An agent based architecture for constructing Interactive Simulation Systems

Zhao, Z.

Publication date
2004

Nederlandse Samenvatting

Interactieve simulaties spelen in toenemende mate een belangrijke rol in het wetenschappelijk onderzoek: in vergelijking met conventionele simulaties maken zij het mogelijk parameterruimten efficiënter te onderzoeken en de berekeningen nauwkeuriger aan te sturen. Op deze manier helpen ze ook om het gebruik van hulpbronnen, zowel voor verwerking als voor opslag te optimaliseren en de communicatie efficiënter te maken. Binnen een interactief simulatiesysteem (ISS) kunnen we meestal de volgende componenten onderscheiden: een of meer simulatie-modules die de eigenlijke berekeningen uitvoeren, een of meer visualisatie-modules die de resultaten van de simulatie meteen voor de gebruiker zichtbaar maken, en een of meer interactiemodules waarmee de gebruiker het simulatie- en visualisatie-proces kan sturen. De visualisatie- en interactie-modules worden veelal gecombineerd. De ontwikkeling en integratie van simulatie- en visualisatiekernels is echter kostbaar. Traditionele ontwikkelingsmethoden resulteren in het algemeen in systemen met een sterke koppeling tussen de systeemfunctionaliteit en de applicatieafhankelijke logische besturing en zijn hierdoor moeilijk aan te passen voor andere problemen. Deze hoge kosten en geringe flexibiliteit hinderen de invoering van dergelijke systemen. Bestaande softwarearchitecturen en middleware richten zich met name op de laag-niveau koppeling van systeemcomponenten, en niet expliciet op "rapid prototyping" van ISS of op een flexibele aansturing van het systeemgedrag.

In dit proefschrift stellen we dat de scheiding van de applicatieafhankelijke logische aansturing en de systeemfunctionaliteit een cruciale stap is naar een verbeterd ontwikkelingsproces voor ISS. We hebben "Interactive Simulation System Conductor" ontwikkeld, een architectuur voor de samenbouw van componenten met behulp van 'agents' (agenten), die een dergelijke scheiding realiseert. Deze architectuur vangt de intelligentie voor de besturing van het geïntegreerde systeem in verschillende rollen voor agenten en maakt het mogelijke met deze agenten een gelaagde verbindingsstructuur op te zetten tussen de componenten. Er is een "proof of concept" implementatie gebouwd op basis van de "High Level Architecture" (HLA), een specifiek voor gedistribueerde simulatie-systemen ontworpen middleware.

De basisarchitectuur van ISS-Conductor wordt in het tweede hoofdstuk gepresenteerd. Door de reken-kernels en ondersteunende structuren van de hoofdmodules van een ISS, zoals simulatie, visualisatie en interactie in te kapselen, staat het wetenschappers toe op een hoog niveau simulatie-experimenten op te zetten zonder zich bezig te hoeven houden met de laag-niveau systeemintegratie. Binnen de architectuur leggen "Communicatieagenten" (ComA) de basis voor de interoperabiliteit van

componenten; "Moduleagenten" (MA) orkestreren het gedrag van het systeem tijdens de uitvoering. De "Run-Time Infrastructure" (RTI) van HLA wordt gebruikt als de "software bus" waarmee alle componenten worden samengebouwd.
In het derde hoofdstuk bespreken we het functionele ontwerp van ISS-Conductor. In een ISS-Conductor component stuurt de MA het gedrag van die component, gebruik makend van een redeneer-systeem; dit systeem heeft in zijn "knowledge base" kennis van zowel de functionaliteit van de diverse componenten, als van de voor de applicatie specifieke constraints op de interacties. De functionaliteit van een component ("capability") wordt gemodelleerd met een eindige automaat die samen met die van de andere componenten kan worden geprogrammeerd, gebruik makend van een mechanisme gebaseerd op een Petri net ("scenario net"). Tijdens de uitvoering interpreteren de MA's van de verschillende componenten gezamenlijk de constraints op de interactie en besturen op deze manier het gedrag van het systeem als geheel.
Hoofdstuk vier is gewijd aan de details van de implementatie en de performance karakteristieken van ISS-Conductor. De ComA koppelt naar de diensten van de HLA RTI die verantwoordelijk zijn voor het delen van gegevens en het versturen van boodschappen. Het redeneer-systeem van de MA is in Prolog geschreven. Op basis van onze metingen aan de prestaties van onze implementatie kunnen we concluderen dat de Communicatieagenten een acceptabele overhead toevoegen aan de RTI. Applicaties gebaseerd op ISS-Conductor kunnen voor grote dataobjecten een netwerkutilisatie behalen, vergelijkbaar aan die van zuivere TCP sockets. De logische besturing van het redeneersysteem veroorzaakt ook slechts een kleine overhead.
In het vijfde hoofdstuk gebruiken we de ISS-Conductor architectuur om een prototype te maken voor een interactieve simulatieomgeving voor het plannen van vasculaire operaties. We bespreken de procedures voor de ontwikkeling van een ISS-Conductor component in detail, evenals die waarmee de componenten worden samengebouwd tot een interactief systeem. Voor een aantal testcases bespreken we de aansturing op het scenario-niveau, automatische tuning van de prestaties in een lopend systeem en de ondersteuning voor interactieve samenwerking. De experimentele resultaten tonen aan dat ISS-Conductor slechts een kleine overhead toevoegt aan reeds bestaande reken-kernels.
In het zesde hoofdstuk bespreken we de mogelijkheid componenten die aan het ISS-Conductor model voldoen te gebruiken als software binnen een zogenaamd "Problem Solving Environment". Een van de cruciale problemen die we daarbij onderzoeken is hoe de componenten voor een interactief simulatie-experiment automatisch gevonden en samengevoegd kunnen worden. In dit hoofdstuk wordt ISS-Studio beschreven, een op Java gebaseerd prototype voor een dergelijke omgeving.
In het zevende en laatste hoofdstuk wordt een samenvatting gegeven van het proefschrift en worden de resultaten geplaatst in het licht van de verwachte toekomstige ontwikkelingen op het gebied van interactie simulatiesystemen.