics/" . $day . "/" . $list[$i]);
        continue;
    }
    system("mv /var/www/html/armadillos/"
        .$mac."/pics/" . $day . "/" . $list[$i] .
        " /var/www/html/armadillos/" . $mac . "/pics/" . $day . "/" .
        $num . ".jpg");
    echo " $i is renamed into $num ";
}

print_r($list);
echo "// START FFMPEG //";

```

```

$ffmpegcommand='nohup ffmpeg -i /var/www/html/armadillos/'$mac.'/pics/'
.$day.'/%05d.jpg -b 1200 -aspect 4:3 -f avi -vcodec mpeg4 -r 10 -y
/var/www/html/armadillos/'$mac.'/pics/'$day.'.avi &';
echo $ffmpegcommand;
system($ffmpegcommand);
echo "// START VLC //";
$vlccommand='sudo -u {GeneralUserName}
nohup cvlc -vvv /var/www/html/armadillos/'
.$mac.'/pics/'$day.'.avi --sout="#transcode{vcodec=theo,vb=1024}:standard
{mux=ogg,dst=/var/www/html/armadillos/'$mac.'/pics/'
$day.'.ogg,access=file}"&';
echo $vlccommand;
system("echo '$day 15' > /php/kill.list");
system($vlccommand);

```

A.3. 野菜育成情報送信用マイクロコントローラー の主なプログラム

A.3.1 post.php

```

<?php
while(1){
$time=time();
system('ifconfig eth0 > /home/guest/mac.dat');
$i=0;
$fp=fopen('/home/guest/mac.dat','r');
while($out[$i]=fgets($fp)){
$i++;}
$out=$out[0];

```

```

$out=explode(" ",$out);
$out=explode(":",$out[10]);
$mac=$out[0].$out[1].$out[2].$out[3].$out[4].$out[5];
fclose($fp);
$i=0;
$mac=$out[0].$out[1].$out[2].$out[3].$out[4].$out[5];
$mac=$out[0].$out[1].$out[2].$out[3].$out[4].$out[5];
fclose($fp);
$out=system("wc -l /home/guest/sensor.dat");
$out=explode(" ",$out);
$j=0;
for($i=0;$i<count($out);$i++){
    if(!empty($out[$i])){
        $out_converted[$j]=$out[$i];
        $j++;
    }
}
$line_num=$out_converted[0];
$command="awk 'NR=="$line_num."' /home/guest/sensor.dat";
$out=system($command);
$out=split(":",$out);
system('wget --spider http://{ApplicatoinServer}/armadillo/
sensors/armadillo/'$mac.'/'$out[0].'/'$out[1].'/'$out[2].
'/'$out[3].'/'$out[4].'/'$out[5].'/'$out[6].'/'$out[7].
'/'$out[8]);
system(":>/home/guest/sensor.dat");
sleep(1);
}

```