Hajautetut sovellusarkkitehtuurit

Markku Vähäaho

5 helmikuuta 2001

Ohjelmiston *arkkitehtuurilla* tarkoitetaan korkean tason kuvausta ohjelmiston alijärjestelmistä, komponenteista ja niiden välisistä suhteista [Bus96, s. 384]. Arkkitehtuurisuunnittelu mahdollistaa keskittymiseen kokonaisuuteen: järjestelmän rakenteeseen ja ohjelmiston kehitysprosessiin. Tämä on sitä tärkeämpää, mitä laajemmiksi ja monimutkaisemmiksi ohjelmistot kasvavat.

Arkkitehtuurin kuvaamiseen on perinteisten epämuodollisten kaavioiden lisäksi olemassa täsmällisiä menetelmiä, kuten *arkkitehtuurikieliä* [MT97]. Ne mahdollistavat arkkitehtuurikuvausten tietokoneavusteisen työstämisen ja automaattisen analysoinnin. Analyysin avulla voidaan varmistua kuvauksen oikeellisuudesta sekä tutkia ohjelmiston ominaisuuksia (esimerkiksi reaaliaikaisuutta) jo suunnitteluvaiheessa.

Arkkitehtuurinäkökulmaa ohjelmistojen suunnittelussa täydentävät uudelleenkäytettävät *ohjelmistokomponentit* [Szy98]. Komponenteilla tarkoitetaan tässä itsenäisesti kehitettäviä ohjelmistoyksiköitä, joista voidaan koostaa laajempia sovelluksia. Komponentin kaikkien rajapintojen ja riippuvuuksien on oltava hyvin määriteltyjä, jotta sitä voitaisiin käyttää erilaisissa ympäristöissä. Ohjelmiston koostaminen komponenteista on pohjimmiltaan arkkitehtuurisuunnittelua.

Darwin [Mag95] ja Olan [Bel96] ovat sovellusalustoja, jotka tukevat komponenttiperustaisten sovellusten kokoamista ja suorittamista. Sovellukset voivat olla hajautettuja; komponentit kommunikoivat yhdistävän ohjelmistokerroksen (engl. *middleware*) avulla. Erona tavanomaisiin järjestelmiin on, että arkkitehtuuri kuvataan eksplisiittisesti arkkitehtuurikielellä, ja suoritusympäristö kokoaa sovelluksen arkkitehtuurikuvauksen perusteella. Darwin ja Olan tukevat jossakin määrin myös dynaamisia arkkitehtuureja, joiden rakennetta voi muuttaa suorituksen aikana. Toistaiseksi tällaiset järjestelmät ovat vasta prototyyppiasteella.

Pilarcos[1]-projektissa [Kut00] tietojenkäsittelytieteen laitoksella tutkitaan avoimia hajautettuja komponenttijärjestelmiä. Tutkimuskohteena ovat erityisesti laajat, organisaatiorajat ylittävät järjestelmät (federoidut järjestelmät), jotka koostuvat itsenäisesti hallinnoiduista palveluista. Pääteemojen taustalla on ODP-referenssimalli [ISO96]. Arkkitehtuuritasoja Pilarcos-sovelluksissa on kolme: palvelujen yhteistyöarkkitehtuuri, kunkin palvelun toiminnallinen arkkitehtuuri sekä toiminnallisen arkkitehtuurin toteuttavat konkreettiset ohjelmistokomponentit. Tässä seminaariesitelmässä kuvataan nämä tasot ja niiden erityispiirteet, keskittyen Darwinin ja Olanin tapaan toiminnallisen arkkitehtuurin vaatimuksiin.

[1] Production and Integration of Large Component Systems

Viitteet

[Bel96] Bellissard, L. et al., Distributed Application Configuration. *Proc. of the 16th International Conference on Distributed Computing Systems*, Hong Kong, May 27–30, 1996.

[Bus96] Buschmann, F. et al, *Pattern-Oriented Software Architecture: A System of Patterns*. John Wiley and Sons, 1996.

[ISO96] ISO/IEC JTC1, *Information Technology – Open Systems Interconnection, Data Management and Open Distributed Processing. Reference Model of Open Distributed Processing.* 1996. IS10746.

[Kut00] Kutvonen, L. et al., Evaluation of OMG technologies for open systems support. To appear as a Technical Report in Series of Publications C, University of Helsinki, Department of Computer Science.

[Mag95] Magee, J. et al., Specifying Distributed Software Architectures. *Proc. of the 5th European Software Engineering Conference ESEC'95*, Barcelona, September 1995.

[MT97] Medvidovic, N., Taylor, R., A framework for classifying and comparing architecture description languages. *Proc. of the 6th European conference held jointly with the 5th ACM SIGSOFT symposium on Software engineering*, Zurich, Switzerland, September 22–25, 1997.

[Szy98] Szyperski, C., *Component Software: Beyond Object-Oriented Programming.* ACM Press and Addison-Wesley, New York, NY, 1998.