Mathematical and Computer Modelling 54 (2011) 944–949

Contents lists available at ScienceDirect

Mathematical and Computer Modelling

journal homepage: www.elsevier.com/locate/mcm

SOA-based precision irrigation decision support system

Xu Liyuan [a,b,1], Chen Liping [b,*], Chen Tianen [b], Gao Yunbing [b]

[a] Capital Normal University, Information Technology Institute, Beijing West 3rd Ring North Road 56, 100048, China

[b] National Engineering Research Center for Information Technology in Agriculture, Beijing Shuguang Garden Road, 11th Beijing Agricultural Building A519, 100097, China

ARTICLE INFO

Article history:
Received 12 August 2010
Accepted 4 November 2010

Keywords:
Precision irrigation decision-making
SOA
UDDI
Web service
Penman–Monteith
Service design

ABSTRACT

At present, irrigation decision-making systems applied to the field of agriculture were mostly aimed at a given area and specific crops. It was difficult to be applied in different areas and different crops. In this paper, a precision irrigation decision-making system has done something to solve this problem. The paper, on the one hand, has synthesized multi-areas and multi-crops in one decision; on the other hand, the key was that it used an advanced design idea, which quickly builds a system by using SOA architecture and fully meets different needs of users based on the maximum reuse of services. The paper arranges services in BPM. First of all, user programmed areas, crop types and the water supply. Secondly, the system selects optimum services from the BPM based on different inputs, and then quickly builds a suitable model. Moreover, it gave precise guidance for crop irrigation processes. The precision irrigation decision support system provides an on-demand decision-making model for agricultural production. Applied results show that, through the test on Xiao Tangshan winter wheat, Da Xing vegetables and rice in central China, the system can quickly build a decision-making model to meet the needs of irrigation for different users. In addition, it shows a good reflection and wide use.

0. Introductions

Decision support systems using databases, human–machine to combine a large number of models realizes scientific decision-making [1–3]. Decision support systems in irrigation management applications began in the early 1990s [4,5]. The developed countries do better in standardization of irrigation decision support systems and current use, and they developed a batch of software for irrigation districts. There are many advanced delegates: the CROPWAT system was an initial single-player based on the DOS environment, then the WISE system supported meteorological data's real time refresh based on an Internet environment, and the SIMIS system which can be applied to various irrigation area gave process from plan, design, water's allocation, maintenance, management and system execution.

In China, water shortage is the basic situation, agricultural water accounts for 70% of the total consumption, but its utilization is only 45% compared with 80% in developed countries. Water consumption per kg of crop is 2–3 times of developed countries [6]. So water-saving agricultural irrigation systems have great potential. The excellent irrigation system abroad is difficult to apply to domestic practical conditions: water shortage and low utilization rates. Therefore, the home also takes up the search for irrigation decision support systems. From the early handmade irrigation water plans to applying programming irrigation management software methods in most areas. Though it lasted 20 years, our country's agriculture irrigation based on the development of software is still in the primary stage [7,8].

* Corresponding author.
E-mail addresses: enhui860227@126.com (L. Xu), chenlp@nercita.org.cnm (L. Chen).

1 Master of The Capital Normal University, Research in Software Engineering and Computer Application System. Tel.: +86 15910669483.

0895-7177/$ – see front matter © 2011 Published by Elsevier Ltd
doi:10.1016/j.mcm.2010.11.020

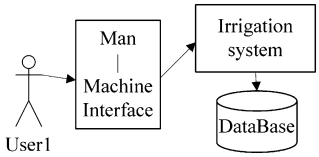

Fig. 1. Framework of the traditional system.

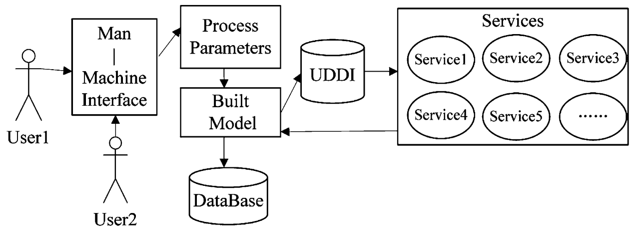

Fig. 2. Framework of the SOA-based irrigation decision system.

At present, irrigation decision-making systems applied in agriculture are aimed at a given area and specific crops, it is difficult to be applied in different areas and different crops [9,10]. In addition, the system is mostly based on networks and results are shown in the form of a web. Business handling logic and decision-making logic are often solidified in code, various processes closely coupled. However agriculture itself is an advanced subject. With the related research methods and means of realization constantly updated, the original treatment method is likely to be eliminated. Therefore, the improvement of code-behind often affects not only consuming financial and material resources, but also causes the system to crash. So, using a framework of flexiblity in decision-making systems is needed; the system should not only implement the basic intelligent decision methods but also easily change related modules without affecting global deployment.

This paper conducts research and implementation on these. Through the study of the Xiao Tang-Shan winter wheat, Da Xing vegetables and the rice in central China, we summarize a crop's characteristics and irrigation regularity, and then design and implement for web service which deploy in BPM for system call. The system's advantage is that it not only synthesizes many areas and various crops on irrigation decisions, but also adopts an advanced design idea; constructing the system with an SOA (Service Oriented Architecture) architecture, provides a decision model interfacing with agricultural production, fully satisfying customer needs on the basis of the maximum service reuse.

1. System design using SOA mode

1.1. Overview of SOA

SOA is a kind of designing style which can share information between heterogeneous systems [11,12]. It solves the software reuse and integration expansion problem in a distributed environment using a new way, that not only improves the efficiency of software development, but also swiftly constructs software components which are open, modular, and reusable; and then accurately and rapidly controls the whole life cycle of the software. SOA is a kind of flexible architecture design idea, and it is also the key technology challenge of flexible business requirements for enterprises [13,14]. Its core concept is service [15]. Some resources in the frame are designed as services which run independently, and these services can be used by other members of the network. SOA provides a loose coupling between packaged services and the means of communication based on news, making services' internal modification and service calling more convenient. It is a software reuse mechanism which is more active, free, and efficient. At present, many international well-known enterprises have used SOA in their product design, such as IBM's Smart SOA, Oracle, HP, etc.

1.2. SOA application in precise irrigation decision support systems

As shown in Fig. 1, the framework applied in irrigation systems currently, aims at developing crop irrigation systems for a given region and specific conditions. Through the man–machine interface, the system only accepts input of the same kind by users, and then guides irrigation strategy by the single decision model of the irrigation system.

By contrast, precise irrigation decision support systems adopt advanced SOA framework ideas to realize the irrigation decision-making process, connecting resources according to the needs. It designs function modules during the process of irrigation as services which can be called by this system and also used by other members of the network. The service is put in a precision agriculture integration platform service registration center (UDDI). Users provide parameters, then the system disposes the parameters simply and chooses a decision-making model automatically. After that, calls corresponding to services from the UDDI, at last return the decision results to the user. In addition, based on the different areas and different crops, this paper implements corresponding services. The system no longer accepts the one region and specific crops, as shown in Fig. 2.

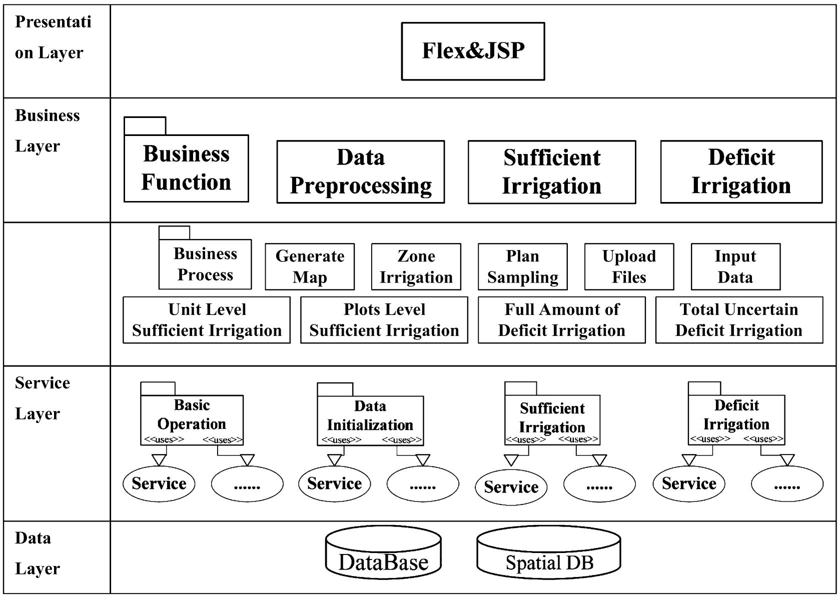

Fig. 3. System level designs.

1.3. System level design using the SOA mode

The system is mainly divided into four levels, the presentation layer, the business layer, the service layer and the data layer, as shown in Fig. 3. The main functions of all levels are as follows:

- Presentation layer: Provides users with a friendly man–machine interaction, handles users and system's interaction, receives user's input, and displays decision-making information.
- Business layer: Processes business logic, composes business by several business processes, business function calls business coarse granularity services.
- Service layer: Coarse granularity services call some fine granularity services that eventually import the business process model, and integrate it with the corresponding business process.
- Data layer: Uses a relational database handling, saving data including spatial data and usual data.

We can see that this paper's major function is sufficient irrigation decision-making and insufficient irrigation decisions. It is mainly based on the limit of water. They use different models in decision-making: sufficient irrigation refers to high yield water supply according to the soil moisture and weather conditions in a crop's growth period, and the deficit irrigation decision-making allocates the limited water reasonably in the crop growth stage, both the decision-making processes are aimed at achieving maximum efficiency.

1.4. The storage container of service—UDDI

In this paper, UDDI (Universal Description Discovery and Integration) is a container used to store and find services. It is developed independently by a precision agriculture project team, and has been deployed in precision agriculture integration platform and put into use. It is mainly used for integration platform service publication, inquiry, subscription, etc., and supervises the life cycle of services from establishment, maintenance to retirement. It is an important way to realize service management. The UDDI is an open service registration center, not only to store services from this system, but also the public services and other services. It can inquire about services quickly and accurately when needed, and then a build decision-making model at any time, and greatly improving availability and service reuse.

2. Service design

2.1. Service design principles

One of the most important service design principles is the service's proper size. Service granularity is the size of the service function [16]. Service granularity directly affects the quality of service, including flexibility and efficiency and so on. Therefore, selecting appropriate size in designing service is very important. How to identify and design a moderate size of service? We are mainly concerned about the following three aspects: reusability, flexibility and performance [17].

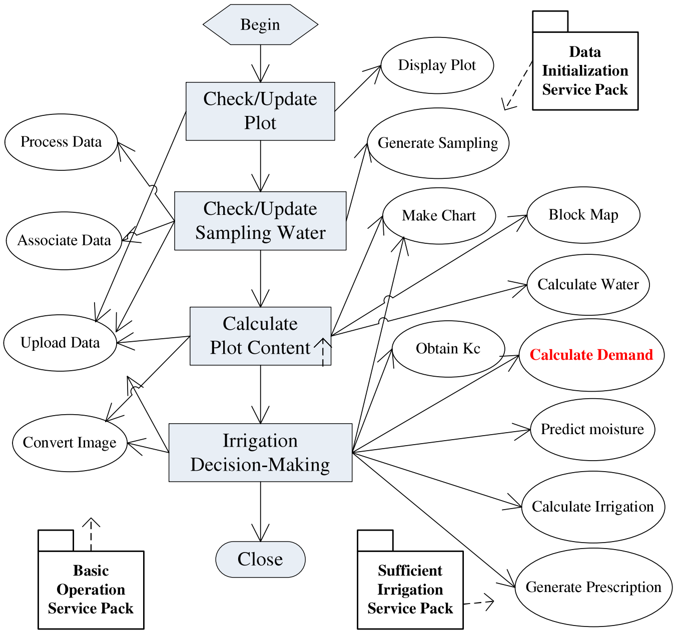

Fig. 4. Service invocations in sufficient irrigation flow.

So-called reusability refers to the ability that can be applied in the process of different departments. Reuse is the core thinking in SOA. Reuse can reduce maintenance cost, and shorten the delivery cycle. The size of the granularity directly affects the reusage of service. Flexibility refers to the ability that can be easily changed when needed. As is known to all, fine granularity service makes the functions contained simple and easily assembled, supplies more flexibility for delivering new functions or changing the business processes. Although individual service efficiency is high, but from the business sense, completing a business task requires more service calls, so the corresponding times of requesting and responding will increase. However, the bigger the granularity size, the more functions it includes, the business logic is more complex, and then it is hard to modify codes.

Therefore, it is necessary to compromise on both sides. SOA often uses a coarse granularity interface (large) as the best practice of external application integration. Combination of services and its layout can be used to create a coarse granularity interface composed by a business process service of fine (small) granularity. In order to solve this problem, SOA often adopts a multi-granularity service design method.

In addition, there are other design principles such as: maximizing the consumption ability of service naming, maintaining the principle of cohesion and integrity, encapsulating detail, not having an interface state and so on. It will be involved in the design service next.

2.2. *Detailed design for service*

A unit level sufficient irrigation process, includes a water demand service, soil moisture content services, irrigation forecast services and more than 30 services. These services are a unified deployment in BPM, and registered in the UDDI so that more users can find and use these services. Next, take the water demand service for example. We will try to introduce it in detail.

Fig. 4 is a unit level sufficient irrigation process, mainly through the four stages: check/update plot, check/update sampling water, calculate plot content and irrigation decision-making. Because each phase involves many services, in order to manage the service better, we put these in three packs: basic operation service pack, data initialization service pack and sufficient irrigation service pack. The water demand service is in the sufficient irrigation service pack, as shown in Fig. 4.

2.2.1. *Service design*

The most important aspect in precision irrigation is accuracy and timeliness. It requires us to calculate the water demand accurately. The universal method, which uses calculation ET_0, obtains the water demand of each stage. This method firstly uses the Penman–Monteith approximation which the FAO (Food and Agriculture Organization) recommends for calculating ET_0, and is then based on crop coefficients K_c to obtain water demand. Use the formula, $ET = K_c * ET_0$. Table 1 displays the import and export parameters.

Table 1
Import and export parameter design of the water demand service.

Service name	Calculate water demand		
Function	Using Penman–Monteith approximation calculate ET_0, and then calculate water demand with $ET = K_c * ET_0$		
Import	Parameter name	Type	Description
	Date	Date	Date
	Daily maximum temperature	Float	Daily maximum temperature, unit is °C
	Daily minimum temperature	Float	Daily minimum temperature, unit is °C
	Average temperature	Float	Absolute humidity, unit is °C
	Average relative temperature	Float	Average relative temperature, unit is %
	Minimum relative humidity	Float	Minimum relative humidity, unit is %
	Sunshine duration	Float	Sunshine duration, unit is h/day
	Altitude	Float	Unit is m
	Wind velocity of weathercock	Float	Unit is m/s
	Altitude of weathercock	Float	Unit is m
Export1	ET_0	Float	Unit is mm
Import2	K_c	Float	Crop coefficients
Export2	ET	Float	Unit is mm

2.2.2. Service implementation

In this system, services are implemented using JAVA. Adopt the local call way in modules, and binding web service way between modules. These connections are written in configuration files; we do not need to realize them in a class. Next, we will give the program codes for the water demand service.

```
   Public double ET (double ET0, double Kc) /*calculate ET*/
   {
   return ET0 * Kc;
   }
   Public double ET0 (ArgImpl arg) /*calculate ET0*/
   {
   double et, x1, x2, x3, x4;
   x1=5.08*Math.pow (10, 7+8.5*(arg.T-273)/arg.T)/ (arg.P*arg.T*arg.T);
    x2 = 0.75 * arg.Ra * (arg.a + arg.b * arg.n / arg.N) - 2 * Math.pow (10, -9)*
Math.pow (arg.T, 4) * (0.56 - 0.079 * Math.pow (6.1 * Math.pow (10, 8.5 *(arg.T - 273) / arg.T) * arg.r, 0.5) * (0.10 + 0.90 * arg.n / arg.N));
    x3 = 0.26 * 6.1 * Math.pow (10, 8.5 * (arg.T - 273) / arg.T) * (1 - arg.r) *(1.0 + arg.C * arg.U);
   x4 = x1 + 1.00;
    et=(x1*x2+x3)/x4;
    x1 = 4098*0.611*Math.exp (17.27*arg.T/(arg.T+237.3))/Math.pow(arg.T+237.3,2);                     return et;
  }
  Public double KC (String name, String baseid, String userid, String date) /*obtain Kc*/
  {
    double kc=-1;
    try{  ManageConfigurationFile MCF=new ManageConfigurationFileImpl ();
          DBImpl db=new DBImpl ();
          db.Connsql (MCF.GetOneConf ("sqlserverconf add"), MCF.GetOneConf ("sqlserverconf db"),
MCF.GetOneConf ("sqlserverconf user"), MCF.GetOneConf ("sqlserverconf pwd"));
          String sql="{call GetCropKc ('"+name+"','"+baseid+"','"+userid+"','"+date+"')}";
          CallableStatement cs=db.conn.prepareCall (sql);
          ResultSet rs=cs.executeQuery ();
          while (rs.next ()){ kc=rs.getDouble(1); }
          rs.close ();
    cs.close ();
    db.Closesql ();
    } catch (Exception e) {System.out.println (e.toString ()) ;}
    return kc;
   }
```

3. System applications

This paper has developed all services in BPM. The decision support system selects the adequate service according to user input from the different BPMs. Then it sets up the model quickly. Finally, it gives accurate instructions in irrigation. The system implementation effect is shown in Fig. 5. Compared with the traditional software, the SOA-based precision irrigation decision support system has advantages on the following two aspects:

(1) It fully integrates existing resources, avoids repeated development and saves development cost.
(2) It establishes flexible processes with the application of SOA and service, and fully embodies the characteristics of an agile business.

And the services of the system have deployed on UDDI for other user calls, so there is improved service reusability.

4. Conclusions

This paper studied the development direction of a precision irrigation technique, analyzed the existing irrigation decision software, and then put forward the advanced concepts of SOA which have been applied in the agriculture industry successfully. Take for example the typical process—precision irrigation unit level sufficient irrigation, we start from irrigation

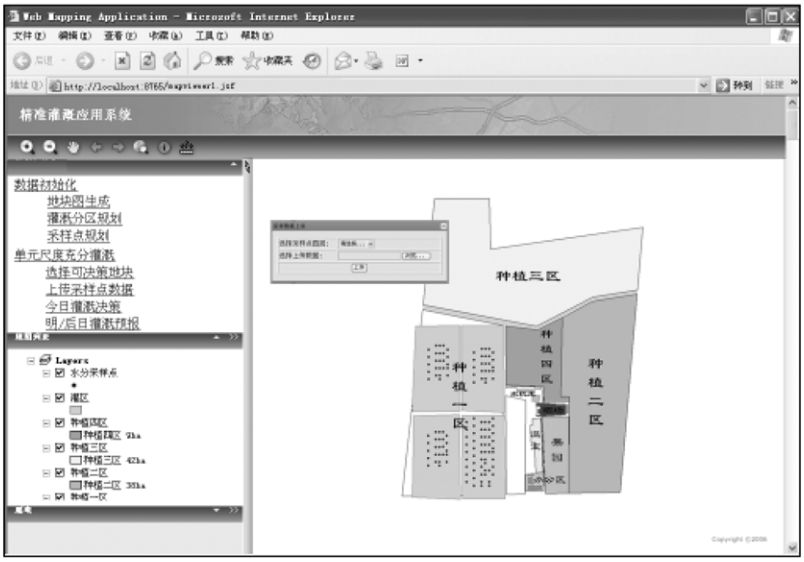

Fig. 5. Precise irrigation system implementation effects.

flow analysis, emphatically discuss the design principle of service, and finally present a design idea and achievement. Through the Xiao Tang-Shan winter wheat, Da Xing vegetables and rice in central China, the system can quickly build a decision-making model to meet the needs of irrigation for different users. In addition, it shows a good reflection and wide usability. Unfortunately, due to the short time available, we only focused on services and the system assembly model, and did not thoroughly ponder about the security of the system and the optimal design of the database. This will be dealt with in a further study.

Acknowledgements

This work is supported by the National science and technology support plan of the "11th five-year plan", "The core technology research and product development of agriculture information collection in tropical regions", ID: 2009BADA1B01 and the State plans to important issue 863" Precision agriculture technology integration platform research and development", ID: 2006AA10A306.

References

[1] Weiwei Zhang, Xifeng Liu, Hongmei Gu, et al., Development on irrigation real-time irrigation decision support system software, Yellow River 29 (4) (2007).
[2] Xiaohong Song, The research and development of decision support system, Computer Knowledge and Technology 36 (2008).
[3] Ruiling Zhang, Yinli Zhang, Application of information technology in precision agriculture, Journal of Anhui Agricultural Sciences 35 (6) (2007).
[4] Jihong Xia, Zhongmin Yan, Mingyao Zhou, et al., Design and application of farm irrigation decision-making support and control system, China Rural Water and Hydropower (8) (2001).
[5] Liping Chen, Chunjiang Zhao, Xiaohong Du, Design and implementation of intelligent decision support system for precision agriculture, Transactions of the CSAE 18 (2) (2002).
[6] Risheng Ding, Development and application of nun-full irrigation management information system for north areas, Northwest Agriculture and Forestry Science and Technology University, 2005.
[7] Mingjun Qin, Precision irrigation and analysis of its prospect, Water Resources & Hydropower of Northeast China (2) (2009) 38–41.
[8] Ding Zhou, Chengbei Zhou, Precision irrigation and existing problem, GX Water Resources & Hydropower Engineering (2) (2009) 77–79.
[9] Dajiang Liu, Jinxiang Feng, Analysis of precision irrigation and its perspective, Water Saving Irrigation (1) (2006) 43–44.
[10] Jinyao Luo, Research progress of water saving irrigation theory and technique, WuHan, 2003.
[11] Peng Chen, Guangyao Li, Research on the software development based on service oriented architecture, Journal of Henan University of Science & Technlogy (Natural Science) 27 (5) (2006).
[12] Dirk Krafzig, Karl Banke, Drik Slama, Enterprise. SOA: Service-Oriented Architecture Best, Prentice Hall PTR, 2007.
[13] IBM Corporation, SOA and Web services [EB/OL], (2007204225) [2008204201]. http://www.ibm.com/developerworks/cn/webservices/newto/.
[14] E.R.L. Thomas, Service-Oriented Architecture: Concepts, Technology, and Design, Prentice Hall PTR, 2007.
[15] Shaomin Xing, Bosheng Zhou, Research progress of SOA, Computer Science 9 (2008).
[16] Junqing Zhang, Strategies and methods used to build enterprise applications based on SOA, Inner Mongolia University, 2006.
[17] Hui Peng, Troubled with service granularity, Information Systems Engineering (1) (2007).