

مقایسه و ارزیابی الگوهای معماری واسط انسان- کامپیوتر برای

سامانه های فضای جنگ نوین

عبد الحمید فطانت-موسسه آموزشی و تحقیقاتی صنایع دفاعی وزارت دفاع و پشتیبانی نیروهای مسلح^۱

abfetanat@gmail.com

مهدی نقیان فشارکی- موسسه آموزشی و تحقیقاتی صنایع دفاعی وزارت دفاع و پشتیبانی نیروهای

مسلح^۲ mehfesharaki@yahoo.com

چکیده

در این مقاله به یکی از مولفه های اصلی سامانه های فضای جنگ پرداخته شده که در بسیاری از موارد همانند زیر دریایی ها نقشی بحرانی دارند. واسط انسان-کامپیوتر یا به اختصار HCI^۱ نه بعنوان یک پوسته بلکه یک سیستم نفوذ پذیر بوده که عامل های انسانی با سامانه مورد نظر تعامل می کنند. جابجایی پذیری، کارایی، اصلاح پذیری از مهمترین صفات کیفی این مولفه بوده که موجب تطبیق پذیری سامانه اصلی شده و سامانه ای از نوع ساپ(سامانه یا سیستم انطباق پذیر پیچیده) را بدست می دهد. از طرف دیگر توسعه کارآمد و موثر این سیستم موجب شده که کاربران با سامانه اصلی تعاملی پایدار و سهل داشته و نمونه مستدلی از سامانه HITL^۲ عامل انسانی در حلقه محاسبه نتیجه شود. به این ترتیب HCI برای سامانه های فضای جنگ نوین از جمله فرماندهی و کنترل اهمیتی بحرانی داشته و لذا طراحی دانه درشت آن یا عبارتی معماری نرم افزار آن بغایت مهم است. دیدگاه دیگری که پرداختن به HCI را در فضای جنگ مهم می کند سکو محور نبودن این فضا است به گونه ای که عامل های رزم در سکو های متعدد مانور کرده و این تغییر نیازمند محاسبات فراگیری است که تطبیق پذیری را ممکن و همه سکو ها را پوشش دهد. HCI کارآمد و موثر انطباق پذیری پویای عامل های انسانی را در مقابل تغییرات گسترده ممکن کرده و اساسا یکی از پاسخ های مناسب به مساله محاسبات فراگیر است. در این مقاله به الگوهای مختلفی از HCI پرداخته و الگوهای یکدست، سی هیم، Arch HCI، PAC، MVC، ARCH/Slinky و AMODEUSPAC- که اصلی ترین الگوهای معماری بوده بررسی شده و آنها را از دستیابی به صفت کیفی گفته شده مقایسه و در نهایت به انتخابی مستدل از آن هادست می یابیم

کلمات کلیدی: محاسبات فراگیر، الگوی معماری، سی هیم، MVC، PAC و AMODEUSPAC

Abstract

In this paper we focus on the important component of warfare space such that in any domain similar submarine has an critical role. Human-Computer Interface(HCI)not only an open shell but is a system such that may be interfaced between computers and human agents and so it must be supported Portability, performance and modifiability as three qualities requirement. the adaptive system will be supported These qualities and every task for adaptively complex will be earned these three qualities. In another way, the system will be performance and sufficient and so users will be have stable and simple interactive relations with system and the result will be human In The Loop (HITL) system, eventually. So, the design of HCI very important for warfare context and software architecture HCI is an critical role. the another viewpoint for focus on the HCI is this fact that warfare context no platform orient but is a multi platform or network centric domain and will be needs pervasive computing

¹Human Computer Intergace

²Human In The Loop

in which either coordinate of forces but will be development the synonymous human agents HCI as a primitive component of pervasive computing will be adaptively for every platforms and may be develop the synonymous human agents. In this paper, the varieties patterns of HCI, monolotique, Seeheim, Arch HCI, PAC, MVC, ARCH/Slinky و AMODEUS as architecture patterns importants of HCI will be analysis and compared in qualities requirements viewpoint, and an correct selection of them will be earned, in final. **Keywords:** Pervasive computing, Architectural pattern, Seeheim, MVC, PAC, AMODEUS-PAC

مقدمه

این حالت حداقل نیازمندی ها شامل سه دسته زیر هستند:

۱- نمایش برای تعامل کاربر (P^4) - برنامه کاربردی برای اهداف سیستم (A^5) - گفتگو یا دیالوگ بین نمایش و برنامه کاربردی (D^6)

در واسط های گرافیکی بطور معمول از کنترل هائی گرافیکی همانند جعبه لیست ها^۷، جعبه های متن^۸ یا حتی tablet ها و کنترل های مشابه برای نمایش استفاده می شود علاوه بر این از این کنترل ها و کنترل های دیگری همانند دکمه های رادیویی برای دیالوگ استفاده می شود. بکار گیری دکمه های فشار یا انتخاب یا حتی در امروز استفاده از داشبورد ها برای اجرای برنامه های کاربردی بهره گرفته می شود. برای اینکه بتوان هر سه نیاز را بطور گرافیکی بدست آورد الگوهای مختلفی وجود داشته که مهمترین آن ها را توصیف نموده و مدل پیشنهادی خود را ارائه می کنیم.

۲- صفات کیفی برای ارزیابی و انتخاب الگوی GUI در فضای جنگ

می توان صفات کیفی مختلف و متنوعی را پیشنهاد داده اما برای توسعه یک سامانه چابک سه صفت کارائی^۹، اصلاح پذیری و جابجائی^{۱۰} (یا قابلیت حمل) اهمیت درجه اول دارند. کارائی بالا به معنی پاسخ گوئی در عین کاهش هزینه است. اگر این صفت کیفی کم باشد در این صورت اساسا الگو محصولی مهندسی نخواهد بود. صفت اصلاح پذیری نشان از همگامی آن با چابکی کل سامانه داشته که نوعی حمایت از تطبیق سامانه با شرایط

از مهمترین چالش های سامانه های فضای جنگ از جمله فرماندهی و کنترل دسترسی به چابکی برای احراز قابلیت سرعت عمل جهت مقابله با رخداد های گوناگون و یکباره ای بوده که در غیر اینصورت غافلگیری محتوم و قطعی بوده و شکست نتیجه طبیعی آن است. برای دستیابی به این قابلیت در فضای جنگ پیچیده و پر از ابهام عامل های رزمنده باید بتوانند بطور خود هماهنگ و با ابتکار و نوآوری عمل کرده و تطبیق پذیری را بعنوان شاکله اصلی رفتار خود کنند. تنها در این صورت است که در جنگ های نامتقارن امکان موفقیت در مواجهه با دشمنی که به لحاظ فیزیکی و اطلاعاتی برتری خورد کننده ای دارد ممکن می شود.

۱- نیازمندی های حداقلی HCI

بطور کلی سه دلیل اصلی برای بکارگیری HCI در سامانه های جنگ های نوین عبارتند از تطبیق پذیری، HITL بعنوان سیستم هائی که عامل انسانی در حلقه محاسبه بوده و بالاخره محاسبات فراگیر (دستیابی به چند سکویی بهمراه توسعه خود مختار عامل ها دهد [2]) است. ضرورت بکارگیری محاسبات فراگیر در فضای جنگ نوین نشان می دهد که نیازمند چشم اندازی بوده که HCI در حالت حداقلی خود مشابه سیستم های GUI^۳ (واسط کاربر گرافیکی) عمل کرده و در حالت های پیشرفته سیستمی هوشمند را عرضه می کنند. در حالت پیشرفته HCI باید یک محیط اشباح شده ای از قابلیت های محاسباتی و ارتباطی برای تجميع کاربران انسانی را فراهم نموده و در عین حال هر کاربر بتواند با مراجعه به آن نقش خود را مأموریت فهمیده و به وضعیت خود آگاهی داشته باشد. در این مقاله ما بطور حداقلی عمل نموده و نیازمند ها را بطور حداقلی سنجش می کنیم. بطور کلی در

⁴Presentation

⁵Application

⁶Dialog

⁷Listbox

⁸Textbox

⁹Performance

¹⁰Portability

³Graphic User Interface

این الگو با اعمال عملیات واحد بر روی یک مولفه یکدست بوجود آمده که شامل سه نوع وظایف نمایش، دیالوگ و کاربرد بوده نتیجه می شود (شکل ۱). این الگو با دو عملیات واحد متوالی کل به جزء بر روی الگوی یکدست نتیجه می شود. برای افزایش صفات کیفی اصلاح پذیری و قابلیت حمل می توان از سه نوع عملیات واحد منابع مشترک، ارث بری و تجرید استفاده کرد: در معماری مونولوگ چون فقط یک مولفه تک وجود دارد پس ارث بری کاربردی ندارد. هر دوی منابع مشترک و تجرید اگر بکار روند تاثیری فوری و مستقیم بر روی کارائی داشته که برای سیستم HCI بسیار مهم و ضروری است. به این ترتیب با دوبار بکارگیری عملیات تجزیه کل به جزء و یکبار عملیات تجرید الگوی سی هیمنتیجه می شود. بکارگیری متوالی سه عملیات واحد (دوبار تجزیه کل به جزء و یک تجرید) اصلاح پذیری همراه با قابلیت حمل را سه بار تقویت می کند.

۵- الگوی MVC^{۱۴}

این معماری که امروزه در سرورهای مختلفی از جمله دات نت^{۱۵}، جی تو ای ای^{۱۶} و کوربا^{۱۷} بکار می روند همانند سی هیم دارای سه مولفه بوده و دارای اهداف مشترک بوده و اصلاح پذیری و قابلیت حمل را بعنوان اهداف خود دارد. در این معماری نقش مدل، مدیریت تغییر داده ها بوده و نقش دیدگاه و کنترل کننده به ترتیب نمایش خروجی و کنترل نمودن درخواست های کاربر است. اگر برنامه کاربردی موجود باشد آنگاه می توان مدل را بعنوان دیالوگ و کاربرد با هم در نظر گرفته که در این صورت کنترل کننده بعنوان ورودی و دیدگاه بعنوان خروجی است از طرف دیگر قبل از هر چیز با عمل واحد کل به جزء در برنامه کاربردی مزبور یک تجزیه برای بدست آوردن مولفه های مختلف HCI فراهم می کنیم. این مشابه سی هیم است که با همان صفات کیفی (قابلیت حمل و اصلاح پذیری) ولی با بکارگیری

و تغییرات جدید است. جابجائی نشان از بکارگیری آن در سکوهای مختلف و محیط های شبکه محور داشته که یکی از مهمترین خصوصیات در فضای جنگ های نوین است.

۳- الگوی یکدست یا مونولوگ^{۱۱}

ساده ترین الگو، الگوی یکدست یا مونولوگ برای واسط کاربر بوده که فقط یا یک مولفه همه نیازمندی ها احصاء می شوند. این الگو تک مولفه ای است که علیرغم کارائی خوب آن، اصلاح پذیر نیست در واقع اصلاح مولفه ای که همه عناصر را در خود جای داده بسیار سخت و عملاً غیر ممکن است. بنابراین این الگو برای توسعه یک معماری چابک یا سبک وزن قابل قبول نیست. بکارگیری همه مدل های داده ای و همه برنامه ها در یک مولفه کارائی را افزایش می دهد ولی تغییرات بر روی داده ها و یا برنامه های متناظر بسختی ممکن خواهد بود. از طرف دیگر هر گونه مولفه سنگین نرم افزاری و سخت افزاری متعارض با داشتن یک سامانه سبک مطلوب در یک جنگ نامتقارن برای غافلگیری دشمن است. نکته دیگر اینکه ارتباط یک مولفه سنگین که دارای همه گونه داده و برنامه ای باشد با زمینه، محکم بوده که بنابراین قابلیت جابجائی را بسیار کاهش می دهد. اساساً برای جابجائی بر روی سکوهای متفاوت باید کمترین وابستگی به آن ها وجود داشته که با یک مولفه سنگین میسر نیست. بنابراین ما معماری GUI مطلوب خود را با اعمال عملیات واحد^{۱۲} بر روی این الگو نتیجه می دهیم، عملیات واحد از عملگرهای اصلی معماری است که بر روی مولفه ها صورت گرفته شده و مولفه های دیگری را با صفات کیفی دیگر نتیجه می دهند عملیات تجمیع، ارث بری، تجرید، ماشین مجازی و منابع مشترک از جمله این ها هستند [1].

۴- الگوی سی هیم^{۱۳}

¹⁴ Model-View-Controller

¹⁵ .net

¹⁶ J2ee

¹⁷ Corba

¹¹ Monologue

¹² Unit Operation

¹³ Seeheim

عملیات واحد های دیگر از یک مولفه تک نتیجه می شود.

۶- مقایسه سی هیم و MVC

هر دو دارای پایه مشترک و تکنیک های همانند هستند در حقیقت این ها دو مدل مرجع بوده که مفروضات متفاوتی دارند. در اولی یعنی سی هیم مهمترین سناریو، حفاظت در هنگام جابجائی اطلاعات بین ابزار ها ی^{۱۸} گوناگون بوده که هر کدام سه لایه را فیلتر (عایق بندی) می کنند به عبارت دیگر گارد اصلی سی هیم ایجاد لایه بندی برای حفاظت از تغییر اطلاعات است. امدار MVC فرض بر این است که اصلاحات بجای اینکه به لایه ها مربوط شوند به اشیائی که آن سه کار را انجام می دهند مربوط می کنیم. در حقیقت MVC با عملیات واحد تجزیه کل به جزء، برنامه اصلی را بهوظایف کاربردی متعددی تقسیم نموده بطوریکه برای هر کدام نیازمند زیر مولفه های ورودی، خروجی و دیالوگ است.

۸- الگوی ARCH HCI

از اصلی ترین عیب های سی هیم عدم حمایت از نمایش های گوناگون و همینطور عدم استفاده از برنامه های کاربردی گوناگون است که در نتیجه آن را برای محیط های محاسباتی متنوع همانند تور ها و بطور کلی محاسبات فراگیر نامناسب می کند. با بکارگیری دو بار عملیات واحد ماشین مجازی^{۱۹} بر مدل اولیه و تک مولفه ای نمایش، برنامه کاربردی و دیالوگ دو مولفه کاربرد مجازی و جعبه ابزار مجازی بدست آمده که نتیجه نهائی مدل ARCH HCI با قابلیت جابجائی بالا (شکل ۲) است، این در شرایطی است که این مدل نیز مشابه سی هیم اصلاح پذیر است. بکارگیری دو بار استفاده از عملیات ماشین مجازی هر چند که قابلیت جابجائی را افزایش می دهد اما از طرف دیگر موجب کاهش شدید کارائی می شود. بنابراین الگوی ARCH HCI مقابل الگوی مونولوگ است. در الگوی تک مولفه ای یکدست فقط کارائی بالا بوده و جابجائی و اصلاح پذیری وجود نداشته که در

ARCH HCI وضعیت معکوس بوده به این ترتیب که کارائی کاهش داشته ولی جابجائی و اصلاح پذیری با نرخ بالائی وجود دارند.

۷- فرا مدل ARCH/Slinky

عیب اصلی الگوی ARCH HCI این است که با بکارگیری دو ماشین مجازی در الگوی سی هیم کارائی بشدت کاهش می یابد. عیب دیگر این است که ساختار توابع تخصیص در هر دو ماشین مجازی متغیر و سیال است. برای کاهش دادن این عیب ها از استعاره Slinky که نوعی فنر بوده استفاده می شود. این فنر به گونه ای است که هر گاه از یک طرف باز شود طرف دیگر پیچ خورده و تا زده می شود. بنابراین فنر در هنگام استفاده لازم نیست که کاملاً باز بوده و فضای زیادی را در بر گیرد. وضعیت الگو های گفته شده تابحال به این شکل بوده که یا کارائی افزایش داشته که همزمان دو صفت کیفی دیگر جابجائی و اصلاح پذیری کاهش می پذیرند و بعکس اگر جابجائی و اصلاح پذیری افزایش داشته باشند آنگاه کارائی کاهش خواهد داشت. هدف از بکارگیری استعاره slinky عبور از فقط دو وضعیت مزبور است. برای این منظور می توانیم از عملیات فشرده سازی همه یا بخشی از سیستم استفاده کرده که در این صورت کارائی افزایش می یابد. می توانیم با بکار گیری این عملیات اصلاح پذیری و همچنین قابلیت حمل را نیز افزایش دهیم. شکل ۳ فرامدل ARCH/Slinky را نمایش می دهد. این یک الگوی معماری نبوده بلکه فقط یک فرامدلی است که بر اساس آن الگوی ARCH HCI با کارائی، اصلاح پذیری و جابجائی بیشتر طراحی می شود. الگوی ARCH HCI و فرامدل ARCH/Slinky هر دو برای بروزرسانی الگوی سی هیم بکار گرفته شده اند. اکنون در ادامه الگوی PAC که توسعه ای از الگوی MVC است را معرفی می کنیم.

۹- الگوی PAC

این الگو شامل سه مولفه نمایش (P)، برنامه کاربردی (A) و کنترل (C) بوده که معادل دیالوگ در ARCH HCI است. اگر یک مولفه شامل نمایش، کاربرد و دیالوگ

¹⁸ TOOLKIT

¹⁹ Virtual machine

[۱] Bass, L., Clements, P. and Kazman, R., "Software Architecture in Practice", Addison-Wesley, 1997.

[2] Cheng, S.W., "Software Architecture-based Adaptation for Pervasive Systems". International Conference on Architecture of Computing Systems: Trends in Network and Pervasive Computing. Karlsruhe, Germany. LNCS Vol. 2299, Schmeck, Ungerer, Wolf, (Eds.) April 2002.

باشد با عملکرد جداسازی^{۲۰} کنترل را جدا نموده و آنگاه در ادامه آن را برای نمایش و کاربرد نیز ادامه می دهیم. نکته مهم این است که این الگو برای تعامل کاربران یا عوامل انسانی در یک سیستم HITL بکار گرفته شده که معمولاً در یک نرم افزار بزرگ بیشتر به مقیاس پذیری نیاز هست بعبارت دیگر مولفه ها و ماژول های مختلف نرم افزار های بزرگ خود به واسطه کاربر ویژه نیازمند هستند که در این صورت یک نمای کلی از این الگو در شکل ۴ آمده است.

۱۰- الگوی AMODEUSPAC

تلفیقی از دو الگوی PAC و ARCH HCI است. در واقع در الگوی ARCH HCI به جای دیالوگ از ساختار PAC نوعی استفاده می کنیم (شکل ۵)

نتیجه گیری

در این مقاله ما پس از معلوم نمودن ارزش سیستم های HCI و اهمیت بحرانی آن به سه دلیل تطبیق پذیری، HITL و محاسبات فراگیر معلوم نموده و آنگاه نیازمندی های سیستم های GUI را بعنوان حداقلی از سیستم های HCI در سه صفت کیفی کارائی، اصلاح پذیری و قابلیت حمل معلوم کردیم. بر پایه دستیابی به این سه صفت کیفی الگوهای معماری مختلفی را بررسی و مقایسه نمودیم. دو الگوی سی هیم و MVC را مبنا گرفته که هر دو با دستکاری یک الگوی تک مولفه ای و مونولوتیک حاوی سه نوع وظیفه مندی نمایش، کاربرد و دیالوگ به صفات کیفی کارائی و اصلاح پذیری دست پیدا کرده اند. الگوهای پیشرفته بعدی در یکی از دو دسته مبتنی بر سی هیم و MVC عرضه شده اند. در دسته اول الگوی ARCH HCI و در دسته دوم الگوی PAC ارائه گردیده که اکنون با تلفیق یا ترکیب این دو در الگویی بنام PAC-AMODEUS می توان سه نوع صفت کیفی کارائی، جابجائی و اصلاح پذیری را در بالاترین نرخ ممکن عرضه نموده که GUI پیشنهادی برای سامانه های فضای جنگ از جمله سامانه فرماندهی و کنترل است.

منابع و مراجع

²⁰Separation



