

RINGKASAN

Sistem perangkat lunak agen bergerak merupakan sebuah sistem yang kompleks. Pengembangan perangkat lunak agen bergerak sejauh ini lebih banyak menggunakan metode, teknik dan pendekatan berorientasi objek. Pada kenyataannya, pendekatan berorientasi objek memiliki masalah bawaan yang mendasar yaitu modularisasi yang memiliki nilai guna-ulang yang tinggi, nilai mampu-pelihara yang tinggi, dan ukuran kode yang besar. Semakin rendah nilai guna-ulang, nilai mampu pelihara, dan ukuran kode maka modularisasi semakin baik. Permasalahan modularisasi tersebut diakibatkan oleh adanya potongan kode atau unit program (perhatian) yang menyebar dan saling terikat atau disebut perhatian yang beririsan (*crosscutting concern*). Selain masalah tersebut, sifat agen yang mandiri sementara entitas objek yang statis membawa ke sebuah kondisi anomali ketika pendekatan objek ini dipaksakan untuk membangun perangkat lunak agen. Penelitian ini bertujuan untuk memperbaiki modularisasi yang kompleks dalam pengembangan agen bergerak dengan memperbaiki nilai guna-ulang, nilai mampu-pelihara dan ukuran kode program. Pendekatan berorientasi aspek menawarkan sebuah solusi yang memperbaiki masalah modularisasi dalam pendekatan berorientasi objek. Pengembangan dari mulai aras arsitektur sampai aras implementasi menjadi solusi yang komprehensif dalam pengembangan perangkat lunak agen. Sebuah arsitektur kerangka agen diusulkan untuk menunjang paradigma agen bergerak berorientasi aspek. Arsitektur yang diusulkan mengadopsi sebuah komposisi yang fleksibel dengan menekankan pada karakteristik mobilitas yang menjadi dasar dalam sistem agen bergerak. Dalam arsitektur tersebut, sebuah sistem agen disusun dengan tiga kelompok perhatian yaitu: perhatian mobilitas, perhatian keagenan, dan perhatian layanan dasar. Tiga kelompok ini digunakan untuk menterjemahkan semua karakteristik yang menempel pada sistem agen. Pada aras desain dan implementasi, dengan menggunakan pendekatan aspek maka hanya ada dua perhatian yaitu perhatian yang beririsan dan perhatian utama. Sehingga dari aras arsitektur ke aras desain dan implementasi diperlukan sebuah transformasi. Hasil pengujian empiris mengelompokkan perhatian mobilitas menjadi perhatian yang beririsan, perhatian keagenan menjadi perhatian utama, dan perhatian-perhatian layanan dasar ada yang tergolong ke perhatian yang beririsan dan ada pula yang tergolong ke perhatian utama. Pengembangan kerangka agen bergerak digunakan untuk membuktikan arsitektur yang dirancang. Hasil pengujian menunjukkan implementasi dari arsitektur yang diusulkan memiliki nilai guna-ulang dan nilai mampu-pelihara yang lebih baik dibandingkan menggunakan pendekatan berorientasi objek.

Kata Kunci: Agen Bergerak, Pendekatan Berorientasi Aspek, Pendekatan Berorientasi Objek, Perangkat Lunak Agen

ABSTRACT

Mobile agent software system is a complex system. So far, development of software agent is more use of methods, techniques and object-oriented (OO) approach. In fact, OO approach has a fundamental problem. The problems are OO has modularization that has high value of reusability, maintainability and size of code. It is caused by scattering and tangling code (crosscutting concern). There is an anomaly condition when OO approach forced to build a software agent. This study aims to improve the complexity of modularization in the development of mobile agent system by improving value of reuse, and value of maintainability and size of code. Aspect oriented (AO) approach offers a solution that fixes the problem modularization in the object-oriented approach. In this research, aspect-oriented approach used to develop software agents. Development that starts from architecture to implementation level can be a comprehensive solution in the development of software agents. An agent framework architecture is proposed to support AO agent paradigm. The proposed architecture adopts a flexible composition with emphasis on mobility characteristics in mobile agent system. In the architecture level, an agent system composed by three groups of concern, namely: mobility concerns, agenthood concerns, and basic functionality concerns. These three groups are used to accommodate all of the characteristics that attach to the agent system. At the design and implementation level, AO approach only have two concerns, namely: main concerns and crosscutting concerns. So it requires a transformation to change three concerns (in architecture level) into two concerns (in design level). The result transformed mobility concerns and some concerns of basic functionality into crosscutting concerns. Another results also transformed agenthood concerns and some concerns of basic functionality into main concerns (in design level). Development mobile agent framework was implemented. It used to prove architecture that was designed. The testing results demonstrate the implementation of the proposed architecture has better values of reusability, maintainability, and size of code than OO approach.

Keywords:

Mobile Agen, Aspect Oriented Programming, Object Oriented Programming, Agent Framework