

## ارزیابی کارایی سیستم‌های اطلاعاتی با بهره‌گیری از الگوی طراحی MVC و زبان نشانه‌گذاری توسعه یافته

شیدا مزارعی<sup>۱</sup>، علی هارون آبادی<sup>۲</sup>، سید جواد میرعابدینی<sup>۳</sup>

۱- دانشگاه آزاد اسلامی، واحد بوشهر، گروه مهندسی کامپیوتر، بوشهر، ایران، sh.mazarei@gmail.com

۲- عضو هیئت علمی دانشگاه آزاد اسلامی، واحد تهران مرکز، گروه مهندسی کامپیوتر، تهران، ایران، a.harounabadi@gmail.com

۳- عضو هیئت علمی دانشگاه آزاد اسلامی، واحد تهران مرکز، گروه مهندسی کامپیوتر، تهران، ایران، j.mirabedini@gmail.com

تاریخ دریافت: ۹۳/۸/۹ تاریخ پذیرش: ۹۴/۳/۱۷

### چکیده:

توسعه سیستم‌های نرم‌افزاری پیچیده خواهان مدل‌هایی است که ارزیابی نیازهای غیروظیفه‌مندی سیستم‌ها را در فرایند توسعه به ویژه در مراحل اولیه انجام دهد. ارزیابی معماری به روش‌های مختلفی انجام می‌گیرد، از این رو نیاز به یک روشی برای ادغام نیازهای غیروظیفه‌مندی می‌باشد. مقاله حاضر، الگوی طراحی MVC را در قالب چارچوبی مبتنی بر زبان نشانه‌گذاری توسعه پذیر برای ارزیابی نیاز غیروظیفه‌مندی کارایی نمایش می‌دهد. این چارچوب بر پایه شبکه‌های پتری بنا شده که به کمک آن می‌توان الگوی طراحی MVC را ارائه داد. تحقیق جاری در مقایسه کارهای پیشین امکان استفاده از زبان نشانه‌گذاری توسعه پذیر، جهت رسیدگی به نیازهای غیروظیفه‌مندی فراهم نموده و یک مدل اجرایی از نرم افزار ترسیم می‌نماید.

**کلید واژه:** الگوی طراحی MVC، زبان نشانه‌گذاری توسعه یافته، شبکه‌های پتری، کارایی، معماری نرم‌افزار

### مقدمه

غیر وظیفه‌مندی در طی فرایند توسعه آشنا نیستند این کار باعث از بین رفتن فاصله ایجاد شده بین معمار نرم‌افزار و تحلیل گران کارایی می‌گردد. با استفاده از این چارچوب که به صورت افزایشی پیاده سازی شده است، معمار نرم‌افزار می‌تواند سیستم خود را با هر نوع علامتگذاری توصیف کند و تحلیل گر نیز با هر علامتگذاری مدلسازی، سیستم را ارزیابی نماید [۲]. لذا اگر بتوان N علامتگذاری برای توصیف معماری را به M مدل قابل اجرا تبدیل نمود آنگاه به N\*M تبدیل نیاز است. اما با استفاده از این چارچوب به N+M تبدیل کاهش می‌یابد [۱۲]. در واقع این چارچوب در بخش میانی توصیفات ورودی را به خروجی نگاشت می‌دهد. در اینجا الگوی MVC را با استفاده از زبان مدلسازی یکپارچه توصیف و خروجی یکی از مدل‌های شبکه‌های پتری می‌باشد.

تحلیل و ارزیابی نیازهای غیروظیفه‌مندی فعالیتی است که بایستی امروزه در فرایند توسعه سیستم‌های نرم‌افزاری بزرگ و پیچیده مورد نظر قرار گیرد [۱]. اگر ارزیابی این نیازهای در فازهای اولیه توسعه به خصوص معماری استفاده شود، بسیاری از هزینه‌های اضافی را کاهش می‌دهد. در این مقاله هدف، ارزیابی نیاز کارایی الگوی طراحی MVC (Model-View-Controller) است که به عنوان یک الگوی پرکاربرد در نرم‌افزار شناخته شده است. با نمایش این الگو در قالب چارچوب مبتنی بر زبان نشانه‌گذاری توسعه پذیر Extensible Markup Language، نیاز کارایی را با مدل‌های مختلف از شبکه پتری می‌توان بیان کرد و مورد ارزیابی قرار داد. از آنجایی که توسعه‌دهندگان نرم‌افزار با ارزیابی و تحلیل نیازهای

مناسب است. ضمن اینکه حاشیه‌نویسی‌های اضافه شده به نمودارهای توصیف‌کننده در معماری را به راحتی می‌تواند نشان دهد و در بخش میانی چارچوب، یک گرامر عمومی برای مدل‌های ارزیابی مبتنی بر شبکه پتری ارائه شده‌است که این مدل نیز مبتنی بر زبان نشانه گذاری توسعه پذیر است [۱۳].

### شبکه‌های پتری

نظریه شبکه پتری برای اولین بار توسط کارل پتری در اوایل ۱۹۶۰ ارائه گردید. این شبکه‌ها یک زبان گرافیکی برای مدل‌های ساختاری از سیستم‌های همروند و آنالیز ویژگی‌های آنهاست. این شبکه‌ها همچنین ابزار مناسبی برای مدلسازی ریاضی و گرافیکی به حساب می‌آیند از این ابزار می‌توان برای مدلسازی، توصیف و تحلیل سیستم‌هایی که دارای ماهیتی همزمان، توزیع شده، موازی و نامعین و اتفافی هستند استفاده نمود. یکی دیگر از خصوصیات شبکه‌های پتری قابل اجرا بودن آنهاست که از همین خاصیت می‌توان برای ارزیابی رفتار و کارایی یک سیستم استفاده نمود [۸]. شبکه پتری دارای بسط‌های متعددی می‌باشد که می‌توان شبکه پتری رنگی، شبکه پتری رنگی زمانی، شبکه پتری تعمیم یافته، شبکه پتری تصادفی و ... را نام برد.

### کارهای مرتبط

در مقاله [۱۲] چارچوب مبتنی بر زبان نشانه گذاری توسعه پذیر را بر اساس شبکه پتری نشان داده است. تفاوتش با مقاله پیش رو، در نشان دادن الگوی معماری MVC با این چارچوب است. در [۲] دو صفت کارایی و قابلیت اطمینان در سطح معماری مورد ارزیابی قرار گرفته است و مدل نهایی بسط‌های مختلف شبکه پتری می‌باشد. بخش میانی چارچوب، قوانین موجود برای تبدیل توصیفات معماری به مدل نهایی را به صورت یک گرامر عمومی بیان می‌کند. گرامر عمومی ارائه شده مشابه PNML است. هدف PNML پشتیبانی از انواع شبکه‌های پتری بین ابزارهای مختلف است. در مقاله [۸] به منظور تسهیل استفاده از نتایج ارزیابی نیازهای غیر وظیفه

در ادامه، قسمت دوم مقاله به اختصار الگوی طراحی MVC و نیاز کارایی، زبان نشانه گذاری توسعه‌پذیر و شبکه پتری را نشان می‌دهد. در بخش سوم کارهای انجام شده در این زمینه را مورد بررسی قرار می‌گیرد. در بخش چهارم روش پیشنهادی شرح داده خواهد شد. ارائه یک مطالعه موردی از سیستم آ؛ زمون برخط در بخش پنجم مورد توجه قرار گرفته است و بخش آخر به نتیجه اختصاص یافته است.

### بستر کار

زمینه انجام کار شامل الگوی طراحی MVC جهت رسیدگی به نیاز کارایی، زبان نشانه گذاری توسعه پذیر و شبکه‌های پتری است که به اختصار به بررسی هر یک پرداخته شده است.

### الگوی طراحی MVC جهت رسیدگی به نیاز کارایی

الگوهای معماری جهت ایجاد طراحی مورد توجه قرار گیرند [۳]. معماری MVC برای سیستم‌هایی که نیاز به داده‌های مشابه دارند ارائه شده است [۴]. الگوی طراحی MVC دسترسی به داده‌ها و پیاده‌سازی سطوح داده و سطوح ابزار را بسیار خوب از هم تفکیک می‌کند و معمولاً به عنوان یکی از مدل‌های توسعه در سیستم‌های برنامه‌های توزیعی استفاده می‌شود [۵]. نمودارهای زبان مدلسازی یکپارچه وجوه ساختاری و رفتاری را جهت توصیف معماری در سطح بالا نشان می‌دهند [۶]. ساختار و رفتار الگوی طراحی MVC را می‌توان با دو نمودار ترتیب و مؤلفه از زبان مدلسازی یکپارچه نشان داد.

### زبان نشانه گذاری توسعه‌پذیر

زبان نشانه گذاری توسعه پذیر یا همان XML در عین سادگی، دارای ایده‌های قوی است. گرامر این زبان بسیار ساده و شبیه به HTML است. دلیل استفاده شدن در چارچوب میانی این است که به راحتی می‌توان با کمک آن داده‌ها را توصیف و دستکاری نمود، از طرفی ساختار سلسله مراتبی موجود در آن برای نمایش مؤلفه‌های معماری اشیا و ساختارهای مشابه بسیار

وظیفه‌مندی کارایی را مورد ارزیابی قرار داد بایستی اطلاعات مربوط به کارایی را به آن اضافه کرد. جهت الحاق کلیشه‌های کارایی به نمودار ترتیب در این مقاله به هر پیغام کلیشه <<PAstep>> با برچسب PAprop, PArep, و PAdelay راکه هر یک بیانگر احتمال اجرای پیغام، تکرار پیغام‌ها و تأخیر در مرکز سرویس الصاق نموده و به اولین پیغام موجود در نمودار کلیشه <<PAcloseload>> اضافه می‌شود تا شدت بارکاری را مشخص کند. به این کلیشه برچسب‌های PApopulation و PAextdelay اضافه می‌شود، تا به ترتیب تعداد کاربران و زمان مورد نیاز برای انجام درخواست را نشان دهد.

### تبدیل توصیفات الگو به چارچوب مبتنی بر زبان نشانه گذاری توسعه یافته

به دلیل اهمیت معماری نرم‌افزار، به عنوان اولین محصول فرایند توسعه نرم‌افزار، در ارزیابی نیازهای غیروظیفه‌مندی در [۱۰،۲] یک چارچوب مبتنی بر XML برای یکپارچه‌سازی آن‌ها ارائه شده‌است. این چارچوب توصیفات معماری یا همان مدل مبدأ را در سه بخش به مدل مقصد یا همان مدل اجرایی تبدیل می‌نماید. بخش اول توصیفات معماری به XML تبدیل می‌شود. در بخش دوم، این توصیفات به یک مدل میانی مبتنی بر XML تبدیل می‌شود و در بخش سوم مدل میانی به یک مدل مقصد برای تحلیل تبدیل می‌شود. جهت انجام این تبدیلات مجموعه قوانینی در بخش میانی قرار می‌گیرد که عناصر مدل مبدأ را به عناصر مدل مقصد نگاشت می‌دهد.

### گرامر کلی بسط‌های مختلف شبکه پتری مبتنی بر زبان نشانه گذاری توسعه‌پذیر

کدها و تگ‌های بیان شده در بخش میانی این چارچوب قابل تبدیل به بسط‌های شبکه پتری می‌باشد. تمام بسط‌های شبکه پتری دارای یک ساختار کلی هستند که یکسری ویژگی‌ها به این ساختار اضافه می‌شود. این شباهت ساختاری منجر به ایجاد یک ساختار کلی شده تا به راحتی بتوان به یکدیگر تبدیل شوند. قسمتی از گرامر BNF مشترک بسط‌های

مندی در جداسازی و ترکیب سیستم‌های مبتنی بر مؤلفه بایستی ابزاری برای پیش‌بینی اتوماتیک ابداع شود، که برخی از ویژگی‌های کیفی کلی نرم‌افزار را بدون نیاز به دانش گسترده از روش تجزیه و تحلیل به طراح پیشنهاد دهد. برای رسیدن به این ایده در این مقاله یک مدل چارچوب انتقال مدل‌رانه تعریف شده است، که هدفش گرفتن اطلاعات مربوط به ارزیابی ویژگی‌های غیروظیفه‌مندی سیستم مبتنی بر مؤلفه است و بر روی کارایی و قابلیت اطمینان متمرکز شده است. همچنین در مقاله‌ای از [۹] یک چارچوب برای یکپارچه‌سازی ارزیابی نرم‌افزار معرفی شده است. هسته این چارچوب نیز مبتنی بر زبان نشانه گذاری توسعه‌پذیر است و از مدل نرم‌افزار و روابط رسمی میان مدل‌های نرم‌افزار استفاده نموده است. تفاوت این چارچوب نیز با چارچوب استفاده شده در مقاله حاضر این است که چارچوب مورد استفاده اطلاعات نیاز غیر وظیفه‌مندی به نمادهای توصیف معماری اضافه شده است. در تمامی این مقالات چارچوب‌ها به طور کلی برای معماری بیان شده است، اما در این مقاله سعی شده است که الگوی طراحی MVC را با استفاده چارچوب مبتنی بر زبان نشانه‌گذاری توسعه‌پذیر جهت ارزیابی نیاز غیروظیفه‌مندی کارایی بیان شود.

### روش پیشنهادی

در این مقاله قصد بر است که کارایی در الگوی طراحی MVC با کمک چارچوب مبتنی بر زبان نشانه گذاری توسعه‌پذیر ارزیابی شود. برای اینکار ابتدا رفتار معماری را با نمودار ترتیب از زبان مدل‌سازی یکپارچه توصیف نموده و سپس اطلاعات مربوط به کارایی را به آن حاشیه‌نوبسی می‌شود و سپس با کمک زبان نشانه‌گذاری توسعه‌پذیر که بر پایه شبکه‌های پتری توسعه یافته الگوی MVC نشان داده خواهد شد. با این کار می‌توان این الگو را با انواع بسط‌های شبکه پتری مدل نمود و سپس مورد ارزیابی قرار داد.

### نمودار ترتیب و حاشیه نویسی کلیشه‌های کارایی

همانگونه که بیان شد برای نمایش رفتار یک سیستم از نمودار ترتیب استفاده می‌شود. اما برای اینکه بتوان نیاز غیر

مختلف شبکه پتری در (جدول ۱) بیان می‌شود:

جدول ۱- گرامر BNF با زبان نشانه گذاری توسعه یافته

NET ELEMENT	::=start ( <u>Albin</u> ) ELEMENT end ::=empty  PLACE ELEMENT  TRANSITION ELEMENT  ARC ELEMENT  TYPEDEF ELEMENT  seeml {FILENAME} ELEMENT ::= STRING ::= place {ID} {NAME INIT CAP}
FILENAME PLACE INIT	::= empty  init{MULTISET} ::= empty  capacity{MULTISET} ::=empty  colour{COLOURSET} ::=ML_DEFINITION ::=INTEGER
CAP	INTEGER'STRING MSLIST INTEGER'STRING   MSLIST@+T_WEIGHT ::= empty  +MULTISET
COLOUR	::=transition{ID}{NAME GUARD PRIO T_WEIGHT} ::=empty guard{BOOLEXP ::=ML_EXPRESSION ::=EXPRESSION ::={ EXPRESSION }and{ EXPRESSION}   EXPRESSION }or{ EXPRESSION }= EXPRESSION }<>{  not{ EXPRESSION }  VALUE  ML_FUNCTION  empty ::=value{true}  value{false}
COLOURSET MULTISET	
MSLIST	
TRANSITION	
GUARD	
BOOLEXP ML_EXPRSSION EXPRESSION	

نیز به آن الحاق شده است. برای نشان دادن الگوی طراحی بر اساس چارچوب مبتنی بر زبان نشانه گذاری توسعه پذیر بایستی این نمودار را توسط فیلتر ورودی به کدهای XML تبدیل کنیم. سپس با استفاده از الگوریتم کدها را به ساختار مشترک بسط‌های شبکه پتری تبدیل کرد. آنگاه فیلتر خروجی این کدها را متناسب با شبکه پتری مورد نظر تبدیل می‌کند. کد XML عمومی نمودار ترتیب را با استفاده از نرم‌افزار VPUML به دست آورد و سپس با ایجاد تغییراتی در آن کد XML مربوط به نمودار را به عنوان خروجی به دست آورد. (شکل ۲) بخشی از کد ساختار مشترک بسط‌های شبکه پتری را برای مورد مطالعه با توجه به الگوی طراحی MVC نشان داده شده است. این کد مطابق با گرامر عمومی برای بسط‌های شبکه پتری به دست آمده است. برای توضیح بیشتر بخشی از کد XML مربوطه می‌توان بیان داشت، طبق تبدیلات مولفه‌ها به شبکه پتری می‌توان مکان‌ها و گذارها را تعیین نمود و حال برای اینکه بتوان یک شبکه پتری رنگی از این کد به دست آورد به برخی مکان‌ها یک مقدار اولیه داده شده به عنوان توکن ورودی که هر توکن دارای رنگی می‌باشد. همچنین به هر کمان یک وزن اختصاص داده شده که اینها در نمودار ترتیب می‌تواند زمان پاسخ را محاسبه نماید. هر یک از این وزن‌ها قابل تغییر می‌باشد و به صورت فرضی وارد شده است. خروجی حاصل از کد XML، شبکه پتری رنگی حاصل از نمودار ترتیب در (شکل ۳) قابل مشاهده می‌باشد. این خروجی با یک سری تغییرات قابل تبدیل به هر یک از بسط‌های دیگر شبکه پتری می‌باشد.

### نتیجه‌گیری

یکی از مهم‌ترین سطوح ارزیابی نیازهای غیروظیفه‌مندی در طی فرایند توسعه نرم‌افزار مرحله معماری نرم‌افزار است. در این مقاله سعی بر این شده است تا قبل از ورود به مرحله پیاده‌سازی نیاز کارایی در الگوی طراحی MVC را بر اساس یک چارچوب مبتنی بر XML بیان کند و این چارچوب بسط‌های شبکه پتری را پشتیبانی می‌کند. با این کار می‌توان الگو را با هر یک از بسط‌های مدل رسمی شبکه پتری به تبدیل

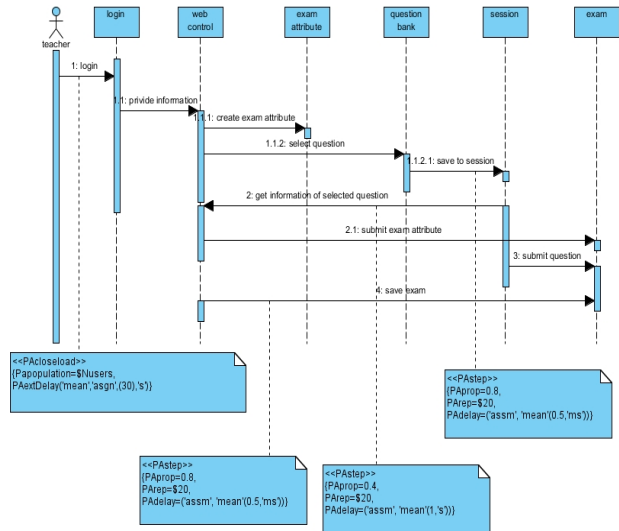
### مطالعه موردی

جهت صحت بررسی و مقاله انجام شده یک سیستم آزمون برخط برای دانشجویان در نظر گرفته شده است [۱۱]. در این سیستم (شکل ۱) نمودار ترتیب از سیستم که مربوط به ایجاد آزمون می‌باشد را نشان می‌دهد. که کلیشه‌های مربوط به کارایی

```

sequenc.xml - Notepad
File Edit Format View Help
<place name=session, init=1, color=0 ></place>
<place name=exam, init=1, color=0 ></place>
<place name=p1></place>
<place name=p2></place>
<place name=p3></place>
<place name=p4></place>
<arc name=a1, weight=x>
<from view teacher to t1></from>
</arc>
<arc name=a2, weight=sh+1>
<from t1 to p2></from>
</arc>
<arc name=a3, weight=sh+1>
<from t1 to p1></from>
</arc>
<transition name=t1></transition>
<transition name=t2></transition>
<transition name=t3></transition>
<transition name=t4></transition>
<transition name=t5></transition>
<transition name=t6></transition>
<transition name=t7></transition>
<transition name=t8></transition>
<transition name=t9></transition>
<place name=p5></place>
<place name=p6></place>
<place name=p8></place>
<place name=p9></place>
<place name=p10></place>
<place name=p11></place>
<place name=p12></place>
<place name=p13></place>
<place name=p14></place>
<place name=p15></place>
<place name=p16></place>
<place name=p17></place>
<place name=p18></place>
<place name=p19></place>
    
```

نمود و سپس آن را ارزیابی کرد.  
 به عنوان کارهای آینده می‌توان به دست‌آوردن نیاز  
 غیروظیفه‌مندی دیگر همانند نیاز قابلیت‌اطمینان را با نیاز  
 کارایی در این چارچوب یکپارچه نمود، همچنین الگوهای  
 دیگر را نیز به عنوان کارهای آتی مورد استفاده قرار داد.

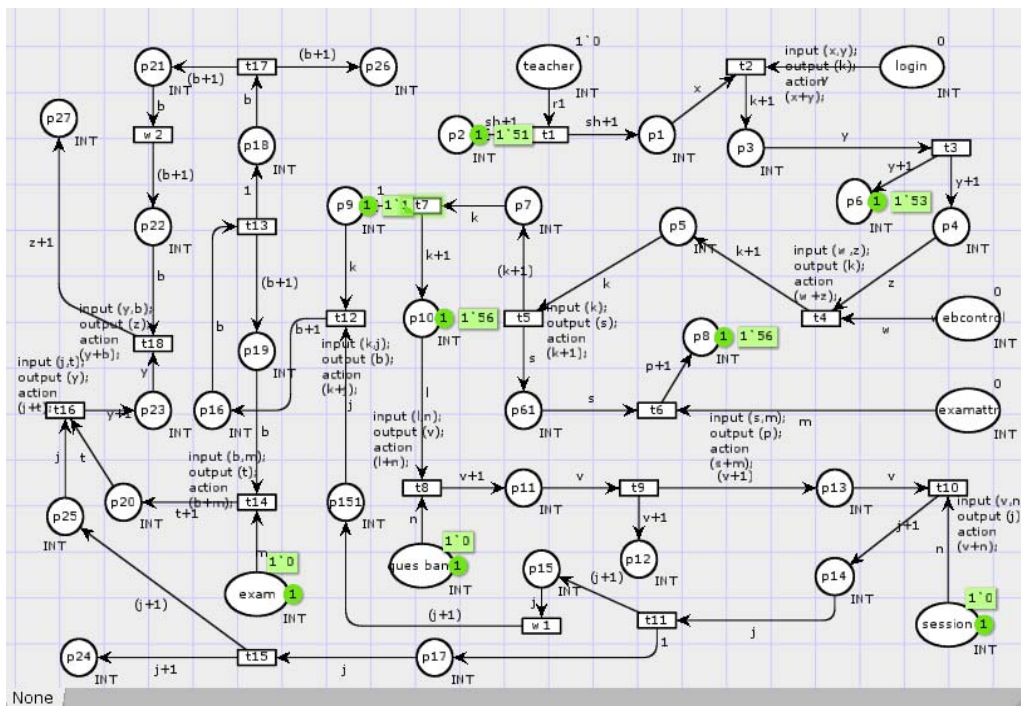


شکل ۲- بخشی از کد XML معادل بسط‌های شبکه پتری

نمودار ترتیب

شکل ۱- نمودار ترتیب ایجاد آزمون برخط به همراه

کلیشه‌های کارایی



شکل ۳- شبکه پتری حاصل از نمودار ترتیب

578-580.2012.

7. Grassi, V., Mirandola, R., & Sabetta, A. " Filling the gap between design and performance/reliability models of component-based systems: A model-driven approach". *Journal of Systems and Software*, Vol.80, No. 4, PP. 528-558.2007.
8. Cortellessa, V., Di Marco, A., Inverardi, P., Mancinelli, F., & Pelliccione, P." A Framework for the Integration of Functional and Non-functional Analysis of Software Architectures". *Electronic Notes in Theoretical Computer Science*, Vol.116, No.1, PP.31-44.2005.
9. Emadi, S., shams, F., & vaziri, s. "General syntax for extensions of petri net," the third international confrence on mathematical sceinces, PP.1133-1138.2008.
10. Liu, C., & Wang, K. "An Online Examination System based on UML Modeling and MVC Design Pattern," the First International Conference on Communication Technology (ICCECT), Liaoning, PP.815-817.2012.

۱۱. سیما عمادی، و فریدون شمس "ارائه روشی برای ارزیابی نیازهای غیروظیفه‌مندی در فرایند توسعه نرم‌افزار" *مجله فنی مهندسی دانشگاه آزاد اسلامی واحد مشهد*، ۱۳۸۷، صفحات. ۷۹-۶۵

۱۲. سیما عمادی "ارائه مدلی به منظور بررسی قابلیت اجرایی معماری نرم افزار" (درجه دکتری) *دانشگاه آزاد اسلامی واحد علوم و تحقیقات تهران*. ۱۳۸۵.

## مراجع

1. Balsamo, S., Marco, A. D., Inverardi, P., & Simeoni, M. "Model-Based Performance Prediction in Software Development: A Survey". *Transactions On Software Engineering*, Vol.30, No.5, PP. 295-310 . 2004.
2. Emadi, S., & Shams, F." An approach to non-functional requirements analysis at software architecture level". *8th Conference International Computer and Information Technology*, Sydney, NSW, PP.736-741. 2008.
3. Bass, L., Clements, P., & Kazman, R. (2012). "Software Architecture in Practice", 3th Edition, Addison Wesley, the United States.2012.
4. Buschmann, F., Schmidt, D., Stal, M & ., Rohnert, H. "Pattern-Oriented Software Architecture a System of Pattern", 1th Edition, John'Wiley & Sons Ltd, England.2001.
5. Yin, X., Zhenga, W., Zhanga, M. & Zhanga, J. "A modularized operator interface framework for Tokamak based on MVC design pattern,"*Fusion Engineering and Design* Vol. 89, No.1, PP. 628-632. 2014.
6. Zhang, C., & Shen, W. "The Studies on Integrated Development Approach Based on the Architecture and MVC Design Pattern," *First Conference of Electrical & Electronics Engineering Symposium*. Kuala Lumpur, PP.