

農業生産履歴システムにおける
多様な情報の統合利用手法に関する研究

2016年1月

伊藤 淳士

農業生産履歴システムにおける
多様な情報の統合利用手法に関する研究

筑波大学大学院
生命環境科学研究科
先端農業技術科学専攻
博士（農学）学位論文

伊藤 淳士

目次

第1章 緒言	1
1.1 はじめに	2
1.2 背景.....	3
1.2.1 食の安全・安心	3
1.2.2 食の安全・安心に関わる取り組み.....	4
1.2.3 トレーサビリティシステム	5
1.2.4 生産履歴管理システム	6
1.3 農業と情報技術.....	7
1.3.1 情報技術の普及	7
1.3.2 農業分野における情報利用	7
1.4 本論文の構成	12
第2章 生産工程管理のためのウェブシステムの構築	13
2.1 背景および目的.....	14
2.2 方法.....	15
2.2.1 生産資材情報のデータベース化	15
2.2.2 生産履歴情報の電子化	15
2.2.3 蓄積情報の検査, 分析	15
2.2.4 農業現場における実証試験	15
2.3 システム構成	16
2.3.1 サーバ側の構成	16
2.3.2 クライアント側の構成	19
2.4 生産資材データベース	20
2.4.1 農薬データベース.....	20
2.4.2 肥料データベース.....	22
2.4.3 作物コードデータベース	26
2.5 機能.....	30
2.5.1 手書き帳票の電子化.....	30
2.5.2 資材データに関する集計, 分析	35
2.5.3 特別栽培への対応.....	40
2.5.4 集計機能.....	43
2.6 PCを用いた記帳システム	46
2.6.1 手書き記帳方式との違い.....	46
2.6.2 生産者向けインターフェース	48

2.7 システムの評価	51
2.7.1 実証試験.....	51
2.7.2 ウェブブラウザに関する問題.....	51
2.7.3 開発言語に関する問題.....	52
2.7.4 クライアント端末に関する問題.....	55
2.8 考察.....	56
第3章 保守性, 拡張性向上のためのシステムの再構築.....	57
3.1 背景および目的.....	58
3.2 情報システムに関する技術的考察.....	59
3.2.1 計算機の変遷.....	59
3.2.2 インターネット.....	60
3.2.3 クラウドコンピューティング.....	60
3.2.4 Web API と分散システム.....	61
3.2.5 HTML.....	63
3.2.6 CSS.....	63
3.2.7 JavaScript.....	64
3.2.8 ウェブブラウザ.....	65
3.3 システムの運用・保守に係る課題.....	67
3.3.1 持続可能性 (Sustainability).....	67
3.3.2 サステイナブルな情報システム.....	68
3.4 生産工程管理システム”apras”の開発.....	71
3.4.1 方法.....	71
3.4.2 旧システムの問題点とその解決方法.....	71
3.4.3 apras のインプリメンテーション.....	74
3.4.4 機能の API 化.....	76
3.4.5 PC 用クライアントアプリケーション.....	76
3.4.6 モバイル端末用 apras-app.....	79
3.4.7 Fax-OCR.....	82
3.5 apras の評価.....	84
3.6 apras の利用実態に関する考察.....	86
3.6.1 利用実態の分析.....	86
3.6.2 生産履歴帳票数.....	86
3.6.3 農薬使用履歴の分析.....	87
3.6 考察.....	93

第4章 API を核としたシステム拡張手法	98
4.1 背景および目的.....	99
4.2 センサデバイスのための拡張	102
4.2.1 フィールドサーバ.....	102
4.2.2 モバイル型計測機器データ	105
4.3 農業機械のための拡張.....	107
4.3.1 農薬散布支援技術.....	107
4.3.2 農業機械の走行軌跡記録.....	111
4.4 考察.....	114
第5章 総合考察	119
5.1 まとめ.....	120
5.2 農業現場における ICT 活用に関する考察	123
5.3 apras の現場活用に関する考察	126
5.4 展望.....	127
5.4.1 営農改善への活用.....	127
5.4.2 ビッグデータの収集.....	127
5.4.3 農業における情報利用の発展方向.....	128
謝辞.....	130
引用文献.....	131
摘要.....	135
Abstract	138

第 1 章

緒言

1.1 はじめに

農業は、人間の生命活動を支えるという点において国の根幹を成す産業であると言える。しかし、わが国における食料自給率はカロリーベースで 39%（農林水産省 2014）と先進国の中で最低の水準であり、また、農業従事者の過半数が 65 歳以上であるなど（農林水産省 2015）、わが国の食料生産基盤は非常に脆弱だと言わざるをえない。担い手が減少する中で農業の持続的な発展を図るには、農業生産の高度化が不可欠である。

本論文では、ICT を導入することで生産履歴を始めとした農業情報を高度に管理するためのシステムを構築すると共に、システムの安定的な運用に必要な技術的要件を明らかにする。また、農業現場における実証試験を通じてシステムの評価および有用性を実証する。

第 1 章では、食と農を取り巻く情勢と計算機やインターネットなど情報通信技術の情勢についての整理を行い、農業分野における情報技術利用の方向性を示す。第 2 章では、JA における生産履歴管理を高度化するためのシステムを構築し、実証試験を通じて問題点を明らかにする。第 3 章では、第 2 章での実証試験の結果を踏まえてシステムの安定運用に必要な技術的側面に関する考察を行い、より安定的な運用のためにシステムへと再構築する。第 4 章では、第 3 章で構築したシステムのいくつかの拡張を実装することで、将来的な発展の方向性を示す。第 5 章では、本研究を通じた考察を行い残された課題について整理すると共に、今後の展望を示す。

1.2 背景

1.2.1 食の安全・安心

食の安全・安心は多くの国民の感心事であり，食料生産を担う農業においては食のリスクを排除すべく様々な対応が求められる．わが国における食の安心・安全の問題は，1996年に大阪堺市で起きたO-157による集団食中毒を機に注目され始め，2001年のBSE感染牛の確認以降その関心は一気に高まった(栗原ら 2005)．2002年には，山形県で無登録農薬（ダイホルタン，プリクトラン）を販売していた業者が農薬取締法違反及び毒物及び劇物取締法違反の容疑で逮捕され，その後の調査で44都道府県において無登録農薬の販売または購入が行われていたことが判明した．これを受けて，農薬取締法の一部が改正され2002年12月に成立した（農林水産省 2002a）．また，2003年の食品衛生法改正により一定量を超えて農薬等が残留する食品の販売等を原則禁止するポジティブリスト制度が制定され，2006年より施行された（厚生労働省 2006）．しかしながら，その後も農畜産物の産地偽装，食品への異物混入等の食をめぐる事件・事故は度々起こり，大きな社会問題となってきた．近年では，福島第一原子力発電所の事故で発生した放射性物質による農産物等への汚染の懸念を払拭するために，様々な機関で食品の放射線量検査が実施されるなど，食の安全・安心への関心はこれまでになく高まっている．

内閣府が平成26年8月に行った食品安全モニター課題報告「食品の安全性に関する意識等について」（内閣府・食品安全委員会 2014）によると，「日常生活を取り巻く分野別不安の程度」の項目における「食品安全」に対する回答は，「とても不安を感じる（23.6%）」，「ある程度不安を感じる（47.4%）」と計71.0%が不安を感じており，これは交通事故（62.3%），犯罪（61.6%）の結果を上回っている．同調査の「食品の安全性の観点から感じているハザードごとの不安の程度」の項目における「残留農薬」に対する回答は，「とても不安である（10.1%）」，「ある程度不安である（48.7%）」と計58.8%が不安を感じており，その理由の上位3位は「事業者の法令遵守や衛生管理が不十分（32.7%）」，「安全性についての科学的

な根拠に疑問（17.5%）」「過去に問題になった事例があるために不安（13.2%）」となっている。このことは、安全とされる食品であってもその科学的根拠への疑問や事業者に対する不信感を持つ消費者に対して安心感を与えることはできないということを示している。食の安全に関するリスクコミュニケーションの在り方に関する研究会（厚生労働省 2005）の報告書において、「安全と安心が同義語と受け止められていた時代から両者が乖離しはじめた今の時代に、両者を調和させるためには、リスクコミュニケーションにより信頼を確保していくことが必要」との指摘がなされている。つまり、食の安全・安心の確保のためには生産者が法令等を順守し安全な農産物を生産するだけでは不十分であり、消費者が安心してもらうだけの情報を蓄積し発信することが不可欠だと言える。

1.2.2 食の安全・安心に関わる取り組み

2001年に起こったBSE問題は、わが国における食と農への不安を一気に高める事となった。時を同じくして、食品の虚偽表示等の食の安全・安心を揺るがす事件が相次いで発生したこともあり、食の信頼回復に向けた様々な取り組みが強く求められることとなった。農林水産省は、2002年4月に「食と農の再生プラン」を発表した（農林水産省 2002b）。このプランは、「食の安全と安心の確保」「農業の構造改革を加速化」「都市と農村漁村の共生・対流」の3本柱からなり、第一番目に挙げられた食の安全・安心に関する項目においては、関連法案の改正、トレーサビリティシステムの導入、食育とリスクコミュニケーション、食品表示の信頼回復、ブランド日本の確立などが謳われており、以降農業現場ではこれに沿った対応が求められることとなった。全国農業協同組合連合会は、2002年7月より生産履歴の記帳運動を開始（社団法人・食品需給研究センター 2006）、以降全国のJAに運動が広がっていく（前田 2004）。現在では、JAや産直組合などの生産者団体において、入荷の条件として生産履歴帳票の提出を義務付けることが一般的になりつつある。

近年は、生産工程管理手法にGAP（Good Agricultural Practices）を用いる例が増えてきている。GAPは、日本語では適正農業規範または農業生産工程管理な

どと訳され、生産履歴の記帳を含む農業生産全般を適正に行うための指針である。GAPは、元々欧州で始まった制度であるが現在では世界的に導入が進み、日本においても農林水産省が指針を示している（農林水産省 2010）。GAPの実践に当たっては、食品安全、環境保全、労働安全等の観点で農業生産上のリスク評価に基づくリスク回避のためのルール作りを行い、それらのルールに則った農業活動の実施と点検を行うことで適切なリスク管理を行うことが求められる。

1.2.3 トレーサビリティシステム

相次ぐ食品関連の事件を受けて、食品産業における情報管理の高度化が求められることとなった。BSE問題や産地偽装問題の発覚以降、食品流通の追跡、遡及を可能にするトレーサビリティシステムの構築が強く望まれた。政府はBSEの蔓延防止措置として、2003年に「牛の個体識別のための情報の管理及び伝達に関する特別措置法（牛肉トレーサビリティ法）」を制定した。これは、わが国におけるトレーサビリティシステムに関する法制化の最初の例となった。この法整備に伴い、国内で飼育されるすべての肉牛に個体識別番号を記した耳標が装着され、食肉加工を経て販売されるまでのすべての流通経路においてこの個体識別番号が付されることとなった。消費者は、独立行政法人・家畜改良センターのウェブサイト上で個体識別番号から牛の生年月日、性別、種別、飼育地、管理者などの情報を得ることができる（独立行政法人・家畜改良センター 2004）。わが国のトレーサビリティに関する法律は牛肉トレーサビリティ法の他に、2011年に制定された「米穀等の取引等に係る情報の記録及び産地情報の伝達に関する法律（米トレーサビリティ法）」がある。この法律は、玄米や精米など米穀はもちろん、米粉や米菓子、おにぎりや赤飯などの米飯類、餅やみりんなどの米加工食品等、米を原料とする広範な食品を対象としている。

牛肉と米以外の食品のトレーサビリティについては、法整備はなされていないものの生産者、事業者らが自主的にシステムを構築した事例は数多く存在する。例えば、農林水産省が2005年から2007年にかけて行った「ユビキタス食の安全・安心システム開発事業」において、多くのトレーサビリティシステムの事例を見

出すことができる(社団法人・食品受給研究センター 2005, 2006, 2007, 2008)。

1.2.4 生産履歴管理システム

食の安全・安心の基礎となる生産履歴情報は、生産工程を通じて随時記録される必要がある。作目、品種等が異なると記録方法は異なる。そのため、複数の作目・品種を栽培すると記帳作業は複雑となり、農業者、生産者団体ともにその業務の負担は大きくなる。肥料や農薬については、その使用方法が適切であるかを確認する必要があるが、その確認には高度な専門的知識が必要である。また、煩雑な計算処理を伴うことから、これは非常に困難な業務である。

これらの問題に対処するため、生産履歴情報を収集・管理するシステムがいくつか開発された。例えば、農薬の適正使用に関しては、農薬適正使用ナビゲーションシステム(南石ら 2005)、農薬使用適正判定・履歴入力アプリケーション(菅原ら 2006)、農薬使用リスク管理システム(南石ら 2006)、作業履歴に関しては圃場地図ベース作業計画管理ソフト(吉田ら 2009)、GIS 機能を備えた営農情報管理システム「FARMS」(林ら 2009)などがある。これらは、GIS の機能をベースに圃場に関わる様々な情報を扱うことができる。ICT を活用した生産履歴管理システムは営農の高度化に寄与するが、次節で述べる通り農業分野における ICT の利用は必ずしも進んでおらず、IT 機器の利用を前提にしたシステムは普及に一定の制限がかかる点には注意が必要である。この点については第2章で検証する。

近年は、クラウドの進展が目覚ましく様々なアプリケーションやサービスがクラウドシステムとして稼働している。農業分野においても、例えば海外では Farm Management System (Kaloxyllos et al. 2012, 2014)が開発されている。これは圃場や生育情報を管理するためのクラウドサービスであり、クラウドサービスの連携によって気象センサなど農業に関わる情報を管理できる仕組みを備えている。クラウドのような新たな技術を取り込むことはシステムの利便性を高める上で重要であり、またシステムの安定的な運用にも寄与すると考えられる。この点については第3章で検証する。

1.3 農業と情報技術

1.3.1 情報技術の普及

総務省の「平成25年通信利用動向調査」（総務省2014）において、平成25年末のわが国のインターネット利用率は82.8%と報告されている（図1-1）。また、インターネット利用端末は、第1位の自宅のパソコン（58.4%）についてスマートフォン（42.4%）が第2位となっている。平成23年の同調査（総務省2012）においては、スマートフォン（16.2%）は第4位であったことから、ここ数年でスマートフォンの普及が急激に進んだことが分かる。世代別に見ると、13才から49才の各年代におけるインターネット利用率はいずれも96%を超えており、若年層から壮年層においてはほぼすべての人がインターネットを利用していると言える。

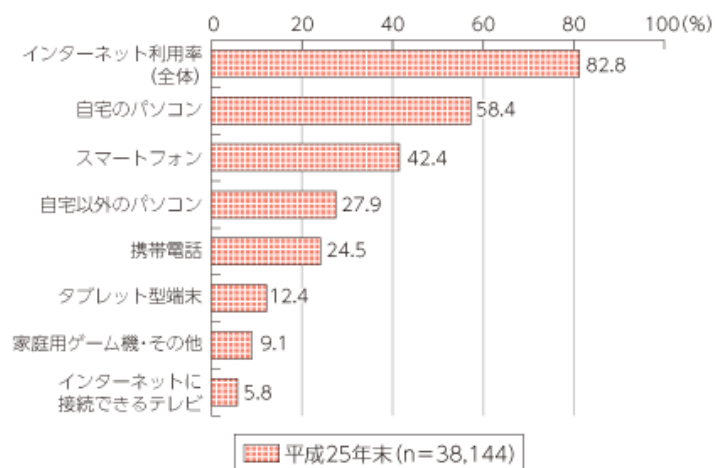
1.3.2 農業分野における情報利用

図1-3に示したとおり、農林水産省(2015)によるとわが国の農業就業人口の過半数は65歳以上であり平成26年における農業従事者の平均年齢は66.7歳となっている。図1-2によると、60歳以上のインターネット利用率は年齢と共に急激に低下し、65-69歳では68.9%、70-79歳では48.9%となっている。これらことから推測すれば、農業従事者のインターネット普及率はおそらく60%前後であり、普及が進んでいるとは言いがたい。

農林水産省(2012)によると、農業従事者のパソコン保有率は76.3%、農業に関してIT機器を利用している割合は50.4%となっている。利用目的は、インターネットを用いて栽培、防除、気象、市況等情報収集（69.2%）、経理事務や経営に関するデータ分析（67.1%）、農作業履歴や出荷履歴の記録（48.8%）などとなっている。農林水産省(2006)では、パソコン保有率は61.2%、IT機器を利用している割合は24.1%と報告されていることから、農業分野においても他の分野ほどはIT機器の利用は盛んではないものの近年急速に利用されるようになってきていることが伺える。

一方海外では、例えば Lazarus(1988)によると 1986 年時点でニューヨークの酪農家のうち 15%がパソコンを所有、8.7%が会計処理にパソコンを利用しているとの報告がある。Park(2003)によると、米国農務省の調査では 2000 年時点での農業従事者のパソコン利用率は 55%、インターネット利用率は 43%と報告されている。また、Batte(2005)によると、2003 年時点でオハイオ州の農家における農業でのパソコン利用率は 44%と報告されている。これらの事例から比較すると、日本におけるパソコンやインターネットの農業利用は海外に比べると高くはないことが分かる。

農業においてパソコン等の IT 機器は、かつては税務処理など経理業務に使われる事が主であったが、近年はインターネット上での気象や市況情報の収集や、農作業に関する記録に利用される事例が増加しており（農林水産省 2006, 2012）、海外においても同様の傾向が報告されている（Batte 2005）。また、パソコンの利用率と農業経営や設備投資の規模との間には相関があることが分かっている（Lazarus 1988, Park 2003, Batte 2005）。わが国においても、農林水産省(2006)によると平成 17 年時点で農業分野全体の IT 機器利用率が 24.1%なのに対して、認定農業者の IT 機器利用率は 43.3%となっており、先進的な農業者が率先して IT 機器を利用してきた傾向が伺える。今後、農業の大規模化、高度化が進む際には、パソコン等の IT 機器の果たす役割がこれまで以上に大きくなると考えられる。



※当該端末を用いて平成25年の1年間にインターネットを利用したことのある人の比率を示す

図 1-1 わが国のインターネット普及率（上図），インターネット利用端末（下図）
（総務省 2014）

Fig 1-1 Internet penetration rate in Japan (above chart), Breakdown of Internet utilization devices (below chart)

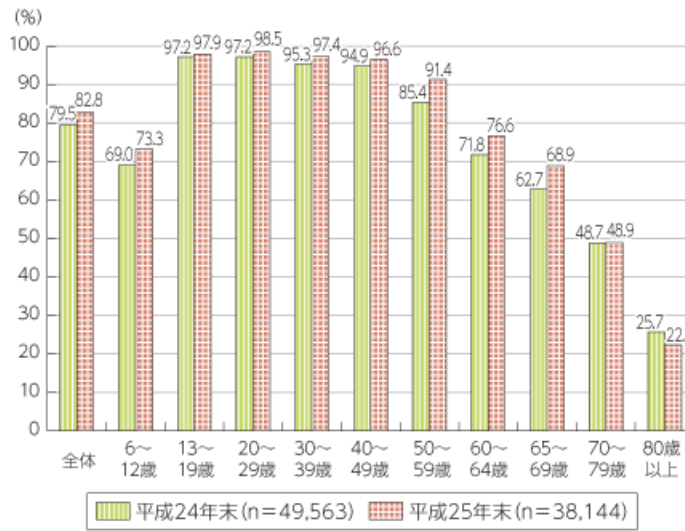


図 1-2 わが国の世代ごとのインターネット利用率（総務省 2014）

Fig 1-2 Internet penetration rate by age in Japan

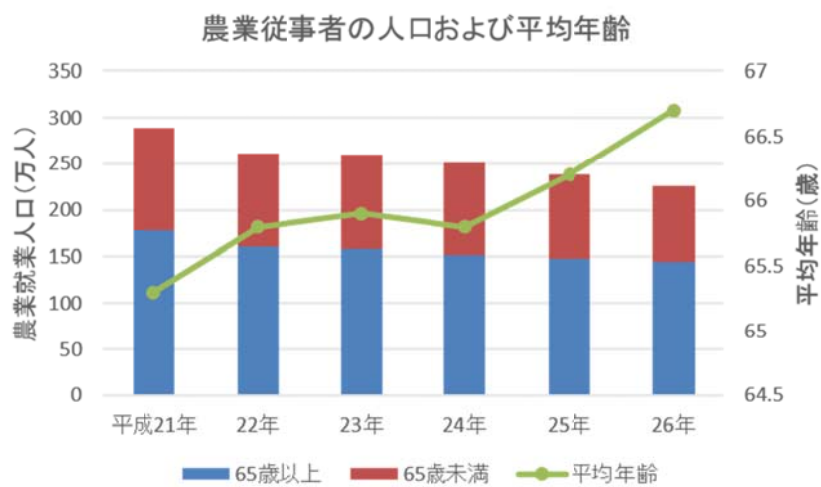


図 1-3 農業就業人口と平均年齢（農林水産省(2015)に基づく）

Fig 1-3 Agricultural workforce and Average age of farmers in Japan

1.4 本論文の構成

本章においては、食と農をめぐる状況と計算機およびインターネットなど ICT に関する状況を整理した。食と農に関わる様々な事件、事故の発生を受けて、食と農への不安感が広がった結果、食と農の再生プランでは食の安全と安心の確保が目標に掲げられ、農業現場では生産履歴の記帳運動が始まった。生産履歴管理には、今後 ICT の活用が不可欠であると考えられるが、インターネットが広く普及した今日においても農業分野では高齢化等の問題を背景に ICT に馴染まない人々が多くいるという点には注意が必要である。

第2章では、JAにおける生産履歴管理を高度化するためのシステムを構築する。本章で指摘したとおり、多くの農業者が IT 機器の利用が難しいことが想定されることから、システムとしてはウェブアプリケーションの形を取るものの農業者は紙ベースでの利用または PC での利用を選択できるようにするなど、すべての農業者が利用できるシステム設計とする。システムは、北海道内において実証試験を行い、その有用性を検証すると共に問題点を明らかにする。

第3章では、ICT に関する技術が指数関数的に発展している現状を分析し、今後安定的にシステムが運用できる条件を検討する。その上で、第2章のシステムを全面的に見直し、保守性、拡張性に優れたシステムとして再構築を行う。

第4章では、第3章で構築したシステムと他のシステムとの連携を行い、システムの拡張性を検証する。センサネットワークや農業機械の情報を生産履歴情報と統合的に扱えるシステムによって、システムが多様な情報を扱うプラットフォームとして機能することを検証する。

第5章では、第4章までに構築したシステムについて考察し、システムの将来展望について述べる。

第 2 章

生産工程管理のためのウェブシステムの構築

2.1 背景および目的

我が国における食の安全性への関心は、BSE 問題、無登録農薬使用問題等の事件をきっかけに 21 世紀に入った頃より急速に高まってきた。また、農産物の産地偽装、残留農薬などの問題が次々と起こったこともあり、農業生産現場に対する消費者等の不信感が蓄積しつつあった。農業政策においても 2002 年 12 月の農薬取締法の改正、2003 年 6 月に発表された「食の安全・安心のための政策大綱」でのトレーサビリティシステム導入への言及、2006 年 5 月より始まったポジティブリスト制の導入など、情勢が目まぐるしく変化したことに対応し、農業生産現場においてはより厳密な生産工程の管理を目的として、JA 等の生産者団体が中心となり、生産履歴の記帳およびそれらの検査を行うといった取り組みが開始された。しかし、各々の栽培ごとに記帳される栽培履歴表は膨大な数になるため、それらの管理業務は生産現場の大きな負担となっており、ICT を活用した農業現場向けの生産工程管理のためのシステム構築が強く望まれていた。

一方で、農業従事者の平均年齢は他の産業に比べて極めて高く、ICT に対する受容性が低い（農林水産省 2006, 2012）といった問題がある。つまり、PC などの IT 機器の利用を前提としたシステムは一部の農業者にしか使われないことが考えられる。しかし、JA 等における生産履歴管理はすべての農業者を対象にすべきであり、ICT の受容度の差がシステム利用の障壁になってはならない。

本章では、農業生産現場における ICT の需要度に関わらず利用可能な生産工程管理システムの要件を明らかにする。その上で、北海道内の JA を対象としたシステムの開発および導入を行い、JA およびその組合員すべてを対象とした実証試験を通じて、その有効性を検証する。

2.2 方法

生産履歴を電子化し効率的に管理する「生産履歴，生産資材マネジメントシステム」を開発し，北海道内のJAにおける実証試験を通じて，農業現場がシステムに求める機能の実装および改良を行う。

2.2.1 生産資材情報のデータベース化

農薬，肥料等の生産資材情報のデータベース化を行なう。農薬データは，社団法人日本植物防疫協会 JPP-NET の提供する農薬データベースを再構築して整備する。肥料データは，本システム利用者を中心にデータの収集・蓄積を行なう。

2.2.2 生産履歴情報の電子化

JAにおいて生産履歴を電子化し効率的に管理できるシステムを開発する。情報機器に不慣れな利用者に考慮して，手書き帳票をスキャナおよびOCRソフトウェアにより電子化する方法を採用する。電子化した情報は，ウェブサーバ上のデータベースで一元管理を行ない，JA職員がウェブブラウザ上で，生産履歴の閲覧・編集等の作業が行えるシステムを構築する。

2.2.3 蓄積情報の検査，分析

電子化した生産履歴のうち，肥料，農薬に関するデータの検査，分析を行なうシステムを開発する。肥料については，成分ごとの集計を行えるものとする。農薬については，農薬取締法における基準に対する適否判断および，特別栽培への対応の一例として北海道内の独自基準であるYES!clean栽培における農薬使用計画に対する適否判断を行なうことができるものとする。

2.2.4 農業現場における実証試験

本システムを，北海道内のJAにて実証試験を行ない，システムの実用性を検証する。

2.3 システム構成

「生産履歴，生産資材マネジメントシステム」は，生産履歴情報および生産資材情報を総合的に管理するシステムである．本システムは，CGI プログラムを実行するためのウェブサーバとウェブサーバに接続するためのクライアント端末からなる．クライアント端末は，複数台用いることが可能であり，少なくとも1台にはスキャナおよびOCR ソフトウェアの導入が必要である．

2.3.1 サーバ側の構成

サーバには，各種の CGI プログラムとデータベースが設置される．推奨動作環境は以下の通りである．

【OS】

Windows Server 2003

【CPU】

インテル社 Pentium III 500MHz と同水準以上．推奨は，インテル社 Pentium4 2.4GHz 以上，または，AMD 社 Athlon XP 2400+以上．

【システムメモリ】

512M バイト以上．推奨は，1G バイト以上．

【ハードディスクドライブ】

システムの導入および運用のために，空き容量 1G バイト以上が必要．

【ウェブサーバ】

Apache 1.3.x または Apache 2.0.x で稼働．ライセンスは，Apache License (フリーウェア)．

【プログラム言語】

Perl 5.8.x で動作可能. Windows 系 OS 向けの Perl としては, ActiveState 社の ActivePerl が利用できる. ライセンスは, ActivePerl Community License (フリーウェア).

【モジュール等】

Perl の追加モジュールとして, Unicode::Japanese の導入が必要. Unicode::Japanese のライセンスは Perl のライセンスに準ずる. また, JavaScript の追加モジュールとして, prototype.js (Version : 1.5.1_rc3) および scriptaculous.js (Version:1.7.0) の導入が必要. ライセンスはいずれも MIT ライセンスで, フリーでの使用が可能.

【プログラム本体】

プログラムは, Perl 言語による複数の CGI が連携して動作することで稼働する. Perl はインタプリタ型言語であるため, ソースコードを保存したファイルをサーバ側に設置するだけで動作可能. プログラム一覧は, 表 2-1 の通り.

表 2-1 プログラム一覧

Table 2-1 Program files

ファイル名	機能
c.pl	全プログラムに共通の Perl モジュール
color.conf	表示色等の設定
cropcode.cgi	作物コードデータベースの管理
culture.cgi	ユーザごとの生産履歴情報の管理
fertdb.cgi	肥料コードデータベースの管理
fertilizer.cgi	肥料使用履歴の管理
fertsearch.cgi	肥料検索プログラム
history.cgi	ユーザごとの履歴情報の管理
member.cgi	ユーザ情報データベースの管理
noyaku.cgi	農薬の適正使用診断プログラム
noyakuinfo.cgi	農薬情報の検索プログラム
noyakudb.cgi	農薬登録情報等のデータベース管理
pesticide.cgi	農薬使用履歴の管理
view.cgi	生産履歴の閲覧画面のコントロール
workday.cgi	生産履歴における作業情報データの管理
yesclean.cgi	YES!clean における資材適用等の検査プログラム
yesdb.cgi	YES!clean 用の基準データ等の管理
ocr.cgi	OCR の読取り修正プログラム
ocr2db.cgi	OCR データのデータベース化プログラム

2.3.2 クライアント側の構成

クライアントは、サーバ上に格納されたデータの閲覧、編集等を行う端末であり、インターネットに接続された一般的な PC であれば種類は問わない。作業は、すべてウェブブラウザ上で行うことができるため、特に新たなソフトウェアをインストールする必要はない。ウェブブラウザは、クライアント 1 同様、Internet Explorer 6 以上、または、FireFox 2.0 に対応。

クライアント端末のうち、少なくとも 1 台には OCR 帳票の読み取りおよびサーバへのデータ転送を行うための端末を設置する必要がある。この端末には、OCR 帳票の読み取りを行うために、市販のソフトウェアである富士通社の DynaEye4.0 およびスキャナを導入する。スキャナは、TWAIN 形式に対応したものであれば種類は問わないが、両面読み取りに対応したオートフィード型のものが望ましい。端末のシステム要件は、DynaEye に準拠するため、OS は Windows、CPU は Pentium 200MHz 以上、メモリは 128M バイト以上が必要である。DynaEye により読み取られたデータをサーバに転送するために、新たに開発した補助プログラムである send.exe を利用する。そのために、Microsoft .NET Framework の実行環境のインストールが必要である。

2.4 生産資材データベース

本システムは、農業資材等の基礎データをデータベースとして構築する機能を有する。ここでは、農薬、肥料に関するデータベース、および、農薬の適正使用診断に必要な、作物コードを取り扱うデータベースについて述べる。なお、データベースはすべて Perl 言語に標準的に組み込まれているデータベースインターフェースモジュールである SQLite (Version: 1.53) を使用して構築している。

2.4.1 農薬データベース

農薬のデータには、社団法人日本植物防疫協会 JPP-NET が有償にて提供する農薬データベースを使用している。このデータベースの利用にあたっては、JPP-NET との契約が別途必要となる。本システムでは、JPP-NET のデータベースを再構築し、最適化した状態でシステム内のデータベースへと変換する。データは、月に5回程度の頻度で内容の更新が行われる（例えば、平成19年4～9月の6ヵ月間では27回の更新があった）。変更は、データの差分がメールにて配信されるので、そのたびに本システムの農薬データベースも更新する必要がある。更新は、すべてウェブブラウザ上で行うことができ、作業時間は一回につき、数十秒～数分である（更新されるデータのサイズにより異なる）。更新作業は、システム管理者が行うこととなる。なお、データベースに登録されたデータについては、農薬名称や農薬登録番号による検索や農薬登録情報の閲覧をウェブブラウザ上で行なうことができる（図2-1）。

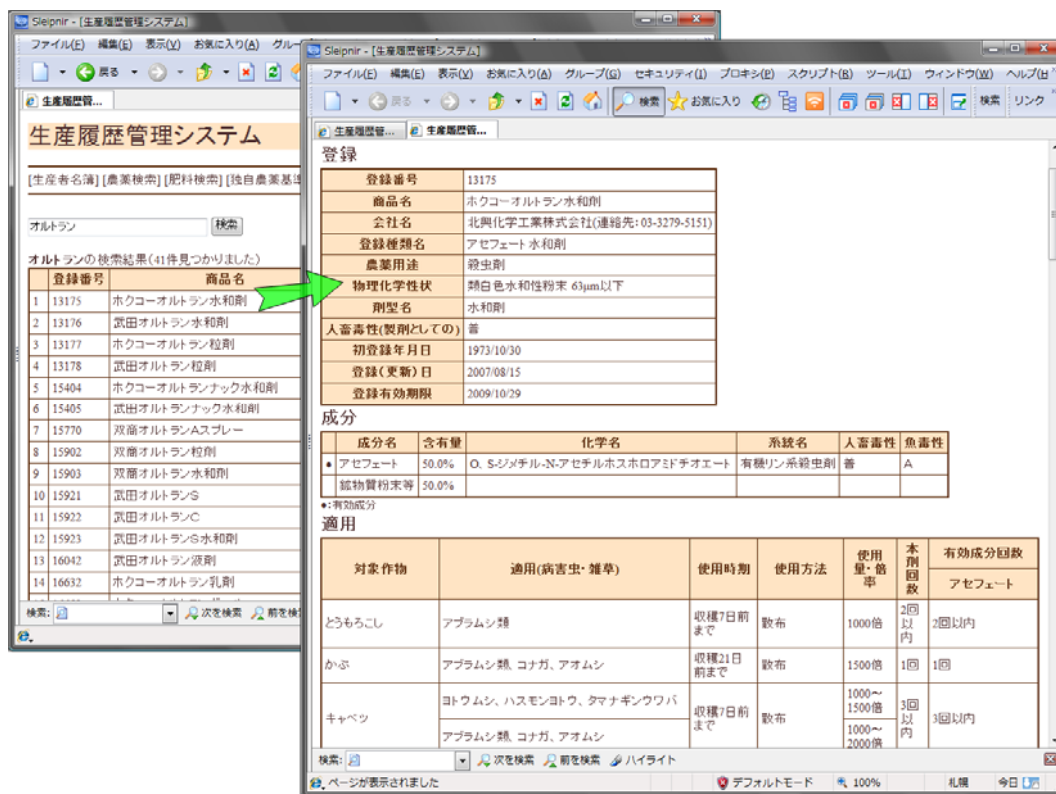


図 2-1 農薬データの検索および登録内容の閲覧画面の例

Fig 2-1 Search and browse of agricultural chemicals data

2.4.2 肥料データベース

肥料に関しては、(独)農林水産消費安全技術センターが肥料データの提供を行っているが、流通しているすべての肥料についてデータが整備されていないため、新たにデータベースを構築した。本システムのために構築したデータベースについては、まず本システムのユーザが主に使用する資材についてデータの整備を行い、その後ユーザがデータベースを保守できる仕組みを備えた。各銘柄の識別子は、アルファベット1文字と3桁の数字からなる(例:A-123)。アルファベットはA~Dを用い、それぞれ、A)ホクレン取扱い銘柄、B)地区銘柄、C)商系銘柄、D)有機肥料、土壌改良剤などその他の銘柄、を表している。各銘柄について、格納されるデータの種類は、資材名、資材略称、適用作物、成分とその含有量である。登録される成分は表2-2の通りである。成分は表2-3に示す各項目についてのデータを持つ。なお、肥料についても農薬同様、データベースに登録された情報をウェブブラウザ上で検索することができる。検索条件には、肥料名のほか、肥料区分、適用作物、各肥料成分の割合を指定することができる(図2-2)。

表 2-2 肥料データの例

Table 2-2 Detailed data of fertilizer

資材コード	A-123
資材名	苦土マンガンほう素入り複合硫加磷安 807
資材略称	807
適用作物	麦類
成分	TN:8, AN:8, TP:30, CP:30, WP:15, TK:17, WK:17, TMg:3, CMg:3, CMn:0.5, TB:0.2, CB:0.2, Cu:0.06

表 2-3 肥料成分一覧

Table 2-3 Constituents of fertilizer

略号	名称	略号	名称	略号	名称	略号	名称
TN	窒素全量	AN	アンモニア性窒素	NN	硝酸性窒素	ON	有機態窒素
TP	りん酸全量	CP	く溶性りん酸	SP	可溶性りん酸	WP	水溶性りん酸
TK	加里全量	CK	く溶性加里	WK	水溶性加里		
TMg	苦土全量	SMg	可溶性苦土	CMg	く溶性苦土	WMg	水溶性苦土
SSi	可溶性けい酸	WSi	水溶性けい酸				
SMn	可溶性マンガン	CMn	く溶性マンガン	WMn	水溶性マンガン		
TB	ほう素全量	CB	く溶性ほう素	WB	水溶性ほう素		
AL	アルカリ分	Fe	鉄	Cu	銅	Ca	カルシウム
S	硫黄						

生産履歴管理システム

ファイル(E) 編集(E) 表示(V) お気に入り(A) グループ(G) セキュリティ(I) プロキシ(P) スクリプト(R) ツール(I) ウィンドウ(W)

生産履歴管...

・検索条件

・肥料区分 A

・適用作物 水稲

・キーワード (AND検索可)

・TNの範囲(%) 10 (下限)~ 30 (上限)

・TPの範囲(%) 10 (下限)~ 20 (上限)

・TKの範囲(%) 10 (下限)~ 15 (上限)

再検索

42件見つかりました

コード	適用作物	略称	資材名	成分
A-38	水稲	264	苦土入り複合塩加燐安264	TN:12, AN:12, TP:16, CP:16, WP:8, TK:14, WK:14, TMg:4, CMg:4
A-40	水稲	365	苦土尿素入り複合硫加燐安365	TN:13, AN:7, TP:16, SP:16, WP:5, TK:15, WK:15, TMg:5, CMg:5
A-43	水稲	042H(ハイブラウン042)	尿素液状複合肥料042H(ハイブラウン042)	TN:10, TP:14, SP:14, WP:13, TK:12, WK:12
A-44	水稲	084	複合硫加安084	TN:10, AN:10, TP:18, SP:18, WP:12, TK:12, WK:12
A-45	水稲	002	苦土入り複合硫加燐安002	TN:10, AN:10, TP:20, CP:20, WP:12, TK:12, WK:12
A-46	水稲	200H(ハイブラウン200)	尿素液状複合肥料200H(ハイブラウン200)	TN:12, TP:10, SP:10, WP:9, TK:10, WK:10
A-48	水稲	S293	複合硫加燐安S293	TN:12, AN:12, TP:19, SP:19, WP:14, TK:13, WK:13
A-49	水稲	352	マンガンほう素入り複合燐加安352	TN:13, AN:13, TP:15, SP:15, WP:9, TK:12, WK:12, CMn:0.75, TB:0.25, CB:0.25

検索: 次を検索 前を検索 ハイライト

ページが表示されました デフォルトモード 100% 札幌 今日

図 2-2 条件を設定した肥料検索の表示例

Fig 2-2 Search and browse of fertilizers based on search criteria

2.4.3 作物コードデータベース

農薬には、それぞれ適用作物が規定されている。また、そのデータは JPP-NET の農薬データベースにも収録されている。適用作物のデータは、農薬登録時の内容がそのまま記載されているため、非常に複雑になっている。例えば、「はくさい」に関連する登録の例を挙げると、当然「はくさい」そのものの適用もあるが、例えば「あぶらな科野菜類」や「野菜類」など大きな括りの登録も見られる。また、「野菜類(きゅうり, なすを除く)」のような但し書きの付いた登録や「野菜類(施設栽培)」のように作型を規定したものも見られる。農薬の使用が適切であるかどうかを判断するような場面においては、これらすべての登録内容を把握しておく必要がある。しかし、登録内容そのものの表記は非常に複雑であり、それらを管理するコードも 10 桁の数字となっていることから(現在ではコード体系が変更となり 15 桁)、本システムの利用者や生産者がそれらの内容を把握することは現実的ではない。

そこで、本システムでは生産者が生産履歴を記帳する際に使用するコードとして、作物に 1 対 1 対応する体系を作成し、そのコードと農薬登録上の作物名およびコードとの対応はシステム管理者が行う仕組みを採用した。生産者が生産履歴の記帳に使用するコードは 2~4 桁の数字とし、稲・麦・豆など主要作物が 2 桁、野菜類が 3 桁、果樹や牧草類が 4 桁となっている(表 2-3)。

表 2-1 作物コード表 (抜粋)

作物コード	作物名
11	移植水稻
12	直播水稻
21	秋まき小麦
...	...
101	はくさい
102	キャベツ
...	...
1201	いちご
1202	なし
...	...
2001	いね科牧草

上記の作物コードは JPP-NET のデータベースにおけるコードとの対応付けを行う必要がある。本システムは、その対応付けをウェブブラウザ上で行うプログラムが準備されている（図 2-3）。手順は、まず対応付けを行う作物に一番近い登録 JPP-NET 上の登録作物名を選択する。例えば、「はくさい」であれば JPP-NET 上の登録にも「はくさい（コード：201000900）」が存在するので、それを選択する。すると、システムが「はくさい」に関連のある作物登録を列挙してくれる。機械により完全に対応付けを行うことは不可能であるため、関連がある項目が列挙から漏れることはないが、関連のない項目が列挙されてしまうことはある。これは、システム管理者が手動でチェックを外す必要がある。これにより、本システム上の作物コードと JPP-NET データベースの作物コードとの対応付けが完了する。

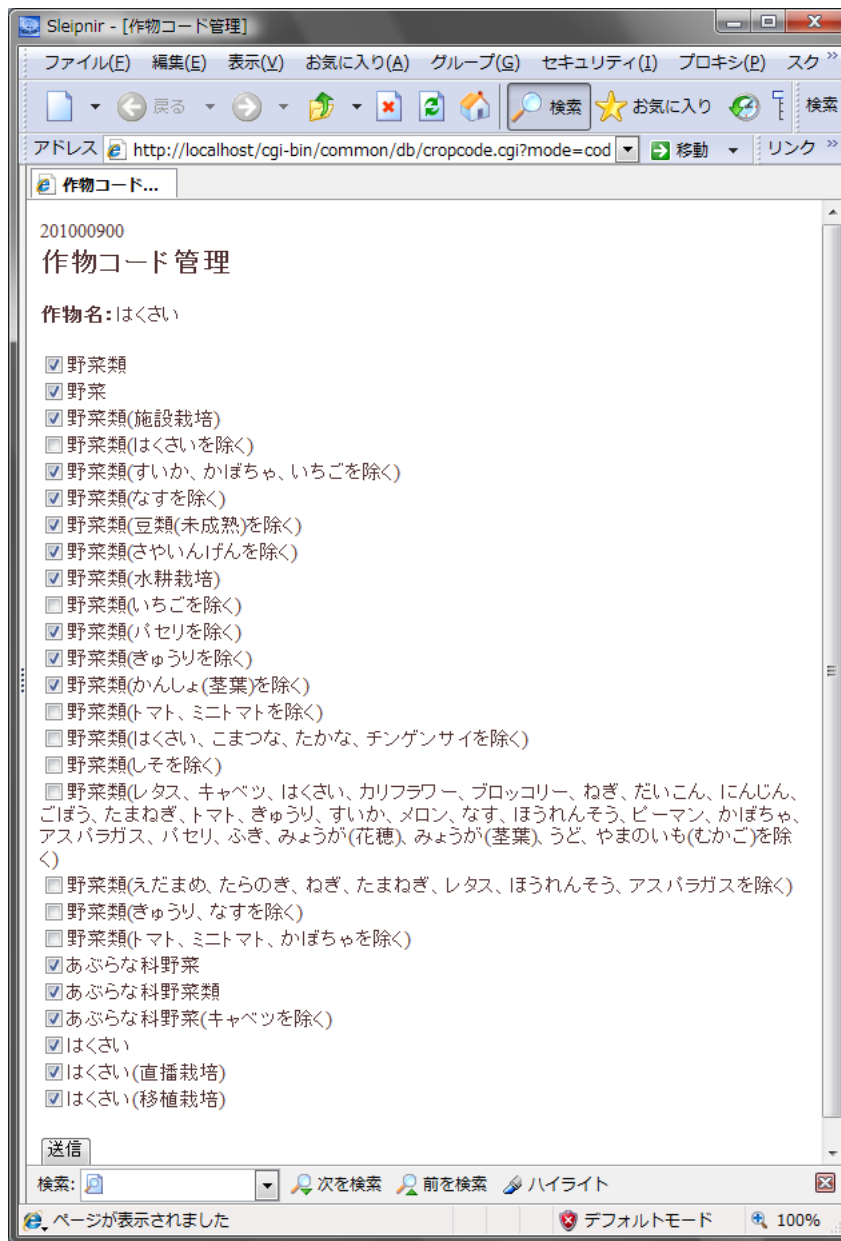


図 2-3 独自の作物コードと JPP-NET のデータベース上の作物コードをウェブブラウザ上で関連付け. 図は「はくさい」の例

Fig 2-3 Association function between the original crop codes and the unique crop codes on JPP-NET database in the browser (Example of Chinese cabbage (*Brassica rapa* var. *pekinensis*))

2.5 機能

以下では、開発したシステムの各機能について述べる。

2.5.1 手書き帳票の電子化

生産者は、定められた手書き帳票（図 2-4）に自信の生産履歴を記帳する。帳票は、氏名、生産者コード等の生産者に関する情報、ほ場番号、面積等の耕種に関する情報、播種や収穫など圃場管理作業に関する情報、肥料の使用履歴に関する情報、農薬の使用履歴に関する情報、からなる。まず、手書き帳票をスキャナにて画像として取り込み（図 2-5）、その後、OCR ソフトウェアを用いて画像から文字情報を抽出する。スキャナによる読み取りは両面同時読み取りのスキャナであれば数秒のうちに完了する（ただし、読み取り時間はスキャナの性能に依存）。また、OCR ソフトウェアでの読み取りも両面合わせて数秒のうちに完了する。なお、OCR ソフトウェアの認識率を高めるために、読み取るべき情報をあらかじめ数値にコード化してある。例えば、品種であれば「1：きらら397、2：ほしのゆめ、3：ななつぼし…」といった具合である。なお、肥料については III 章 2 節で述べた肥料コードを、農薬については農薬取締法で定められた農薬登録番号を農薬コードとして用いている。

OCR ソフトウェアによって読み取られた文字情報は、即座にサーバへと転送される。データがサーバに転送された後は、ウェブブラウザが起動し、後の作業はすべてウェブブラウザ上で行うこととなる（図 2-6）。OCR ソフトウェアでの読み取りはすべて正しいとは限らないため、ウェブブラウザ上で再度読み取りデータを確認することになる。データの誤りを訂正した後、データベースへの登録を行い、データを確定させる。帳票の読み取りからデータの確定までに要する時間は、1～2分程度である。

データベースへの登録が終わった栽培履歴データは、ウェブブラウザ上で閲覧、編集することができる（図 2-7）。この作業は、スキャナおよび OCR ソフトウェアが導入されていないクライアント端末でも行うことができる。

除草剤使用・病害虫防除の履歴（農業使用履歴）

氏名 * 肥料の種類は、播種時の番号を選び記入
1. [指] 2. [kg/10a] 3. [g/10a] 4. [g/10a] 5. [ml/10a] 6. その他

使用年月日	農薬登録番号	農薬名	使用倍率・使用量	散布水量 (l/10a)
記入例 19 6 22	18010	アザラシ	1000	1100

0710 平成19年度 生産履歴票 《水稲用》

氏名 印 出荷コード 電話

* 品種・肥地管理が同じであれば、荷役場を1枚の履歴票に記入しても可（共済地団地場番号を記入）

ほ場番号

* 田舎番号はカンマ「,」で区切り、連続田舎は「5~9」の欄に記入、休耕地は「0」で区切り、田舎、カンマ、田舎「,」で1字に1字 記入例: 2, 3, 5~9, 10 (5) 合計面積 a

水稲品種 1. 倉ら397 2. ぼしのゆめ 3. なつづき 4. ややひめ 5. おぼろづき 6. 福米 9. その他 ()

たね更新 1. 全量 2. 一部更新 5. 自家採種 育苗様式 1. ボット 2. マット 3. 型枠 5. 直播

播種開始月日 月 日 播種量 kg/10a 前作 11. 水稲 41. 大豆 81. 雑草 * 此欄以外の作物はコード表を参照

移植開始月日 月 日 幼穂形成月日 月 日

出穂期 月 日 収穫開始予定月日 月 日

栽培区分 1. 慣行 2. YesClean 3. 特別栽培 9. その他 ()

施用年月日 平成 年 月 日	資材コード * 資材コード表参照	資材名	施用量 (10a当りkg またはl)
記入例 19 5 10	A-60	水稲 888	50 kg
堆肥・肥料・土壌改良資材施用・葉面散布の履歴			

出荷予定月日 月 日 ver.2.1

裏面

表面

図 2-4 専用の手書き帳票

Fig 2-4 Special handwritten checking form

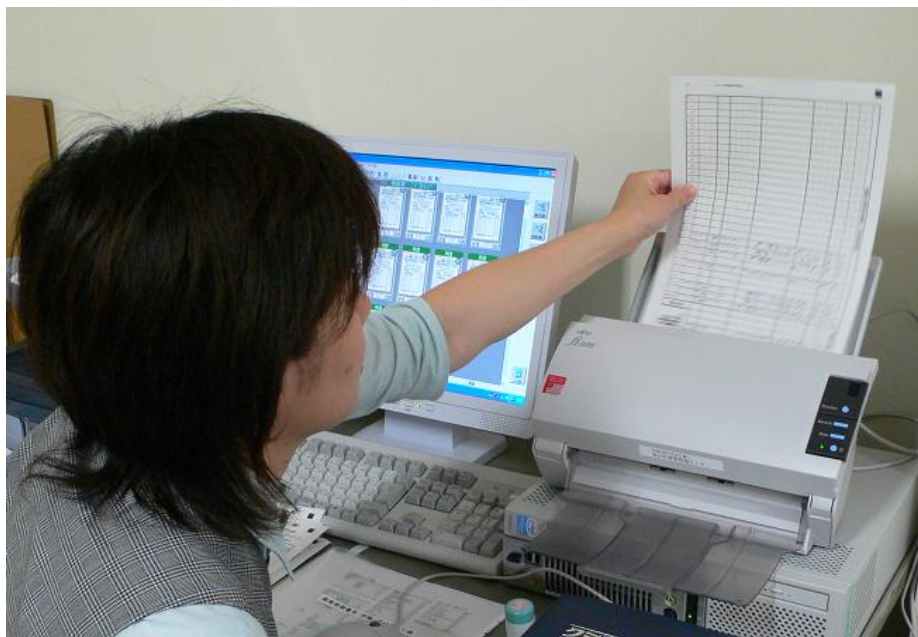


図 2-5 オペレータによる手書き帳票のスキャン

Fig 2-5 Scan operation of handwritten checking forms

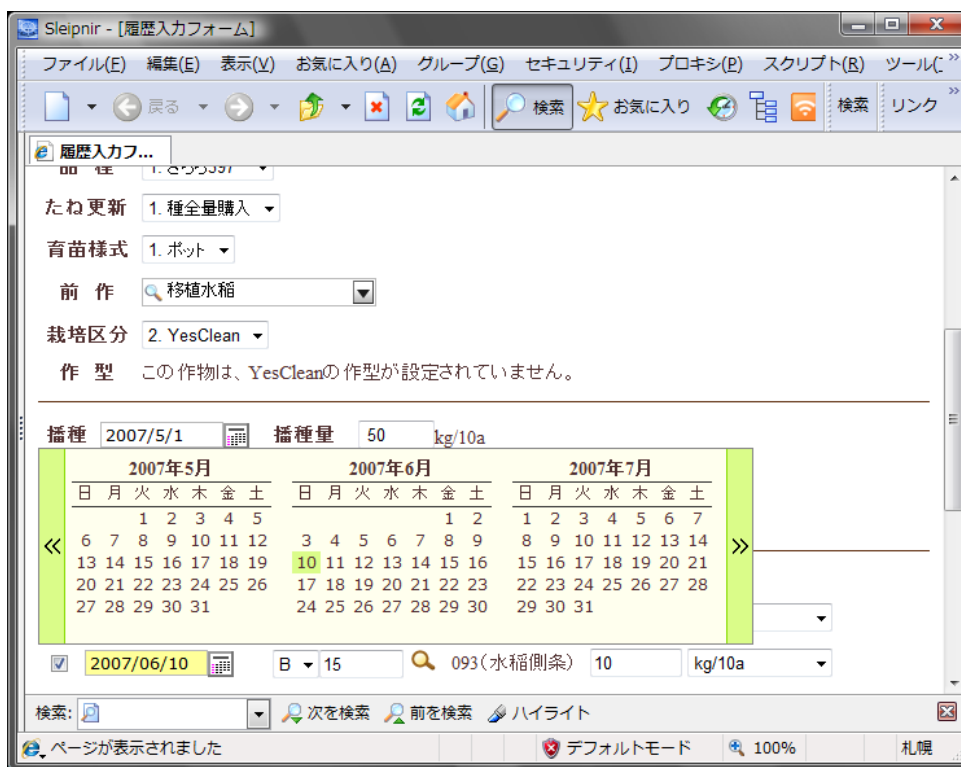


図 2-6 ウェブブラウザ上で読み取られた内容の確認と修正

Fig 2-6 OCR error correction in the browser

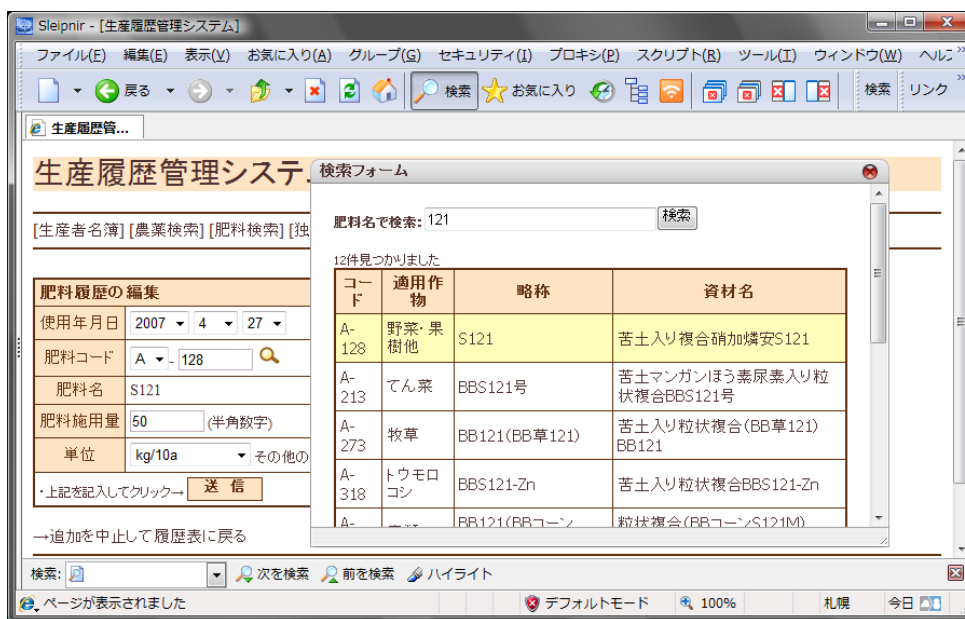


図 2-7 クライアント端末での生産履歴データの管理

Fig 2-7 Management operation of production history data on the client terminal

2.5.2 資材データに関する集計, 分析

データベースに登録された生産履歴データは、資材データベースの情報を用いて、各種の集計、分析を行なうことができる。

肥料履歴については、肥料データベースに基づき、チッソ、リン酸、カリ、マグネシウムの各成分について、総施用量が自動的に計算されて表示される(図 2-8)。

農薬の使用履歴については、農薬データベースに基づき、入力された農薬使用履歴が適正であるかどうかの判断を支援する機能を備えている(図 2-9a, 図 2-9b, 図 2-9c)。検査される項目は、農薬登録の有無、有効期限、製剤ごとの使用回数制限、成分ごとの使用回数制限、農薬使用前日数制限、使用倍率・量、となっている。これらの検査は、本システムによって自動で行なわれるが、農薬の登録基準の中には機械で自動的に判断することが不可能な項目も多く含まれているため、最終的な判断は担当者の確認の元で行なわれるべきである。

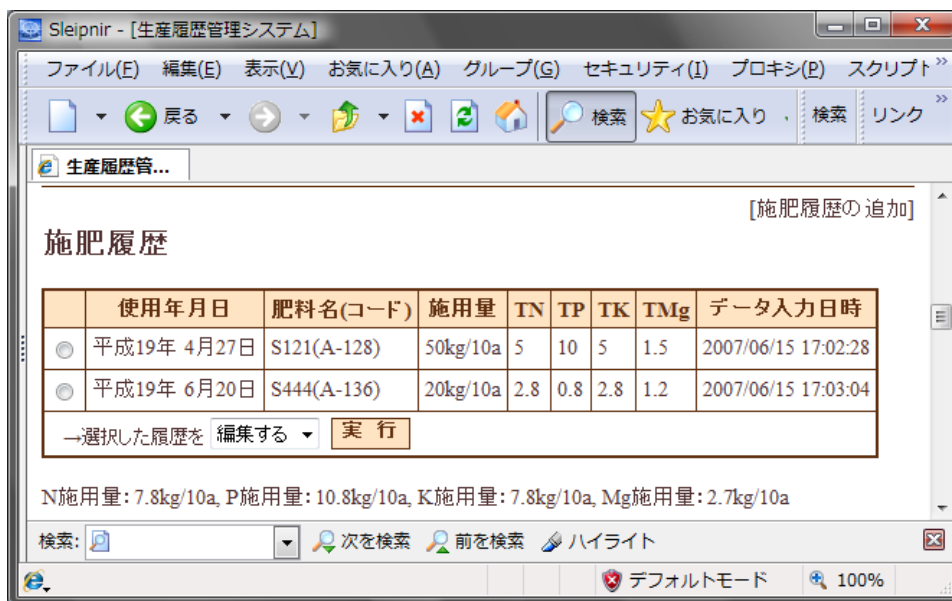


図 2-8 肥料の成分ごとの投入量の自動計算

Fig 2-8 Automated calculation of fertilizer input by constituents

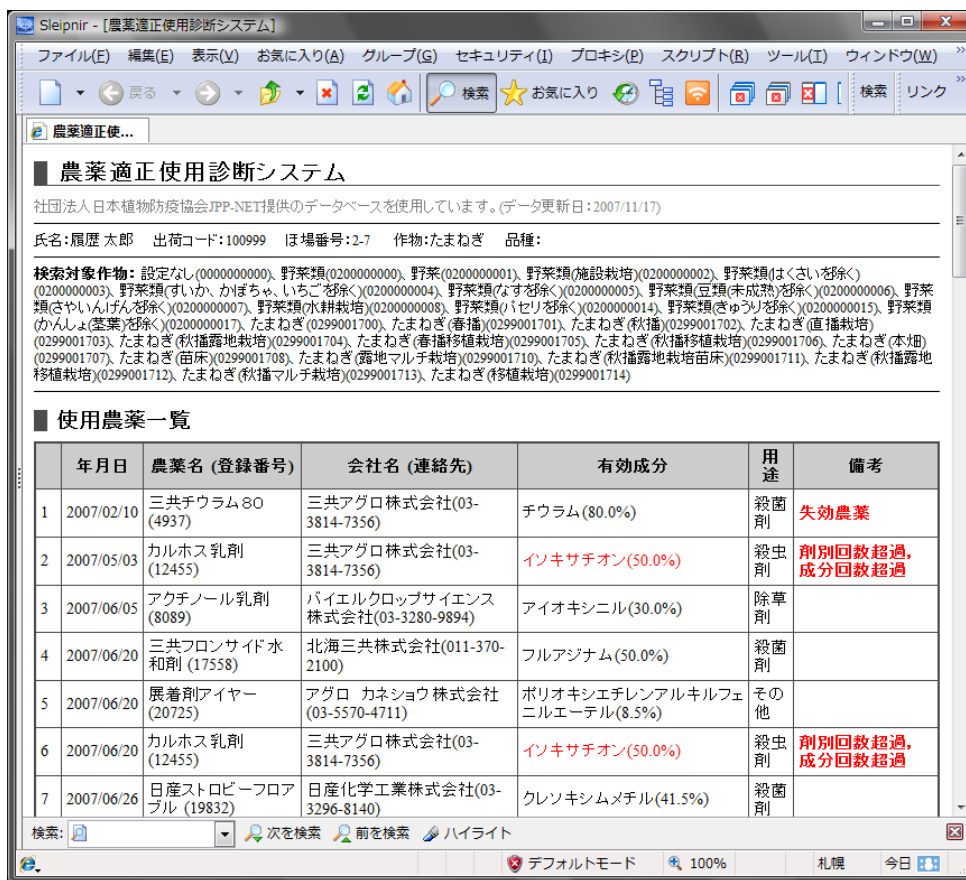


図 2-9a 農薬使用の適正診断の例 (使用農薬一覧)

Fig 2-9a Diagnoses of agricultural chemicals use (List of agricultural chemicals use)

剤別

	農業名	使用回数	回数制限	詳細条件
1	トップアム水和剤	1	6	但し、定植後は5回以内
2	！カルホス乳剤	2	1	
3	トクチオン乳剤	1	3	
4	アグレプト液剤	1	5	
5	アブローチB1	1	-	
6	三共フロンスайд水和剤	2	5	
7	日産ストロビーフロアブル	1	3	
8	オルトラン水和剤	1	5	
9	農着剤アイヤー	1	-	
10	三共デウラム80	1	1	
11	アクチノール乳剤	1	3	

成分別

	成分名	使用回数	回数制限	詳細条件
1	ストレプトマイシン硫酸塩	1	5	
2	デウラム	1	-	
3	チオファネートメチル	1	7	但し、種子への処理は1回以内、苗根部浸漬は1回以内、葉面散布は3回以内、散布は5回以内
4	フルアジナム	2	5	但し、苗根部浸漬は1回以内
5	アイオキシニル	1	3	
6	アセフェート	1	5	
7	ポリオキシエチレンアルキルフェニルエーテル	1	-	
8	ポリオキシエチレン脂肪酸エステル	1	-	
9	ポリオキシエチレンヘキサン脂肪酸エステル	1	-	
10	クレソキシムメチル	1	3	
11	！イソキサチオン	2	1	
12	プロチオホス	1	3	

図 2-9b 農薬使用の適正診断の例（剤別，成分別使用回数制限の判断）

Fig 2-9b Diagnoses of agricultural chemicals use (Detecting the excess use of agricultural chemical)

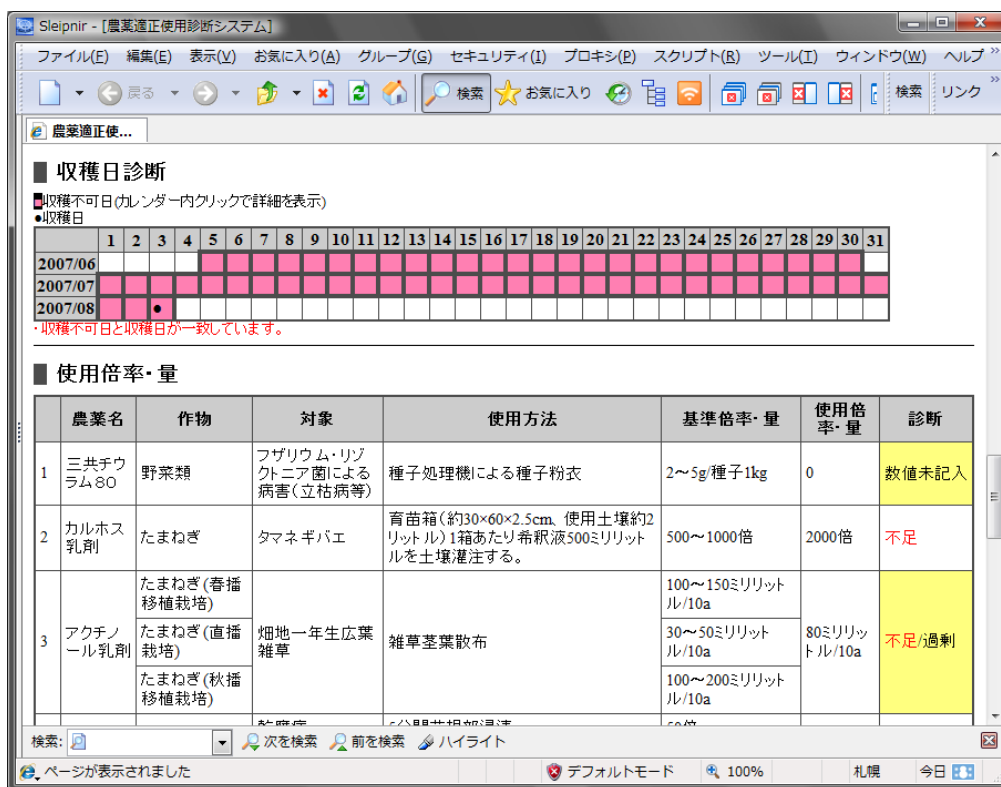


図 2-9c 農薬使用の適正診断の例 (収穫前日数, 使用倍率・量の判断)

Fig 2-9c Diagnoses of agricultural chemicals use (Checking the rightness of harvest date, Checking the concentration or amount of chemicals)

2.5.3 特別栽培への対応

本システムは、北海道内の特別栽培基準 YES!clean に基いて、農薬の使用履歴が適切であったかどうかの判断を支援する機構を備えている。YES!clean では、農薬を適切な計画のもとに使用することが定められているが、その計画は剤の種類ごとに使用する農薬とその成分回数を規定するなど、やや複雑である。本システムは、使用計画と実際の農薬使用履歴との照合を自動的に行なうことができるため、煩雑になりがちであった農薬使用履歴の検査を容易に行なうことができる。

利用者は、あらかじめ作物・作型ごとに農薬の使用計画をデータベースに登録しておく。登録作業は、ウェブブラウザ上で行なうことができる（図 2-10a）。対象病虫害等ごとに、防除の種類、農薬の名称、成分回数を規定すると、自動的にグループ化され番号が与えられる（例えば、殺虫剤の基幹防除はグループ1など）。システムによる検査は、そのグループごとに使用した農薬の種類や回数が適切であるかどうかの判断を行ない、登録のない農薬が使用されていた場合や成分回数に超過が見られた場合などには、警告を行なう（図 2-10b）。

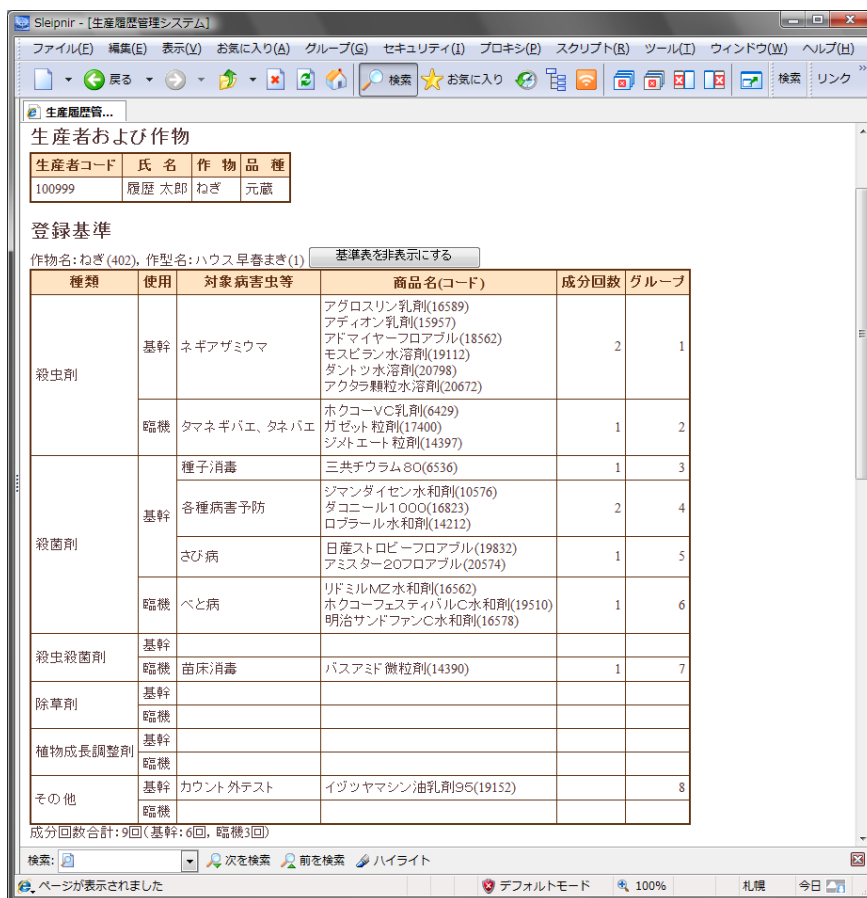


図 2-10a ウェブブラウザ上での YES!clean 栽培の農薬使用基準の登録

Fig 2-10a Registration of agricultural chemical use for YES!clean (special cultivation standard in Hokkaido) in the browser

年月日	薬剤名(コード)	成分回数								コメント	
		1	2	3	4	5	6	7	8		
2007/01/01	三共テウラム80(6536)			1							登録あり
2007/04/29	ベンレート水和剤(20889)										登録なし
2007/06/20	ジマンダイセン水和剤(10576)				1						登録あり
2007/06/20	アグロスリン乳剤(16589)	1									登録あり
2007/06/20	展着剤アイヤー(20725)										展着剤(カウント外)
2007/06/26	明治サンドファンC水和剤(16578)							2			登録あり
2007/06/26	バンガードKS-20(17214)										展着剤(カウント外)
2007/07/05	明治サンドファンC水和剤(16578)							2			登録あり
2007/07/05	ホクコーアディオン乳剤(15959)	1									アディオン乳剤(15957)と同一成分
2007/07/05	バンガードKS-20(17214)										展着剤(カウント外)
2007/07/14	明治サンドファンC水和剤(16578)							2			登録あり
2007/07/14	モスピラン水溶剤(19112)	1									登録あり
2007/07/14	バンガードKS-20(17214)										展着剤(カウント外)
2007/07/21	アグロスリン乳剤(16589)	1									登録あり
2007/07/21	アプローチB(15763)										展着剤(カウント外)
2007/07/30	アグロスリン乳剤(16589)	1									登録あり
2007/07/30	アプローチB(15763)										展着剤(カウント外)
合計		5/2	0/1	1/1	1/2	0/1	6/1	0/1	0/0	2個のグループで回数オーバーあり	
成分回数合計: 13/9 回数オーバー											

注:
 ・農薬取締法上の使用基準を農たしているかは、別途確認のこと。
 ・本システムは、YES!cleanの記録を行うものではありません。
 ・カウント外農薬は、有効成分等で自動判別しています。実際の基準は、農水省特別栽培農産物ガイドライン等で確認のこと。

図 2-10b 登録された基準と実際の農薬使用履歴との照合

Fig 2-10b Verification of YES!clean cultivation

2.5.4 集計機能

本システムは、登録された生産履歴の一覧を取得する機能を有している。全生産者のすべての履歴を集計、統計処理等を行なうことで、営農指導等へのデータ活用が可能となる。集計表は、A表、B表、C表の3種からなる(表2-2)。表は、ウェブブラウザ上で閲覧できるほか、表計算ソフトで閲覧可能な形式でも取得できる(図2-11)。

表 2-2 集計表の種類

Table 2-2 Types of spreadsheets for production history data

種類	項目
A表	生産者コード, 生産者, 住所, 作物コード, 作物, 品種, ほ場番号, 面積, 栽培区分, 播種日, 定植日, 移植日, 収穫日, 最終更新日時
B表	生産者コード, 生産者, 作物コード, 作物, 品種, ほ場番号, 農薬使用日時, 農薬登録番号, 使用倍率・量, 最終更新日時
C表	生産者コード, 生産者, 作物コード, 作物, 品種, ほ場番号, 肥料使用日時, 資材コード, 施用量, 最終更新日時

1	生産者コード	生産者	作物コード	作物	品種	ほ場面積	農業使用日	農業登録番号	使用倍率	最終更新日時
2			103	みずな	京水菜	5	2007/5/15	19842	2000 倍	2007/6/12 8:06
3			103	みずな	京水菜	5	2007/5/25	19842	2000 倍	2007/6/12 8:06
4			110	こまつな	わかみ	5	2007/5/15	19842	2000 倍	2007/6/12 8:10
5			110	こまつな	わかみ	5	2007/5/25	19842	2000 倍	2007/6/12 8:10
6			403	軟白ねぎ	北洋一本	17	2006/1/1	6536	倍	2007/5/17 13:24
7			403	軟白ねぎ	北洋一本	17	2006/8/15	6429	1500 倍	2007/5/17 13:24
8			403	軟白ねぎ	北洋一本	17	2006/9/10	16589	2000 倍	2007/5/17 13:24
9			403	軟白ねぎ	北洋一本	17	2006/9/20	16589	2000 倍	2007/5/17 13:24
10			403	軟白ねぎ	北洋一本	17	2006/10/15	14212	1000 倍	2007/5/17 13:24
11			901	ほうれんそう	ブライトン	10	2007/1/1	6536	0	2007/5/23 16:41
12			901	ほうれんそう	ブライトン	10	2007/1/1	15699	0	2007/5/23 16:41
13			901	ほうれんそう	ブライトン	10	2007/5/1	19842	2000 倍	2007/5/23 16:41
14			901	ほうれんそう	ブライトン	10	2007/5/4	9521	1000 倍	2007/5/23 16:41
15			901	ほうれんそう	ルーカス	20	2007/1/1	6536	倍	2007/5/17 13:39
16			901	ほうれんそう	ルーカス	20	2007/1/1	15699	倍	2007/5/17 13:39
17			901	ほうれんそう	ルーカス	20	2006/11/10	9521	1500 倍	2007/5/17 13:39
18			901	ほうれんそう	ルーカス	5				2007/5/17 13:14
19			901	ほうれんそう	寒アジ、まほ	6	2006/1/1	6536	倍	2007/5/17 13:26
20			901	ほうれんそう	寒アジ、まほ	6	2006/1/1	15699	倍	2007/5/17 13:26
21			901	ほうれんそう	寒アジ、まほ	6	2006/1/1	10529	倍	2007/5/17 13:26
22			901	ほうれんそう	ニューアンナ	1				2007/5/21 9:19
23			102	キャベツ	YR青春	25	2007/1/1	15699	0	2007/6/13 11:58
24			102	キャベツ	YR青春	25	2007/4/30	16575	1 g	2007/6/13 11:58
25			102	キャベツ	YR青春	25	2007/6/10	20863	1000 倍	2007/6/13 11:58
26			407	アスパラガス	ウエルカム	27				2007/5/17 16:38
27			407	アスパラガス	ウエルカム、9、10、17					2007/5/17 15:23
28			51	馬鈴薯	シヤガキップ	80	2006/4/2	5655	100 倍	2007/5/17 14:51

図 2-11 集計表 (B 表) の表計算ソフト (Excel) 上での閲覧

Fig 2-11 View of the spreadsheet on Microsoft Excel

2.6 PC を用いた記帳システム

システムの開発当初は、JA のすべての組合員に問題なくシステムを利用してもらうことが目標であったことから、生産履歴の記帳方法を専用の手書き帳票に統一してきた。しかし、実証試験を続けるうちに特に若い生産者を中心に PC 等での記帳への要望が強く寄せられるようになった。そこで、これまで JA の職員のみを対象としてのシステムであったものを、個々の生産者もインターネット上でシステムを利用できるよう改良を行った。

2.6.1 手書き記帳方式との違い

手書き帳票をスキャナ、OCR を用いて電子化する従来の方法に加え、インターネットに接続した PC 上で生産者が直接生産履歴を記帳する方法を導入した。新方式ではウェブブラウザ上のフォームを利用することで、すべての生産履歴を記帳することが可能となった。その際、自動的に電子化、データベース化が行なわれるため、手書き記帳方式で必要であったスキャナ、OCR を用いた作業は不要となる。生産履歴の提出も履歴 ID を報告するだけで完了するため簡便である。また、リアルタイムに情報が蓄積されるため、農薬使用適否の診断を随時行なうことが可能となった。手書き帳票方式とウェブブラウザ上での記帳方式の違いを表 2-3 に示す。

表 2-3 手書きと PC とでの記帳の違い

Table 2-3 Differences of usage between handwritten checking form and web form

	手書き記帳	PCで記帳
記帳方法	専用の手書き帳票に記入	ウェブブラウザ上のフォームに記入
電子化方法	スキャナ, OCRを使用	不要
農業使用適否の診断	履歴提出後(最終防除から出荷までの間)	PC上で随時
提出方法	帳票をJAIに提出	履歴IDをJAIに報告
受入れ方法	帳票を受取り電子化	履歴IDの報告を受けると自動的にシステムに反映
履歴票の保管	帳票を保管	履歴印刷モードを使用して帳票印刷し保管

2.6.2 生産者向けインターフェース

生産者は、管理者より発行されたパスワードを入力する事により自分専用のサイトを閲覧することができる（図 2-12）。また、「耕種概要」「裁判管理履歴」「肥料履歴」「農薬履歴」の各項目の記帳，編集も同サイトで行なう。「閲覧モード」と「編集モード」の2つが用意されており，これらは同一画面で自由に切換えられる（図 2-13）。また「編集モード」で変更が加わるとサーバとの通信が行なわれデータベースに即座に反映される。生産資材の入力は資材コードを用いて行なう。コードを入力すると，サーバ上のデータベースを検索し，即座に資材名が帳票上に表示される。コードが分からない場合は，資材名の一部をキーワードに検索し入力することができる（図 2-12 のポップアップ参照）。

また，農薬使用履歴については，使用履歴の適否を即座に診断することができる。作業前に記帳し診断を行えば，農薬の誤使用を未然に防ぐことが可能である。



図 2-12 ウェブブラウザ上での記帳画面

Fig 2-12 Editing form in the browser

肥料履歴 >>編集モードに切り替える

年月日	肥料名(コード)	施用量	TN	TP	TK	TMg
2007/09/30	大豆粕(D-8)	80 kg/10a	4.8	0.8	0.8	
2008/04/21	燐安17-45(A-37)	70 kg/10a	11.9	31.5		
	硫安(粒)(A-2)	40 kg/10a	8.4			
合計			25.1	32.3	0.8	

閲覧モード

肥料履歴 保存 キャンセル

	年月日	肥料コード	肥料名	施用量
<input checked="" type="checkbox"/>	2007/09/30	D 8	大豆粕	80 kg/10a
<input checked="" type="checkbox"/>	2008/04/21	A 37	燐安17-45	70 kg/10a
<input checked="" type="checkbox"/>	2008/04/21	A 2	硫安(粒)	40 kg/10a
<input checked="" type="checkbox"/>	2008/4/21			

+ 記入欄を追加する

記入方法:

- 履歴を削除するには、左側のチェックをはずします。
- 資材コード「X」を選択すると、資材名を自由に記述できます。
- 数字は半角文字で記入してください。それ以外の文字は反映されません。
- 虫眼鏡マークをクリックすると資材検索ができます。
- 年月日は「2007/2/10」のように入力します。また、カレンダーから選択することもできます。
- 年月日の表記が正しくない時には、「！」マークが表示されます。
- 年月日と肥料コードが正しく入力されていない履歴は登録されません。

編集モード

図 2-13 閲覧モード（上図）と編集モード（下図）

Fig 2-13 View mode (above) and Edit mode (below) in the browser

2.7 システムの評価

2.7.1 実証試験

システムの有用性を検証するために、北海道内の JA で実証試験を行った。実証試験は、2005年に1つの JA で開始し、その後2012年までに7つの JA に拡大した。システムは、JA における生産履歴管理業務に用いられ、すべての組合員の生産履歴情報を対象とした。

システム導入当初は、すべてのユーザが問題なく利用できることに配慮し、紙ベースの帳票に記帳された情報を OCR にて電子化し、JA の職員のみが電子化された情報にアクセスできる形態を取った。その後、PC を利用した記帳への要望が高まったことから、2008年からは組合員自らがウェブブラウザ上で記帳できるシステムを実装した。

システム導入以前に JA で行われていた紙ベースの生産履歴管理業務は、システム導入によってすべて電子化管理できるようになったことで、より緻密な管理業務を実現できた。特に、農薬の使用適否をシステムが即座に判定できることは、農薬誤使用に伴う食品事故を未然に防ぐなどの効果を上げた。

実証試験開始当初より、システムは JA の既存の業務を行うのに十分な機能を備えていたが、ユーザビリティに対するユーザの要望やウェブブラウザのバージョンアップなど動作環境の変化に対応するため、順次改良が行われた。

2.7.2 ウェブブラウザに関する問題

実証試験開始時の2005年時点では、Internet Explore が世界のウェブブラウザのシェアの8割以上を占めていた。当時の IE の最新バージョンは6であり、それ以前のバージョンを利用するユーザも多数存在した。同時期に Mozilla Firefox が徐々にシェアを獲得してきたことから、当時のウェブアプリケーションは IE のバージョン6およびその以前と Firefox で動作するものが多く、本システムもこの2つのウェブブラウザをサポートしていた。

2006年には IE のバージョン7と Firefox のバージョン2がリリース、2007年

には Opera や Safari も僅かなからシェアを獲得していたことから、ウェブアプリケーションは多くのウェブブラウザへの対応を迫られることとなった。各ブラウザは、ブラウザ間の互換性が確保されていないどころか、ブラウザのバージョンが異なると動作が大きく違うことも多々あったため、アプリケーションのクロスブラウザ対応は困難を極めた。この混乱は、2012年以降各ブラウザが次世代規格を意識したいわゆるモダンブラウザをリリースする頃まで継続することとなる。

2.7.3 開発言語に関する問題

インターネットが一般に広く普及し始めたのは 21 世紀に入ってからであるが、それ以前のウェブサイト（ウェブページ）はアクセスカウンタなど一部の動的コンテンツを除けばページの大半が静的なものがほとんどであった。2000 年以降、インターネットが一般利用されるにつれて動的に変化するページが増加する。本システム開発当初の 2005 年頃は、ユーザのリクエストに応じてサーバ側でコンテンツを動的に生成する手法として perl 言語による CGI が広く用いられており、本システムもこの方式を採用した。

同時期に Google が発表した Google Map は、それまでの CGI 方式のウェブアプリケーションと全く異なるユーザビリティを有した画期的なものであったため、その後のウェブアプリケーションに多大な影響を与えることとなる。Google Map のように、非同期通信でデータを取得しクライアント側でページを動的に変化させる手法は Ajax と呼ばれ、それまでのサーバ側でのページ生成手法と大きく異なる。

本システムにおいても、Ajax がもたらすユーザビリティの改善効果に着目し、クライアント側でページを動的に変化させるソースコードなどを既存のプログラムに多く組み込んでいったため、ユーザビリティは徐々に改善していったが、ソースコードが複雑に入り組むいわゆるスパゲッティ化が起こったため（図 2-14）、プログラム本体の維持管理が徐々に困難になった。また、Ajax の普及に伴いかつての主流であった Perl 言語等を用いた CGI 方式のアプリケーションは減少の一途をたどり、代わりにサーバ側では PHP を、クライアント側では HTML+JavaScript

を用いる例が急増した。これに伴い、JavaScript を用いたライブラリの開発が盛んに行われる一方、Perl 用のライブラリの開発が終了する事例も増えたため、近い将来システムの維持に支障をきたすことが明らかとなった。

```
1: sub dupHistory{
2:   my $dbh = DBI->connect("dbi:SQLite:$c::USERDB", "", "", {AutoCommit => 0});
3:   my ($data) = $dbh->selectrow_array("SELECT data FROM history WHERE
code='$CODE' AND id='$ID'");
4:   my $obj = decode_json($data);
5:   my ($crop, $variety, $field) = ($obj->{'crop'}, $obj->{'variety'}, $obj->{'field'});
6:   my $new_field = '(C)' . $field;
7:   $obj->{'field'} = $new_field;
8:   my $new_id = time . $$;
9:   $obj->{'id'} = $new_id;
10:  $obj->{'update'} = &c::current_date;
11:  my $new_data = JSON::XS->new->utf8(0)->encode($obj);
12:  $dbh->do("INSERT INTO history VALUES ('$new_id', '$YEAR', '$CODE', '$crop',
'$variety', '$new_field', '$new_data', '0')");
13:  $dbh->commit;
14:  $dbh->disconnect;
15:  my $saibai = &saibai;
16:  $saibai =~ s//¥/g;
17:  &ajaxOut("¥$(saibai).innerHTML = '$saibai';new Effect.ScrollTo('pagetop', {duration:
0.2});");
18: }
```

図 2-14 システム拡張に伴いソースコードが複雑化

Fig 2-14 Repeated system expansion cause the complicated source codes

SQL 言語で書かれたサーバ側で行われるデータベースの操作 (2, 3, 12 行目) のためのソース、HTML 言語で書かれたクライアント側で行われるページの操作 (17 行目) のためのソースが、Perl 言語で書かれた CGI の中に混在している。

This CGI source code written by Perl language contains mixed program language i.e. SQL for database manipulation on server side (line 2, 3 and 12), HTML for web page generation on client side (line 17)

2.7.4 クライアント端末に関する問題

インターネットの利用は、PCで行うことが一般的であったが、ここ数年でスマートフォンやタブレット端末が普及したことで、状況が大きく異なってきた。スマートフォンの登場以前でも、フィーチャーフォン（いわゆるガラケー）でインターネットに接続することはできたし、ウェブサイトには接続することはできた。しかし、ディスプレイのサイズや解像度がPCに比べ大きく劣り、ユーザインタフェースも貧弱であったことから、広く使われるには至らなかった。しかし、スマートフォンにおいてはPCと同等のサイトを閲覧することが可能であり、また最近ではスマートフォンの画面サイズや操作方法に適したサイトを構築することが一般化したことにより、スマートフォンでのインターネット利用が急増した。

また、スマートフォンは4Gなどの高速インターネットに接続できるものが多く、外出先でもインターネットを利用することが容易であるため、屋外での活動が多い農業者との親和性も高い。そのため、本システムにおいてもスマートフォンでの利用を希望するユーザが次第に増えていった。もちろん、PCのウェブブラウザを念頭に開発されたウェブアプリケーションであればスマートフォンでも一通りの利用は可能であるが、高いユーザビリティを確保するにはスマートフォン専用のアプリケーションの開発は不可欠である。

スマートフォンの普及が進み始めた頃から、スマートフォン用のアプリケーションを作成するための優れたJavaScriptライブラリが多く発表されており、それらを利用することで以前とは比べ物にならないほど容易にアプリケーションを構築できるようになった。しかし、本システムのようなCGI方式のアプリケーションと新たな手法で書かれたライブラリとでは、システムの構築手法があまりにも異なるため、本システムの改良にライブラリを用いるのは困難であった。そのため、本システムにおけるスマートフォン対応は断念せざるを得なかった。

2.8 考察

「生産履歴，生産資材マネジメントシステム」によって，JAにおける生産工程管理業務を高度化することができることが示された．本システムは，JA職員および組合員すべてがユーザとなる必要があるため，IT機器に馴染まない組合員であっても問題なく利用できる必要があった．紙ベースでの利用を基本とする本システムの設計は，ITリテラシーの程度に関わらず誰でも利用可能であるという点で，JAにおける運用に際して非常に有効に機能した．一方，IT機器に慣れたユーザに対しては，PCを用いたシステム利用の手段が有効に機能した．紙ベース，PCベース双方を並行稼働させる事が，すべてのシステム利用者の利便性向上に寄与した．

すべてのユーザの情報が電子化されたことで，JAにおける生産工程管理業務を高度化することができた．情報が電子化することで，農薬の使用適否の自動判断に基づく農薬誤使用等に起因する食品事故の未然防止，肥料の投入量の自動計算に基づく的確な営農指導等への活用が可能になり，システムが営農改善に寄与することが示された．

一方，プログラム言語やライブラリといったソフトウェア，PCやスマートフォンといったハードウェアなど，ICTをめぐる状況は急速に変化をしており，システムもそれらへの対応を常に求められた．システムを安定的に運用するためには，かつては主流であったPC上での動作のみを念頭に置いたCGI方式のウェブアプリケーションでは不十分であることが明らかになった．本システムは，生産工程管理システムとしての機能は備わっており，農業現場における要求は満たしていたが，ICTの環境変化への対応が困難な設計であったため，システムが早晚陳腐化することは避けられないと考えた．そこで，保守性，拡張性を考慮した新たなシステム設計を行う必要があるとの結論に達した．

第3章

保守性, 拡張性向上のためのシステムの再構築

3.1 背景および目的

パーソナルコンピュータやインターネットといった情報技術は 21 世紀に入った頃から広く一般に利用され始めたが, その発展速度は目覚ましく, ソフトウェア, ハードウェア共に次々に新技術が登場しては程なくして陳腐化するといった状況が続いている. しかも情報技術の発展速度が指数関数的であることから, 陳腐化する期間も指数関数的に短くなっている. 長期間にわたって生産履歴データを収集・保存しなければならない生産履歴システムにとってこれは極めて大きな問題である.

また, 総務省(2014)によると, 平成 25 年末時点でのインターネットの利用端末の割合は, 自宅のパソコンが 58.4%, スマートフォンが 42.4%, タブレット型端末が 12.4% (総務省, 2014) と, スマートフォンなどの新たな情報端末が台頭してきていることが見て取れる. 今後も, ICT を取り巻く環境は常に変化していくことが想定され, 生産履歴システムもそういった変化に柔軟に対応していくことが求められる.

本章の目的は, 前章で構築したシステムの機能を情報技術の発展に耐えうるウェブアプリケーションとして再構築することである. そのために, ウェブを取り巻く社会的側面, 技術的側面の考察を行うとともに, システム運用における問題点を抽出し, 保守性, 拡張性を考慮した新たなウェブシステムを実現する. また, 情報技術の発展に対してより堅牢なシステムとなる要件を明らかにする.

3.2 情報システムに関する技術的考察

3.2.1 計算機の変遷

計算機の起源は、1645年にブレーズ・パスカルが発明した機械式計算機「パスカリーヌ」に見出すことができる。パスカリーヌは、連結された複数のダイヤルの回転により加算のみを行うものであったが、後にライプニッツにより加減乗除が可能な機械式計算機が発明される。19世紀に入り機械式計算機が商用化されると、20世紀にいたるまで各国で様々な製品が作られたが、それらはいずれも歯車やドラムを物理的に回転させることで計算を行う機械仕掛けの計算機であった。1936年にアラン・チューリングによって提唱されたチューリングマシンは、仮定の機械ではあるものの初めて自動計算機概念を示したものとされ、現在の計算機科学の原点とされる。1940年代に入ると、ABC, Colossus, ENIACなどリレーや真空管を用いた電子計算機が発明され、弾道計算や暗号解読などの軍事目的に活用されることとなる。1945年にジョン・フォン・ノイマンによって提唱されたプログラム実行型の計算機はノイマン型コンピュータと呼ばれ、現在のコンピュータの基本動作原理とされる。1949年には世界初のノイマン型コンピュータEDSACが完成、その後もIBMなどにより数多くのコンピュータが開発される。それらの計算機は、商用計算や科学技術計算といった用途に限られた専用機であったのに対し、1964年にIBMが発表した「System/360」は特定用途を持たない汎用機として開発された。汎用機の登場により、計算機は企業の基幹システムなどに広く使われるようになった。その後、集積回路(IC)の発明により計算機の小型化が実現、1977年にアップル社がApple IIすると計算機は技術者などの専門家でない人々でも利用できる機器へと変わっていった。1980年代に入りIBMが「IBM PC/AT」を発表、その後数多くのPC/AT互換機が登場したことにより計算機は急速に一般に普及することとなった。1995年にマイクロソフト社がWindows 95を発表すると、その発売が世界中で大きく注目されるなど一種の社会現象を引き起こし、一般家庭へのパソコンの普及拡大の一つのきっかけとなった。2007年、アップル社はiPhoneを発表した。iPhoneを始めとした携帯電話にコンピュータ

を融合させた新しい端末はスマートフォンと呼ばれ, 計算機を一層身近なものとした.

3.2.2 インターネット

インターネットの歴史は, 1969 年にアメリカで運用が開始された ARPANET に始まり, 大学その他の研究機関を結ぶ通信網として発展してきた. 現在では, 誰でもインターネットに接続できることが当たり前になっているが, UUNET が 1989 年に世界初の商用インターネットサービスプロバイダ事業を始める以前は, インターネットは研究者などの一部の人が利用できるのみであった. 1980 年台には, PC 間のデータ通信を行う手段としてパソコン通信が用いられることがあったが, パソコン通信は通信業者と契約者のみが利用する閉じたネットワークであったという点でインターネットとは性格がまったく異なる. 1990 年に入り Web の概念が誕生しインターネットが一般に解放されると, インターネットは不特定多数を対象としたコミュニケーションツールとしての側面が大きくなる. その後, FTTH などの高速インターネット回線の普及, クラウドの概念の提唱, 各種ソーシャルメディアの普及など, Web をめぐる情勢は急速に変化した. 現在では様々なサービスが Web を介して提供されており, Web が今日の ICT の核であると言える.

3.2.3 クラウドコンピューティング

クラウドコンピューティングとは, インターネット上に存在するアプリケーションやストレージ等のリソースを活用してサービスを構築する技術である. 一方, 使用者がサーバやアプリケーションをすべて自身で管理する従来の運用形態は, クラウドコンピューティングと区別するためにオンプレミスと呼ばれる. クラウドコンピューティングの普及は, 2006 年にグーグル社によって提唱されたこと始まりとされるが, それ以前にもネットワークを介してリソースを共有する仕組み自体は古くから使われてきた. 例えば, 1980 年代に広く使われた UNIX ワークス

ーションは、ネットワーク経由でサーバのリソースをクライアントが利用するクライアントサーバモデルである。また、1990年代以降に多く誕生したASP(Application Service Provider)は、インターネット経由でストレージやアプリケーションを提供するサービスである。それらのサービスにおいては、ハードウェアとそれに付随するOS、データベース、アプリケーションなどが一体的に扱われていたのに対して、クラウドコンピューティングにおいてはこれらの構成に関する自由度がより高くなっている。

クラウドサービスは、その形態の違いからIaaS(Infrastructure as a Service)、PaaS(Platform as a Service)、SaaS(Service as a Service)の3つに大別される。また、これらのインターネットを介してサービスを提供する仕組みを総称してXaaSと呼ばれることもある。IaaSは、ハードウェアやOSといったシステム基板を提供するサービスでAmazon EC2などがその代表である。IaaSは、従来のレンタルサーバに近いが仮想化されたマシンを提供することで利用者のニーズに応じてリソースの種類や量を柔軟に変化させることができるという点で異なる。PaaSは、アプリケーション開発のためのプラットフォームを提供するサービスでGoogle App Engineなどがその代表である。PaaSは、定められたプラットフォーム上でアプリケーション開発を行う必要があるが、OSの保守等が不要なため開発のみに専念できる利点がある。SaaSは、サービス(アプリケーション)を提供するサービスでウェブメールなど多くのウェブアプリはこれに当たる。従来のASPも、クラウドサービスとしてはSaaSに分類される。利用者は必要に応じてこれらのサービスを選択することになるが、いずれのサービスもハードウェアやOSの保守を利用者が行う必要がないことから、オンプレミス型のサービスと比較すると開発に係る工数や経費を節減できる利点がある。

3.2.4 Web API と分散システム

現在クラウド上で公開されているアプリケーションには、不特定多数の利用者

がそれぞれの利用目的に応じてカスタマイズ可能な形で提供されているものが多くある。これは、特定の利用者が必要な機能をパッケージ化して提供してきた従来型のアプリケーションと大きく異なる点である。例えば Google マップは、ウェブブラウザ上で動作する地図アプリとして利用できる他に、Google マップの持つ機能を他のアプリケーションの機能を組み込んで利用することも可能である。具体的には、Google マップ上に天気などの他のサービスが提供する情報を重ねて表示するというのがその典型例である。これは、Web 上に API(Application Programming Interface)を公開することで、そのサービスの機能を自由に使うことができるようになっていていて実現できる。複数の Web API を組み合わせて新たなサービスを構築することはマッシュアップと呼ばれ、クラウド時代以降の新しいアプリケーションのあり方である。

クラウドコンピューティングにおいては、前述したようなハードウェアや OS などのリソースをインターネット越しに共有利用することの利点に加え、マッシュアップなどによるサービス自体の共有利用も大きな利点である。複数の計算機を強調動作させることは分散システムと呼ばれ、クラウドコンピューティングにおける大きな側面の一つである。分散システム自体は、クラウド時代のはるか昔から存在した。例えば、UNIX で別のコンピュータ上の手続きを実行する技術である RPC(Remote Procedure Call)は 1980 年代から使用されてきたし、1990 年代には異なるプログラム言語間でもオブジェクトを相互利用できる仕組みとして、CORBA(Common Object Request Broker Architecture)に代表されるような分散オブジェクトが様々開発された。しかし、それらの技術は仕様が複雑であることに加え、互換性を持たない同様のシステムが乱立したこともあり、汎用的な分散システムとなるには至らなかった。2000 年代に入ると、Web 上で分散システムを実現する動きが広がる。マイクロソフト社などは、1999 年に Web 上での連携動作のための通信プロトコルである SOAP を発表、2000 年には W3C が勧告を行った。SOAP は、その後仕様が非常に複雑化したため徐々に敬遠され、現在では後に Fielding(2000)によって提唱された REST(REpresentational State Transfer)が広く使われるようになっていく。

3.2.5 HTML

HTML(HyperText Markup Language)は, ウェブページを記述するためのマークアップ言語である. ハイパーテキスト(hypertext)という名前が示す通り, テキスト(文書)をハイパーリンクによって結びつけることでテキスト間の参照を可能にすることが初期のウェブでの大きな特徴であったが, 現在ではウェブ上で音声や動画など様々なメディアを扱うことが一般化していることから, HTML はハイパーメディアのための言語であるといった方がより適切だと言える.

HTML は, 業界団体が仕様の検討を行い W3C の勧告により仕様を確定する方式でこれまでいくつかの仕様が誕生してきた. IETF(Internet Engineering Task Force)のワーキンググループが 1993 年に HTML 仕様書を発表するが, 当時はまだ W3C が HTML を管理してなかったため正式な勧告はなかったが, 事実上これが最初の HTML とされる. その後, 1995 年に HTML2.0, 1997 年に HTML3.0, 1998 年に HTML4.0 改訂版, 1999 年に HTML4.01 が, いずれも W3C から勧告され, HTML4.01 が事実上の決定版として用いられることとなった. 2005 年にティム・オライリーによって Web2.0 の概念が提唱された頃から, ウェブの多様な利用方法への要求が高まり, HTML も新たなニーズに対応した仕様策定が望まれるようになる. W3C は 2008 年 1 月に HTML5 の最初のドラフトを発表するが, ウェブをめぐる情勢の急速な変化や各関係者団体との調整が難航したことにより, 正式な勧告は 2014 年 10 月にまでずれ込んだ. ただし, 2008 年以降に発表された多くのウェブブラウザが W3C の勧告を待たずに HTML5 への対応を順次進めた結果, 2010 年頃には HTML5 を用いたウェブサイトが多く現れ始めた. 2016 年以内に HTML5.1 の勧告が予定されていることから, 今後は HTML5 が標準的地位を占めるものと思われる.

3.2.6 CSS

CSS(Cascading Style Sheets)とは, ウェブページのスタイル(色, サイズ, 位置, 装飾など)を記述するための言語である. 1996 年以降, CSS も様々な勧告が

行われたが、仕様書の定義が曖昧であったりウェブブラウザ間での実装が異なっていたりしたこと、CSSで記述できることの多くがHTMLでも記述可能であったことから、ウェブデザインをCSSのみによって完全にコントロールするという目的は達成しているとは言いがたい。現在では、CSS3をHTML5と共に用いられることが標準になっているが、ウェブブラウザ間の実装の差異が現在でも存在することから、後述のJavaScriptのライブラリがCSSの補完的機能を果たすことでウェブデザインをコントロールする事例が増えている。

3.2.7 JavaScript

JavaScriptは、プロトタイプベースのオブジェクト指向プログラミング言語でNetscape Communicationsによって開発され、1995年にNetscape Navigator 2.0に実装された。JavaScriptは、ウェブページを動的に変化させるといったHTMLのみでは行うことができない動作を比較的容易に実装できることなどから、1996年にマイクロソフト社がInternet Explorer 3.0に実装した頃から急速に利用されるようになった。しかし、当初のJavaScriptは各ウェブブラウザで独自仕様が組み込まれるなどしてブラウザ間の互換性が著しく損なわれていたこと、また、クライアントサイドで悪意のあるコードを実行させることができるというセキュリティ上の問題点もあったことなどにより、JavaScriptの使用は次第に敬遠され、JavaScriptをウェブブラウザに実行させない設定を推奨するといった動きも一部で見られた。その後、Ecma Internationalによって仕様の策定が行われたことによりウェブブラウザ間の互換性が向上し、次第に利用が加速する。さらに、2005年以降にAjaxと呼ばれるJavaScriptを用いた非同期通信を用いた高機能なウェブアプリケーションが登場したことで、JavaScriptは再び大きく注目を集めることとなった。

現在では、リッチなウェブアプリケーションの構築にはJavaScriptは欠くことができない技術になっているが、それに加えて重要な役割を果たしているのがJavaScriptを用いたライブラリである。代表的なJavaScriptライブラリには、jQuery, Ext JS, Google Web Toolkit, Prototype JavaScript Framework, YUI

などがある。ライブラリは、JavaScript の仕様を擬似的に拡張する、非同期通信や DOM の操作をシンプルなコードで記述できるようにする、高機能な GUI を簡易に作成するなど、それらの機能はライブラリによって異なる。Ext JS や YUI の様に、一つのライブラリでアプリケーション作成に必要な機能を網羅したものもあれば、単機能のみを提供するものもあるが、それらのほとんどがライセンスをフリーに設定しているのも大きな特徴である。また、現在でもウェブブラウザ間の互換性の問題は完全には解決していないため、JavaScript や CSS の挙動の違いを吸収する役割を担う JavaScript ライブラリも数多く存在する。

2015年8月現在、W3Thechs の調査によるとクライアントサイドプログラム言語の利用シェアは、1位が JavaScript で 98.8%、2位が Flash で 10.3%、3位が Silverlight で 0.1%である。また、同調査によると JavaScript ライブラリの市場シェアは、1位が jQuery で 95.4%、2位が Modernizr で 12.9%、3位が Bootstrap で 12.5%である。つまり、現在は JavaScript および jQuery がウェブアプリケーションを構築するプログラム言語におけるデファクトスタンダードであると言って差し支えない。

3.2.8 ウェブブラウザ

ウェブブラウザは、ウェブ (World Wide Web) を閲覧するためのソフトウェアである。1991年に公開された NeXT 上で動作する WorldWideWeb が世界初のウェブブラウザとされ、当時のブラウザはハイパーテキストのみを閲覧するシンプルなものであった。1993年に画像を扱うことができるブラウザである NCSA Mosaic が登場したのをきっかけにウェブブラウザの普及が進み、その後 Netscape Navigator や Internet Explore など多くのウェブブラウザが誕生することとなった。現在では、画像に加え音声や動画といった様々なハイパーメディアをウェブブラウザ上で扱うことが一般的である。

1990年代後半には、ウェブブラウザ上で動作する様々なコンテンツが作られたが、ウェブブラウザ間の互換性が著しく低かったことや、圧倒的シェアを有した Internet Explore の Web 標準への対応が進まなかった事などから、ウェブブラウ

ブラウザ上で動作するアプリケーションの機能は限定的であった。そのため、異なるプラットフォームでも同一の動作が行える **Flash** や **Java** を用いて作られたアプリケーションをウェブブラウザ上のプラグインとして動作させる手法も多く用いられた。現在では、各ブラウザの **Web** 標準への対応が進んだことにより、**WWW** の機能を用いたアプリケーションが一般的となり、**Flash** や **Java** のブラウザ上での利用は著しく減少した。なお、**iOS** 上で圧倒的なシェアを占める **Safari** は **Flash** をサポートしていない。

現在、世界のウェブブラウザのシェア 1 位は **Google Chrome**、続いて **Internet Explore**, **Firefox**, **Safari** となっている。これらのウェブブラウザは **HTML5** などの新しい仕様に対応しており、ウェブアプリケーション開発に当たってはかつて行われてきたクロスブラウザ対応のためのコーディングはほとんど不要となっている。逆に、近年はスマートフォンやタブレット端末が普及したことで、ディスプレイのサイズや操作方法の違いに考慮したアプリケーション開発が求められるようになってきており、それらの開発を支援する **JavaScript** ライブラリ等も盛んに開発されている。

3.3 システムの運用・保守に係る課題

3.3.1 持続可能性 (Sustainability)

20世紀後半に科学技術は飛躍的に進歩したが、同時に地球規模での大きな環境変化をもたらした。そういった中で、1987年に国連世界環境開発委員会によって「持続可能な開発(Sustainable Development)」の概念が提唱された。「持続可能な発展」は、「将来の世代が自らのニーズを充足する能力を損なうことなく、現在の世代のニーズを満たすような発展(開発)」(知恵蔵 2015)と定義され、主に環境に配慮した開発の基調となる概念であったが、この考えは政治経済、文化、教育など様々な分野に影響を与え、現在ではそれらの概念をまとめて「持続可能性(Sustainability)」と呼ばれる。

情報システムは、現在でも急速な発展を続けており、持続可能性を考慮することは非常に重要である。例えば我が国においては、2001年に施行された「資源の有効な利用の促進に関する法律(資源有効利用促進法)」(経済産業省 2001)に基づきPCをはじめとする多くのハードウェアがリサイクルの対象となる。また欧州連合においては、ハードウェアのリサイクルに関する規制として Waste Electrical and Electronic Equipment(WEEE)指令があり、2003年に施行されている(European Commission 2003)。

Piotrowicz(2009)は、IT(information technology), IS(information system)における持続可能性への貢献の評価を試みている。Piotrowiczは、IT/ISに対する評価がそれまでは経済面にのみ着目したものであったのに対して、新たに経済(economic)、社会(social)、環境(environmental)の3つに着目した評価手法を用いて既存のサプライチェーンシステムの評価を行っている。Piotrowiczの評価は概念的ではあるものの、持続可能性がIT/ISの評価軸になる可能性を示している。この評価手法を本研究で構築する生産履歴を扱うISに対して適用すると、表3-1のようにまとめることができる。このように、農業の生産活動を支えるISにおいても経済面のみならず社会面、環境面での評価を与えることができることから、システムを適切に運用することで社会の求める持続可能性への要求に応えること事

が十分可能であると言える。

3.3.2 サステイナブルな情報システム

情報システムが上で述べたような社会的責任を果たす要件を備えていることは重要であるが、安定的なシステム運用を行うためにはそれだけでは不十分である。ICT を巡る情勢は常に大きく変化しており、周辺の状態の変化にシステムが取り残されれば早期に陳腐化してしまう可能性がある。そのため、システムが環境の変化に柔軟に対応できる仕組みを備えておくことが重要である。Mărușter(2008)は、既存の SOA(Service Oriented-based Architecture)で作られた情報システムが安定的に運用できる条件の検討を行い、システムの持続可能性(Sustainability)の評価を試みている。ここで言う持続可能性は、上で述べたような社会の中で存在し続けられるかといった意味とは異なりシステムそのものが持続的に運用可能であるかを指している。Mărușterによると、情報システムの安定的運用にはシステムを取り巻く環境の変化を検知することと、それに対処すべくプラットフォームを柔軟に変化させることが重要だと指摘している。また、Schelp(2009)も同様に SOA に対する評価を行い、システムを少ないコストで改変、拡張を行うことができる敏捷性(agility)が安定運用には必要であると述べている。具体的には、ドキュメントの整備、ミドルウェアによる保守性の向上、標準化されたインターフェースの利用などが敏捷性の向上に必要な要素としている。また、過去の資産（ソースコードなど）の再利用は敏捷性の向上に寄与するよう思えるが、想定されるシナリオが複数あり実際に寄与するかは事例ごとに異なり、過去の資産を再利用する SOA は Schelp が調査した 650 の SOA においては 34% とそれほど多くないことが分かっている。

2.7 節で述べたとおり、これまでに筆者らが開発したシステムは現場で問題なく稼働できたものの、周辺の変化への対応が困難になったことでシステムの陳腐化が避けられなくなった。これは、Schelp の述べる敏捷性がシステムに不足していたことが大きな原因であったと考える。また、過去の資産の再利用によるシステムの改良は図 2-14 に示したようなソースコードの複雑化を招き、システムの敏捷性低下の一因になったと言える。そこで、過去の資産の再利用を行わずにゼロか

らシステムを再構築する必要があると結論付けた。

表 3-1 経済, 社会, 環境の3つの軸による生産履歴を扱う IS の評価

Table 3-1 Evaluations of IS for agricultural production history based on relation to economic, social, environmental dimensions

経済	<ul style="list-style-type: none">・適切な栽培管理に基づく資材費の削減・栽培管理の高度化による作物の品質向上・農薬誤使用の未然防止による潜在的な経済損失の除去
社会	<ul style="list-style-type: none">・食の安全・安心を目指した生産活動・実需者に向けた生産履歴情報の提供
環境	<ul style="list-style-type: none">・適切な施肥管理に基づく土壌等の環境汚染の低減・適切な農業機械の運用による燃料消費量の低減

3.4 生産工程管理システム” apras” の開発

3.4.1 方法

旧システムの運用結果から現場における問題点を抽出した。その改善方法をソフトウェアアーキテクチャの面から検討することで、新システムの構成を決定した。また、新システムは **apras** (**a**gricultural **p**roduction **a**ssist **s**ystem) と命名した。具体的には、旧システムをクラウドサービスとクライアントアプリに分離することで保守性を高め、同時にスマートフォン等多様なデバイスに柔軟かつ迅速に対応できる構成とした。また今後も進展が続く ICT に対応して **apras** を低コストにアップデートして行くため、できるだけ少ない工数で **apras** を開発することを目指した。

次に、旧システムで不足している機能及び今後必要と考えられる機能について検討し、それを **apras** の拡張機能として実装した。具体的には、クラウドサービス化、モバイル端末への対応などの機能を付加した。また、将来における **apras** の拡張を考え、センサデータの格納のための機能を実装した。センサデータへの対応は、第4章で詳述する。

3.4.2 旧システムの問題点とその解決方法

旧システムでは、PC が使えないユーザ向けに手書きで記入した帳票をスキャンし、OCR で読み取って自動的に入力する機能を用意した。しかし、手書きであっても帳票への記入は煩雑な作業であり、多くのユーザについて、積極的に詳細なデータを入力しようとする姿勢は見られない。この問題を解決するにはユーザインタフェース (UI) の改良だけでは不十分であり、データ入力モチベーションを向上させる仕組みが必要であると考えられた。

モチベーションを向上させるには、例えば生産履歴データを入力することで生産性及び収益が将来的に向上する仕組みが想定される。そのためには、マニュアル入力による生産履歴データだけでは不十分であり、圃場における作物の生育、気象環境等のデータと組み合わせて分析できるようにする必要がある。そこで、

センサデータ等との連携が将来的に行うことができる拡張性を確保しておくことが重要である。

`apras` の具体的実装方法を表 3-2 のようにした。ここ数年で主要な Web ブラウザが HTML5 に対応したことにより、HTML5 のメリットを活かした高機能な Web システムが増加している。旧システムは、HTML4.01 に基づいているため急速に陳腐化することが懸念された。そこで、`apras` では HTML5 を用いることとし、HTML5 が持つ機能を出来るだけ活用することとした。旧システムは、Web サーバ上でユーザプログラムを動作させる CGI(Common Gateway Interface)を用いていた。CGI は、かつては Web プログラミングの主流であったが、現在はより柔軟なシステム設計が可能な Web API に取って代わられつつある。また、`apras` の機能を API で提供することで、他の API サービスとのマッシュアップが可能になる。そこで、`apras` は、Fielding(2000)によって提唱された REST (REpresentational State Transfer) 型の Web API 群として設計した。また、データ交換フォーマットには、現在多くの Web サービスが採用している JSON(JavaScript Object Notation)を用いた。JSON の採用により、他のサービスとの連携の簡便性、Machine to Machine(M2M)の連携動作における高速性が期待できる。

表 3-2 新旧システム比較

Table 3-2 Comparative table of old and new systems

	apras	旧システム
OS	Windows Server 2008	Windows Server 2003
Web サーバ	Apache HTTP 2.2	Apache HTTP 2.2
データベース	PostgreSQL9.2	SQLite(ActivePerl5.8 標準)
サーバプログラム	PHP5.3(Rest API 方式)	ActivePerl5.8(CGI1.1 方式)
HTML	HTML5	HTML4.01
CSS	CSS3	CSS2
使用する HTTP メソッド	GET, POST, PUT, DELETE	GET, POST
データ交換	JSON	独自形式
CORS	対応	非対応
外部のプログラムとの連携	API を利用することで, 自由に連携することが可能	CGI であるため, サーバプログラムの改変を伴う

3.4.3 apras のインプリメンテーション

図 3-1 に apras のシステム構造を示す。apras は、インターネット上で生産工程の管理を行うための Web サービスで、JA 等の生産者団体での利用を想定している。プログラムは、REST 形式の API 群を提供するサーバプログラムとそれらの API を利用したユーザインタフェースを提供するクライアントプログラムからなる。サーバは、OS は Windows Server 2008 Standard 64bit, Web サーバは Apache HTTP Server 2.2 である。サーバプログラムは、PHP5.3 により実装、データベースは PostgreSQL 9.2 64bit を用いている。データリソースは、階層化された URI によって表現され、リソースの操作は HTTP の 4 つのメソッド(GET, POST, PUT, DELETE)によって行う。サーバプログラムは、セキュリティ上必要なユーザ制御が行われた状態で各リソースへの CRUD(Create, Read, Update, Delete)を安全に行うことができる API を提供する。API におけるデータ交換フォーマットには JSON を用いている。JSON は、元々は JavaScript のために定義されたオブジェクト表記法だが、現在では多くの言語で JSON を扱うことができるため、JavaScript 以外の言語で記述されたプログラムとのデータ交換も容易である。クライアントプログラムは、上記の API の仕様にに基づき誰でも任意に作成することができるが、apras としては Web ブラウザ上で動作するユーザインタフェースを標準クライアントプログラムとして提供している。

apras への情報入力は、生産者が Web ブラウザ上で動作するクライアントプログラムを操作するか、専用の帳票に手書き記帳することで行う。手書き記帳された帳票は、TIFF 画像化したデータをサーバにアップロードすると、サーバ上に設置された OCR プログラムにより電子化される。入力されたデータは、生産者、生産者団体の担当者の双方とも Web ブラウザ上で閲覧、編集することができるため、両者の情報共有をリアルタイムに行うことが可能である。

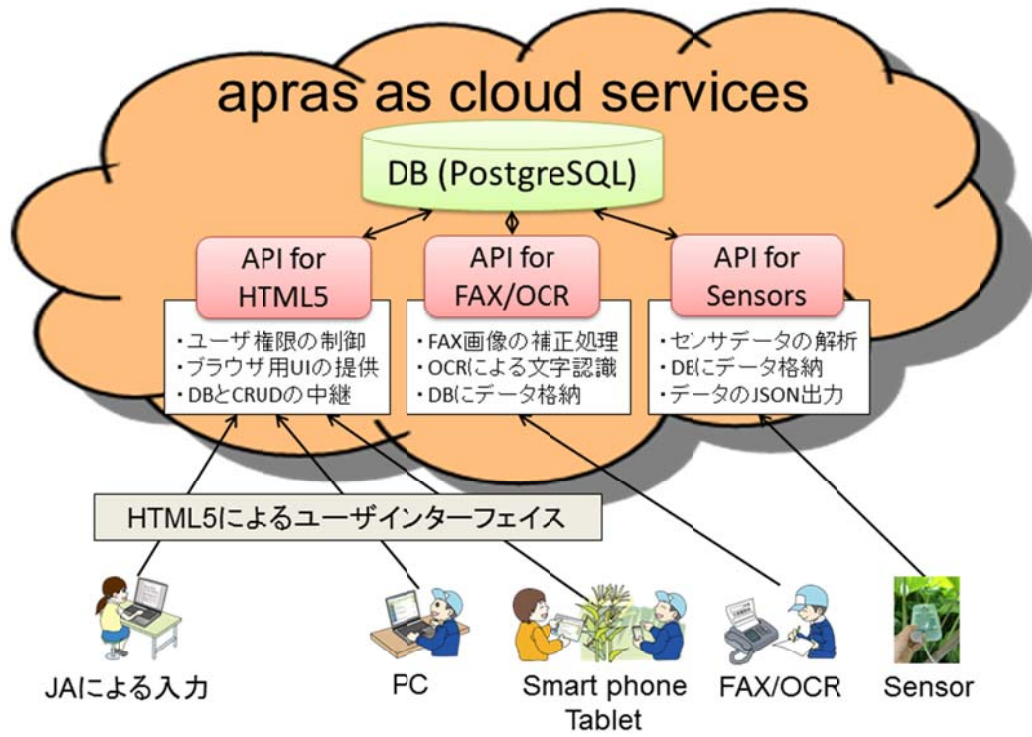


図 3-1 apras の構成

Fig 3-1 Structure of apras

3.4.4 機能の API 化

サーバプログラムは, サーバに格納されたリソースに対する各種の操作を行う API を提供する(章末の付録参照). `apras` が扱うリソースは, 肥料, 農薬, 作物, 生産者など多岐に渡る. リソースは, サーバ上のデータベースのテーブルにそれぞれ格納される. リソースの実体はデータベース上のテーブルであるが, 外部に対しては階層化された URI でリソースを表現する. 各リソースに対する CRUD(Create, Read, Update, Delete)操作は, 各 URI に対して HTTP の4つのメソッド(GET, POST, PUT, DELETE)を発行することで行う.

リソースを表す URI は不特定多数に公開されているため, 適切なアカウント制御を行う必要がある. アカウント制御を行うことで, アクセスしてきたユーザの所有するリソースのみを扱うといったユーザごとのリソースの切り替えや, 不適切なアクセスを防ぐといったセキュリティ制御が可能になる. `apras` は, UID (ユニーク ID) を URI に含めることでアカウント制御を行うことができる. UID は, 「ユーザ ID」および「パスワード」を認証用 API に送信することで得られる.

3.4.5 PC 用クライアントアプリケーション

前述の通り, `apras` は Web ブラウザ上で動作するクライアントアプリケーション(以下, `apras-app`)と連携して動作する. `apras-app` は, インターネットに接続可能なブラウザのみで使用でき, その他のアプリケーションをインストールする必要はない. `apras-app` は JavaScript で実装し, ユーザインタラクションが伴う処理はすべて `ajax` を用いることで高いユーザビリティを実現している. JavaScript のライブラリには, `jQuery`, `jQuery Mobile` および, その関連ライブラリを採用した. クライアントプログラムは, 現在はサーバプログラムと同じサーバに設置されているが, サーバサイドが CORS (=Cross-Origin Resource Sharing) に基づき設計されているため, 異なる OS のサーバに実装することもできる.

`apras-app` は Web ブラウザ上で動作し, モダンブラウザであればその種類は問

わない。なお, `apras-app` で扱うのは, 耕種概要, 圃場情報, 肥料記録, 農薬記録, 収穫記録, 作業記録, 生育記録の7項目で, 一度システムにアクセスした後のサーバとのやりとりは `ajax` で行うため, 書き込み, 編集, 削除などのすべての操作が画面遷移を伴う事なく行うことができる (図 3-2)。



図 3-2 ウェブブラウザ上でのクライアントプログラムの表示例

Fig 3-2 View of the client application in the browser

3.4.6 モバイル端末用 apras-app

スマートフォンやタブレット等の高機能かつ使いやすいモバイル用端末の普及によって、マウス操作が必須だった Web アプリケーションを現場で使いやすいものに改善できる。モバイル端末の多くは PC での利用を前提とした Web サイトも問題なく利用できるが、ディスプレイの大きさやマウスやキーボードなどのデバイスの制限がある。そのため、近年はモバイル端末に対応させて高いユーザビリティを有する専用のアプリケーションとクラウドサービスを用意する方式が増えつつある。PC, モバイル端末双方からの利用を想定する場合、両者はサーバ上の同一のデータリソースに対する操作を行うことになるため、その操作は同一の API を介して行うことが望ましい。

そこで、タッチパネル等を備えたモバイル端末の豊富な機能に対応した `apras-app` を開発した (図 3-3, 3-4)。API は PC 上の Web ブラウザ用クライアントプログラムと同様のものを使用したため、サーバプログラムに対する変更は全く必要なかった。また、JavaScript のライブラリには多くのモバイル端末をサポートする `jQuery Mobile` を使用した。



図 3-3 モダンブラウザが動作する各種端末への対応

Fig 3-3 apras on devices which have modern browser



図 3-4 スマートフォンに最適化されたクライアントアプリケーションの表示例

Fig 3-4 Client application Displays optimized for smartphone

3.4.7 Fax-OCR

旧システムにおいて過半数のユーザが手書き帳票を用いていた。また、JAによっては農業者になるべく窓口へ来てもらうため、手書き帳票を対面で受け取ることを推奨する事例も見られた。この場合、未記入部分の追記、誤記入やOCRの誤認識がないかどうかを職員が確認できるというメリットがあった。これらのことから、**apras**にも手書き帳票のOCR読取り機能を持たせると共に、ユーザの利便性をさらに向上させるために**Fax**送信データ(TIFF画像)のOCR読取り機能も追加した。

生産者によって記帳された手書き帳票は、スキャナを用いた読み込み、あるいは**Fax**による送信を受けた**Fax**サーバによりTIFF画像化した後、**apras**サービスにアップロードされる。**apras**サーバは**Fax**送信等に伴う画像のノイズ除去、OCRによる文字認識を行い、適切な形式にデータを変換した後、データベースに格納する。このデータは**apras-app**から記帳されたデータと同様にブラウザ上で扱うことができる(図3-5)。なお、スキャナの読み込みやTIFF画像のアップロードは、生産者団体の担当者が行うことを想定している。

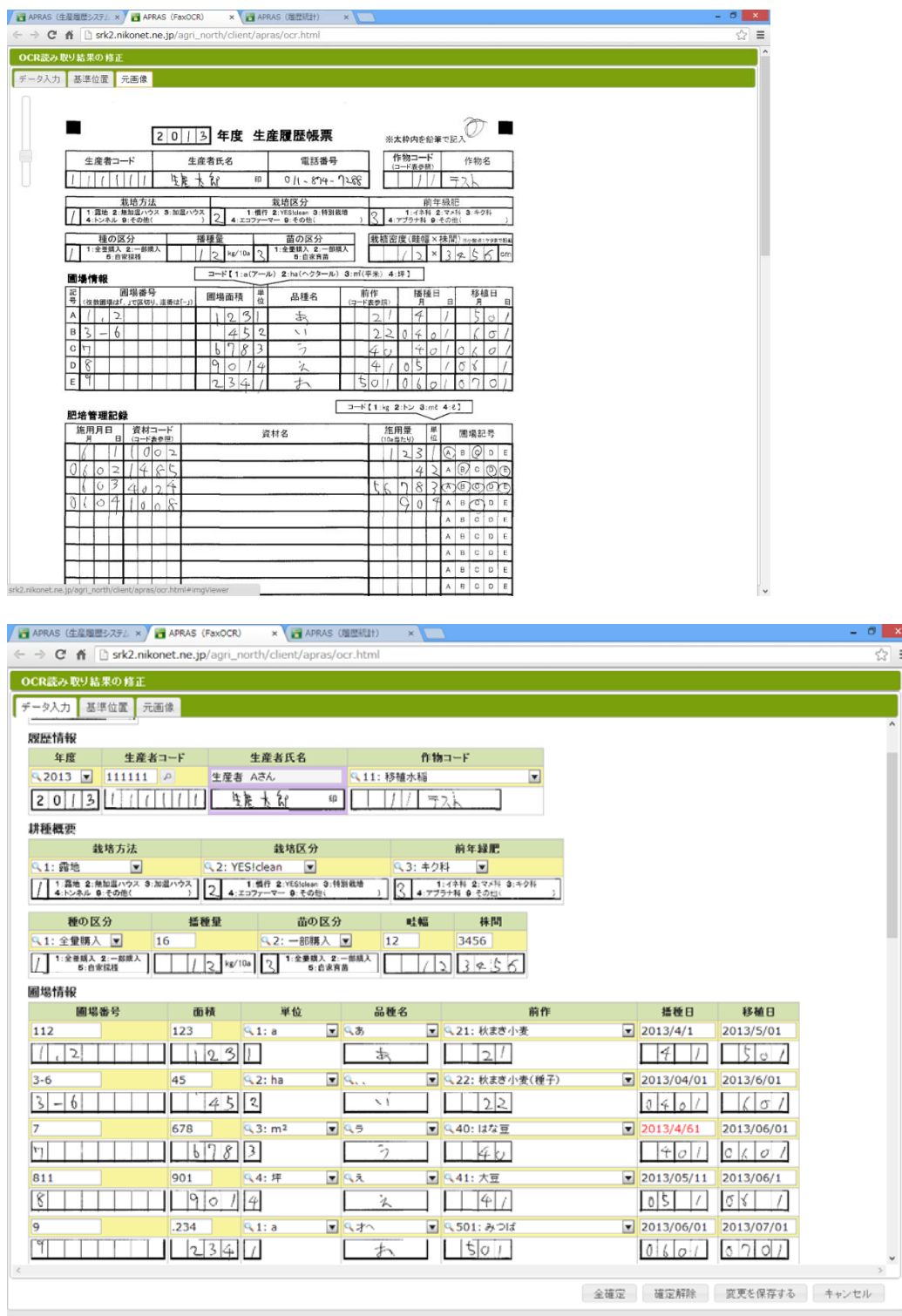


図 3-5 ウェブブラウザ上で Fax 送信画像の閲覧 (上図), 読取り確認と修正 (下図)
 Fig 3-5 View of the Fax-sent image (above) and OCR error correction (below) in the browser

3.5 apras の評価

サーバプログラムを REST API 形式にしたことで, 異なるクライアントアプリケーション(apras-app)を効率的に作成できた. 当初, PC 上の Web ブラウザで apras-app を作成することを想定し, サーバの API (付録) の仕様を策定したが, 後にモバイル端末用の apras-app を作成するに当たっては, この仕様を一切変更することなく実装することができた. これは, サーバプログラムではデータベースへの読み書きといった必要最小限の機能だけを持たせ, データの加工や表示といった複雑な機能はすべて apras-app に委ねるようにしたために可能となった.

また, apras-app の作成に当たっては, 既存の JavaScript ライブラリである jQuery, jQuery Mobile を使用することで, 高機能なプログラムを効率良く作成することができた. W3Techs(<http://w3techs.com/>)の調査によると, 2014 年 5 月 28 日現在, 世界のウェブサイトにおける jQuery の使用率は 59.2%, 市場シェアは 93.8%となっており JavaScript ライブラリにおける事実上標準の地位を確立している. jQuery は Web ブラウザ間の仕様の差異を吸収する形で多くの機能を提供しているため, それらを使用することで高機能な apras-app を作成するための工数を大幅に削減することができた. jQuery Mobile は, jQuery の派生ライブラリであるため, jQuery を利用した PC 用 apras-app のソースの多くをモバイル用 apras-app に僅かな改変で流用することができた. モバイル用 apras-app は, PC 用 apras-app と同等の機能を有するが, PC 用 apras-app の実装に約 3 人月を要したのに対し, そのソースコードの多くを流用したモバイル用 apras-app はわずか約 7 人日で実装できた. apras-app は, 端末間の互換性をライブラリによって維持しており, 現在 Web ブラウザは Internet Explorer, Chrome, Firefox, Safari (いずれも最新バージョン) で, モバイル端末は iOS 6+, Android 4+で動作を確認している. 今後, 普及が期待されるウェアラブルデバイスにも同様の方法で柔軟に対応できることが期待される.

apras は, 旧システム同様北海道の 8 つの JA において 2013 年 4 月より実証試験を行った. 旧システムの運用では, JA の担当者に対する聞き取り調査を随時行い, 改善要望をシステムに反映させてきた. 旧システムにおいても, 生産工程管理情報を適切に管理するための必要最低限の機能は備わっており, システムを稼働さ

せたすべての JA のすべての組合員 (約 6,000 人)を対象に全作物の生産工程管理記録を処理できていた。しかし、スマートフォンやタブレットなどの新たな端末への迅速な対応、洗練されたユーザインタフェースの構築、柔軟なカスタマイズ等への要望への対応がかなり困難であった。これは、旧システムが旧来の CGI 方式を取っていたためシステムを変更するにはサーバプログラムの大規模な改修が必要であり、新たなユーザインタフェースの構築や個々のカスタマイズの対応をすると維持管理にかかるコストを大幅に押し上げるためであった。それに対し、apras では REST API 形式を採用したため、ユーザインタフェースの変更やカスタマイズはクライアントプログラムへの対応だけで済むことに加え、HTML5 およびそれを活用したライブラリの利用により、低コストで洗練されたユーザインタフェースの実現が可能であった。2014 年 3 月、apras を利用する 8 つの JA に対してアンケート調査を行った結果、「apras に満足していますか」との間に、「はい」が 6 件、「いいえ」が 1 件、「無回答」が 1 件であった。また、いくつかの要望事項が各 JA から挙げられたが、その多くは JA ごとのシステムのカスタマイズを求めるものであった。apras は、容易にカスタマイズが可能な設計になってはいるが、実証試験において各 JA の要望に対応することは困難であるため、これらへの対応は今後実運用を通じて行うこととしている。

なお、実運用に当たってはサーバ等の機器の保守管理、セキュリティ面に万全を期す必要があることから、専門知識を有する機関での運用が必要である。また、それに伴う費用負担は、数百名程度の生産者団体での稼働の場合、初期導入費用 200 万円、保守管理費用 60 万円/年程度を見込んでいる。

3.6 apras の利用実態に関する考察

apras の利用者は、栽培期間を通じて自身の記録を apras に残すこととなる。システムを利用して記帳することは一定程度の負担が利用者に生じていると思われるため、記帳行動を分析することでその程度を把握する必要があると考えた。そこで、apras のログを解析することでシステム利用の実態を調査し、記帳の頻度、各作物における記帳の特性、個人間の差異等の基礎データを提示する。

3.6.1 利用実態の分析

調査は、apras の実証件を行った JA のうち、道央圏水田転換畑地帯の 3 つの JA (A 農協, B 農協, C 農協とする)、道東圏畑作地帯の 3 つの JA (D 農協, E 農協, F 農協とする) の 6JA (図 3-6) の全組合員を対象とし、apras のシステムログの分析を行う。

生産履歴のデータには、耕種概要、圃場管理履歴、施肥履歴、農薬使用履歴などのデータが含まれているが、今回はこの中で最も頻繁に記帳の機会がある農薬使用履歴を中心に解析を行う。なお、システム上では生産者は個別の ID で識別されるが、1 つの ID を 1 つの家族経営や 1 つの法人で共有している事例もあるため、1 生産者が必ずしも 1 名の作業員ではない場合もある。また、生産履歴は同一の管理がなされている栽培に対して 1 つの帳票が作成されるため、帳票 1 枚に複数の圃場の情報が含まれる場合がある。反対に 1 つの圃場内で複数の異なる管理がなされた場合は、複数の帳票が作成される。

3.6.2 生産履歴帳票数

A,B,C 農協に提出された帳票数、D,E,F 農協に提出された帳票数をそれぞれ図 3-7 に示す。6JA における生産者 1 人あたりの帳票提出枚数は平均 2.37 枚となっている。A,B,C 農協で野菜類の帳票数が多い傾向にあるが、施設園芸における長期取りの作物はその栽培期間内で管理が分割されることで複数の帳票を作成する必要があること、また、産直向けに小ロットで複数の野菜を生産していることな

どが原因として考える。一方、畑作地帯の D,E,F 農協においては麦, 大豆, いも類の帳票数がやや多い傾向にある。これは, 大規模圃場で品種や管理を分けていることが要因となっている。

生産者 1 人あたりの平均提出履歴帳票数と栽培作物数を図 3-8 に示す。1 枚だけの生産者から 15 枚以上の生産者まで幅広く分布しているが, 当然ながら栽培作物数が多い生産者の帳票数が多くなる傾向がみられ, 作物数が増えると履歴帳票の記帳, 管理の負担が増加するといえる。

3.6.3 農薬使用履歴の分析

栽培期間を通じて恒常的に発生する記帳作業として, 農薬使用履歴の記帳がある。図 3-9 (上図) に生産者 1 人あたりの平均農薬使用日数を示す。農薬を使用したその日に記帳を行うと考えると, この日数がそのまま記帳業務を行う日数と言える。1 日から 90 日以上まで幅広く分布しており, その負担は生産者によって大きく異なることがわかる。また, 図 3-9 (下図) に示すとおり生産者 1 人あたりが使用する農薬の種類数には大きなばらつきがある。また, 図 3-10 は, 作物別の農薬使用回数を示しているが, 図から分かるように農薬使用がとうもろこしで数回程度, たまねぎで地域によっては 30 回前後になるなど, 作物によって農薬使用回数に大きな開きがある。30 を超える数の農薬を使用している生産者も少なからず存在するが, 多くの種類の農薬を数多く使用しなければならない作物を扱う生産者は, 農薬の使用方法を正確に把握し, 記帳しなければならないことから, その負担は非常に大きくなる。そういった生産者にとっては, apras の農薬使用適否を支援する仕組みは非常に有効だと言える。

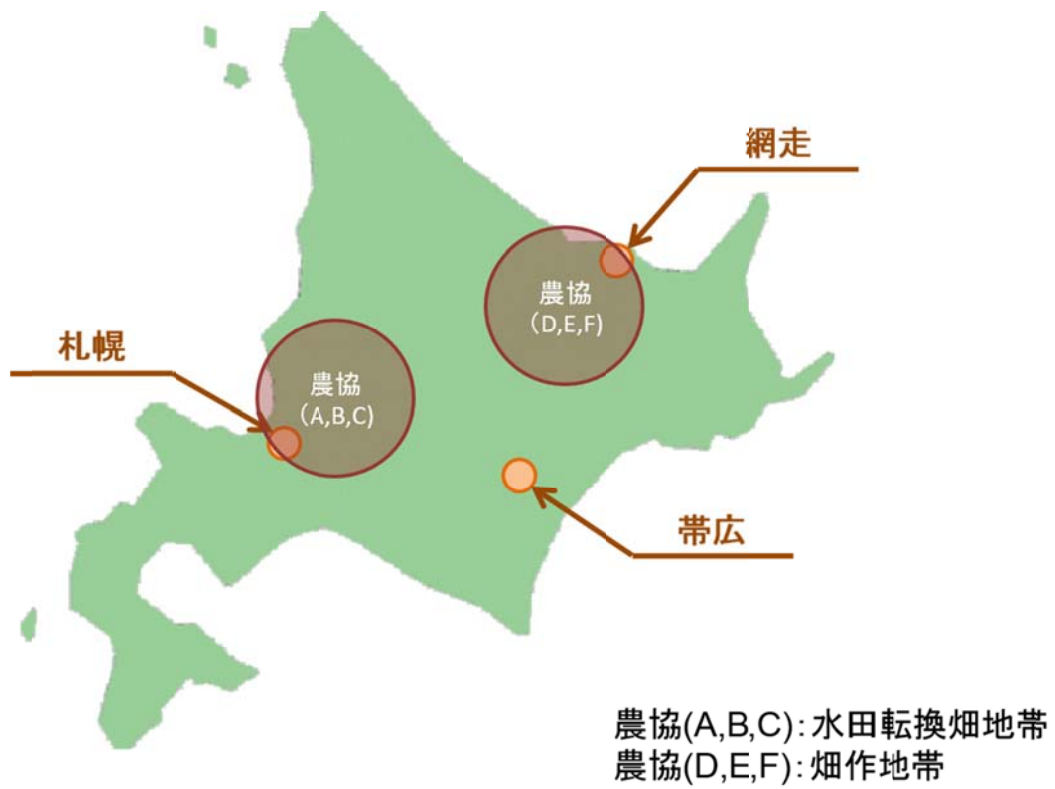


図 3-6 調査 JA

Fig 3-6 Surveyed JAs

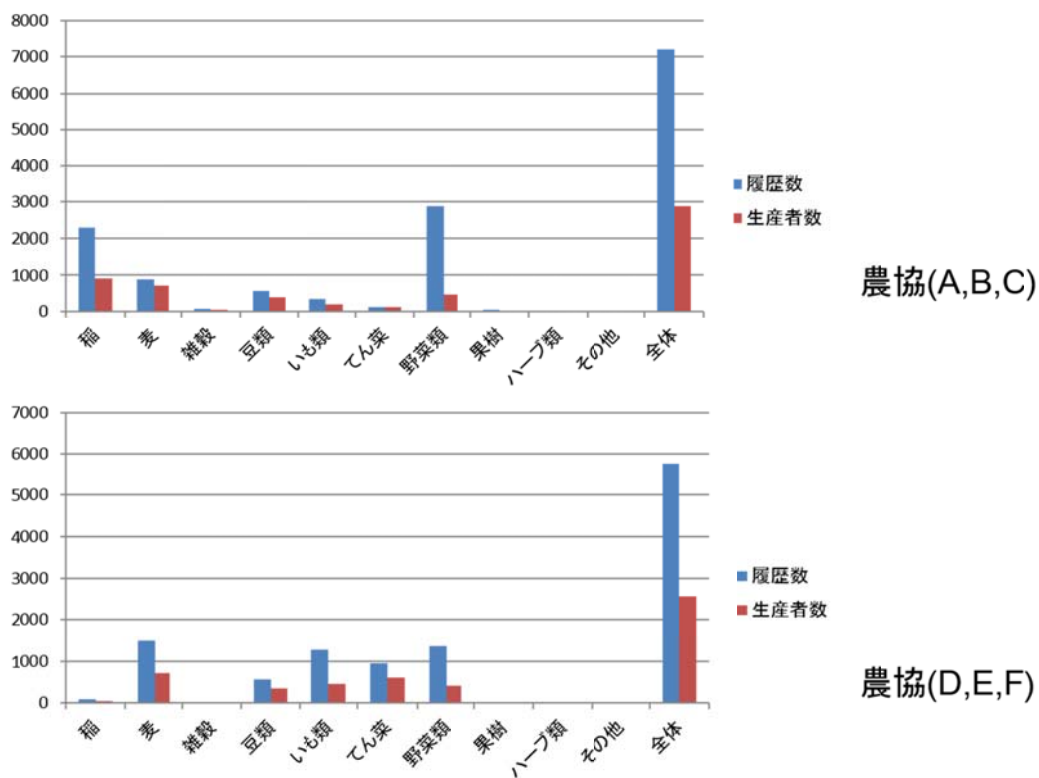


図 3-7 作物種類別の提出履歴数と生産者数

Fig 3-7 Number of submitted sheets and number of farmers by crop

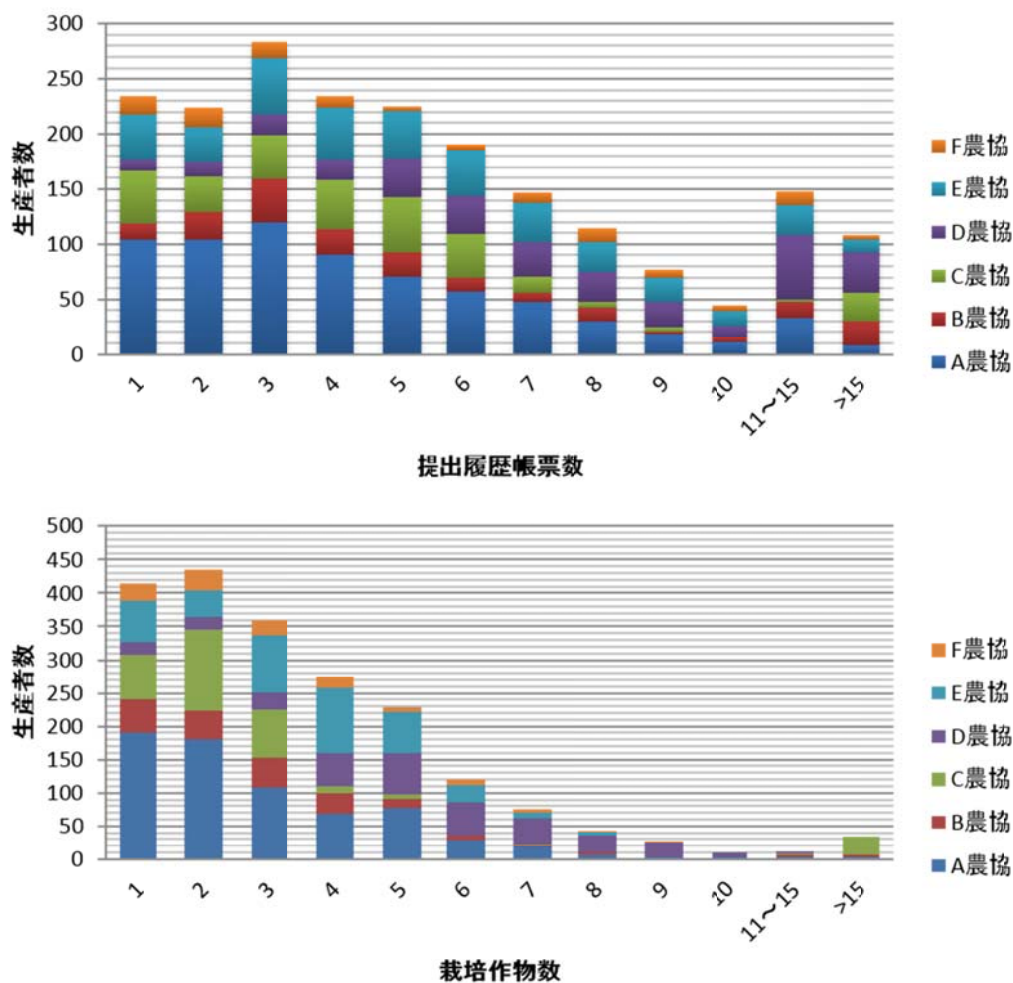


図 3-8 生産者一人あたりの提出履歴帳票数（上図）と栽培作物数（下図）

Fig 3-8 Number of submitted sheets (above) and number of cultivated crops (below) of each farmer

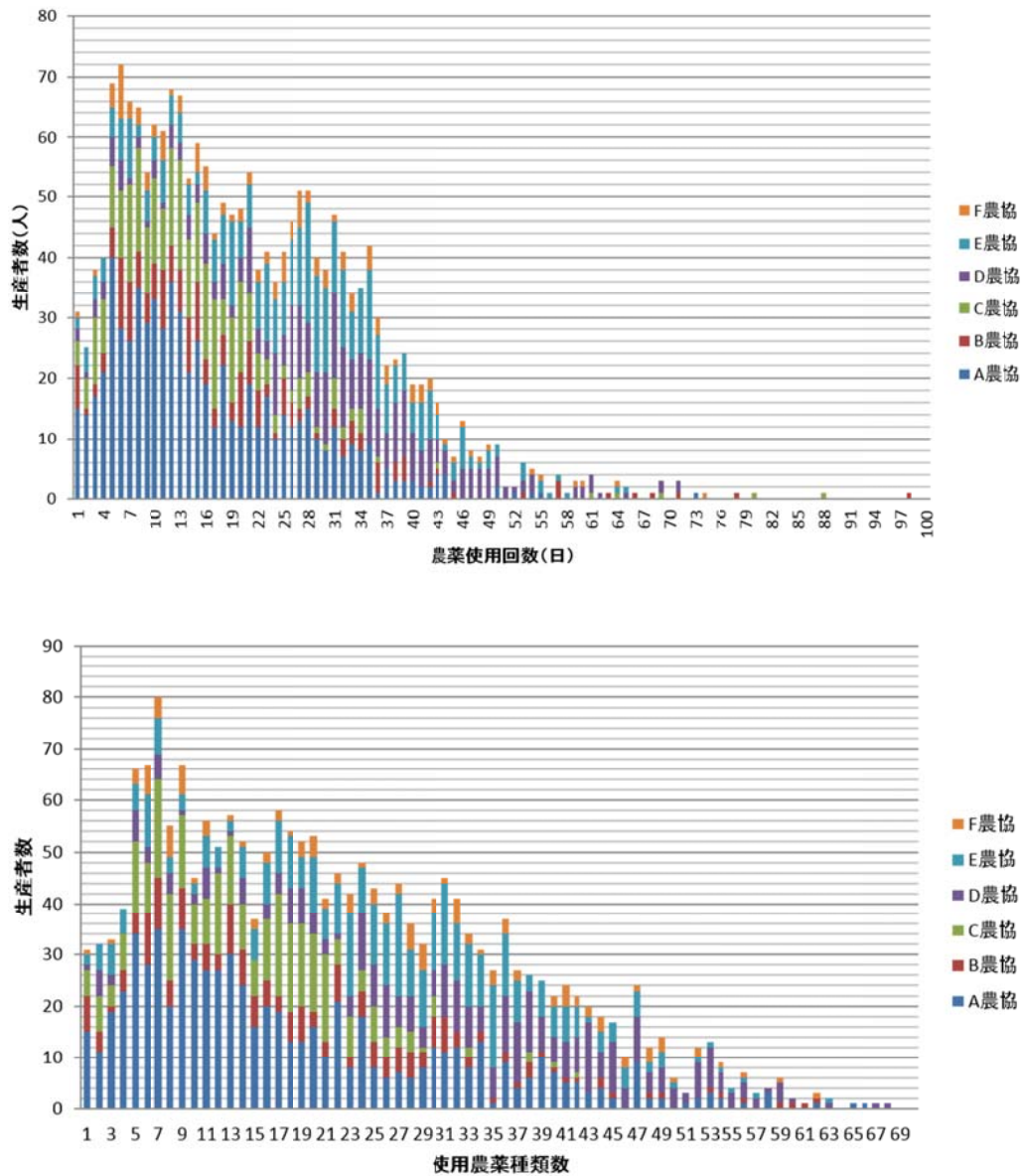


図 3-9 生産者一人あたりの農薬使用日数（上図）と使用農薬種類数（下図）

Fig 3-9 Number of days of agricultural chemical application (above) and number of types of agricultural chemicals (below) of each farmers

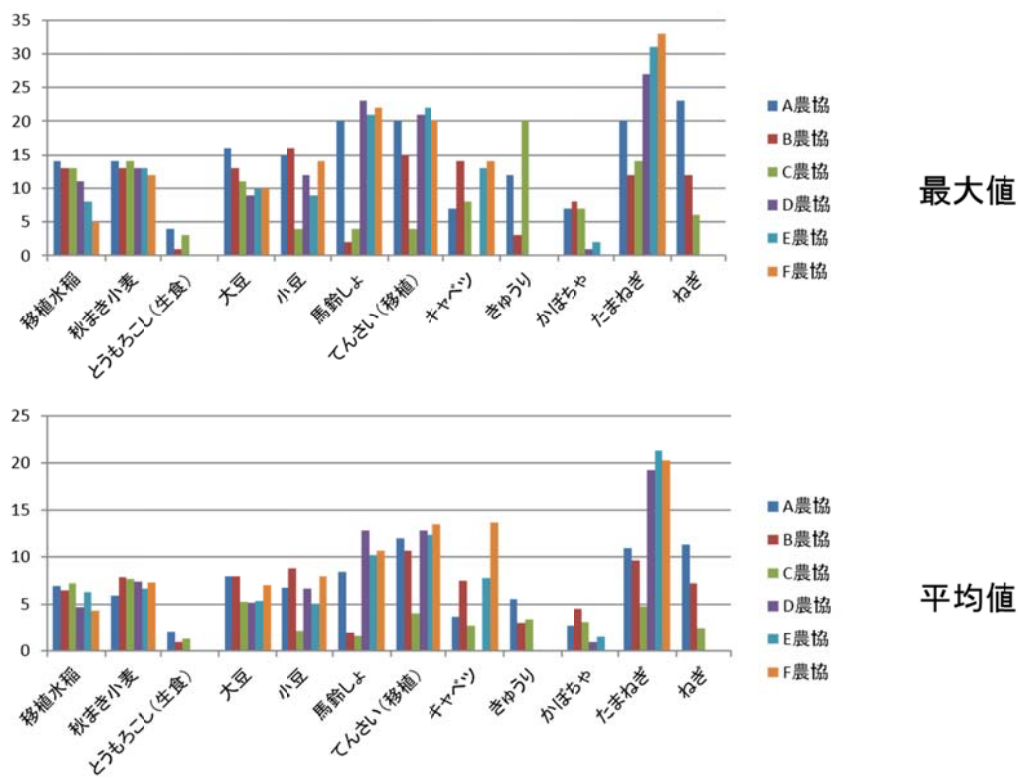


図 3-10 作物別の農薬使用回数の最大値（上図）と平均値（下図）

Fig 3-10 Max count (above) and Average count (below) of the number of agricultural chemical applications by crop

3.6 考察

前章で開発した「生産履歴, 生産資材マネジメントシステム」を全面的に見直し, 保守性, 拡張性を向上させた **apras** として再構築を行なった. **apras** のサーバプログラムは REST 形式の API としたが, これはシステムを保守性, 拡張性を実現する上で非常に有効であった. サーバプログラムにおいては各リソースを個別に扱う API を準備し, 各リソースを選択的に取得する機能や相互に作用させる機能などはクライアント側の実装に任せることとした. このことにより, **apras-app** の作成及びカスタマイズが柔軟にできるようになったため, スマートフォンなど新たな端末への対応も非常に容易になる事が示された.

apras-app の作成時には, コミュニティサイトに蓄積された技術情報やトラブルシューティング等に関する膨大なノウハウがプログラミング時の非常に大きな助けとなった. 現在も, コミュニティによりライブラリの開発, 改良も活発に行われているため, Web ブラウザのバージョンアップ等でブラウザの仕様が追加, 変更になった際はライブラリがそれに対応することが想定される. その結果, この種のライブラリを利用したプログラムは個別にブラウザの仕様変更等に対応する必要がなくなり, 以後のメンテナンス性が向上することが期待できる. 大規模なプログラムほど時間とともにレガシー化しやすいが, 極力多くの機能を汎用的なライブラリに委ねておけば, そのライブラリの開発, 維持が継続する間は, レガシー化の歩みを遅らせることが可能だと考える.

apras の利用実態を調査した結果, 調査対象とした北海道地域は, 一生産者あたりの圃場所有面積が他の地域と比較して格段に大きいため, 生産履歴管理業務が非常に大きな負担になっているのではないかと懸念もあったが, 図 3-8 で示した通り管理すべき生産履歴帳票の数はそれほど多くはなかった. これは, 一つの圃場が他の地域に比べて非常に大きいこと, また, 複数の圃場にまたがって同一管理で同一の作物を作付する事が多いため, 一つの生産履歴管理に大きな圃場面積を含められるためである. 逆に, 小さな面積で行う施設園芸の方が一回のシーズンで複数の品種や作型を導入するため, 生産履歴帳票は多くなる傾向が見られた. 今後, **apras** を都市近郊の施設園芸が盛んな地域に導入する際などは, 再度シ

ステム利用の負担等を調査する必要があると考える.

付録 サーバプログラム API 一覧

Appendix: List of server side program APIs

No	リソース	Method	URI
1	農薬一括	GET	http://{server}/{UID}/noyakus/登録番号
2	農薬一括	GET	http://{server}/{UID}/noyakus/番号?update=jpp 登録日付
3	肥料	GET	http://{server}/{UID}/hiryos/肥料コード
4	作物	GET	http://{server}/{UID}/sakumots/作物コード
5		GET	http://{server}/{UID}/sakumots
6	JA	GET	http://{server}/{UID}/jas/農協コード
7	生産者	GET	http://{server}/{UID}/seisansyas/生産者コード
8	履歴	GET	http://{server}/{UID}/rirkis
9		GET	http://{server}/{UID}/rirkis/履歴 ID
10		POST	http://{server}/{UID}/rirkis
11		PUT	http://{server}/{UID}/rirkis
12		DELETE	http://{server}/{UID}/rirkis
13	認証	POST	http://{server}/auth/
14	品種	GET	http://{server}/{UID}/hinshu
15	認証確認	POST	http://{server}/{UID}/authcheck
16	認証無効化	POST	http://{server}/{UID}/authclear
17	アップロード	POST	http://{server}/parent_upload.html
18	帳票定義	GET	http://{server}/ocrdef/1
19		GET	http://{server}/ocrdef/2
20	OCR 履歴テーブル	GET	http://{server}/{UID}/ocrrirkis/履歴 ID
21		PUT	http://{server}/{UID}/ocrrirkis
22		DELETE	http://{server}/{UID}/ocrrirkis
23	履歴概要テーブル	GET	http://{server}/{UID}/rirkis_gaiyo/履歴 ID
24		POST	http://{server}/{UID}/rirkis_gaiyo
25		PUT	http://{server}/{UID}/rirkis_gaiyo
26		DELETE	http://{server}/{UID}/rirkis_gaiyo
27	履歴圃場テーブル	GET	http://{server}/{UID}/rirkis_hojos/履歴 ID
28		POST	http://{server}/{UID}/rirkis_hojos
29		PUT	http://{server}/{UID}/rirkis_hojos

30		DELETE	http://{server}/{UID}/rerekis_hojos
31	履歴肥料テーブル	GET	http://{server}/{UID}/rerekis_hiryos/履歴 ID
32		POST	http://{server}/{UID}/rerekis_hiryos
33		PUT	http://{server}/{UID}/rerekis_hiryos
34		DELETE	http://{server}/{UID}/rerekis_hiryos
35	履歴農薬テーブル	GET	http://{server}/{UID}/rerekis_noyakus/履歴 ID
36		POST	http://{server}/{UID}/rerekis_noyakus
37		PUT	http://{server}/{UID}/rerekis_noyakus
38		DELETE	http://{server}/{UID}/rerekis_noyakus
39	履歴収穫テーブル	GET	http://{server}/{UID}/rerekis_shukakus/履歴 ID
40		POST	http://{server}/{UID}/rerekis_shukakus
41		PUT	http://{server}/{UID}/rerekis_shukakus
42		DELETE	http://{server}/{UID}/rerekis_shukakus
43	農薬診断用	GET	http://{server}/{UID}/shindan/履歴 ID
44	農薬, 肥料検索	GET	http://{server}/{UID}/hiryos_search?word=
45		GET	http://{server}/{UID}/noyakus_search?word=
46	履歴コピー	GET	http://{server}/{UID}/rireki_dup
47	履歴ステータス更新	PUT	http://{server}/{UID}/rerekis_status
48	生産者一覧取得	GET	http://{server}/{UID}/seisansyas_all/
49	生産者名称検索	GET	http://{server}/{UID}/seisansyas_search?word=
50	履歴一覧取得 (生産者指定)	GET	http://{server}/{UID}/rerekis?uid
51	履歴データ CSV 出力	GET	http://{server}/{UID}/rerekis_csv/年度?sakumotsu=作物
52	農薬データ CSV 出力	GET	http://{server}/{UID}/noyakus_csv/年度?sakumotsu=作物
53	肥料データ CSV 出力	GET	http://{server}/{UID}/hiryos_csv/年度?sakumotsu=作物
54	農薬成分取得取得	GET	http://{server}/{UID}/seibunlist
55	残留成分一覧取得	GET	http://{server}/{UID}/zanryulist
56	履歴受け入れ待ち一覧	GET	http://{server}/{UID}/rerekis_unapp?usr=
57	PDF 出力	GET	http://{server}/{UID}/pdfmake/履歴 ID?hojos=ABCDE
58	EXCEL 出力	GET	http://{server}/{UID}/excelmake/履歴 ID?hojos=ABCDE
59	特別栽培診断	GET	http://{server}/{UID}/tokusai/履歴 ID?sakugata=&hojos=A
60	作型テーブル	GET	http://{server}/{UID}/sakugata/作物コード
61		POST	http://{server}/{UID}/sakugata

62		PUT	http://{server}/{UID}/sakugata
63		DELETE	http://{server}/{UID}/sakugata
64	基準テーブル	GET	http://{server}/{UID}/ki jun/作物コード
65		POST	http://{server}/{UID}/ki jun
66		PUT	http://{server}/{UID}/ki jun
67		DELETE	http://{server}/{UID}/ki jun
68	作業一覧取得	GET	http://{server}/{UID}/sagyou
69	生育ステージ一覧取得	GET	http://{server}/{UID}/seiiku
70	履歴作業テーブル	GET	http://{server}/{UID}/rirekis_sagyous/履歴 ID
71		POST	http://{server}/{UID}/rirekis_sagyous
72		PUT	http://{server}/{UID}/rirekis_sagyous
73		DELETE	http://{server}/{UID}/rirekis_sagyous
74	履歴生育テーブル	GET	http://{server}/{UID}/rirekis_seiikus/履歴 ID
75		POST	http://{server}/{UID}/rirekis_seiikus
76		PUT	http://{server}/{UID}/rirekis_seiikus
77		DELETE	http://{server}/{UID}/rirekis_seiikus

第 4 章

API を核としたシステム拡張手法

4.1 背景および目的

第3章で述べたとおり、`apras`は、JA等で広く行われている生産履歴の記帳および管理業務を、ICTを活用することで高度化することができた。しかし、農産物の生産に係る情報は生産履歴のみにとどまらず、気象、土壌、生育状況など多岐にわたり、それらを統合的に扱うことでより高度な農産物生産を行う仕組みを構築することが必要だと考えられる。

今日においては、CPUやメモリなどが指数関数的な発展を続けており、各種情報のデータ量が急速に増大する情報爆発と言われる状況が続いている。農業分野においても、気象等のセンサデータ、農業機械の稼働記録、人工衛星等を用いたりリモートセンシングデータなど多くの情報が以前に比べて安価にかつ容易に入手できるようになってきており、それらの活用も進みつつある。現在、食料生産地域再生のための先端技術展開事業「土地利用型営農技術の実証研究」(先端農業情報ステーション, 2012)においては、農業に係る様々なアプリケーションをクラウド上で統合利用するための標準化技術として、プラットフォームとしてCLOP (Cloud-Open-Platform, 農業オープン・クラウド・プラットフォーム) 提唱されており(図4-1)、研究が進んでいるところである(Hirafuji et al., 2012)。

`apras`においては、APIによる拡張が容易に行える様に設計されているため、センサデータ等もAPIによる扱いが可能となるシステムを構築することで`apras`と連携することができる(図4-2)。本章では、センサや農業機械が生成する情報と`apras`とを統合するIoTのための拡張を行う。



図 4-1 農業オープン・クラウド・プラットフォームの概念

Fig 4-1 Concept of CLOP (Cloud-Open-Platform)

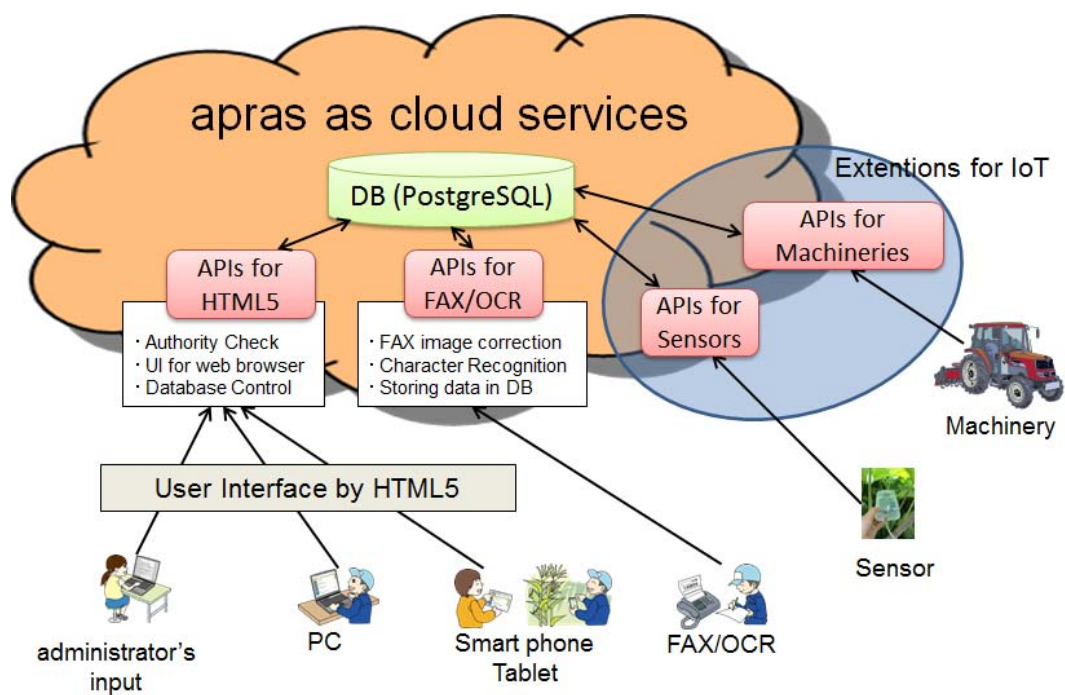


図 4-2 IoT のための apras の拡張

Fig 4-2 Extentions of apras for IoT

4.2 センサデバイスのための拡張

生産管理を科学的に行うには生産履歴情報と同時に気象データや作物生育情報を入力すべきであるが、測定値をマニュアルで入力するのは煩雑であり、入力を自動化する必要がある。apras は API で任意のデータを格納できるクラウドサービスであり、apras にセンサデータ等を柔軟に格納できる API 機能を持たせた。これにより、ユーザが農作業の合間や圃場巡回時に作物に関する様々なセンサデータを効率的に収集できる。









4.2.1 フィールドサーバ

気象データは、農業現場における最も重要なセンサデータの一つであり広く利用されている気象データの一つに AMeDAS がある。AMeDAS は、日本全国に約 17km 間隔で約 1,300 箇所を設置されているが、多くの農業地帯では現地の実体に近い観測地点がないのが実情であり、また、僅かな気象の差が作物に及ぼす影響を考えると観測されるべき箇所にセンサを設置するのが望ましい。フィールドサーバ (平藤ら, 2013) は、圃場に気象センサを設置できるデバイスであり、簡便に気象データを収集することが可能である。また、データを Twitter の API 経由で取得できるため apras のような API による設計がなされたシステムと容易に連携することが可能である。しかし、フィールドサーバは独自のフォーマットによるキャラクタベースのログしかないため、汎用利用のためには工夫が必要となる。

図 4-3 は、圃場に設置されたフィールドサーバ、Twitter 上でのフィールドサーバのログ、Twitter の API から取得したデータに基づくグラフを示している。グラフは、Twitter の API からログを取得した後、独自のデータフォーマットを JSON にシリアルライズし直し、JavaScript のグラフ描画ライブラリを用いてグラフ表示させたものである。グラフ描画ライブラリには Sencha Touch Chart を用いたことで、僅かな工数でモバイル環境での閲覧が可能なグラフ表示アプリを作成することが可能であった。API 経由でのデータ取得はライブラリとの連携が容易であるため、非常に単純なコーディングでリッチなアプリケーションが用意に実

装できた (章末付録参照)。このように、センサデータを API 経由で扱うことができれば他の API システムや既存のライブラリとの連携動作が簡単に実現できるため、`apras` のような API 形式のシステムであれば将来のセンサネットワークの発展に対しても十分対応可能であることが示された。



- 
nfa4324c @nfa4324c 11月10日
 Battery 11.6 [V], Board-T 8.8 [deg-C], S-Moist 87694 [Hz], Top-FD 751, Middle-Up-FD 57, Middle-Down-FD 18, Bottom-FD 4
- 
nfa4324c @nfa4324c 11月10日
 Battery 10.9 [V], Board-T 3.4 [deg-C], S-Moist 100246 [Hz], Top-FD 771, Middle-Up-FD 76, Middle-Down-FD 30, Bottom-FD 5
- 
nfa4324c @nfa4324c 11月10日
 Battery 10.9 [V], Board-T 3.9 [deg-C], S-Moist 104551 [Hz], Top-FD 774, Middle-Up-FD 91, Middle-Down-FD 49, Bottom-FD 7
- 
nfa4324c @nfa4324c 11月10日
 Battery 10.6 [V], Board-T 2.0 [deg-C], S-Moist 108355 [Hz], Top-FD 778, Middle-Up-FD 105, Middle-Down-FD 70, Bottom-FD 9
- 
nfa4324c @nfa4324c 11月10日
 Battery 11.1 [V], Board-T 12.7 [deg-C], S-Moist 89039 [Hz], Top-FD 759, Middle-Up-FD 140, Middle-Down-FD 112, Bottom-FD 14
- 
nfa4324c @nfa4324c 11月10日
 Battery 12.7 [V], Board-T 27.8 [deg-C], S-Moist 88955 [Hz], Top-FD 741, Middle-Up-FD 166, Middle-Down-FD 151, Bottom-FD 21
- 
nfa4324c @nfa4324c 11月9日
 Battery 10.8 [V], Board-T 2.0 [deg-C], S-Moist 101398 [Hz], Top-FD 770, Middle-Up-FD 72, Middle-Down-FD 24, Bottom-FD 6
- 
nfa4324c @nfa4324c 11月9日
 Battery 10.0 [V], Board-T 0.4 [deg-C], S-Moist 106823 [Hz], Top-FD 779, Middle-Up-FD 81, Middle-Down-FD 35, Bottom-FD 6

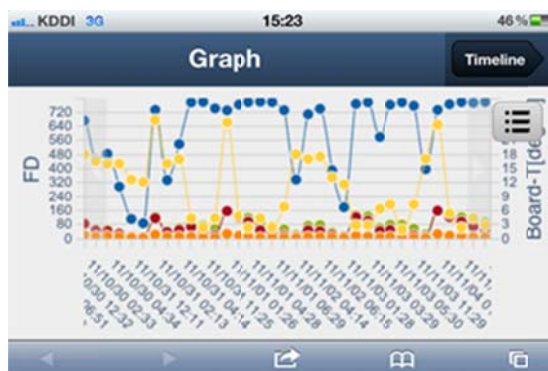


図 4-3 圃場に設置されたフィールドサーバ（上図）とフィールドサーバの Twitter ログ（下左図）に基づくグラフ化（下右図）

Fig 4-3 Field server placed in the field (above), Twitter logs of field server (below left) and chart application based on twitter logs (below right)

4.2.2 モバイル型計測機器データ

図4-4は圃場におけるバレイショ（キタアカリ）の群落内の透過光や反射光を超小型分光センサ C11708MA（浜松ホトニクス 2014）で測定した場合の例である。分光センサの受光部の位置及び方向を変えることによって、群落内透過光及び群落の反射光を簡単に測定できる。この分光センサの測定波長範囲は 640～1050 nm、波長分解能 20 nm である。このスペクトラムのデータセットはほぼリアルタイムに USB 経由で PC に伝送され、ファイルとしてフォルダに保存される。このファイルは Dropbox 等のクラウドサービスを通じて容易に共有できるが、そのままでは作目・品種・農作業等との関連づけができない。そこで、これを `apras` のデータフィールドに記録できるよう、システムの拡張を行った。拡張にあたっては、データベースに新たなテーブルを作成し、そのテーブルへのデータ入出力のための API を新たに作成する必要がある。REST API は、各 API の機能がシンプルでかつ似通っているため、新たな API を作成する時は既存の API のソースコードの多くを流用することができ開発は容易であった。分光センサ用の API 作成には約3人日を要したが、これは今回使用した分光センサのデータ保存形式が独自であったため、データ解析のためのパーサを独自で実装したことによってかかった工数がほとんどである。XML など汎用的なフォーマットを出力できるセンサであれば、システムの拡張は更に容易になると考えられる。

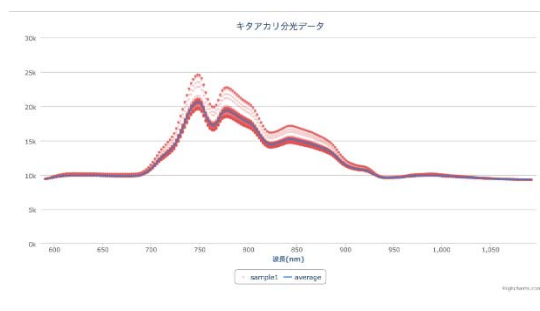


図 4-4 apras に格納した現場測定データの閲覧画面の例。(左図:超小型分光センサによる透過光スペクトラムデータの収集. 右図:スマートフォンでの閲覧例)

Fig 4-4 Spot measurements of crop for apras (Left: measurement of transmitted light spectrum, Right: View of measured data on smartphone)

4.3 農業機械のための拡張

近年、農業機械にカメラやセンサ類を設置したり、コンピュータを搭載するなど、農業機械のロボット化が進みつつある。また、CAN (Controller Area Network) により農業機械上のセンサや各種の駆動装置をコントロールすることも可能になってきている。ロボット化された農業機械においては、センサや機械の状態に関する様々な情報を電子データとして取得することが容易であるため、それらのデータは農業機械の制御に使用できると共に農作業の記録として活用することも可能である。そこで、農業機械と apras とを連携させることで作業支援や農作業情報の自動取得を行う手法を開発する。

4.3.1 農薬散布支援技術

前章までで述べたとおり、農薬の使用には詳細な使用方法が定められており、また農薬の誤使用は食品事故に直結することから、その使用に当たっては細心の注意が求められる。多くの農薬において使用倍率や使用量が定められており、使用に際しては使用者自らが原液を希釈し、必要量を散布することが一般的であるが、使用者の錯誤によって正しい倍率に希釈されない可能性は否定できない。そこで、農業機械によって希釈倍率を正しく制御することができればヒューマンエラーの排除につながると考えられる。

従来の農薬散布作業は、あらかじめ使用者が薬液と水を薬液タンク内で規定の倍率に希釈して散布するのが一般的であるが、近年、散布時に薬液と水を自動混合しつつ散布する防除機が普及しつつある。そこで、自動混合装置を備えた農業機械と apras を連動することで適切な倍率での散布を支援するシステムを開発した (図 4-5)。農業機械のキャビン内に設置された車載 PC は、携帯電話回線経由で apras にアクセスできる。また、CAN を介して農薬散布機への制御命令を送受信できるアプリケーションを開発し、同 PC にインストールした。農薬希釈倍率制御装置への命令伝達は、PC の USB ポート経由で USB バスに接続できるケーブル (CAN USB) 経由で行われる (図 4-6)。車載 PC 上のアプリケーション上で、利用者が散布予定農薬の対象病害虫、

散布方法、対象作物等の検索条件をタッチパネル操作で入力すると、アプリケーションが **apras** の API を呼び出して過去の農薬使用履歴を照会することで、使用予定農薬の使用可否に関する情報がウェブブラウザ上で参照できる。農薬が使用可能であった場合は、適切な使用倍率や量が提示される。提示された使用倍率は車載 PC 上のアプリケーション上にセットされ、CAN を介して農薬散布機への制御命令として伝達される。散布作業が開始されると、**apras** の API を介して農薬使用履歴が自動的に記録される。このことは、記帳の手間の軽減につながるだけでなく、自動で記録されることで記録の信頼度の向上にも寄与する。

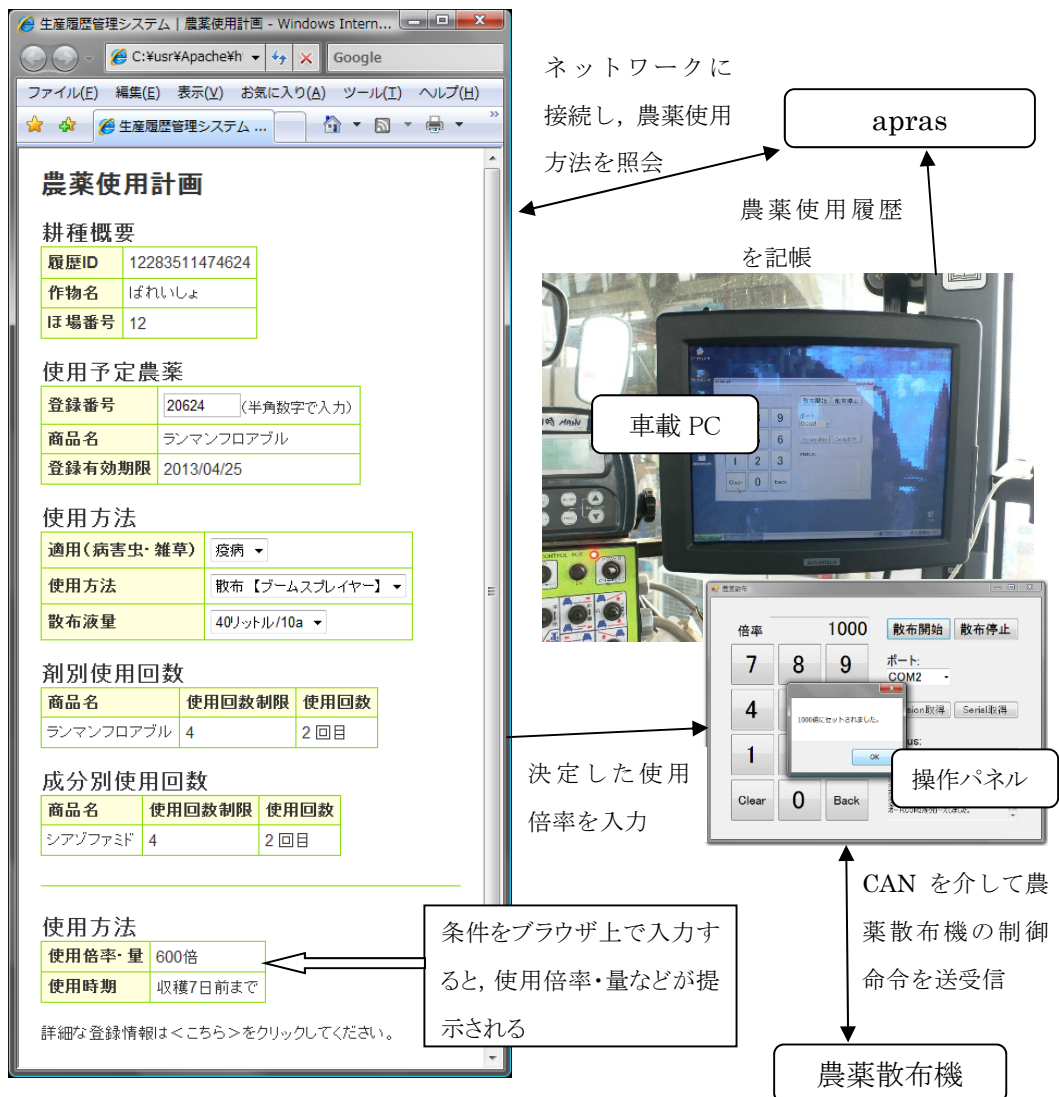


図 4-5 apras と農業機械の車載 PC との連携

Fig 4-5 Cooperation system between apras and agricultural machinery with vehicle-mounted PC

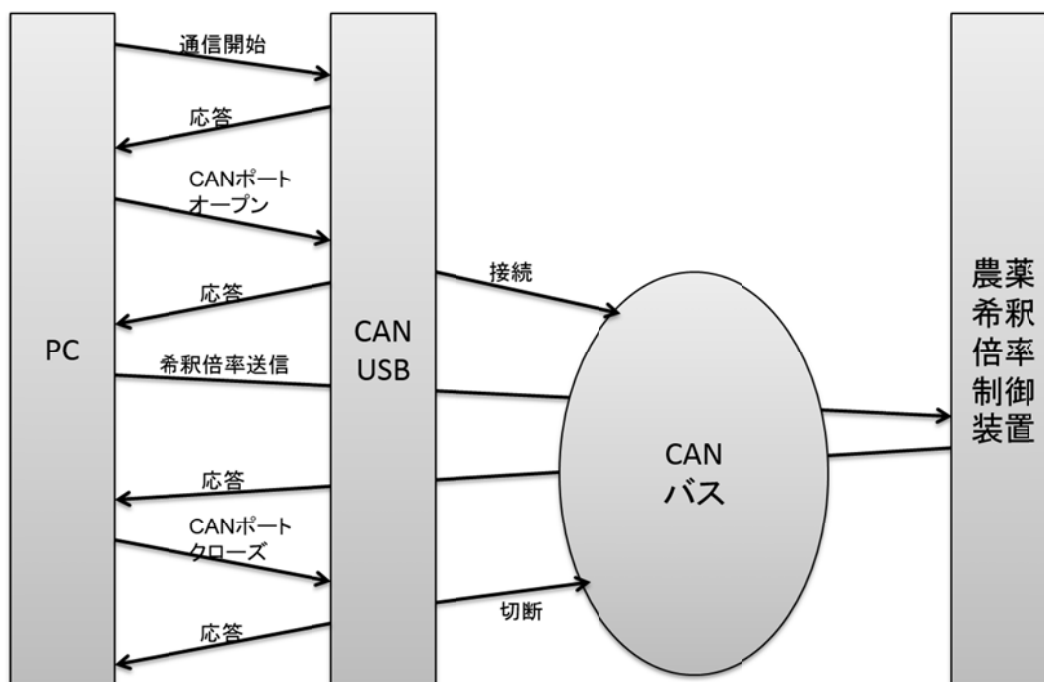


図 4-6 車載 PC と農薬散布機間の通信制御

Fig 4-6 Communication control between vehicle-mounted PC and pesticide applicator

4.3.2 農業機械の走行軌跡記録

近年、北海道などの大規模農業地帯を中心に農業機械の自動操舵装置やガイダンスシステムの普及が進んでおり、それに伴い GPS 受信機が農業機械に設置される事例が増加している。GPS データは、農業機械の走行経路を適切に保つなどの用途に活用されるが、データが使用されるのは作業中のみであり、データを蓄積し活用するといったことは通常行われなない。しかし、GPS データは作業の記録そのものであり、適切に処理することで精密な作業ログとして活用することが期待できる。

図 4-7 は、キャベツの収穫機に市販の GPS 機能を搭載したカメラを設置し収穫作業を記録した様子である。カメラは、農業機械上での収穫物の選別作業を記録するために導入されたものであるが、本研究では GPS による走行ログの解析を試みた。図 4-8 にその結果を示す。上図は、GPS のログに基づく農業機械の走行軌跡を Google マップ上に描画したものである。下図は、走行速度および走行距離をグラフ化したものである。さらに、走行ログの解析から農業機械の稼働時間のうち収穫を行っている時間帯の推定を行った。収穫を行っている時間帯は、上図においては青線の軌跡、下図においては黄色で色付けされた背景で示されている。旋回や停止など収穫以外の作業時間は、上図においては赤線の軌跡、下図においては白色の背景で示されている。収穫時間帯の推定は、10 秒前からの移動距離と 10 秒前からの 1 秒ごとの移動距離の積算の比がしきい値（ここでは 0.7）を超え、かつ移動速度がしきい値（ここでは 0.3m/s）を超えた時に収穫作業中であると判断する比較的単純な方法で行ったが、概ね実情を反映していると考えられる。上記の結果は、`apras` に新たに作業ログを蓄積できるよう API を拡張しデータの蓄積を行っている。現在では、農業機械の作業内容（例えば収穫など）を自動で判断できないため一部手動での入力を伴うが、農業機械の稼働記録が `apras` に問題なく蓄積できることを検証できた。



図 4-7 GPS 機能付きカメラを設置したキャベツの収穫機

Fig 4-7 Cabbage harvester and GPS camera on machinery

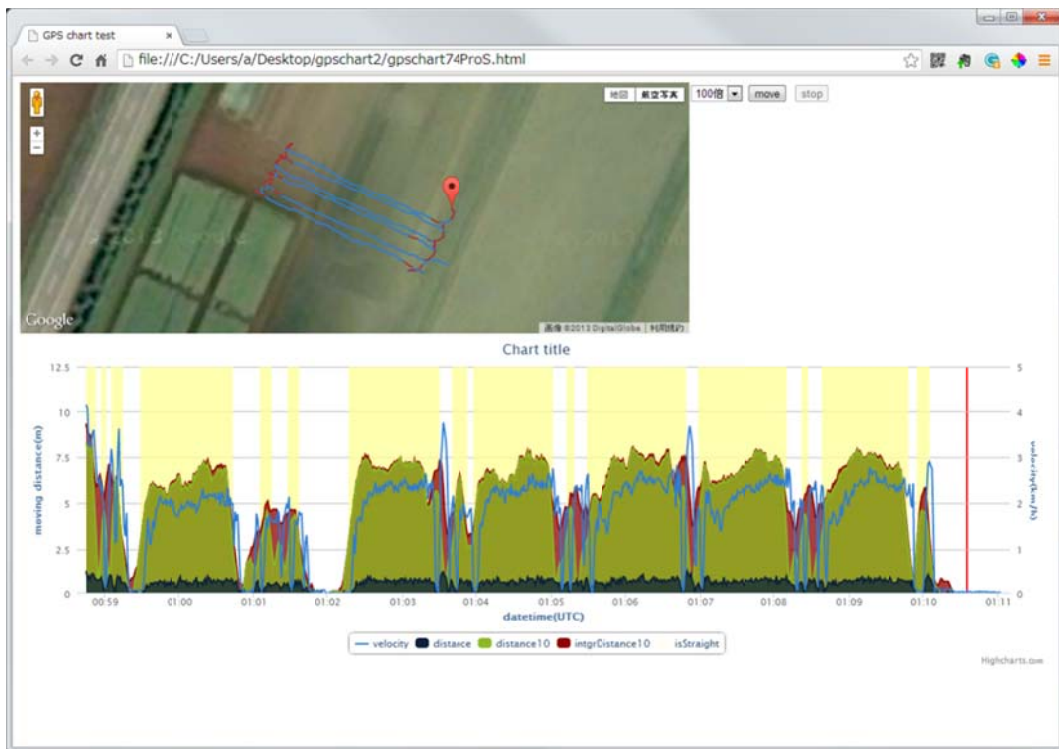


図 4-8 Google マップ上での走行軌跡の表示（上図）と走行ログの解析（下図）
 Fig 4-8 Trajectory view on Google maps (above) and analysis chart of machinery log (below)

4.4 考察

第3章では、`apras`の当初の目的である従来JAで行われていた生産履歴管理に係るシステム化を行ったが、本章では生産履歴の周辺情報への対応を`apras`の拡張により行うことができることを示した。`apras`と他のシステムとの連携は、APIによって行われるため、互いのシステムの内部仕様を把握したり、あるいは変更したりといったことは必要ない。本研究においては、すべてのシステム構築を筆者が行っているためこのことはあまり重要ではないが、第三者との連携の際はこのことは非常に重要である。第三者が、`apras`から情報を取得し、`apras`に情報を返すといったシステムを構築する際には、開発者は第3章付録のサーバプログラムAPI一覧のみを把握していれば良い。また、`apras`側も新たなシステムに連携に当たってAPIを一部追加したりデータベースのテーブルを一部追加したりという作業はあるが、システム連携に際して`apras`の内部仕様を変更するといった開発は必要ない。本章における開発についても、`apras`に対する拡張機能実装にあたって`apras`の内部仕様の変更や追加は一切行っていない。センサデータのグラフの描画、GPSデータの可視化チャートの描画といった新たな機能は、すべて`apras`とは切り離された新たなクライアントソフトを作成することで実現している。こうすることで、`apras`自体の保守性が担保されると共に、万一新たに作成したクライアントソフトにおいて不具合が存在した場合でも`apras`への影響を排除することができる。

付録 スマートフォン向けグラフ描画アプリケーションのためのソースコード

Appendix: Source code of chart drawing application for smartphones

```
Ext.setup({
  tabletStartupScreen: 'tablet_startup.png',
  phoneStartupScreen: 'phone_startup.png',

  onReady: function() {

    var timeline = new Ext.Panel({
      id: 'timeline',
      cls: 'timeline',
      scroll: 'vertical',
      dockedItems: [{
        xtype: 'toolbar',
        dock: 'top',
        title: 'Timeline',
        items: [{
          ui: 'back',
          text: 'Graph',
          handler: function() {
            Ext.getCmp('cards').layout.setActiveItem('graph', { type: 'slide', direction:
'right' });
          }
        ]
      }
    ]},
    ],
    tpl: [
      '<tpl for=". ">',
      '<div class="tweet">',
      '<div class="tweet-content">',
      '<h2>{created_at}</h2>',
      '<p>{text}</p>',
      '</div>',
      '</div>',
      '</tpl>'
    ]
  });

  var store = new Ext.data.JsonStore({
    fields: ['date', 'TOP-FD', 'Middle-Up-FD', 'Middle-Down-FD', 'Bottom-FD',
'Board-T']
  });

  var generateData = function(data) {
    var value = [];
    for(i = 0; i < data.length; i++){
      var date = data[i].created_at.split(' ');
      var tweet = data[i].text.split(',').join("").split(' ');

      value.push({
        'date': new Date(date[1] + ' ' + date[2] + " " + date[5] + ' ' + date[3]),
        'TOP-FD': parseFloat(tweet[10]),

```



```

        'Middle-Up-FD': parseFloat(tweet[12]),
        'Middle-Down-FD': parseFloat(tweet[14]),
        'Bottom-FD': parseFloat(tweet[16]),
        'Board-T': parseFloat(tweet[4])
    })
}
return value.reverse();
}

var refresh = function() {
    Ext.util.JSONP.request({
        url: 'http://twitter.com/statuses/user_timeline/nfa4324c.json',
        callbackKey: 'callback',
        params: {
            callback: '?',
            count: 100
        },
        callback: function(data) {
            if (data && !!data.length) {
                timeline.update(data);
                store.loadData(generateData(data));
            }
        }
    });
}
refresh();

var graph = new Ext.Panel({
    id: 'graph',
    layout: 'fit',
    items: [
        new Ext.chart.Panel({
            title: 'Graph',
            dockedItems: {
                dock: 'top',
                xtype: 'toolbar',
                items: [{
                    ui: 'forward',
                    text: 'Timeline',
                    handler: function(){
                        Ext.getCmp('cards').layout.setActiveItem('timeline', { type:
'slide', direction: 'left' });
                    }
                }
            ]
        },
        items: {
            cls: 'stock1',
            theme: 'Demo',
            legend: {
                position: 'right'
            },
            interactions: [{
                type: 'panzoom',
                axes: {
                    bottom: {

```

```

        maxZoom: 200
      }
    }
  },
  animate: false,
  store: store,
  axes: [{
    type: 'Numeric',
    position: 'left',
    fields: ['TOP-FD', 'Middle-Up-FD', 'Middle-Down-FD',
'Bottom-FD'],
    title: 'FD'
  }, {
    type: 'Numeric',
    position: 'right',
    fields: ['Board-T'],
    title: 'Board-T[deg-C]'
  }, {
    type: 'Time',
    position: 'bottom',
    fields: ['date'],
    dateFormat: ' y/m/d h:i ',
    groupBy: 'year,month,day,hour',
    label: {
      rotate: {
        degrees: 45
      }
    }
  }
  ]],
  series: [{
    type: 'line',
    showMarkers: true,
    smooth: true,
    axis: 'left',
    xField: 'date',
    yField: 'TOP-FD'
  }, {
    type: 'line',
    showMarkers: true,
    smooth: true,
    axis: 'left',
    xField: 'date',
    yField: 'Middle-Up-FD'
  }, {
    type: 'line',
    showMarkers: true,
    smooth: true,
    axis: 'left',
    xField: 'date',
    yField: 'Middle-Down-FD'
  }, {
    type: 'line',
    showMarkers: true,

```

```
        smooth: true,
        axis: 'left',
        xField: 'date',
        yField: 'Bottom-FD'
    }, {
        type: 'line',
        showMarkers: true,
        smooth: true,
        axis: 'right',
        xField: 'date',
        yField: 'Board-T'
    }
    ]
    });

    new Ext.Panel({
        id: 'cards',
        layout: 'card',
        fullscreen: true,
        activeItem: 0,
        items: [graph, timeline]
    });
} //onReady()
});
```

第 5 章

総合考察

5.1 まとめ

本論文では、農業現場において ICT を活用し生産履歴情報を始めとした様々な農業情報を収集、管理するための手法およびシステム化について論じた。

第1章では、はじめに農業における情報システムの必要性を考える上で最も重要な要素の一つである食の安全・安心を巡る情勢について整理を行った。21世紀以降大きく揺らいだ食と農に対する信頼を回復するためにはリスクコミュニケーションが重要であり、流通段階における情報管理を行うトレーサビリティシステムを食に関する情報システムの先行事例として紹介した。農業分野における ICT の利用を考える際、わが国の農業就労人口の高齢化が普及に対するネガティブな要因となりうることが考えられ、実際にわが国の農業に関する ICT の利用は諸外国に比べて遅れていることが分かった。しかし、今後 ICT の活用が農業分野においても重要になることが想定されるため、システム構築に当ってはわが国の農業事情に即した戦略を取ることが必要であると考えた。

第2章では、生産履歴情報を電子化するためのシステムを構築し、現場での実証試験によりシステムの有用性の検討を行った。システムは、JAにおける生産履歴情報の管理を目的とし、すべての組合員のすべての生産履歴情報を電子化すると共に農薬の使用可否の判断支援など、営農支援を行う仕組みを備えた。情報を電子化する手法に OCR による手書き帳票の読取りを用いたことは、IT 機器に慣れない農業者であってもシステムを利用できるという点で JA におけるシステム運用においては非常に有効に機能した。システムは「生産履歴、生産資材マネジメントシステム」と命名され、2004年度より北海道内の JA で実証試験を継続し、JAにおける生産履歴および生産資材の管理システムとして問題なく運用できることを確認している。しかし、システムの有用性は示されたものの、一方でシステムを支える ICT の状況が大きく変化したことから、システムを安定的に運用し続けることに対するコストが著しく増大したため、システムの根本の設計を見直す必要性に迫られた。

第3章では、第2章で構築したシステムを再構築することで、システムの保守性、拡張性の向上を実現した。システム再構築にあたり、HTML や JavaScript

といったウェブを支える技術の情勢を整理し、さらにシステムの持続可能性についての考察を行った。また、第2章のシステムにおける問題点を整理し、使用言語や設計手法の見直しを行った。それを踏まえ、新たなシステムは REST 形式の Web API を核としたクラウド型システムとし、次世代のウェブの中核をなす HTML5 を用いた柔軟性の高い設計を行った。新システムは **apras** と命名され、旧システムの機能をすべて盛り込み HTML5 の機能を活用したリッチなインターフェースを実現した。また、システムの拡張性が向上した結果、スマートフォンなど新たな情報端末への対応が非常に低いコストで実現できることも実証できた。

第4章では、**apras** の拡張手法を検討した。**apras** は、生産履歴情報を扱うシステムとして設計されたが、農業に係る情報はそれ以外にも様々あることから、これらの情報も統合的に利用できることが望ましい。本章では、センサと農業機械の情報を **apras** と連携動作するシステムを複数構築し、気象センサ、分光センサ、農業機械の CAN、農業機械に設置された GPS のそれぞれについて、**apras** に情報を問題なく受け渡せることを確認した。Web API で設計されたシステムであれば **apras** と容易に連携できるため、今後様々な情報を **apras** で扱う、あるいは逆に **apras** の情報を他のシステムで扱うことができることが示唆された。

一般的に、ICT を活用したシステムはアーリーアダプターと呼ばれる一部の先行したユーザによって利用が開始されることが多いが、本システムにおいては第一に紙ベースでの記帳を前提としたシステムを構築した後に、ウェブベースでの記帳の仕組みを追加導入し並行稼働させることで、あらゆるタイプの農業者をユーザとすることに成功した。このことは、JA においては共同出荷などの際に全情報が等しく管理される必要があることから、非常に重要な利点であった。旧システムの実証試験を開始した 2004 年は、Web 2.0 や Ajax といった用語が生まれた 2005 年の前年に当たり、実証試験開始後にウェブを巡る環境の大きなパラダイムシフトが起こった。当初は、旧システムの改良により新技術への対応を行っていたが、旧来の技術をベースとする改良はシステムの維持コストを増大させ、永続的な運用を困難にした。そこで、2010 年より新たな技術に基づくシステムをゼロから構築しなおし、保守性、拡張性を向上させた新システム **apras** を構築した。

これにより、同時期に普及が急速に進んだスマートフォンなど新たな情報端末への対応も非常に低いコストで行うことができた。今後、センサネットワーク等のさらなる発展により様々な情報がさらに容易に取得できるようになることが期待されるが、APIでの連携により **apras** を通じてより多くの情報にアクセスする環境を構築することで、**apras** が農業情報を統合的に扱うプラットフォームとなることが期待できる。

5.2 農業現場における ICT 活用に関する考察

農業分野において、高齢化等の問題で ICT の活用が進みにくい現状にあることは第1章で述べたとおりであるが、では実際にどの程度 ICT の活用が進んでいるかを検証するために apras での ICT 活用の実態を調査した。

図 5-1 は、各ユーザがどのような手段でデータを入力したのかを apras のシステムログから解析し、その割合を各チャートに示している。「紙ベース」は、手書き帳票そのものを JA に提出、あるいは Fax-OCR の利用を表す。「PC 等」は、PC、スマートフォン、タブレットなどウェブベースでの利用を表す。C と F の2つの JA では、「PC 等」の利用が 100%となっている。これは、すべてのユーザが PC 等を利用しているということではなく、紙ベースのユーザの記帳を JA 職員が PC で入力し直していることを表す。これは、紙の帳票をスキャンし OCR の誤認識がないかを視認するよりは直接入力したほうが速い、あるいはストレスが少ないと考えているからだと考えられる。C、F 以外の JA について、それぞれ「併用」が一定割合存在する。紙ベースと PC 利用を併用しているユーザが多数いるとは考えにくく、「併用」のほとんどは上で述べた C、F の事例と同じく JA 職員が紙ベースのものの一部を PC 等で入力し直している結果だと考えられる。これは、C、F 以外の JA においても手入力を選択する職員が一定割合いることを示している。あるいは、集荷所などスキャナがない環境で即座にデータを電子化したいという需要があることも関係していると思われる。「併用」は、ユーザ自身はほとんど紙ベースでの記帳を行っていると考えられることから、ユーザ段階でのニーズは紙ベースでの記帳がかなりの割合を占めていることが分かる。現時点では、あらゆる農業者を対象にそのデータを電子化するという場合には、JA 等の団体における役割が非常に大きいと言える。

以上より、農業分野における ICT の活用は依然として大きなハードルであることが示された。北海道地域は、他の地域と比較すると若い農業者の割合が多いため、他の地域への apras の導入に際しては ICT 活用の課題は一層高いハードルになることが想定される。apras は、「手書き帳票」でも「スマートフォン等での記帳」でも同一システムで並行稼動可能であるため、ICT の活用割合の低さは apras

の導入に際しての大きな問題にはならないと考えられるが、手書き及び PC によるデータ入力は生産者の大きな負担となっている。しかも、今後はより詳細なデータの inputs が求められることも予想されるため、第4章で述べた **apras** の拡張手法により普及が期待されるウェアラブルデバイスやさまざまなセンサに迅速な対応することが今後重要になってくると考えられる。

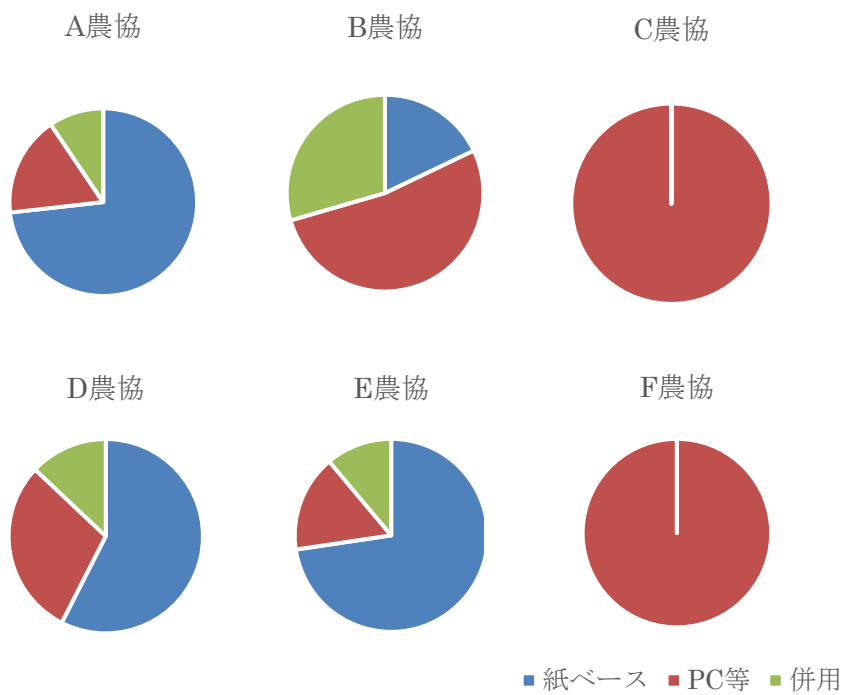


図 5-1 生産者が用いる記帳方式の割合

Fig 5-1 Rate of record keeping method by farmer

5.3 apras の現場活用に関する考察

第2章、第3章で述べたシステムは、その開発期間を通じてすべて現地（JA）での実証試験を行ってきた。また、現地の要望をシステムにフィードバックすることでシステムの改良を重ねた。農業現場では、生産履歴情報の収集、管理業務が大きな負担となっていたが、**apras**を導入することでその負担を軽減し、さらに高度な情報管理が実現することが実証試験を通じて確認できている。

現在、多くのJAでは収穫物の出荷時に生産履歴帳票の提出を義務付ける協定を生産者と締結している。**apras**を導入しているJAにおいては手書きの生産履歴帳票の提出、または**apras**へのデータ入力が生産者に求められるが、導入JAへの聞き取り調査の結果、すべての生産者の情報が**apras**に入力された状態で出荷を行う運用ができています。つまり、**apras**がすべての生産者が問題なく使用することができ、また、あらゆる作物の情報を**apras**が問題なく扱うことができることが実証された。

aprasは、2014年より商用利用を開始し、現在は北海道と大阪府において運用されている。現在、東北、北陸、中部など日本各地で導入に向けた動きがあり、今後全国的な普及が見込まれる。

5.4 展望

5.4.1 営農改善への活用

apras は、入力された情報は半永久的に記録を残す運用をしているため、日々生産履歴等の情報が蓄積を続けている。旧システムによって蓄積された情報も apras に引き継がれているため、早い時期より実証試験を開始した JA においては 10 年近い年月における情報がすでに蓄積されている。今後、それらの情報を営農改善につなげることが期待される。現在、apras を運用している一部の JA では、普及指導員などの専門家が apras に蓄積されたデータを元に生産者に次年度以降の施肥、農薬散布の方法の改善を指導している事例がある。そういった専門家の指導のノウハウをシステム化できれば、システムに農作業の計画を入力すればより良い作業計画への変更をシステムが提案するといった事が可能になるかもしれない。あるいは、大量に蓄積された施肥や農薬散布などの作業履歴と、収量、品質などの関係を解析し、より適切な栽培管理方法をシステム自らが導き出せる可能性もある。また、田中(2011)が開発した農業シミュレーションモデルのフレームワークにおける作物生育モデルと apras を連携させれば、例えばユーザが自らの圃場のフィールドサーバを設置し apras に気象データを蓄積することで、自らの圃場における作物生育モデルを参照しながら栽培管理を行うといったことも可能になると考える。

5.4.2 ビッグデータの収集

これまでにシステムが扱ってきた生産履歴などの情報は、記録すべき項目は多岐にわたるものの蓄積される情報のサイズは特に大きくなく数もさほど大きくなることはない。また、第 4 章で扱った気象センサや農業機械に係る情報も基本はテキストベースであるためサイズは小さく数もあまり多くはなく、それらを記録し蓄積することに問題はない。しかし、例えば日々の生育を画像で記録するといった場合はこの限りではない。

現在、UAV（無人航空機）の農業利用が始まりつつあり、筆者らも UAV による

圃場空撮画像の収集を試みている。筆者らが対象とした北海道十勝地区においては、圃場の1筆の平均面積は5ha前後、生産者1戸あたり40ha程度の圃場を所有している。5haの圃場を地上100mで10分間飛行し市販のデジタルカメラ（SONY α6000）により2秒間隔で撮影し、それらの画像をオルソモザイク処理すると、地上分解能約12cmの画像を得ることができる。そうして得られた画像は、これまでに用いられてきた人工衛星の画像より分解能が数十倍も優れており、より高度な栽培管理への活用が期待できる。しかし、1回の撮影で発生する画像サイズの合計は約1GBにもなり、例えば40haの圃場を1週間間隔で5ヶ月間撮影するとすれば、合計は160GB程度の容量になってしまう。現状では、そのようなデータをaprasで扱うこと、あるいは解析することは不可能であるため、今後、ビッグデータへの対応を考えるに当たっては、PCクラスタ等を構築しaprasと連携することを検討すべきである。

5.4.3 農業における情報利用の発展方向

わが国の農業就業者はその6割以上が65歳以上であり、今後も高齢化が進むことが予想される。そのため、手書きベースでの生産履歴の記帳といった従来型の情報収集方法は当面は必要不可欠であると言える。一方で、農業分野においても各種のセンサが導入されるなど、人手を介すことなく生成される情報は年々増加している。喜連川(2011)が指摘するとおり、情報の生成がセンサなどの人工物によりなされるようになったことにより情報爆発が今後一層加速するものと想定され、その膨大な情報を逆手に取って利用することが重要になってくるであろう。これまで情報は、人間が記録を残そうと思ったものだけが人手により情報として残された。例えば、生産履歴であれば施肥や農薬散布に関する記録など、農業生産上重要とされる情報が選んで記録される。トラクタの運行状況などの記録も残す気になれば残せるはずだが、何に使うか分からない記録は残してこなかった。今では、ロボット化された農業機械であればエンジンをかけてからストップするまでの速度、エンジン回転数、燃料消費量などを自動的に記録することができる。それらの記録を元に農作業記録を自動生成するといったことは可能であるし、例え

ばエンジン負荷の変化を解析することで圃場内の土壌の不均一性を推定するといったこともできるかもしれない。

これまで述べたとおり, **apras** はその拡張性により新たなセンサデータ等への対応を容易に行うことができる。従って, 今後様々なデータが **apras** を介して統合利用され解析されることで, **apras** をビッグデータ分析のプラットフォームとして機能させることが期待できる。

謝辞

本研究の遂行および本論文の取りまとめるにあたり、数々のご指導ならびにご助言を頂いた、筑波大学大学院 生命環境科学研究科 教授 平藤雅之 博士（国立研究開発法人 農業・食品産業技術総合研究機構 北海道農業研究センター 大規模畑作研究領域 領域長）に深甚なる感謝の意を表します。また、本論文の取りまとめにあたりご指導頂いた、筑波大学大学院 生命環境科学研究科 教授 林武司 博士（国立研究開発法人 農業・食品産業技術総合研究機構 中央農業総合研究センター 情報利用研究領域 上席研究員）、筑波大学大学院 生命環境科学研究科 准教授 竹澤邦夫 博士（国立研究開発法人 農業・食品産業技術総合研究機構 中央農業総合研究センター 情報利用研究領域 上席研究員）、筑波大学大学院 生命環境科学研究科 教授 瀧川具弘 博士、東京大学 農学生命科学研究科 教授 二宮正士 博士に感謝の意を表します。

本研究の遂行にあたりご支援いただいた、国立研究開発法人 農業・食品産業技術総合研究機構 北海道農業研究センター 大規模畑作研究領域 上席研究員 澁谷幸憲氏、国立研究開発法人 農業・食品産業技術総合研究機構 北海道農業研究センター 大規模畑作研究領域 主任研究員 西脇健太郎氏、(株) 農業情報設計社 濱田安之氏（元国立研究開発法人 農業・食品産業技術総合研究機構 北海道農業研究センター 大規模畑作研究領域 主任研究員）に感謝の意を表します。また、本研究におけるシステム構築についてご協力いただいた北海道日興通信株式会社 システム開発部 遠藤智章氏に感謝の意を表します。また、本研究の実証試験に際してご協力いただいた北海道農産物生産履歴管理システム利用協議会 元会長 高橋浩氏（そらち南農業協同組合 営農部長）、北海道農産物生産履歴管理システム利用協議会 会長 清水則孝氏（津別町農業協同組合 経済部長）に感謝の意を表します。

なお、本研究の一部は、総務省 SCOPE「戦略的情報通信研究開発推進事業」、農林水産省「食料生産地域再生のための先端技術展開事業」の支援によって行われた。

引用文献

独立行政法人・家畜改良センター (2004) 牛の個体識別情報検索サービス,
<https://www.id.nlbc.go.jp/top.html>

European Commission (2003) “Waste Electrical & Electronic Equipment (WEEE)”,
http://ec.europa.eu/environment/waste/weee/index_en.htm

Fielding, R. T. (2000) “Architectural Styles and the Design of Network-based Software Architectures”, <http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm>

林和信・西村洋・堀尾光広 (2009) 様々な営農形態に柔軟に対応できる GIS 機能を備えた 営農情報管理システム「FARMS」,
<https://www.naro.affrc.go.jp/project/results/laboratory/brain/2008/brain08-05.html>

Hirafuji, M., Y. Hamada, T. Yoshida, A. Itoh and T. Kiura (2012) “Strategy and Concept of Open Cloud Application Platform in Agriculture” Proc. of AFITA/WCCA2012.,
http://www.afita.org/graph/web_structure/20130128110552/files/Semianr%20%2801%29-01.pdf

平藤 雅之・世一 秀雄・三木 悠吾・木浦 卓治・深津 時広・田中 慶・松本 恵子・星 典宏・根角 博久・澁谷 幸憲・伊藤 淳士・二宮 正士・Adinarayana J.・Sudharsan D.・斉藤 保典・小林 一樹・鈴木 剛伸 (2013) オープン・フィールドサーバ及びセンサクラウド・システムの開発, 農業情報研究, 22(1), 60-70

伊藤淳士・村上則幸 (2007) 生産履歴, 生産資材マネジメントシステム,
https://www.naro.affrc.go.jp/patent/program/patent_category/patent_cate070/006882.html

伊藤淳士・遠藤智章・平藤雅之 (2014), クラウドサービス型農産物生産工程管理システム「apras」の開発, 農業情報研究, 23(4), 154-164

Kaloxylos, A., R. Eigenmann, F. Teye, Z. Politopoulou, S. Wolfert, C. Shrank, M. Dilinger, I. Lampropoulou, E. Antoniou, L. Pesonen, H. Nicole, F. Thomas, N. Alonistioti and G. Kormentzas (2012) Farm management systems and the Future Internet era. Computers and Electronics in Agriculture, 89: 130–144.

Kaloxylas, A., A. Groumas, V. Sarris, L. Katsikas, P. Magdalinos, E. Antoniou, Z. Politopoulou, S. Wolfert, C. Brewster, R. Eigenmann and C. M. Terol (2014) A cloud-based Farm Management System: Architecture and implementation, *Computers and Electronics in Agriculture*, 100:168-179

経済産業省 (2001) 資源の有効な利用の促進に関する法律, <http://law.e-gov.go.jp/htmldata/H03/H03HO048.html>

喜連川優(2011) 情報爆発のこれまでとこれから, *電子情報通信学会誌*, 94(8), 662-666

厚生労働省・食の安全に関するリスクコミュニケーションの在り方に関する研究会 (2005) 食の安全に関するリスクコミュニケーションの今後の進め方について, <http://www.mhlw.go.jp/topics/bukyoku/iyaku/syoku-anzen/riskcom/dl/01.pdf>

厚生労働省 (2006) 食品に残留する農薬等に関する新しい制度 (ポジティブリスト制度) について, <http://www.mhlw.go.jp/topics/bukyoku/iyaku/syoku-anzen/zanryu2/dl/060516-1.pdf>

栗原伸一・丸山敦史・霜浦森平・西山未真・A.E. Luloff・廣瀬牧人・松田友義(2005) 食の安全性情報と消費者行動に関する基礎的研究, *食と緑の科学* 60, 99-108

Lazarus, W. F., Smith, T. R. (1988) "Adoption of Computers and Consultant Services by New York Dairy Farmers", *Journal of Dairy Science* 71(6), 1667-1675

前田健喜 (2004) JA グループの生産履歴記帳運動への取組, *公庫月報* 52(9), 10-15

Mărușter, L., Faber, N. R., Peters, K. (2008), "Sustainable Information Systems: a knowledge perspective", *Proceeding of CAiSE'08 Forum*

内閣府・食品安全委員会(2014) 食品安全モニター課題報告「食品の安全性に関する意識等について」, <http://www.fsc.go.jp/monitor/2608moni-kadai-kekka-yoyaku.pdf>

南石晃明・菅原幸治・渡邊朋也・大口鉄雄・菊地宏之・鈴木剛伸・遠藤宏幸 (2005) 農薬適正使用ナビゲーションシステム: 構想および実装, *農業情報研究*, 14(3), 207-226.

南石晃明・木村浩・平石武・高橋利美 (2006) 農薬使用リスク管理システムの開発実証. 農業情報研究, 15(4), 359-371.

農林水産省 (2002a) 無登録農薬問題の経緯について,
http://www.maff.go.jp/j/nouyaku/n_sizai/mutoroku_keii.html

農林水産省 (2002b) 食と農の再生プラン,
http://www.maff.go.jp/j/study/other/keiei/noukyo_study/pdf/1s6.pdf

農林水産省 (2006) 平成17年度農林漁家におけるパソコン等の利用状況調査結果,
<http://www.maff.go.jp/j/finding/mind/pdf/20060308cyosa.pdf>

農林水産省 (2006) 農業生産工程管理 (GAP) の共通基盤に関するガイドライン,
http://www.maff.go.jp/j/seisan/gizyutu/gap/guideline/pdf/guide_line_120306.pdf

農林水産省 (2012) 農業分野におけるIT利活用に関する意識・意向調査結果
<http://www.maff.go.jp/j/finding/mind/pdf/itrikatu.pdf>

農林水産省 (2014) 平成26年度食料自給率の概要
http://www.maff.go.jp/j/zyukyu/zikyu_ritu/pdf/26gai.pdf

農林水産省 (2015) 農業労働力に関する統計
<http://www.maff.go.jp/j/tokei/sihyo/data/08.html>

Park, T., Mishra, A. (2003) "Internet Usage by Farmers: Evidence from a National Survey", Paper for Presentation at the American Agricultural Economics Association Annual Meeting, Montreal, Canada

Piotrowicz, W., Cuthbertson, R. (2009) "Sustainability – a new dimension in information systems evaluation", Journal of Enterprise Information Management, 22(5) 492-503

Schelp, J., & Aier, S. (2009), "SOA and EA - sustainable contributions for increasing corporate agility", HICSS'09. 42nd Hawaii International Conference,

1-8

社団法人・食品受給研究センター (2005) トレーサビリティシステム導入事例集,
http://www.fmric.or.jp/trace/h16/casestudy/casestudy_full.pdf

社団法人・食品受給研究センター (2006) トレーサビリティシステム導入事例集 第2集,
http://www.fmric.or.jp/trace/h17/casestudy2/casestudy2_full.pdf

社団法人・食品受給研究センター (2007) トレーサビリティシステム導入事例集 第3集,
<http://www.fmric.or.jp/trace/h18/casestudy3.htm>

社団法人・食品受給研究センター (2008) トレーサビリティシステム導入事例集 第4集,
<http://www.fmric.or.jp/trace/h19/casestudy4.html>

先端農業情報ステーション(2012) 先端技術展開事業,
http://www.ais-sentan.jp/h_kadai.html

総務省(2012) 平成 23 年通信利用動向調査,
http://www.soumu.go.jp/menu_news/s-news/01tsushin02_02000040.html

総務省(2014) 平成 25 年通信利用動向調査,
<http://www.soumu.go.jp/johotsusintokei/statistics/statistics05.html>

菅原幸治・田中慶・大塚彰・南石晃明 (2006) 携帯電話上で作動する農薬使用適正判定・履歴入力アプリケーション, 農業情報研究, 15(4): 381-393.

田中慶(2011) 農業シミュレーションモデルにおける分散協調システムのフレームワークに関する研究, 博士論文 (筑波大学)

横田敏恭 (2004) 農薬取締法の改正と農薬の適正使用の遵守(残留農薬のリスクの評価・管理・コミュニケーション), 日本農薬学会 29(4), 395-396.

吉田智一・高橋英博・寺元郁博 (2009) 圃場地図ベース作業計画管理ソフトの開発, 農業情報研究, 18(1):187-198.

摘要

食の安全・安心は多くの国民の感心事であり、食料生産を担う農業においては食のリスクを排除すべく様々な対応が求められる。わが国においては、O-157 による集団食中毒（1996 年）、BSE 感染牛の確認（2001 年）、無登録農薬（ダイホルタン等）問題（2002 年）などの事件、事故を契機に農薬取締法改正（2002 年）、食品衛生法改正（ポジティブリスト制度の導入）（2006 年）が行われるなど、食と農に関する管理の厳格化が進んだ。また、農林水産省は 2002 年に「食と農の再生プラン」を発表し、その中では「食の安全と安心の確保」が柱の一つとして掲げられた。そうした流れを受けて、2002 年以降農業現場においては JA や産直組合などの生産者団体が入荷の条件として生産履歴帳票の提出を義務付ける動きが広がった。しかし、生産履歴情報は、生産工程を通じて随時記録される必要があり、複数の作目・品種を栽培すると記帳作業は複雑となる。また、肥料や農薬については、その使用方法が適切であるかを確認する必要があるが、その確認には高度な専門的知識が必要であり、煩雑な計算処理を伴うことから、これは非常に困難な業務となっている。

上記の問題を解決するためには、ICT を活用することで生産工程管理を支援するシステムを導入することが有効であると考えた。しかし、農業分野においては就労人口の過半数が 65 歳以上であるなど高齢化が進行しており、他の分野と比較すると ICT の受容度がかなり低い現状がある。そこで、筆者らは ICT に対する受容度に関わりなく JA における生産履歴情報の管理業務を支援するシステム「生産履歴、生産資材マネジメントシステム」を開発し、2004 年より実証試験を行った。本システムは、生産履歴と生産資材に関わる情報を電子化しサーバ上で一元管理することができるウェブシステムである。ICT の受容度が低い利用者に対しては紙ベースの専用帳票を利用してもらい OCR によって電子化する仕組みを導入することで、すべての生産者が問題なく利用できるシステムとした。実証試験を通じて、すべての生産者のすべての作物に関する情報を問題なく処理することが確認でき、JA における生産履歴管理業務の高度化に有用であることが示された。また、農薬の誤使用を未然に防ぐ、あるいは誤使用のあった生産物の出荷を防ぐ

など、食の安全にも寄与することが確認できた。

一方、インターネットを取り巻く技術は日々急速な進歩を遂げ、2005 年以降 Web 2.0 と呼ばれる新たな潮流が生まれた。その後、Ajax (2005 年)、クラウドコンピューティング (2006 年) など新たな技術革新が起こったことで、旧来のウェブシステムの急速な陳腐化は避けられない状況になった。そこで筆者らは、システムを安定的に運用し続ける技術的要件を整理し、保守性、拡張性に優れた新システムを再構築することとした。新システムでは、かつてウェブアプリケーションの主流であった CGI(Common Gateway Interface)方式をやめ、Web API 方式によるクラウドサービス方式を採用した。新システムは、**apras** (**a**gricultural **p**roduction **a**ssist **s**ystem) と命名され、旧システムと同じく北海道内の 8 つの JA で実証試験を行った。新システムでは、JavaScript の事実上標準となっているライブラリ jQuery を用いて HTML5 に対応した設計とすることで保守性が格段に向上した。また、API 方式としたことで拡張性も向上させることができた。具体的には、3 人月の工数をかけて開発した PC 向けクライアントソフトをスマートフォン向けに改変する拡張を 7 人日で行えるなど、将来的な新たな IT 機器の登場にも柔軟に対応できることが示された。一方、IT 機器に馴染まない農業者向けには新たに Fax-OCR 用の API を実装することで利便性を向上させた。システムログの分析の結果、依然として紙ベースでの利用への需要が高いことが分かっており、紙ベースとウェブベースの仕組みを並行稼働させる本システムが現状でも農業現場では有用であることが示された。

生産履歴は、一般的に人の手によって情報が記録されるが、将来はセンサネットワークやロボット技術などにより多くの情報を得ることが期待できる。そこで、生産履歴に関わる周辺情報を **apras** と連動させる仕組みを実装した。具体的には、気象や分光などのセンサデータ、農業機械の CAN (Controller Area Network) や GPS のデータとの連携が可能となった。

このように、**apras** は生産履歴情報を高度に管理できるほか、センサや農業機械のデータと連携できるなど、農業情報全般を扱うプラットフォームとなり得ることを示した。現在、UAV 等の新たなツールを用いた圃場データの収集に関する研究も盛ん

に行われていることから、将来的には様々なデータがさらに容易に収集できることが期待できるため、**apras** の有用性はさらに向上すると考えられる。

Integrated-Utilization Methods for Diverse Information in Agricultural Production History Management Systems

Abstract

Food-related accidents feature in the headlines across the world, occurring in numerous countries, including Japan. For example, in Japan, a serious incident occurred in 2002 regarding the sale of agricultural products containing hazardous agricultural chemicals. Since then, consumers have become much more aware of the safety of their food.

Furthermore, the revision of the Agricultural Chemicals Regulation Act and the Food Sanitation Act, has caused people and organizations engaged in farming to manage production processes more strictly at the farming sites. In addition, producers' groups, such as the JA (Japanese Agricultural Cooperatives), have begun to check and manage the production processes of individual farmers. As a result, many producers' groups have obliged farmers to submit management work reports, as well as records of farming materials and agricultural chemicals used in the production processes. Production history must be recorded at all times in every part of the production process; it creates a high volume of paperwork and consequently, it exerts a heavy burden on both farmers and producers' groups, who support farmers.

To meet these requirements, we considered the desirability of introducing a web system to support the production process management of agricultural products by leveraging ICT (Information and Communications Technology). The average age of the population engaged in agricultural practices in Japan is over 65; the acceptance of ICT in agriculture is lower than that in other industries. Therefore, we developed a web system called "The Management System for Agricultural Material and Production," accessible for both users and non-users of IT devices. We have conducted demonstration experiments of the system in Hokkaido since 2004. By using this system, users can easily digitize and unify information on production processes. Non-IT device users can refer to a tailored checking

form to be completed by hand. The form is scanned to access the data that are then automatically digitized by OCR functions. Thus, everyone can use this system regardless of IT literacy. Through the demonstration experiments, we confirmed that the system worked effectively in JA's practices. Additionally, this system supported safe agricultural production by evaluating the appropriate application of agricultural chemicals.

On the other hand, technologies involving the Internet are improving rapidly; since 2005, a new trend called "Web 2.0" has been launched. New technological innovations such as Ajax (2005) and cloud computing (2006) rapidly made the past web technologies obsolete. Therefore, we decided to restructure the system that we developed in the past to increase the maintainability and scalability of the system. We have developed a new system for cloud services based on "Web API" instead of Common Gateway Interface (CGI) that we used for the previous system. We named the new system "apras" (agricultural production assist system), and returned to conduct demonstration experiments in Hokkaido once again. The introduction of "jQuery" (a standard library of JavaScript) and HTML5-compliant structure, allowed the maintainability of "apras" to improve drastically. Moreover, the introduction of the Web API method contributed to the expansion of the scalability of "apras." Because of improved scalability, we succeeded in developing the client software for smartphone usage in just 3 days. We also developed the API for Fax-OCR system for users who are not IT-literate. Analysis of the system log indicates that the demand for paper-based systems remains high. Accordingly, this indicates that it is effective for the agricultural sector to operate both the paper-based system and web-based system at present.

Currently, humans are responsible for the gathering of production information; however, in the near future, we can expect to gain more information from sensor networks and robot technologies. We have implemented the mechanisms to aggregate a wide variety of agricultural information to "apras." Now we can assemble the sensor data of meteorology or crop spectrum, CAN (Controller Area Network), and GPS data of agricultural machinery to "apras."

We have shown that “apras” manages production history information effectively. Additionally, we have suggested the possibility that “apras” works as a platform for a wide range of agricultural information such as sensors or agricultural machinery data. The intensive collection of field data, utilizing new technologies such as UAV (Unmanned aerial vehicle), is currently in progress. In the near future, we will be able to collect substantial amounts of data with ease; hence, we believe that “apras” will progress significantly.