UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE CIÊNCIA DA COMPUTAÇÃO

ANDERSON DIDONÉ FOSCARINI

Development of a Context Broker with High Availability resource

Work presented in partial fulfillment
of the requirements for the degree of
Bachelor in Computer Science

Advisor: Prof. Dr. Taisy Silva Weber
Coadvisor: M.Sc. Marcos Rates Crippa

Porto Alegre
June 2015

ABSTRACT

Este documento é um exemplo de como formatar documentos para o Instituto de Informática da UFRGS usando as classes LaTeX disponibilizadas pelo UTUG. Ao mesmo tempo, pode servir de consulta para comandos mais genéricos. *O texto do resumo não deve conter mais do que 500 palavras.*

Keywords: High availability. context broker. context computing. fault tolerance.

Desenvolvimento de Broker de Contexto com Recurso de Alta Disponibilidade

RESUMO

Neste trabalho será definido e desenvolvido um Broker de Contexto, e posteriormente aplicada técnica de Tolerância à Falhas ao mesmo. Serão realizados testes para fins de comparação entre as soluções, e avaliações quando aplicadas a um ambiente crítico, onde a Tolerância à Falhas é fundamental.

Palavras-chave: Alta disponibilidade. Broker de contexto. Computação de context. Tolerância a Falhas.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

CxB	Context Broker
CxC	Context Consumer
CxP	Context Provider
REST	Representational State Transfer
ABNT	Associação Brasileira de Normas Técnicas

CONTENTS

1 INTRODUCTION

Bla (ARSANJANI, 2004) (AVIŽIENIS et al., 2004) (BALDAUF; DUSTDAR; ROSENBERG, 2007) (BARRERA et al.,) (DEY, 2000) (KIAN et al., 2010) (KNAPPMEYER et al., 2010) (SKEEN, 1981) (CRIPPA, 2010)

1.1 Motivation

1.2 Goals

1.3 Organization of Thesis

2 DEFINITIONS

This work proposes the application of a high availability technique to a context broker system. For a better understanding of the system and its development, definitions of Context, Context-Aware System, Context Representation, Fault Tolerance and High Availability related terms are presented.

2.1 Context

Context has had many definitions throughout the years. The first time Context was defined regarding the human-computer interaction was by (SCHILIT; THEIMER, 1994), where context is defined related to location, identities of nearby people and objects, and the changes happening to those. Similarly, (BROWN; BOVEY; CHEN, 1997) define context as location, people around the user, time of day, season, temperature, etc. Many authors have also defined context using synonyms, as (BROWN, 1995) had the idea of "environment", i.e., what the computer knows about the user's environment, or (FRANKLIN; FLASCHBART, 1998) that see context as user's situation. Thus, a lot of definitions have existed, but they all end up being too specific. Context is about the whole situation of an application and its users, and we can't really.

(DEY, 2000) defined context as follows: "Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and application themselves." This is the definition this work uses.

2.1.1 Context-Aware System

Like context, context-awareness has had several definitions over the years. The first definition was done by (SCHILIT; THEIMER, 1994), and it restricted to applications informed about context and applications that adapt themselves to context. Later on, synonyms have been used to define a context-aware system: reactive (COOPERSTOCK et al., 1995), responsive (ELROD et al., 1993), situated (HULL; NEAVES; BEDFORD-ROBERTS, 1997), context-sensitive (REKIMOTO; AYATSUKA; HAYASHI, 1998) and environment-directed (FICKAS; KORTUEM; SEGALL, 1997). All these definitions refer to either using context, adapting to context, or both.

The definition used in this work is provided by (DEY, 2000): ”A system is context-aware if it uses context to provide relevant information and/or services to the user, where relevancy depends on the user’s task.”

2.1.2 How to represent Context

According to (DEY, 2000), context-aware applications deal with the who’s, where’s, when’s and what’s (the activities that are occurring) of entities, and interpret this information to define why a situation is occurring. The designer of the application must decide what to do with the information. Once we have the information available, either through automated sensors or through user’s interference, we need to represent it in a way a machine can process and store it.

(BALDAUF; DUSTDAR; ROSENBERG, 2007) presents and defines the most relevant context modeling options: key-value, markup scheme, graphical, object oriented, logic based and ontology based models. As this work is developed over the same base as (CRIPPA, 2010), it uses the same context representation model: a markup scheme variation, ContextML (KNAPPMEYER et al., 2010). It is chosen as it provides **ver artigo do marcos pq escolheu cxml

2.1.3 ContextML

ContextML is an XML-based representation schema for context information, where it is categorized into scopes and related to different types of entities. It is designed to be used with REST-based communication between the framework components (KNAPPMEYER et al., 2010).

The system consists on three core components: Context Provider, Context Consumer and Context Broker. They are briefly described below, based on definitions made by (KNAPPMEYER et al., 2010).

2.1.3.1 Context Representation

2.1.3.2 Context Provider

A Context Provider (CxP) provides context information of a certain type, e.g. weather, location, activity, etc. It gathers data from sensors, network, user interactions, or other sources.

A CxP is specialized in a specific domain of context information (location, weather etc).

2.1.3.3 Context Consumer

A Context Consumer (CxC) queries for and uses context data, e.g. is a context-aware application. A CxC can retrieve context information asynchronously through a subscription method, or by a synchronous method where it requests the Broker for a specific information or for a particular Provider interface, to query the Provider directly.

2.1.3.4 Context Broker

A Context Broker (CxB) is the central component of the architecture, and is the focus of this work. It handles and aggregates context information, and is an interface between the other architecture components. The CxB allows CxCs to subscribe to context information, and CxPs to provide this information. It also provides a lookup service, where the CxCs can query the CxB for CxPs that have a particular capability, depending on the CxC's interest.

2.1.3.5 Entity and Scope

An entity is the subject of interest which context data refers to, and it is composed of two parts: a type and an identifier. The type refers to the category of the entity: username for human users, imei for mobile devices, room for a room with sensors, etc. The identifier specifies a particular item within a set of entities of the same type.

A scope is a set of closely related context parameters. Every context parameter has a name and belongs to only one scope. The parameters of a scope can only be requested, updated, provided and store at the same time, making the data always consistent. For example, a scope 'position' has latitude, longitude and accuracy attributes; any operation on this scope is performed on all these attributes: if the latitude is updated, so is the longitude and accuracy, what is correct, because otherwise it would not make sense. Entity-scope association is illustrated in Figure 2.1.

2.1.3.6 Context Messages

- A - B

2.2 Fault Tolerance

2.2.1 Dependable System

2.2.2 High Availability

2.2.2.1 Uses in real world

2.2.2.2 Problems involving High Availability

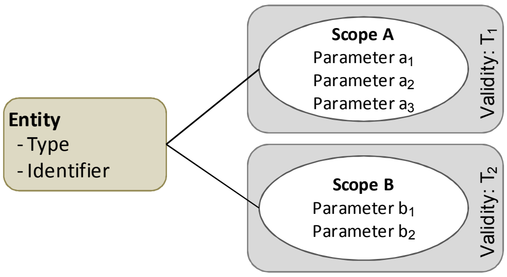

Figure 2.1 – Entity and Scope relationship. Source: (KNAPPMEYER et al., 2010)

3 DEVELOPMENT

Bla.

3.1 Developing a Context Broker

3.1.1 Platform Choice

3.1.2 System Architecture

3.1.3 Interfaces

3.1.3.1 Context Advertisement

3.1.3.2 Context Update

3.1.3.3 Get Context Providers

3.1.3.4 Get Context

3.1.3.5 Subscribe

3.2 Introducing High Availability Technique

3.2.1 Objective

3.2.2 Approach used

3.2.3 Protocol created

4 TESTS

Bla.

5 CONCLUSION

Bla.

5.1 Future Work

REFERENCES

ARSANJANI, A. Service-oriented modeling and architecture. **IBM developer works**, p. 1–15, 2004.

AVIŽIENIS, A. et al. Basic concepts and taxonomy of dependable and secure computing. **Dependable and Secure Computing, IEEE Transactions on**, IEEE, v. 1, n. 1, p. 11–33, 2004.

BALDAUF, M.; DUSTDAR, S.; ROSENBERG, F. A survey on context-aware systems. **International Journal of Ad Hoc and Ubiquitous Computing**, Inderscience Publishers, v. 2, n. 4, p. 263–277, 2007.

BARRERA, J. et al. The design and architecture of the microsoft cluster service.

BROWN, P. J. The stick-e document: a framework for creating context-aware applications. **ELECTRONIC PUBLISHING-CHICHESTER-**, Citeseer, v. 8, p. 259–272, 1995.

BROWN, P. J.; BOVEY, J. D.; CHEN, X. Context-aware applications: from the laboratory to the marketplace. **Personal Communications, IEEE**, IEEE, v. 4, n. 5, p. 58–64, 1997.

COOPERSTOCK, J. R. et al. Evolution of a reactive environment. In: ACM PRESS/ADDISON-WESLEY PUBLISHING CO. **Proceedings of the SIGCHI conference on Human factors in computing systems**. [S.l.], 1995. p. 170–177.

CRIPPA, M. R. **Design and implementation of a broker for a service-oriented context management and distribution architecture**. 2010.

DEY, A. K. **Providing architectural support for building context-aware applications**. Tese (Doutorado) — Georgia Institute of Technology, 2000.

ELROD, S. et al. Responsive office environments. **Communications of the ACM**, ACM, v. 36, n. 7, p. 84–85, 1993.

FICKAS, S.; KORTUEM, G.; SEGALL, Z. Software organization for dynamic and adaptable wearable systems. In: IEEE. **Wearable Computers, 1997. Digest of Papers., First International Symposium on**. [S.l.], 1997. p. 56–63.

FRANKLIN, D.; FLASCHBART, J. All gadget and no representation makes jack a dull environment. In: **Proceedings of the AAAI 1998 Spring Symposium on Intelligent Environments**. [S.l.: s.n.], 1998. p. 155–160.

HULL, R.; NEAVES, P.; BEDFORD-ROBERTS, J. Towards situated computing. In: IEEE. **Wearable Computers, 1997. Digest of Papers., First International Symposium on**. [S.l.], 1997. p. 146–153.

KIAN, S. L. et al. A federated broker architecture for large scale context dissemination. In: IEEE. **Computer and Information Technology (CIT), 2010 IEEE 10th International Conference on**. [S.l.], 2010. p. 2964–2969.

KNAPPMEYER, M. et al. Contextml: a light-weight context representation and context management schema. In: IEEE. **Wireless Pervasive Computing (ISWPC), 2010 5th IEEE International Symposium on**. [S.l.], 2010. p. 367–372.

REKIMOTO, J.; AYATSUKA, Y.; HAYASHI, K. Augment-able reality: Situated communication through physical and digital spaces. In: IEEE. **Wearable Computers, 1998. Digest of Papers. Second International Symposium on**. [S.l.], 1998. p. 68–75.

SCHILIT, B. N.; THEIMER, M. M. Disseminating active map information to mobile hosts. **Network, IEEE**, IEEE, v. 8, n. 5, p. 22–32, 1994.

SKEEN, D. Nonblocking commit protocols. In: ACM. **Proceedings of the 1981 ACM SIGMOD international conference on Management of data**. [S.l.], 1981. p. 133–142.