

## تصمیم گیری مبتنی بر ابر و اتوماسیون برای بهبود دقت کشاورزی در باغ ها

### چکیده

تحولات تکنولوژیکی و تجاری اخیر، محاسبات ابری را به یک تکنولوژی و پلت فرم مقرون به صرفه، مقیاس پذیر و بسیار قابل دسترس تبدیل کرده است. در عین حال افزایش دقت در کشاورزی و بهبود عملیات کشاورزی با تصمیم گیری بهتر داده ها، نشان دهنده توانایی های این تکنولوژی است. با این وجود، توسعه بیشتر دقت کشاورزی نیاز به تکنولوژی و ابزارهای بهتر برای پردازش داده ها به صورت موثر با هزینه معقول و انتقال داده ها به تصمیم گیری ها و اقدامات بهتر در یک زمینه دارد. ما یک چارچوب برای سیستم های تصمیم گیری مبتنی بر ابر و سیستم های اتوماسیون که می توانند داده ها را از منابع مختلف به دست آورند تنظیم کرده ایم ، سنتز تصمیم های خاص برنامه ، و دستگاه های کنترل میدان از ابر را توسعه دادیم. یکی از ویژگی های متمایز چارچوب ما، معماری نرم افزار قابل گسترش آن است: ماژول های تصمیم گیری می توانند برای یک عملیات خاص اضافه شوند و یا پیکربندی شوند. این پلتفرم دارای ظاهرا یک دستگاه-آگنوستیک است که می تواند داده های ورودی را در قالب ها و معانی مختلف پردازش کند. در نهایت، این پلت فرم شامل کنترل نرم افزاری است . یک پارادایم طراحی جدید نرم افزاری است که ما پیشنهاد کردیم تا کنترل های همه جانبه و ایمن دستگاه های زمینه را از یک پلت فرم محاسبات ابری ، کنترل کنیم. یک نسخه اولیه از این سیستم با حمایت USDA توسعه و آزمایش شده است.

© 2016, IFAC (فدراسیون بین المللی کنترل خودکار) میزبانی توسط Elsevier Ltd. تمام حقوق محفوظ

است.

**کلید واژه ها:** محاسبات ابری، پشتیبانی تصمیم گیری و اتوماسیون، کشاورزی دقیق، کنترل نرم افزاری.

## 1. معرفی

کشاورزی دقیق یک نوع کشاورزی خاص است، که از فن آوری ها برای اندازه گیری و واکنش به تغییرات بین و درون زمینه در محصولات کشاورزی استفاده می کند. این کار به عنوان یک ابزار حیاتی برای افزایش بهره وری کشاورزی و حفظ منابع طبیعی دیده می شود. "مگز" دقت کشاورزی یک سیستم پشتیبانی تصمیم (DSS) است که به فرآیند رشد کمک می کند و به داده های داخلی و بین زمین پاسخ می دهد. با پیشرفت های تکنولوژیکی اخیر، به ویژه فن آوری های حسگر و خدمات داده آنلاین، کشاورزی به طور فزاینده ای تبدیل به یک عملیات غنی با داده ها می شود. صنایع ویژه صنایع خاص در ایالات متحده، از طریق یک تغییر و تحول از طریق استفاده از فناوری اطلاعات است. با تعریف USDA، محصولات تخصصی عبارتند از "میوه ها و سبزیجات، آجیل درخت، میوه های خشک، باغبانی و محصولات پرورشگاهی (از جمله گلخانه ای)". محصولات تخصصی نیمی از ارزش دروازه تولید محصولات کشاورزی ایالات متحده را به عهده دارند [1] در مقایسه با محصولات خاص، محصولات تخصصی بیشتر و کارآمدتر هستند. این امر به تغییر در زمینه وضعیت کار و بسیاری از عوامل دیگر بستگی دارد. در حالی که بسیاری از عملیات کشاورزی می توانند از یک سیستم پشتیبانی تصمیم استفاده کنند، محصولات مخصوص می توانند سیستم پشتیبانی بهتر بهره مند شوند؛ ولی امروزه اکثر باغ های هنوز هم از یک فرآیند تصمیم گیری سنتی انسان محور استفاده می کنند.

توسعه، یک سیستم پشتیبانی بهتر تصمیم برای صنایع تخصصی، چالش های خاصی را در زمینه فناوری اطلاعات ارائه می دهد. عملیات های تخصصی محصول بسیار فصلی هستند. درخواست های پشتیبانی تصمیم می تواند به شدت در فصول و فصلی نوسان یابد. یکی از چالش ها این است که چگونه با خواسته های نوظهور مواجه شویم در حالیکه خدمات بسیاری در دسترس وجود دارد. علاوه بر این، هر باغچه ویژگی خاص خود را دارد. چالشی دیگر این است که چگونه سیستم پشتیبانی تصمیمی تنظیم کنیم که بتواند برای یک عملیات خاص دوباره تنظیم شود. در نهایت یک عملیات دقیق کشاورزی یک سیستم کنترل نزدیک است که ورودی ها (مثلا حسگرها و سایر منابع داده) را از مزرعه می گیرد و بازخورد ها (مثلا اعمال مزرعه) به مزرعه می دهد. سیستم پشتیبانی تصمیم گیری

سنتی تنها بر بخش اول حلقه یعنی ورودی ها و تصمیم گیری های تلفیقی کنترل و تأکید دارد. یک چالش بزرگ هم چگونگی بستن حلقه کنترل با دستگاه های کنترل مزرعه است به طوری که با خیال راحت با تصمیمات بهینه بگیرد.

با توجه به تعریف رسمی NIST از محاسبات ابری "یک مدل برای دسترسی به شبکه گسترده و راحت و بر اساس تقاضا برای دسترسی به یک پایگاه مشترک از منابع محاسباتی قابل تنظیم (به عنوان مثال شبکه ها، سرورها، ذخیره سازی، برنامه ها و سرویس ها) که می تواند به سرعت در اختیار قرار گیرد و با حداقل تلاش مدیریت یا ارائه دهنده خدمات . محاسبات ابر از تحویل ابزار الهام گرفته شده است. [ 2 ] مدل: محاسبات قدرت ممکن است به صورت تقاضا مانند برق یا گاز تحویل داده شود، در تلاش برای پاسخگویی به تقاضا برای محاسبات مقیاس پذیر، محاسبات اینترنت به شدت در دسترس و مقرون به صرفه است. سرویس محاسبات ابری کاربران را قادر می سازد درخواست و انتشار منابع را براساس تقاضا و تنها برای آنچه که مورد استفاده قرار گرفته است، پرداخت شود. با جمع آوری منابع با هم، سرویس محاسبات ابری معمولاً دسترسی بیشتری نسبت به شبکه های سرورهای معمولی کاربر دارد.

محاسبات ابری همچنین برای حمایت از تصمیم گیری در کشاورزی دقیق برای تولید محصولات تخصصی کاربرد دارد. اول ، کشاورزی دقیق در یک باغ یک عملیات غنی با داده است. یک سیستم پشتیبانی تصمیم باید حجم زیادی از اطلاعات را از حسگرها و سایر منابع را دریافت و آنها را اداره کند. محاسبات ابر مقیاس پذیری لازم برای مدیریت این داده ها را در زمان واقعی فراهم می کند. دوم ، تقاضا برای حمایت از تصمیمات به طور چشمگیری در فصول متغیر است. با ارائه منابع . محاسبات ابری به سرعت می تواند تعداد نمونه های سرور و سایر منابع را بر اساس تقاضا تغییر دهد. در نهایت، سیستم های پشتیبانی از تصمیمات کشاورزی به طور فزاینده ای در اینترنت برای استفاده از دستگاه های متصل به اینترنت [ 4 ] و ایجاد یک انجمن آنلاین میزبانی می شوند. ارائه دهنده خدمات ابری پیچیدگی سخت افزار را اجرا می کند و میان افزار را برای افزایش دسترسی و امنیت نگه می دارد. این توسعه دهندگان بر منطق یک سیستم پشتیبانی مبتنی بر وب تمرکز دارند.

حرکت به سوی پشتیبانی تصمیم‌گیری مبتنی بر ابر، فرصت‌ها و چالش‌های جدید را ارائه می‌دهد. اول، کشاورزی دقیق از انواع حسگرها و منابع داده استفاده می‌کند، که هر کدام ممکن است فرمت و معانی خاص خود را داشته باشند. یک سیستم پشتیبانی مبتنی بر ابر نیاز به مدیریت یک نمایه متنوع از انواع و فرمت‌های مختلف داده دارد. دوم، به طور سنتی یک سیستم پشتیبانی تصمیم نرم افزار خاص است. یک کشاورز ممکن است نیاز به دسترسی به سیستم‌های مختلف برای یک برنامه خاص (مانند آبیاری، لقاح و غیره) داشته باشد. برای ارائه یک تجربه کاربری ساده، یک سیستم پشتیبانی مبتنی بر ابر، باید بتواند برای برنامه‌های مختلف تمديد و پیکربندی شود؛ و در نهایت، توسعه اخیر اینترنت همه چیز (IoT) دستگاه‌های زمینه را از طریق اینترنت پیوند می‌دهد. انتظار می‌رود که یک سیستم پشتیبانی آینده تصمیمی برای کنترل پیشرفت در IoT، به طور ایمن از محیط ابر کنترل شود

در این مقاله چارچوب ما برای پشتیبانی از تصمیم‌گیری مبتنی بر ابر و سیستم‌های اتوماسیون (DSAS) و تجربه ما در اجرای آن در DSAS, Agrilaksy و ما از زمین تا قبل از استفاده از مقیاس پذیری و در دسترس بودن یک پلت فرم محاسبات ابر بحث می‌کنیم. ما تکنیک‌های جدیدی را برای رسیدگی به چالش‌های طراحی با DSAS مبتنی بر ابر طراحی کردیم. بقیه مقاله به شرح زیر برگزار می‌شود: بخش 2 خلاصه‌ای از چارچوب ما را ارائه می‌دهد. در بخش 3 ما در مورد ظاهر واردات داده‌ها در دستگاه-آگنوستیک بحث می‌کنیم که با منابع مختلف داده با فرمت داده‌ها و معانی شناخته شده تعریف می‌شود. در بخش 4 ما معماری نرم افزار قابل گسترش برای ماژول‌های تصمیم‌گیری را مورد بحث قرار می‌دهیم. در بخش 5 ما پارادایم کنترل فرآیندهای نرم افزاری را معرفی می‌کنیم، یک پارادایم جدید طراحی نرم افزار برای کنترل دستگاه‌های فیزیکی در زمینه‌ای از ابر. بخش 6 در مورد اجرای فعلی Agrilaksy بحث می‌کند، سرانجام، در بخش 7 این مقاله را به پایان می‌رساند.

## 2. بررسی اجمالی سیستم

شکل 1 نمای کلی چارچوب ما برای DSAS های مبتنی بر ابر را نشان می دهد. این دستگاه ظاهری وارداتی اطلاعاتی دارد. قسمت جلویی از مدل داده ها برای تعریف فرمت ها و معنانشناسی مجموعه داده های ورودی استفاده می کند. جزئیات مدل داده در بخش 2 ارائه شده است. ماژول های تصمیم گیری آن با یک معماری نرم افزاری قابل برنامه ریزی طراحی شده است که شامل یک طراحی مدولار سلسله مراتبی می باشد. یک ماژول، که به عنوان یک برنامه وب در DSAS نامیده می شود، توسط رابط های آن مشخص می شود. یک برنامه وب جدید می تواند به سلسله مراتب تصمیم گیری بر روی پرواز اضافه شود یا برای جایگزینی یک ماژول موجود با یک رابط سازگار استفاده شود. جزئیات معماری نرم افزار قابل گسترش ما در بخش 3 معرفی شده است. در بخش 4، کنترل نرم افزاری تعریف شده، یک پارادایم جدید طراحی نرم افزار برای مدیریت پیچیدگی کنترل نمایه متنوع دستگاه ها را معرفی می کنیم. کنترل تعریف شده توسط نرم افزار، مجازی سازی دستگاه های فیزیکی از طریق لایه های انتزاعی است. هر لایه جزئیات اجرایی لایه زیر را خلاصه می کند، در حالی که یک رابط کنترل، کنترل را به لایه بالا ارائه می دهد.

## 3. وارد کردن اطلاعات مربوط به دستگاه آگنوستیک

کشاورزی دقیق با استفاده از سنسورهای مختلف (مانند سنسور رطوبت خاک، سنسور نیتروژن) و منابع داده (به عنوان مثال، مکان آب و هوا) برای اندازه گیری تغییرات درون و بین مزرعه مشخص می شود. DSAS مبتنی بر ابر نیاز به کار با طیف وسیعی از سنسورها و منابع داده دارد. برای رسیدگی به تنوع داده ها از منابع مختلف، ما یک روش مبتنی بر متد داده را که در ابتدا ارائه شده است گسترش می دهیم. این تکنیک از یک مدل داده برای مشخص کردن نوع قالب و داده یک مجموعه داده ورودی استفاده می کند، مانند دما، رطوبت خاک، و غیره. [ 15 ] مدل متادون مدل داده است. این نوع داده های مجاز برای ستون ها و معانی عملیاتی مربوط به هر نوع داده را مشخص می کند. معنانشناسی های عامل، یک نوع داده خاص عملیاتی که می تواند در نوع داده ها را مشخص کند. به عنوان مثال، معنای عملیاتی برای درجه حرارت .

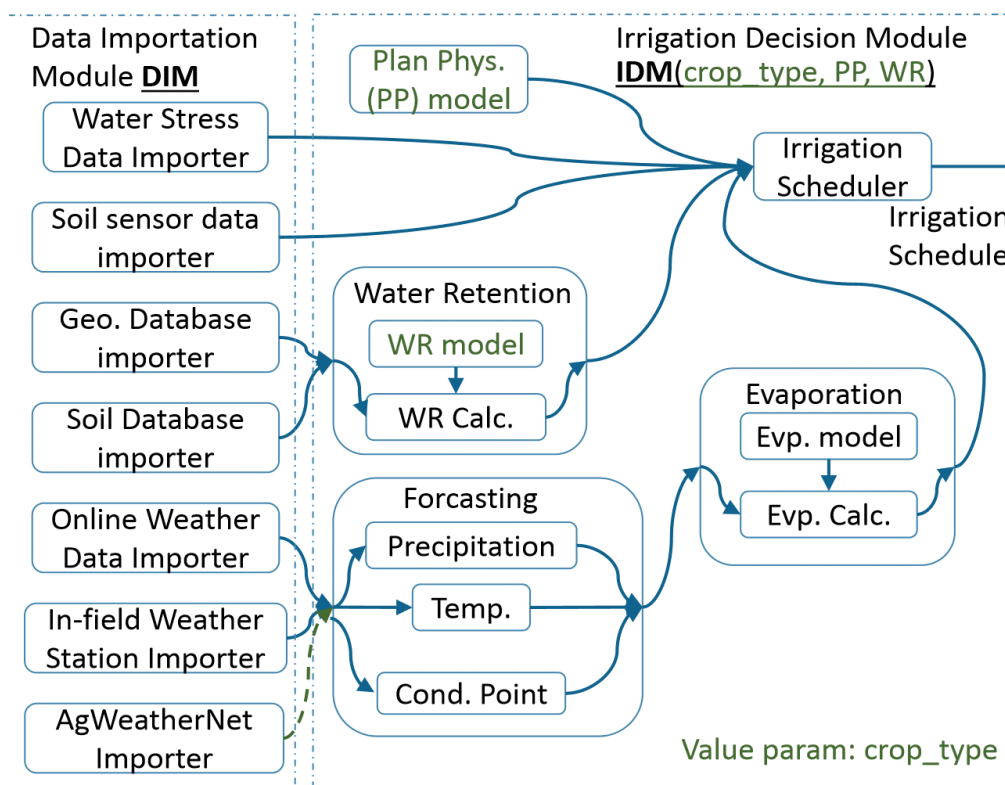
تعریف نحوه ترجمه شکل 2: مدل داده متا در داده های XSD \* به قالب متوسط در سلسیم قرار دهید . برای بهبود قابلیت تعامل، یک مدل داده در XML را مشخص می کنیم [6] که یک فرمت داده غالب است ؛ برای تبادل بیش تر اطلاعات ساختاری از اینترنت استفاده می کنیم . یک مدل داده متادم در یک پسوند طرح (XSD) XML [7] مشخص شده که به عنوان XSD \* نامگذاری شده است XSD \* XSD را با معانی عملیاتی برای نوع داده گسترش می دهد. شکل 2 یک مدل فراداده در XSD پیشنهاد شده را نشان می دهد. عناصر مدل ، با معانی عملیاتی خود (به عنوان مثال 'op') گسترش می یابند. به عنوان مثال، عنصر 'time' دارای date\_seg و date\_single است که هر کدام یک فرمت زمان ممکن را نشان می دهند. زمان "data\_seg" شامل "year"، "month" و غیره است. 'op' مرتبط با 'date\_seq' یک الگوریتم ترجمه در یک زمان date\_seq را به یک نمایش داخلی واحد پذیرفته شده توسط ماژول های تصمیم می دهد. برای هر مدل داده، یک وارد کننده داده برای دسته بندی داده ها تعریف شده که توسط خود مدل داده تولید می شود. وارد کننده داده، معانی عملیاتی را که در مدل داده متا داده شده است، بر روی مجموعه داده ورودی اجرا می کند. این مجموعه داده ها را به یک قالب متوسط که توسط ماژول های تصمیم، پذیرفته می شود، ترجمه می کند. برای معرفی نوع جدیدی از منبع داده (به عنوان مثال نوع جدیدی از سنسور)، تنها باید مدل داده خود را در XML تعریف کرد. DSAS مدل داده XML را به یک وارد کننده داده تبدیل می کند.

#### 4. یک معماری نرم افزاری گسترده برای ماژول های تصمیم گیری

منطق تصمیم گیری هسته ی سیستم پشتیبانی تصمیم گیری است. در یک سیستم پشتیبانی تصمیم گیری سنتی منطق تصمیمات سخت براساس یک برنامه خاص است. با این وجود، عملیات هر باغ متفاوت است و تحت تأثیر بسیاری از عوامل مانند انواع محصولات باغی، محیط زیست و حتی مدل کسب و کار قرار دارد. ما یک معماری نرم افزاری قابل انعطاف پیشنهاد دادیم که DSAS را به راحتی با منطق تصمیم گیری جدیدی گسترش می دهد . معماری نرم افزار به دنبال اصل باز / بستن است ، به همین خاطر یک اصل مهم در طراحی نرم افزار شی گرا بودن

آن است، به عنوان مثال، نرم افزار باید گسترش یابد، اما برای اصلاح هم آماده باشد [8]. چارچوب ماژول های تصمیم گیری، به این معنی است که سیستم را بتوان به راحتی با ماژول های تصمیم گیری جدید و با تغییرات جزئی در اجرای آن توسعه داد. به همین خاطر، یک معماری نرم افزاری قابل گسترش برای ساخت ماژول های تصمیم گیری بسیار قابل تنظیم را ایجاد کردیم. شکل 3 انواع مختلفی از مکانیزم های بازپشتیبانی را که توسط معماری نرم افزار قابل برنامه ریزی باشد، از جمله پیکربندی جریان داده ها و پارامترهای ارزش و بلوک را نشان می دهد.

Figure



شکل 3 قابل انعطاف است و معماری نرم افزاری جریان داده با داده های قابل گسترش را نشان می دهد یک کاربر با روش بصری می تواند و پیکربندی یک منطق تصمیم گیری را مشاهده کند. در مثال نشان داده شده در شکل 3 یک کاربر می تواند مسیر «for-casting» را دوباره به مسیر وارد کند تا به جای ورودی های وارد کننده agweathernet، وارد کننده ایستگاه هوا در فضا وارد کند. این باعث می شود decodeman با استفاده از اطلاعات آب و هوا از agweathernet شبکه ای از آب و هوا که در WA می باشد استوار است [9] طراحی جریان داده رانده شده نیز باعث رفتار و وارث ساخت و ساز معماری استفاده می کند. archies (همانطور که در شکل 3

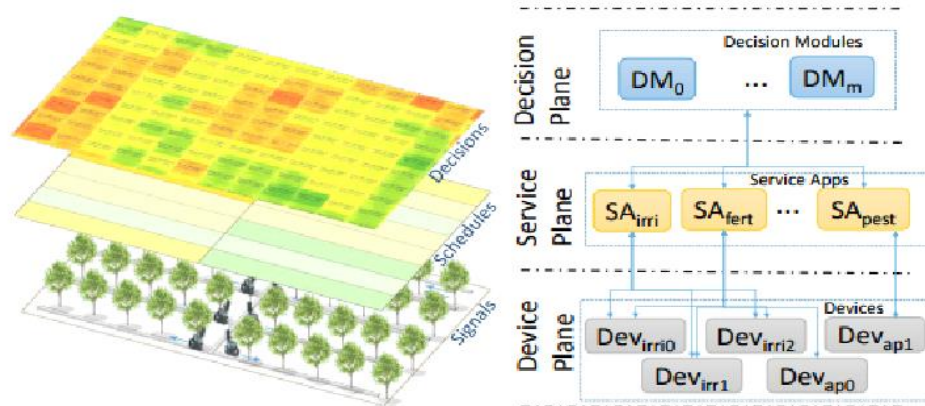
نشان داده شده است) (3) برای کاهش سربار ساخت یک ماژول تصمیم جدید از ابتدا، معماری نرم افزار فقط به کاربر اجازه می دهد که بخشی از منطق را به صورت ضروری از طریق پارامترهای مقدار و بلوک را جایگزین کند. برای مثال، در شکل 3 پارامتر "crop\_type" برای انتقال نوع محصول به ماژول تصمیم استفاده می شود و در یک سلسله مراتب بلوک با استفاده از یک قانون محدوده، انتقال داده می شود.

## 5. کنترل تعریف شده توسط نرم افزار: یک پارادایم طراحی نرم افزاری برای کنترل مبتنی بر ابر است.

در مقایسه با DSS های موجود، یک ویژگی متمایز DSAS مبتنی بر ابر قابلیت کنترل، ستگاه ها به طور مستقیم از ابر آن است. کشاورزان اغلب انواع دستگاه ها را برای برنامه های کاربردی مختلف (به عنوان مثال آبیاری، کنترل آفات). این دستگاه ها با رابط های کنترل اختصاصی خود می آیند. یک چالش در اتوماسیون کنترل مبتنی بر ابر، نحوه کنترل انواع دستگاه های زمینه از ابر است. برای شناخت این چالش، ما ایده کنترل نرم افزار را پیشنهاد کردیم. البته با الهام از مفهوم شبکه تعریف شده توسط نرم افزار [10] کنترل از راه دور نرم افزاری باعث می شود که دستگاه ها با استفاده از یک نرم افزار یکپارچه استفاده کنند و باهم سازگار باشند. این باعث می شود دستگاه ساز انعطاف پذیر باشد (رابط کاربری توسط نرم افزار و نه سخت افزار تعریف می شود) و بدون نفوذ (بدون اصلاح رابط دستگاه، خود لازم است) راه را برای دستگاه های خود که با یک DSAS کار می کنند هموار سازد. کنترل نرم افزاری انطباق پذیری نرم افزاری چند مرحله ای را از تصمیم گیری های مستقل دستگاه برای سیگنال های کنترل خاص دستگاه فراهم می کند. جدا از نگرانی های طراحی، ساختارهای کنترل شده و تعریف شده توسط نرم افزار در یک معماری سه لایه، هر سه لایه باید با هم سازگار (1) یک تصمیم گیری سنتز مستقل از دستگاه. این تصمیمات اقدامات مورد نیاز را در سطح بالا مشخص می کند، مثلاً توزیع جغرافیایی حجم آبیاری؛ (2) یک سرویس دهنده در حال ترجمه یک تصمیم برای برنامه برای هر دستگاه. یک برنامه نشان دهنده پیکربندی و طرح دستگاه است؛ و (3) هواپیمایی دستگاه برنامه را روی آن اجرا می کند. شکل 4، کنترل نرم افزاری را نشان می دهد، به عنوان مثال، با استفاده از یک سیستم آبیاری 3 منطقه ای. هر منطقه آبیاری دارای کنترل کننده خاص خود (Dev<sub>irri1</sub>، Dev<sub>irri2</sub>).



و  $Dev_{irri0}$  به منظور کنترل مجموعه ای از دریاچه ها است. تصمیم گرفته شده توسط هواپیما روش توزیع آب در یک باغ را مشخص می کند. هواپیما سرویس تصمیم گیری برای برنامه های آبیاری را برای هر کنترلر ترجمه می کند و در نهایت هواپیما شامل سه کنترل کننده از برنامه های کنترل شیر می شود.



شکل 4 کنترل نرم افزاری تعریف شده: یک نمای کلی

لایه های انتزاعی که توسط کنترل نرم افزار تعریف شده اعمال می شود، هر لایه را قادر می سازد تا یک نمایه انتزاعی از لایه زیر را از طریق یک رابط به خوبی تعریف کند. به عنوان مثال، یک سرویس هواپیما دارای مجموعه ای از سرویسهای (SA) است. هر SA یک انتزاع از سرویس ارائه شده توسط دستگاه های زمینه فراهم می کند. در شکل 4،  $SA_{irri}$  یک نمایه انتزاعی از یک سیستم آبیاری ارائه می دهد و جزئیات را از قبیل منطقه بندی و انواع دستگاه دور می کند.

کنترل تعریف شده توسط نرم افزار همچنین امکان بازبینی و نظارت مستمر را در هر زمان فراهم می کند. به عنوان مثال، در رابط یک سطح سرویس هواپیمای تصمیم را برای کشف SA در لایه سرویس انتزاع می دهد. هواپیما، برای ارسال تصمیم به SA، و همچنین برای دریافت آمار زمان اجرا از آنها به آنجا می رود. رابط یک صفحه دستگاه، ترافیک دو طرفه را فراهم می کند که در آن هواپیمای سرویس می تواند دستگاه های مزرعه را برنامه ریزی کند و بازخورد را از این دستگاه ها را دریافت کند این رابط اجازه می دهد تا هواپیمای تصمیم گیری از بازخورد دستگاه ها برای تطبیق منطق خود از طریق معماری سازگار استفاده کند.

## 6. پیاده سازی سیستم

نسخه اولیه Agrilaxy در ماه مه 2014 اجرا شده است. که اجزای سازنده ی اطلاعات شناسایی را به پایان می رساند. این سیستم با استفاده از رومی ساخته شد. رومی یک زبان برنامه نویسی پویا است که زبان مورد علاقه توسعه دهندگان وب است. بخشی از محبوبیت آن با جامعه وب به علت ریلز است، رومی یک چارچوب کتابخانه است که به طور خاص برای توسعه وب سرور طراحی شده است [11]. ما Agrilaxy را بر روی سرویس AWS مبتنی بر ابر آمازون آزمایش کردیم [12] و در حال حاضر ما در حال کار بر روی یک نسخه جدید از Agrilaxy، با تاکید بر اجرای یک کنترل مبتنی بر ابر کنترل و ماژول های تصمیم سازگار هستیم .

## 7. نتیجه گیری

محاسبات ابر در حال تبدیل شدن به یک انتخاب اصلی برای میزبانی وب سیستم های اطلاعاتی است. محاسبات ابر یک پلت فرم محاسباتی قابل مقیاس و بسیار قابل دسترس با تخصیص منابع تقاضا را فراهم می کند و با چالش هایی که با سیستم پشتیبانی تصمیم گیری کشاورزی مبتنی بر وب مطابقت دارد و کمک می کند . برای استفاده از محاسبات ابری، و همچنین برای بستن حلقه کنترل در کشاورزی دقیق، ما یک چارچوب طراحی جدید برای سیستم های تصمیم گیری مبتنی بر ابر و سیستم های اتوماسیون پیشنهاد می دهیم. چارچوب ویژگی های ظاهری واردات داده ها به صورت آگاهانه . معماری نرم افزار قابل گسترش برای ماژول های تصمیم گیری و اتوماسیون کنترل مبتنی بر ابر با استفاده از کنترل طراحی نرم افزار . . ظاهرا دستگاه آگنوستیک از داده های ورودی از منابع مختلف داده پشتیبانی می کند . معماری نرم افزاری قابل گسترش، DSAS را برای برنامه های مختلف تصمیم گیری گسترش می دهد . توماسیون مبتنی بر ابر، حلقه تصمیم گیری، از ورودی های سنسور، به تصمیم گیری، برای کنترل سیگنال ها به نمایه متنوع دستگاه های زمینه بسته می شود. در نهایت ما یک نسخه اولیه از چارچوب ما با استفاده از Ruby on Rails اجرا کردیم و آن را بر روی خدمات ابری آمازون (AWS) آزمایش کردیم.

## References

- [1] United States Department of Agriculture, "Specialty Crop Research Initiative." [Online]. Available: [http://nifa.usda.gov/sites/default/files/resources/SCRI\\_Self-Study\\_document.pdf](http://nifa.usda.gov/sites/default/files/resources/SCRI_Self-Study_document.pdf). [Accessed: 14- Jun-2015].
- [2] P. Mell and T. Grance, "The NIST definition of cloud computing," 2011.
- [3] C. Li and L. Y. Li, "Optimal Resource Provisioning for Cloud Computing Environment," *J. Supercomput.*, vol. 62, no. 2, pp. 989–1022, 2012.
- [4] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," *Futur. Gener. Comput. Syst.*, vol. 29, no. 7, pp. 1645–1660, Sep. 2013.
- [5] L. Tan, R. Haley, and R. Wortman, "An Extensible and Integrated Software Architecture for Data Analysis and Visualization in Precision Agriculture," in the proceedings of IEEE International Conference on Information Reuse and Integration (IRI'09), 2009.
- [6] W3C Working Group, "RDF 1.1 XML Syntax," W3C Recommendation. 2014.
- [7] W. C. W. D. December, "W3C XML Schema Definition Language ( XSD )," *Language (Baltim.)*, 2009.
- [8] B. Meyer, "Tell less, say more: the power of implicitness," *Computer (Long. Beach. Calif.)*, vol. 31, no. 7, pp. 97–98, Jul. 1998.
- [9] Washington State University, "AgWeatherNet," 2015. [Online]. Available: <http://weather.wsu.edu/awn.php>.
- [10] D. Kreutz, F. M. V. Ramos, P. E. Verissimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-Defined Networking: A Comprehensive Survey," *Proc. IEEE*, vol. 103, no. 1, pp. 14– 76, Jan. 2015.
- [11] "Ruby on Rails." [Online]. Available: <http://rubyonrails.org/>. [Accessed: 15-Jul-2015].
- [12] Amazon, "Amazon Web Services," 2016. [Online]. Available: <http://aws.amazon.com>. [Accessed: 03-Jan-2016].