

ON THE USE OF WEB SERVICES FOR BUILDING DISTRIBUTED AUTOMATION SYSTEMS

MITKO SHOPOV, HRISTO MATEV, GRISHA SPASOV

Abstract. The paper discusses the applicability of Web services for building distributed automation systems and the benefits they bring. An adaptation of some of the well-proven enterprise architecture models and their applicability in distributed measurement is presented. The presented sample implementation is based on open and standardized approaches – high-level programming languages, object-oriented platforms, Internet technologies, and standardized communication interfaces.

Key words: distributed automation, embedded systems, multi-tier models, web services.

ПРИЛОЖЕНИЕ НА WEB УСЛУГИТЕ В ИЗГРАЖДАНЕТО НА РАЗПРЕДЕЛЕНИ СИСТЕМИ ЗА АВТОМАТИЗАЦИЯ

1. Въведение

Тенденция от последните години е да се избяга от използването на частни, затворени хардуерни и софтуерни платформи за изграждане на системи за разпределена автоматизация и да се премине към отворени и стандартизирани подходи и решения. В това отношение проектирането на подобни системи е силно повлияно от използването на езици за програмиране от високо ниво, обектно-ориентирани платформи, Интернет технологии и стандартни комуникационни интерфейси [1, 2, 7].

Допълнителна подкрепа осигурява бързият темп на развитие на хардуера, осигурявайки пазара с множество нови вградени системи с интегриран TCP/IP стек, вграден Web сървър и продължително увеличаващи се изчислителни възможности. Моделите, прилагани в разпределената автоматизация, въпреки своите специфични изисквания, следват доказаните в бизнес среда модели за обмен на данни. Специфичността на разпределените системи се проявява в две основни насоки – специфичните изисквания към актуалността на данните и ограничените ресурси на вградените системи [5, 8].

Внедряването на web услугите в системите за разпределена автоматизация е сравнително нова и интересна област за изследвания. Авторът на [7] предлага модел за адаптиране на web технологиите в индустриалната автоматизация, в който функциите на вградените системи са представени като услуги от сървър работещ в междинния слой (Remote Service Server). В [6] авторите разглеждат два метода за реализиране на отдалечено управление на газов хроматограф – с CGI приложение и посредством web услуга. Текущия проект SIRENA [1] има за цел създаването на рамка за да се улесни

изграждането и внедряването на SOA архитектурата и в частност web услугите в индустриалната автоматизация, в следващото поколение вградени системи, както и приложенията и услугите, които те предлагат [1].

В настоящата статия се разглеждат популярни модели за изграждане на разпределена автоматизация базирани се на използването на TCP/IP протоколния стек, многослойни архитектури и Web технологии. Предложена примерна реализация използва четири-слоен модел [3] и стандартизиран интерфейс на Web услугите [4] за комуникация между средните слоеве на системата.

2. Модели за разпределена автоматизация

Сред моделите за изграждане на разпределена автоматизация, най-голям е дялът на трислойните архитектури. Те се характеризират с различни реализации на междинния слой [5]:

- **Модел със сървър за транзакции** – сървърът за транзакции има за цел да разпределя заявките към мрежата от контролери. Той може да осъществява групиране на еднотипните заявки и приоритизиране на отделните заявки.
- **Модел със сървър за отдалечени услуги** – функциите на отделните контролери се представят като услуги привързани към софтуерни компоненти на междинния сървър. По този начин клиентът комуникира с компонентите на сървъра, а не директно с контролерите [7].
- **Модел с Web сървър** – една от най-разпространените реализации поради популярността на сървърните технологии. Базира се на динамични HTML документи и скриптове изпълнявани на сървъра.
- **Трислоен модел с Web услуги** – Този модел залага на универсалността на описанието на данните чрез XML и универсалността на предаване им чрез HTTP. Функциите на вградените системи се представят чрез Web услуги описани на стандартен език – WSDL.

Всеки от гореизброените модели притежава различни предимства и недостатъци за различни конкретни приложения. Представения в [3] многослоен модел има за цел да обедини предимствата на отделните модели. Този модел се състои основно от четири слоя отделени по функционалност и администриране (фигура 1).



Фиг.1. Четири-слоен модел за разпределена автоматизация.

Функциите на отделните слоеве са [3]:

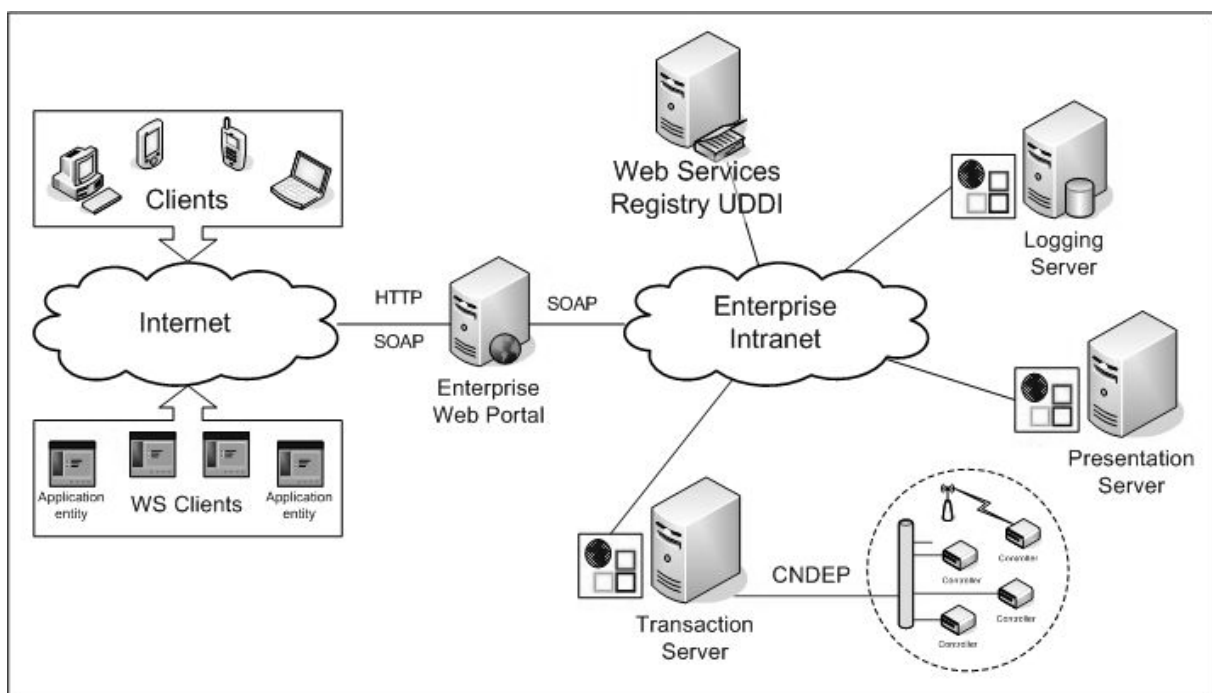
- **Клиентски слой** – На този слой са разположени различните клиенти на системата. Те отправят заявки за услуги към системата посредством стандартен Web браузър или чрез Web услуги. Отговорът на системата зависи от платформата на клиента. (PC, PDA, cellphone).

- **Презентационен слой** – Този слой е отговорен за обработването на заявките и формирането на отговорите. Заявките се анализират и препращат към подходящ сървър от слоя за услуги. Сървърът работещ на този слой се нарича Web портал и е базиран на портал технологията.
- **Слой за услуги** – Този слой дава основната функционалност на системата. Различните услуги на този слой работят на отделни машини следователно срыв на един сървър се отразява само на конкретната услуга, която той предлага. Този модулен подход позволява гъвкавост и надеждност на системата.
- **Даннов слой** – Ролята на този слой е да продуцира и съхранява данни. Конкретната му реализация зависи от слоя за услуги. При услуга за събиране на журнална информация се представя от база данни. В случай на автоматизационна услуга – от мрежа от контролери.

3. Експериментална реализация на система за разпределена автоматизация

В тази точка е представена експериментална система за изграждане на разпределена автоматизация, реализирана върху представения в предишната точка четирислоен модел. Архитектурата на системата (фигура 2) е изградена от следните компоненти:

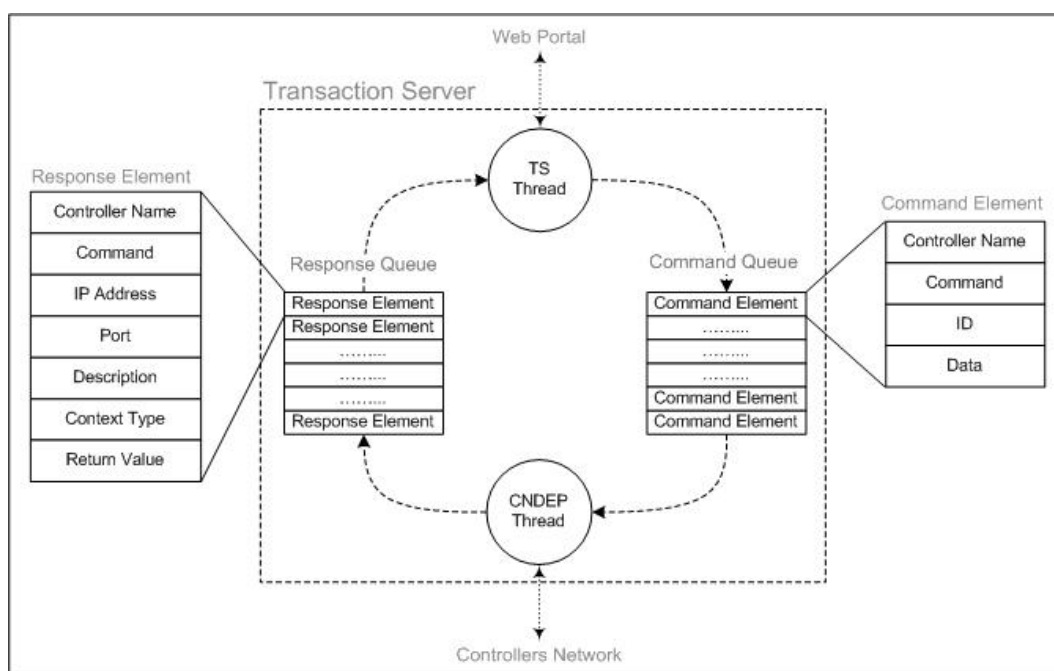
- мрежи от контролери със сензори за мониторинг на температура и влажност. Използваните вградени системи са съответно DS TINI на Dallas Semiconductors [13] и IPC@Chip SC12 на Beck IPC [12].
- сървър за транзакции – реализиран чрез Apache Tomcat 5.5.14 servlet контейнер и Apache Axis работещи на машина със следните параметри - 333MHz Pentium II, 256 MB памет и ОС Debian Linux 2.4.27-2-386.
- сървър за събиране на журнална информация – реализиран чрез Apache Tomcat 5.5.14 servlet контейнер и сървър за управление на база данни MySQL 5.0.22.
- услуга за изчертаване на графики [10]
- UDDI регистър – реализиран чрез Microsoft UDDI Services.



Фиг. 2: Обща архитектура на системата.

3.1. Сървър за транзакции

Сървър за транзакции (transaction server) представлява една от възможните реализации на слоя за услуги (Service tier). Неговата роля е да осъществява свързващо звено между предлаганите от система за разпределена автоматизация услуги и данните, които постъпват от мрежата от сензори. Сред възможните му приложения са осъществяване на контрол и измервания във фабрики (plant control and monitoring) и автоматизиране на дома и офиса. В предложената реализация, сървър за транзакции получава заявки за изпълнение на определени услуги от представителния слой (presentation tier), които заявки трансформира в серия от транзакции към слоя за данни. Комуникацията с представителния слой и останалите сървъри на същия слой се осъществява посредством web услуги. В конкретния случай, слоя за данни е представен от мрежа от контролери, комуникацията с които се осъществява чрез специално разработен протокол – CNDEP [9].



Фиг. 2: Вътрешна архитектура на сървъра за транзакции.

Структурата на сървъра за транзакции включва два компонента (нишки), които комуникират помежду си посредством синхронизирани опашки (фигура 3). Компонентата 'TS thread' се използва за предоставянето на функционалността на мрежата от сензори чрез Web услуги. За различните web услуги се генерират различни последователности от команди към слоя данни. Компонентата 'CNDEP thread' има за цел да осъществи изпълнението на командите над мрежата от контролери. В конкретния случай е използван протоколът CNDEP. Подмяната на протоколът за комуникация с контролерите ще изисква единствено подмяната на тази компонента.

3.2. Сървър за събиране на журнална информация

Предназначението на този модул е да събира и съхранява и предоставя журнална информация от вградените системи. Възможностите на сървъра за събиране на журнална информация са: заявка за събиране на конкретна информация от мрежа от контролери или конкретна вградена система; регистриране на заявката и стартиране на работна нишка; тази нишка има две задачи – събиране на информацията през зададен

период и съхранението и в база данни; предоставяне при заявка на информация от базата с данни. Всички тези задачи са реализирани като Web услуги. Извличаната от базата данни информация е XML кодирана, което улеснява по-нататъшното и обработване.

3.3. Услуга за изчертаване на графики

За изчертаване на графики се използва web услугата представена в [10]. Тази услуга получава серия от данни и ги организира в таблици или графики. Таблиците се съхраняват в comma separated values (.csv) формат, а графиките като картинки (bmp формат). Построяването на графиката се извършва от Graphic Server .NET 3.0 компонента от Graphic Server Technologies. Графиките могат да се синтезират за различни данни получени от Web-услугата за регистрация.

3.4. Откриване на услугите

Един от популярните методи за откриване и локализиране на услуги е използването на централизиран справочник. При него се използва отделен сървър за съхранение на информацията за наличните отдалечени услуги. Популярни технологии за реализиране на централизиран справочници са Jini, Universal Plug and Play (UPnP) и Universal Description, Discovery, and Implementation (UDDI). От тях последния е част от стандартите за Web услуги.

Стандартът UDDI дефинира правилата за взаимодействие с централизирания регистър и формата на записите в него. Операциите над регистъра са за регистриране и за търсене на информация. Съществуват два типа UDDI регистри: публични и частни. Публичните са свободно достъпни в Интернет и служат за свободно търсене на Web услуги. Частните UDDI регистри се използват от организации за регистриране и търсене само на частни, използвани изключително от тях или партньорски организации Web услуги [11].

4. Заключение

В статията са разгледани популярни модели за изграждане на системи за разпределена автоматизация. Разгледани са предимствата произтичащи от прилагането на SOA архитектурата и технологията на web услугите в тези системи. Предложена е реализация на четириислоен модел позволяваща географското разпръскване на отделните звена без това да се отразява съществено върху функционалността на цялата система. Реализацията се основава на отворени и стандартизирани подходи, което позволява улеснена интеграция със съществуващите информационни системи както и със следващото поколение програмни системи за управление а автоматизационни контролери.

5. Благодарности

Изследванията в настоящата работа са финансирани от Фонда за научни изследвания към МОН – проект “ВУ-966/2005”, “Интегриране на web услуги и данни в разпределени информационни системи за автоматизация в интернет среда”, договор “ВУ-МИ-108/2005”.

ЛИТЕРАТУРА

1. **F. Jammes and H. Smith.** Service-Oriented Paradigms in Industrial Automation, *IEEE Transaction on Industrial Informatics*, Vol. 1, Issue 1, Feb. 2005, 62 – 70.

2. **M. Fowler.** Patterns of Enterprise Application Architecture, Addison-Wesley Professional, 2002, ISBN: 0-321-12742-0.
3. **N. Kakanakov, M. Shopov and G. Spasov.** A new Web-based multi-tier model for distributed automation systems. *Information Technologies and Control*, Vol. 2, 2006.
4. **N. Kakanakov, M. Shopov and G. Spasov.** Distributed Automation System based on Java and Web Services, *Proceedings of the CompSysTech'06*, Veliko Tarnovo, Bulgaria, June 15-16, 2006, III-A.24-1-6.
5. **Н. Каканакoв.** Web базирани модели за разпределена автоматизация. *Автоматика и Информатика*, бр. 3, 2006, 40-44, ISSN: 0861-7562.
6. **U. Topp, and P. Müller.** Web based service for embedded devices. 2001.
7. **N. Jazdi.** Component-based and Distributed Web Application for Embedded Systems. *International Conference on Intelligent Agents, Web Technology and Internet Commerce*, Las Vegas, USA, 9-11 July 2001.
8. **G. Borriello, R. Want.** Embedded Computation Meets the World Wide Web. *Communications of ACM*, Vol. 43, Issue 5, May 2000, pp. 59-66.
9. **Kakanakov, N., I. Stankov, M. Shopov, and G. Spasov.** Controller Network Data Extracting Protocol – design and implementation. *Proceedings of the CompSysTech'06*, Veliko Tarnovo, Bulgaria, June 15-16, 2006, pp. IIIA-14.1 – IIIA-14.6.
10. **Kakanakov, N.** An Application of Web Services for Distributed Measurement. *Proceedings of the Young Researchers Session, International Symposium on Modern Computing.*, 4-6 October 2006, Sofia, Bulgaria, pp. 55-60, ISBN: 954-91743-5-2.
11. **Alonso, G.** Myths around Web Services. *IEEE Data Engineering Bulletin*, Vol.25, number 4, 2002.
12. <http://www.beck-ipc.com/ipc/> - IPC@Chip Homepage [October 2006].
13. <http://www.maxim-ic.com/products/tini/> - DS TINI Homepage [October 2006].

Department of Computer Systems and Technologies
 Technical University–Sofia, Branch Plovdiv
 25, Tsanko Dystabanov Str.
 4000 Plovdiv
 BULGARIA
 E-mail: mshopov@tu-plovdiv.bg
 hristo.matev@gmail.com
 gvs@tu-plovdiv.bg