

A UML Profile for Software Architectures and Peer to Peer Dependable Applications

Aphrodite Tsalgatidou, George Athanasopoulos, Periklis Liaskovitis *

Dept. of Informatics & Telecommunications, National & Kapodistrian University of Athens
(NKUA), Panepistimiopolis, Ilisia 157 84, Greece
{[atsalga](mailto:atsalga@di.uoa.gr), [gathanas](mailto:gathanas@di.uoa.gr), [pliaskov](mailto:pliaskov@di.uoa.gr)}@di.uoa.gr

Abstract. Peer-to-peer (p2p) applications exhibit characteristics such as dynamic network topology and configuration, heterogeneity and scalability that make their development a complicated task. This is further aggravated due to the required satisfaction of dependability properties, i.e. security, availability, reliability, etc. Furthermore, p2p application requirements are driven by the software architecture adopted for their development while in other kinds of applications, the requirements drive the architecture. It is therefore only natural to adopt an architecture-based approach for the development of p2p dependable applications in order to ensure at the architectural level that dependability requirements are met. An essential part for the effective architecture-based development is the use of an appropriate modelling mechanism. In this paper we argue on the use of a UML profile that provides appropriate constructs and notations for modelling dependable p2p software architectures.

1. Introduction

Peer-to-peer (p2p) computing is a form of networking that eliminates the need for servers and leverages computing capabilities, by enabling individual computers to share resources (applications, drive space, processing power, etc.) and to communicate with each other as peers. One of the major challenges for the wide applicability and use of p2p applications, especially in the business domain, is the satisfaction of their dependability requirements, i.e. reliability, security, survivability, responsiveness and availability. This is of paramount importance since p2p applications operate in an open environment such as the Internet where there is no control over the participating nodes or users.

The nature of p2p applications is such, that their software architecture drives their requirements while in other kind of applications the requirements drive the architecture. Therefore, an architecture-based approach provides a good basis for the development of p2p applications as it can facilitate developers to predict the quality of a final p2p application before proceeding to its implementation and to ensure at the architecture level that dependability requirements are met. This requires appropriate

* P. Liaskovitis is currently at the *ECE Dept. of the University of California San Diego, USA*, pliaskov@ucsd.edu

modelling mechanisms and notations that can support the effective modelling of p2p software architectures and their dependability properties.

A popular mechanism for modelling software architectures is the Architecture Description Languages (ADLs), [2], [3], [4] which provide appropriate notations and formalisms to specify software architectures. Their supporting tools and environments provide additional facilities like automated analysis and simulation that leverage the architecture specification process. However, their intrinsic complexity and strict formalism prevent their wide adoption by industry. Another modelling mechanism that can be used is UML which is widely accepted by industry due to its simplicity and flexibility. Furthermore, its intrinsic extensibility mechanism and the extensions provided by OMG [9] make it possible to use it for modelling systems and concepts that were not initially supported.

Therefore, we decided to use UML as the underlying modelling mechanism and to develop appropriate extensions in order to cater for modelling software architectures. As regards the dependability properties of p2p applications, these are modelled by using the QoS UML profile proposed by SINTEF [15] and by providing some UML extensions for aspects not covered by it. The developed UML extensions for software architectures and p2p dependable applications are offered in a UML profile that we call “UML P2P Profile”.

In the following section we describe the properties and characteristics of dependable p2p applications and discuss about the need of architecture-based development for dependable p2p applications. In section 3 we describe the architectural views that are facilitated by the elements that we have introduced as extensions to UML v1.4 [6]. Finally section 4 summarizes our results.

2. P2P Applications and Architecture-Based Development

P2P applications are built around a collection of nodes that are networked together in various configurations depending on the architectural model they adopt. There are two main architectural models adopted by p2p systems: A purely *decentralized p2p model* where all nodes are equal and a *semi-centralized p2p architectural model* where there is at least one peer node which performs some central functions, e.g. indexing, for better serving the rest of the peers. These two main architectural p2p models can be further classified depending on the type of interaction between the participating peers. A detailed report on the various p2p architectural models may be found in [16].

Some key characteristics of most p2p applications are the following: ***symmetric interaction between peers*** since they are simultaneously clients and servers; ***non-deterministic topology*** since the set of peers participating in a p2p network varies constantly with time; ***heterogeneity***, an inherent characteristic in p2p applications since the participating nodes differ in many respects, including communication bandwidth, available memory, etc.; ***grouping*** since peers are usually self-organized into groups (created and evolving dynamically) with a logical, physical, security or context related scope (peers and groups have a many-to-many relation); ***dynamic and virtual allocation of communication paths*** between peers based on factors like

network conjunction or intermediate peers state. There are many other characteristics of p2p applications, e.g. scalability or anonymity, which have been analysed in detail in [16]. Here we focus mainly on the characteristics that need to be modelled and tested during architecture modelling and that we have taken into account during the construction of the UML profile presented in this paper. Along these lines, another crucial characteristic is *dependability*. Dependability is actually the trustworthiness of a system and it depends on the context the system is used, as well as on other system properties like security, availability, responsiveness and survivability [22].

In general, the requirements of a p2p application are driven by the selected p2p architecture, while in other types of systems the requirements drive the architecture. It is therefore only natural to use an *architecture-based approach for the development of dependable p2p applications*. In this way, application requirements can be better clarified and it is possible to ensure at the architectural level that dependability requirements (an essential characteristic for the acceptance and usage of p2p applications) are met.

The specification of the software architecture is an essential step during architecture-based development and the modelling mechanism used for carrying out this step is of paramount importance. Therefore, we investigated thoroughly all the existing mechanisms for modelling software architectures [19] and we decided to use UML v1.4 as the underlying modelling mechanism and to further extend it in order to satisfy our specific needs. The extensions that we have introduced to UML v.1.4 are grouped in a profile which we call “UML P2P Profile”, which is briefly described in the following section. A detailed description of the profile can be found in [19].

3. UML P2P Profile

According to [17], the architectural description of a system is given in a number of architectural views which address the concerns of one or more system stakeholders. The views that we have identified for building the architecture of a p2p application are [19], Logical Network Architecture (LNA), Layered Architecture, Application Architecture and Dependability Properties views.

The primary purpose of the *LNA* view is to enable the systematic description of various network architectures of p2p systems e.g. decentralized or semi-centralized, and behavioural aspects of the application, such as the role of each node in it or the protocol that will govern the communication channels. Specifically, the LNA view provides an abstract model of the network topology of a p2p application and the interactions between all participating peers.

The *Layered Architecture* view is a representation of the application architecture in layers. This view provides an abstract representation of a p2p application capabilities¹ and a way to identify its constituent subsystems.

The *Application Architecture* view depicts the computational and data storage elements of a software application as well as the interactions between them towards

¹ A capability represents abstract functionality that is desirable to be possessed by a system, e.g. encryption/decryption of exchanged messages or joint file editing [23].

achieving specific functionality. In our profile, these aspects are modelled using architectural elements such as those used by Architecture Description Languages [3][18].

Finally, the *Dependability Properties* view enables the specification of non-functional aspects of a system. Dependability properties may be imposed on individual system elements as well as on a system as a whole. Dependability properties can be thought of as Quality of Service (QoS) properties, therefore, in order to model them, we may reuse the existing conceptual models of the QoS frameworks [12][13] and the modelling constructs offered by the proposed QoS UML Profiles [14][15] or the recently accepted by OMG profile for modeling QoS [21].

These four architectural views, not only illustrate various aspects of a p2p application in different levels of detail but they are also interrelated (see Fig 1):

- The *LNA* view depicts the nodes of a p2p application and their interactions
- The corresponding *Layered Architecture* view provides a layered representation of the structure of *each* p2p node type.
- The *Application Architecture* view depicts the structure of a p2p node in a lower level of detail by showing its constituting architectural elements and their structure.
- The *Dependability Properties* view is orthogonal to all these views in the sense that dependability properties apply to all these views.

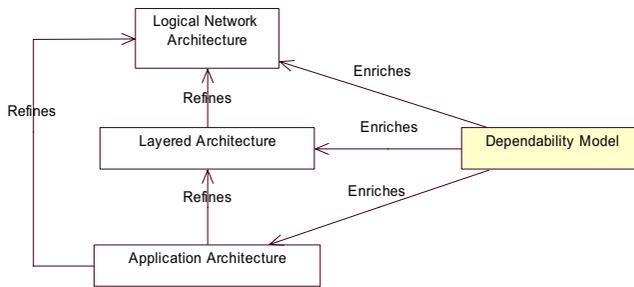


Fig. 1. Architectural Views Relationships

This logical relationship among the architectural views has been reflected in the various modelling constructs that have been produced for their description and are thoroughly presented in [19].

4. Conclusions

The initial motivation of this work was to provide notations and specification primitives in order to support the architecture-based development of dependable P2P applications. After an initial investigation of existing modelling notations for software architectures, we decided to use as a basis the UML v.1.4 (as by that time the UML v2.0 was not available) and to extend it by introducing constructs appropriate for modelling dependable p2p software architectures and applications. The result was a UML profile that can support the modelling of:

- Logical Network Architectures
- Layered Architectures
- Application Architectures and
- Dependability Properties

The first two architectural views exhibit two different facets of the architectural backbone of a p2p application while the third view provides a more detailed description of the application architecture. These views are enriched with the modelling of the dependability properties required by p2p architectures and applications.

We can view the architecture based development of a p2p application in a number of steps that produce various architectural views in an increasing level of detail, starting from the construction of an LNA model, proceeding to the construction of Layered Architecture models and then producing Application Architecture models which provide a detailed architectural view of all subsystems implementing the functionality specified in the Layered Architecture Models. Dependability Modelling is taking place in all these steps.

The provided profile, although it is targeted to p2p dependable applications, it provides all necessary constructs for architecture modelling and thus it can be used for modelling not only p2p dependable architectures but any kind of software architecture.

Finally, we compared our profile with the modelling constructs provided by UML v.2.0 [7][8] and we found out that UML v.2.0 directly supports some of the elements offered by our profile such as “Component” or “Connector”, while other elements, such as “Group” or “Layer” can still not be modelled directly. As regards the dependability concepts introduced in our profile, these are fully addressed by the “UML Profile for Modeling Quality of Service and Fault Tolerance Characteristics and Mechanisms” [21] accepted by OMG in September 2004.

5. Acknowledgement

This work has been supported by the IST Project P2P_Architect (IST-2001-32708)[20]. We would like to thank the rest of the project consortium for their contribution in this work and especially Prof. Ian Sommerville and his group at the University of Lancaster for their work on specification of the dependability properties of p2p applications and layered architectures.

6. References

- [1] “Is UML an Architecture Description Language” OOPSLA99
- [2] Overview Of The Rapide Prototyping Project
<http://pavg.stanford.edu/rapide/overview.html>
- [3] The Acme Architectural Description Language <http://www-2.cs.cmu.edu/~acme/>
- [4] David Garlan. “An introduction to the AESOP system”. Version of 11 July 1995.

- [5] James Walkerdine, Lee Melville, Ian Sommerville, "Dependability Properties of P2P Architectures", Presented in 2nd International Conference on Peer to Peer Computing (P2P 2002), September 2002, Sweden.
- [6] OMG "Unified Modelling Language Specification", Version 1.4, September 2001, available from www.omg.org
- [7] OMG, "Unified Modeling Language: Superstructure, version 2.0", OMG Final Adopted Specification, ptc/03-08-02
- [8] OMG, "Unified Modeling Language Specification: Infrastructure, version 2.0", ptc/03-09-15
- [9] OMG Analysis and Design Platform Task Force, "White Paper on the Profile mechanism", Version 1.0, OMG Document ad/99-04-07 April 1999.
- [10] ENG, LANC, UoA, ATC, "Comprehensive Survey of contemporary P2P technology", P2P Architect Project, IST-2001-32708, Project Deliverable D1
- [11] David Garlan, Robert T. Monroe, David Wile. "Acme: An Architecture Description Interchange Language". Proceedings of CASCON '97, November 1997.
- [12] ITU-T "Information Technology – Open Distributed Processing – Reference Model: Overview" ITU-T Recommendation X.901 (ISO/IEC DIS 10746-1), December 1997.
- [13] ITU-T "Information Technology – Quality of Service Framework" ITU-T Recommendation X.641 (ISO/IEC IS 13236), December 1997.
- [14] J.I. Asensio, V.A. Villagrà, "A UML Profile for QoS Management Information Specification in Distributed Object-based Applications", HP-OVUA Plenary Workshop 2000, Santorini Greece, June 2000
- [15] Chris Sluma, Jan Øyvind Aagedal, Mass Soldal Lund, Earl F. Ecklund, Jr. "Response to the OMG RFP for Modeling Quality of Service and Fault Tolerance Characteristics and Mechanisms", OMG document number realtime/2002-09-02, Sept 2002.
- [16] James Walkerdine, Lee Melville, Ian Sommerville, "Report on the Dependability Properties of P2P Architectures", P2P Architect Project, , IST-2001-32708, Project Deliverable D5
- [17] IEEE Software Engineering Standards Committee "IEEE Recommended Practice for Architectural Description of Software-Intensive Systems", IEEE Std 1471-2000, Sept 2000
- [18] Nenad Medvidovic, David S. Rosenblum, "Domains of Concern in Software Architectures and Architecture Description Languages ", Proceedings of the 1997 USENIX Conference on Domain-Specific Languages, Santa Barbara, California, October 1997
- [19] G. Athanasopoulos et al, "Models and specification primitives for building dependable p2p systems/applications", P2P Architect Project, IST-2001-32708, Project Deliverable D8, June 2003
- [20] P2P Architect Project, IST-2001-32708, http://www.atc.gr/p2p_architect/
- [21] OMG, UML Profile for Modeling Quality of Service and Fault Tolerance Characteristics and Mechanisms, OMG Final Adopted Specification, ptc/04-09-01
- [22] I. Sommerville, Software Engineering, Addison Wesley, 2001
- [23] James Walkerdine, Lee Melville, Ian Sommerville, "P2P Reference Architectures", P2P Architect Project, , IST-2001-32708, Project Deliverable D9