یک روش ترکیبی مبتنی بر مفاهیم بلوک تابع و شئ‌گرایی برای مدل‌سازی سیستمهای کنترل بلادرنگ توزیعی

روزبه واحدی‌پور[۱]، محسن صدیقی مشکنانی[۲]
۱- کارشناس ارشد مهندسی برق، کنترل، دانشگاه صنعتی اصفهان، rvahedipour@ec.iut.ac.ir
۲- استادیار گروه کامپیوتر، دانشکده فنی پردیس کیش، دانشگاه صنعتی شریف، sadighim@gmail.com

چکیده

در این مقاله یک متدولوژی مدل‌سازی جدید برای سیستمهای کنترل بلادرنگ توزیعی، ارائه می‌شود که حاصل بکارگیری همزمان مدل بلوکهای تابع استاندارد IEC 61499 مربوط به مدل‌سازی سیستمهای کنترلی صنعتی و روشهای شیءگرا برای تولید نرم‌افزار است. روش مدل‌سازی جدید، OO+FB نامیده شده است. این متد، ویژگیهای توزیع و محدودیتهای بلادرنگ را با دقت مناسبی مدل نموده و زمینهٔ تحلیل و طراحی کنترل کننده‌ها را فراهم می‌آورد. متدولوژی جدید، با بهره مندی از نقاط مثبت هر یک از دو روش پایه و در جهت حذف نقاط ضعف آنها، چارچوبی مناسب و جامع برای رویارویی با فرآیندهای صنعتی امروزی که اغلب خصوصیات توزیعی و بلادرنگ دارند، مهیا می‌سازد. در این مقاله با معرفی مراحل مختلف روش OO+FB و تشریح مزایایی که OO+FB به همراه می‌آورد، یک رهیافت مدل‌سازی مدون پیش روی قرار می‌گیرد.

کلمات کلیدی

کنترل صنعتی، سیستمهای بلادرنگ توزیعی، بلوک تابع، شیءگرایی.

۱- مقدمه

سیستمهای کنترل صنعتی مدرن به سرعت به سمت توزیعی شدن پیش می‌روند. از طرفی، در عمل تمامی این سیستمها دارای محدودیتهای زمانی نرم یا سخت می‌باشند. بنابراین در صنعت به طور وسیعی با کنترل بلادرنگ توزیعی روبرو هستیم. نخستین گام در برخورد با چنین سیستمهایی، داشتن یک مدل مناسب از آن است که در اولویت اول، یک توصیف گرافیکی و پس از آن یک مدل ریاضی مورد نظر است. معتبرترین روشی که برای مدل‌سازی سیستمهای کنترل توزیعی صنعتی مرسوم شده است، استفاده از استاندارد IEC 61499 و مفهوم بلوک تابع است[۱]. این استاندارد با پرداختن به مدل‌های مختلف سیستم، در همه‌ی سطوح، از کلی تا بسیاری از جزئیات، توصیف مناسبی با سطوح تجرید گوناگون در اختیار قرار می‌دهد[۲]. اما این استاندارد در برخورد با ویژگیهای زمانی سیستمهای بلادرنگ دچار نقصان است و ابزاری برای مدل نمودن این محدودیتهای دارا نمی‌باشد. علاوه بر ویژگیهای بلادرنگ ناشی از خواسته‌های غیرعملکردی مسئله‌ی کنترل، خصوصیات زمانی دیگری نیز در رابطه با خود بلوکهای تابع وجود دارد که استاندارد IEC 61499 بدون پرداختن به جزئیات از کنار آنها می‌گذرد[۳].

در مقابل روش بلوک تابعی، رهیافت دیگری در این مقاله معرفی می‌شود که از اصول مدل شئ برای مدل‌سازی سیستم کنترل استفاده می‌نماید. در این راستا، زبان مدل‌سازی یکپارچه، UML، به عنوان ابزاری نیرومند در فرآیند مدل‌سازی به کارگرفته می‌شود[۴، ۵]. مهمترین ویژگی این روش، امکانات بسیار مناسب برای مدل‌سازی سیستمهای بلادرنگ است. تعمیمی از زبان UML تحت عنوان UML برای سیستمهای بلادرنگ یا UML-RT موجود است که با جزئیات فراوان، محدودیتهای زمانی را وارد مدل می‌نماید[۶].

اما بنا به دلایلی که در مقاله ذکر خواهد شد، مدل شئ نمی‌تواند بطور کامل پاسخگوی نیاز مهندسان کنترل و ابزار دقیق

Archive of SID

باشد و دارای نقاط ضعفی است که جامعیت آن را در راستای مدل‌سازی سیستمهای کنترل بلادرنگ توزیعی از بین می‌برد[۷]. بدین دلیل، روش مدل‌سازی جدید OO+FB حاصل از تلفیق مدل بلوک تابع یا به اختصار FB، و روش شئ‌گرا یا به اختصار OO، در این مقاله پیشنهاد و توسعه داده شده است[۸]، که مبنای اصلی را مدل بلوک تابع، به عنوان استاندارد رایج در جامعه‌ی مهندسی کنترل نهاده و نقاط ضعف آن را در برخورد با سیستمهای کنترل بلادرنگ توزیعی، به خوبی پوشش داده است. ضمن ارزیابی عملکرد هریک از روشهای مدل‌سازی بلوک تابعی و شئ‌گرا در مدل کردن سیستمهای کنترل بلادرنگ توزیعی، تحلیل کاملی از ویژگیهای روش OO+FB و نحوه‌ی برطرف کردن نقایص دو روش اولیه در این متد جدید، انجام گرفته است. در ادامه، قسمت ۲ توصیفی گذرا از سیستمهای بلادرنگ توزیعی در اختیار می‌گذارد. سپس در قسمت ۳ به معرفی روشهای مدل‌سازی بلوک تابعی و شئ‌گرا برای سیستمهای کنترل صنعتی می‌پردازیم. بخش ۴ به تشریح روش جدید OO+FB جهت مدل‌سازی سیستمهای بلادرنگ توزیعی اختصاص یافته است. در این قسمت، فازهای مختلف روش مذکور بیان خواهد شد. بخش ۵ به ارائه مثالی از مدل‌سازی یک سیستم کنترل صنعتی توزیعی در حضور برخی محدودیتهای بلادرنگ، با استفاده از روش OO+FB می‌پردازد. در نهایت قسمت ۶ بر مبنای مثالی که ارائه شده، به توصیف مزایای متعدد روش جدید و چگونگی برطرف شدن نقایص هر یک از دو مدل بلوک تابعی و شئ-گرا در برخورد با سیستمهای بلادرنگ توزیعی، در روش مدل‌سازی OO+FB مطرح شده است.

۲- کنترل بلادرنگ توزیعی

بکارگیری علمیات توزیعی برای استفاده از همروندی در سیستمهای بلادرنگ، منجر به یک سیستم «بلادرنگ توزیعی» می‌گردد. برای مثال، یک سیستم بلادرنگ که می‌تواند همه‌ی سقف زمانی‌ها را با پنج فرآیند همزمان در یک پردازنده برآورده می‌کند، ممکن است نتواند دو فرآیند اضافی دیگر که بطور ناگهانی لازم می‌شوند، مدیریت کند. بنابراین، در چنین موردی یک سیستم توزیعی می‌تواند راه حل مناسبی باشد. در این مثال، می‌توان یک پردازنده را به پردازش آن دو فرآیند اضافی که بصورت غیرقابل پیش‌بینی ظاهر می‌شوند، اختصاص داد و یک پردازنده را نیز به همان پنج فرآیند همزمان اصلی محدود نمود.

یک سیستم بلادرنگ توزیعی، سیستم توزیعی است که صحت عملکرد آن به برآورده کردن محدودیتهای زمانی، درست مانند نیازهای منطقی، بستگی دارد[۹]. عملیات توزیعی ممکن است برای بهبود پاسخ زمانی یک سیستم بلادرنگ یا بالاتربردن قابلیت اطمینان آن مورد استفاده قرار گیرد. با تقسیم یک سیستم بلادرنگ بزرگ به مجموعه‌ای از فرآیندها، فرآیندهای حاصل از تجزیه می‌توانند بطور همزمان روی منابع مختلف اجرا شوند و از طریق ارتباطات بین فرآیندی و همزمان‌سازی، هدف نهایی سیستم را برآورده کنند. بنابراین، پاسخ زمانی را می‌توان با عملیات موازی بهبود داد. قابلیت اطمینان نیز توسط سیستمهای توزیعی، افزایش می‌یابد.

۳- مدل‌سازی سیستمهای کنترل صنعتی

منظور از مدل‌سازی در این مقاله، مدل‌سازی گرافیکی است. یعنی توصیف یک سیستم و ارتباط اجزای آن از طریق شکلها و نمودارهای گرافیکی انجام می‌شود که دیدگاه شهودی و البته مناسبی، از طرزکار سیستم در اختیار مهندسان قرار می‌دهد. یک مدل گرافیکی، اگرچه مبنای طراحی‌های ریاضی نمی‌تواند باشد، ولی بدون شک باید در کنار یک مدل ریاضی قرار گیرد تا بتوانند با هم یک مدل کامل از سیستم ارائه دهند و به این ترتیب عملکرد سیستم قابل تحلیل باشد. برای مدل‌سازی فرآیندهای صنعتی، دو روش مهم موجود، یکی مدل بلوک تابعی و دیگری مدل شئ و نمودارهای UML است.

۳-۱- مدل‌سازی مبتنی بر بلوک تابع

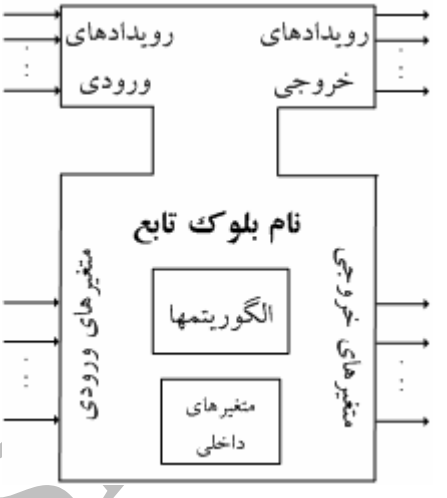
کمیتهٔ بین‌المللی الکتروتکنیک، IEC، استاندارد IEC 61499 را برای تعریف چگونگی استفاده از بلوکهای تابع در فرآیندهای صنعتی توزیعی و سیستمهای اندازه‌گیری و کنترل توزیعی را توسعه داده است. این استاندارد کمک زیادی در جهت حل بخش عمده‌ای از مشکل «هماهنگی معنایی» ارائه می‌دهد.

در سیستمهای صنعتی، بلوک تابع، مفهومی است که برای تعریف اجزاء نرم‌افزاری مقاوم و دارای قابلیت استفاده مجدد معرفی شده‌اند. یک بلوک تابع ممکن است یک راه حل نرم‌افزاری برای یک مسئلهٔ کوچک مثل کنترل یک شیر و یا یک مسئله‌ی در مقیاس بسیار بزرگ و در حد کنترل یک واحد بزرگ از یک فرآیند، مانند کنترل کامل خط تولید باشد.

شکل (۱) نماد یک بلوک تابع و اجزای آن را نمایش می‌دهد. یک بلوک تابع زمانی وارد عملیات می‌شود که رویداد ورودی خاصی آن را آتش کند. در این هنگام، بلوک بر مبنای متغیرهای ورودی و با استفاده از الگوریتمی که در اختیار دارد، متغیرهای خروجی را برای استفاده‌ی سایر بلوکها تولید می‌کند و جهت ادامه عملیات، سیگنالهایی را روی خطوط رویدادهای خروجی ارسال می‌کند تا بلوکهای بعدی فعال شوند. استفاده از بلوک تابع، با مؤلفه‌ای کردن مدل سیستم، علاوه بر قابل درک‌تر کردن مدل، پیاده‌سازی و نگهداری را بهبود می‌بخشد. به علاوه، در مدل‌سازی سیستمهای توزیعی، با تعریفی که استاندارد IEC 61499 برای بلوک تابع ارائه

Archive of SID

کرده است، فرآیند مدل‌سازی بدون هیچ ابهامی قابل انجام است.



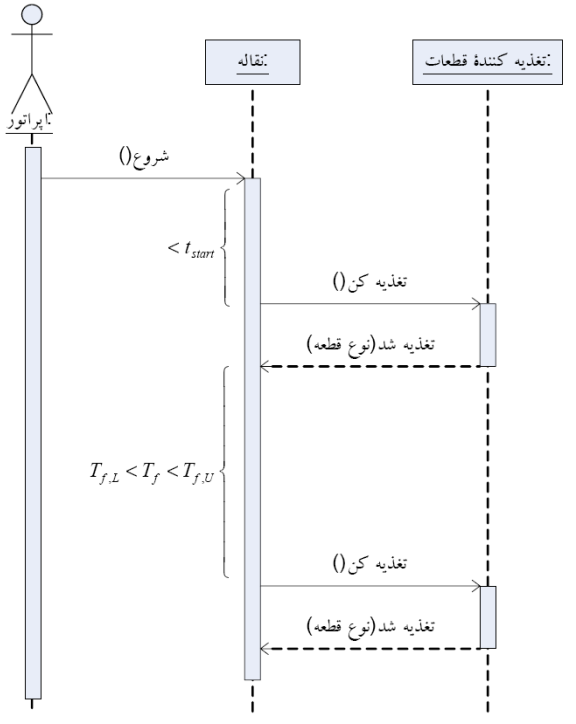
شکل (۱): نماد بلوک تابع و قسمتهای مختلف آن

۳-۲- مدل‌سازی شیءگرا

رهیافت شئ‌گرا در اصل برای سیستمهای نرم افزاری ابداع شد ولی ایده‌ی آن کلی است و با دید به جهان واقعی است[۱۰]. هم اکنون، مدل شئ، بطور گسترده‌ای در مدل‌سازی سیتسمهای فیزیکی به کار گرفته می‌شود[۱۱، ۱۲].

بکارگیری مدل‌سازی شئ‌گرا با مفاهیم پایه، به خاطر نیاز به بیان محدودیتهای زمانی، با مشکلاتی مواجه می‌شود. در جهت رفع این مشکلات، گسترشی برای مدل‌سازی شئ‌گرا و زبان UML با عنوان «زبان مدل‌سازی یکپارچه برای سیستمهای بلادرنگ» یا به اختصار UML-RT ارائه شده است[۱۳، ۱۴]. بدین ترتیب می‌توان در چارچوب یک روش استاندارد و فراگیر به مدل‌سازی شئ‌گرای سیستمهای کنترل بلادرنگ پرداخت و از مزایای مدل شئ، در این سیستمها نیز بهره گرفت[۱۵، ۱۶].

شکل (۲) مثالی از نمودار ترتیب را نمایش می‌دهد که مربوط به مکانیزم تغذیه قطعات در یک نورد است. محدودیتهای زمانی مختلفی که در تحلیل سیستم مورد نظر مشخص می‌شود، همانگونه که در شکل (۲) مشاهده می‌شود، در نمودارهای مدل وارد شده است[۸].



شکل (۲): نمودار ترتیب (از نمودارهای UML) با نمایش محدودیتهای زمانی. در این مثال، مکانیزم تغذیه قطعات در یک سیستم نورد مدل شده و محدودیتهای بلادرنگ روی آن به نمایش درآمده است.

۴- روش مدل‌سازی OO+FB

متدولوژی مدل‌سازی OO+FB دارای چهار فاز است:

- فاز اول: مهندسی خواسته‌ها؛
- فاز دوم: مدل‌سازی استاتیک و مدل‌سازی توزیع؛
- فاز سوم: مدل‌سازی دینامیک؛
- فاز چهارم: آزمایش مدل.

مانند آنچه در شئ‌گرایی مرسوم است، از تکامل «تکراری - افزایشی» برای مدل‌سازی استفاده می‌کنیم. بدین معنی که هر فاز در چند تکرار کامل می‌شود ولی هر تکرار دارای محصول می‌باشد، گرچه این محصول ممکن است تمام جزئیات لازم را در بر نداشته باشد و نیاز به تکرارهای دیگری برای تکمیل آن باشد. بدین‌ترتیب نمی‌توان مرز دقیقی بین فازهای مختلف در نظر گرفت، اما بطور مشخص، از لحاظ بُعد زمانی، فازی که تقدم دارد با تکیه‌ی بیشتری مورد توجه قرار می‌گیرد و سایر آنها بصورتی کمرنگ در نظرگرفته می‌شوند. اما به هر حال امکان کار موازی روی فازهای مختلف وجود دارد. شکل (۳) فرآیند مدل‌سازی OO+FB را بصورت تصویری نمایش می‌دهد.

۴-۱- فاز اول، مهندسی خواسته‌ها

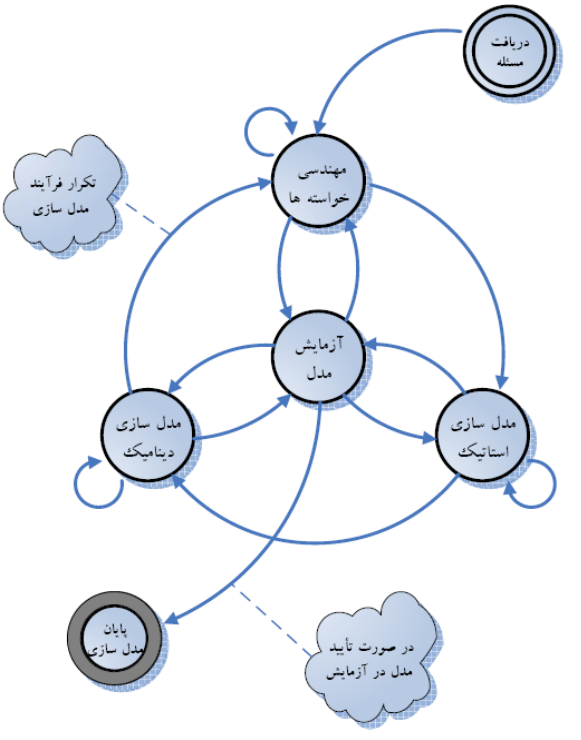
۱) در متدولوژی OO+FB، گامهایی که برای فاز مهندسی خواسته‌ها در نظر گرفته شده است، عبارتند از: لیست کردن خواسته‌های عملکردی و غیرعملکردی؛
۲) رسم نمودار مورد کاربرد و نوشتن سناریوی هر یک از موارد کاربرد؛
۳) تعیین محدودیتهای زمانی و توزیع سیستم؛
۴) بازنگری و بهبود محصولات مراحل قبلی.

Archive of SID

محصولات مورد نظر در فاز مهندسی خواسته‌ها، شامل این موارد می‌شود:

۱) مستندات خواسته‌های عملکردی و غیرعمکردی؛
۲) نمودار مورد کاربرد؛
۳) سناریوها.

درنهایت اینکه اگر از اشکال منابع دیگری در مقاله خود استفاده می کنید، ضروری است که نام و نشانی منبع در زیر شکل ذکرشده یابه شماره مرجع درفهرست مراجع ارجاع شود.



شکل (۳): فرآیند مدل‌سازی OO+FB[۸]

۴–۲– فاز دوم، مدل استاتیک و مدل توزیع

در متدولوژی OO+FB، گامهایی که برای فاز مدل‌سازی استاتیک و توزیع در نظر گرفته شده است، عبارتند از:

۱) تعیین کاربردها و زیرکاربردها؛
۲) رسم نمودار استقرار (مدل توزیع)؛
۳) رسم نمودار P&ID برای فرآیند؛
۴) تشخیص بلوکهای تابع و تعریف آنها؛
۵) تعیین روابط ساختاری بلوکها، شامل ارثبری، تجمع و ترکیب (به مفهوم رایج در مدل شیء و نمودار کلاس)؛
۶) بازنگری و بهبود محصول هریک از گامهای قبلی.

محصولات مورد نظر در فاز مدل‌سازی استاتیک، شامل این موارد می‌شود:

۱) مدل کاربرد (از مدلهای IEC 61499)؛
۲) نمودار استقرار (از نمودارهای UML)؛
۳) نمودار P&ID؛
۴) لیست و تعریف بلوکهای تابع؛
۵) نمودار روابط استاتیکی بلوکها (نمودار کلاس)؛



شکل (۴): نمودار فعالیت برای مدل‌سازی استاتیک

شکل (۴) یک نمودار فعالیت (از نمودارهای UML که توسعه‌ای از مفهوم عمومی فوچارت است)، برای فاز مدل‌سازی استاتیک نشان می‌دهد. نمایش هر فاز با یک نمودار فعالیت درک روند اجرای آن را بهتر میسر می‌سازد که به خاطر طولانی شدن جزئیات، در اینجا تنها یک نمونه از این نمودارها، تنها برای فاز مدل استاتیک آورده شده است.

۴–۳– فاز سوم، مدل دینامیک

در متدولوژی OO+FB، گامهای فاز مدل‌سازی دینامیک بدین صورت در نظر گرفته شده است:

۱) رسم نمودار همکاری مکانیزمها و منظورکردن ویژگیهای زمانی (به کمک زبان OCL)؛
۲) رسم نمودار فعالیت برای هر مکانیزم و رسم نمودار حالت برای مکانیزمهای وابسته به حالت؛

۳) رسم نمودار ترتیب برای هر یک از مکانیزمها با نمایش محدودیتهای بلادرنگ (مطابق UML-RT)؛
۴) بازنگری و بهبود محصول هریک از گامهای قبلی.

محصولات این فاز عبارتند از:
۱) نمودارهای همکاری با نمایش ویژگیهای زمانی؛
۲) نمودارهای فعالیت مکانیزمهای مختلف سیستم؛
۳) نمودارهای حالت برای مکانیزمهای وابسته به حالت؛
۴) نمودارهای ترتیب به همراه نمایش محدودیتهای زمانی.

۴-۴- فاز چهارم، آزمایش مدل

مراحلی که برای فاز چهارم، یعنی آزمایش مدل، در متدولوژی OO+FB در نظر گرفته شده است، عبارتند از:
۱) تعیین یک الگوی از پیش آماده‌ی آزمایش مدل شئ[۱۷]؛
۲) آزمایش و ارزیابی مدل براساس الگوی انتخابی؛
۳) امتیاز دادن به مدل بر اساس نتایج حاصل از آزمایش؛
۴) بررسی رضایت بخش بودن نتایج آزمایش؛
۵) بررسی انطباق مدل با استانداردهای کنترل صنعتی؛
۶) بازنگری در فازهای قبلی مدل‌سازی و بهبود مدل؛

محصولات فاز آزمایش مدل عبارتند از:
۱) مستندات الگوی انتخابی برای آزمایش مدل؛
۲) مستندات نتایج آزمایش؛
۳) تحلیل آماری و کیفی نتایج آزمایش و امتیاز دادن به قسمتهای مختلف مدل؛
۴) نتیجه‌گیری‌های حاصل از تحلیل آماری و کیفی؛
۵) مستندات مقایسه‌ی مدل با استانداردهای موجود؛
۶) مستندات مربوط به نقاط ضعف مدل برای بازنگری.

۵- مثال‌های مدل‌سازی با OO+FB

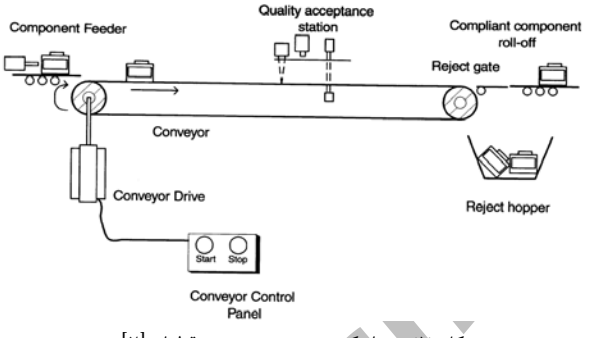
در این قسمت، به بیان دو مثال برای توضیح ویژگیهای ممتاز متدولوژی OO+FB در مدل‌سازی سیستمهای کنترل صنعتی، پرداخته می‌شود. در هر دو مثال، تأکید بر این است که مدل‌سازی با استفاده از استاندارد IEC 61499 به تنهایی، در مقیاس سیستمهای واقعی، به سمت پیچیدگی زیاد می‌رود و برای عموم مهندسان غیرقابل درک‌تر می‌شود. در حالی که به‌کارگیری تکنیکهای شیءگرایی در قالب زبان UML، در کنار امکانات استاندارد مذکور، با توجه به عمومیت بیشتر و نمایش مفهوم‌تر نمودارهای UML، مدل را برای مهندسان مختلف قابل درک‌تر خواهد نمود.

۵-۱- سیستم نورد تست قطعات

یک سیستم نورد تست قطعات را به عنوان مثال، درنظر می‌گیریم. در این سیستم، یک نورد، قطعات تولیدی را به عنوان ورودی می‌گیرد و از یک ایستگاه تست کیفیت عبور می‌دهد و در صورت پذیرفته شدن قطعه، آن را به طرف انبار هدایت می‌کند و در غیر اینصورت آن را به واگن قطعات رد شده می‌فرستد. نتایج آزمایش‌ها در یک پایگاه داده خارجی که روی یک ماشین سرور راه دور قرار دارد، ثبت می‌شود.

نوار با یک موتور راه انداز به حرکت در می‌آید. قطعات یکی یکی توسط یک تغذیه کننده، به نقاله تحویل داده می‌شوند و از زیر ایستگاه تست کیفیت عبور می‌کند در ادامه مسیر، اگر قطعه مورد قبول باشد، قطعه روی نوار دیگری قرار داده می‌شود که قطعه را به انبار منتقل می‌کند و در صورتی که در آزمایش رد شود، دریچهٔ ردکردن، عمل می‌کند و قطعه داخل واگن مخصوص قرار می‌گیرد. عمل ورود قطعه به نقاله، تست و الصاق بارکد و در نهایت خروج قطعه از نقاله به صورت پی در پی صورت می‌گیرد. راه اندازی و توقف کل سیستم با یک پانل ساده که دارای دکمه‌های «Start» و «Stop» می‌باشد، انجام می‌گیرد.

شکل (۵) یک طرح کلی شامل اجزای اصلی سیستم نورد تست قطعات را نمایش می‌دهد.



شکل (۵): شماتیک سیستم نورد تست قطعات[۲]

مرحله بعدی تبیین خواسته‌هاست (فاز اول) که در اینجا یک نمودار «مورد کاربرد» از مجموعه نمودارهای UML برای این سیستم رسم می‌شود. این نمودار در شکل (۶) مشاهده می‌شود.

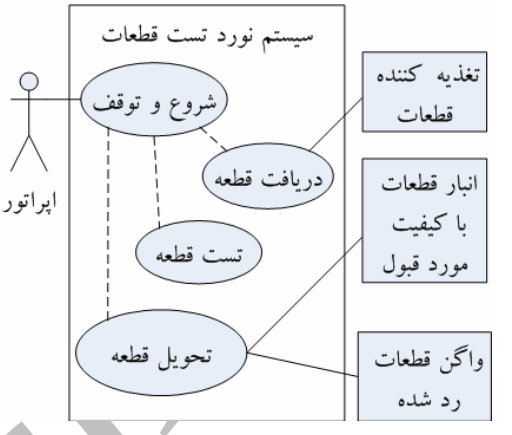
در مرحله بعد، به تعریف بلوکهای تابع و در نهایت، براساس تحلیلی که از خواسته‌ها به عمل آمده، به رسم دیاگرام بلوک تابع پرداخته می‌شود. بلوکهای تابعی که برای این سیستم قابل تعریف هستند، عبارتند از:

پانل فرمان، تغذیه کننده، انبار، ایستگاه تست، انتقال دهنده به انبار، دریچه رد قطعه، واگن قطعات رد شده، درایو کنترل نقاله، سرور دیتابیس و ثبت کننده در پایگاه داده.

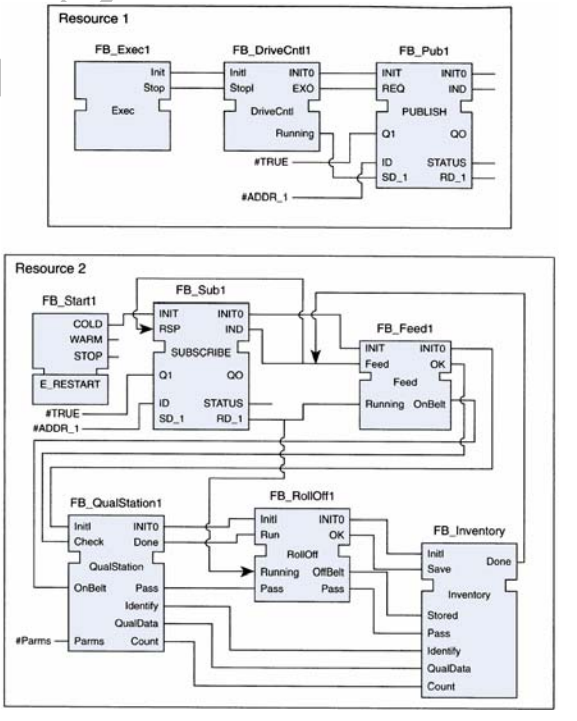
یک دیاگرام بلوک تابع قابل رسم برای سیستم نورد تست قطعات که در مرجع [۴] پیشنهاد شده است، در شکل (۷) مشاهده می‌شود. همانگونه که در شکل مشخص است، سیستم توزیعی دارای دو منبع است، یکی مجموعه خود سیستم نورد و دیگری سرویس

Archive of SID

دهنده پایگاه اطلاعاتی که اطلاعات حاصل از تست کیفیت قطعات برای آن ارسال و در آن ثبت می‌گردد.



شکل (۶): نمودار مورد کاربرد برای تحلیل خواسته‌های سیستم کنترل نورد تست قطعات که طرح آن در شکل (۵) ارائه شده است.[۸]

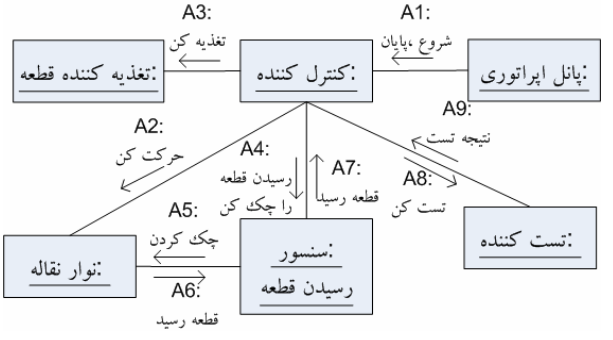


شکل (۷): دیاگرام بلوک تابعی سیستم نورد تست قطعات با در نظرگرفتن توزیع بین منابع، براساس استاندارد IEC 61499.[۲]

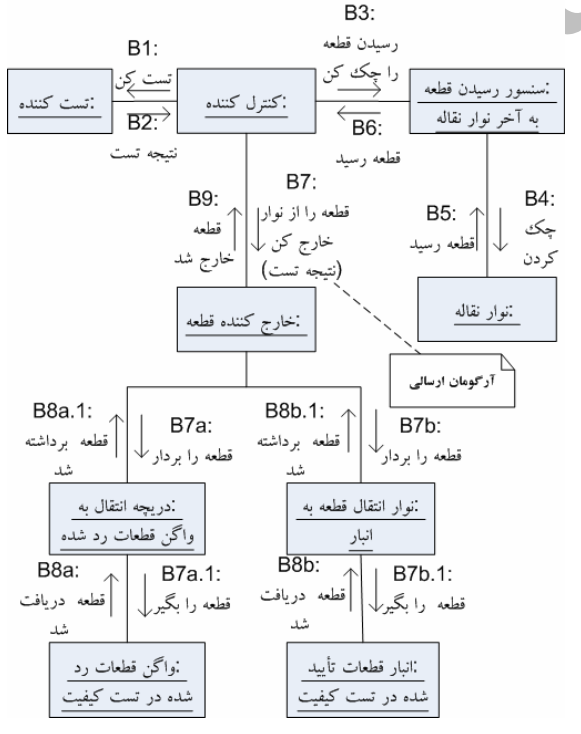
در مراحل بعدی، به مدل‌سازی دینامیک سیستم کنترل پرداخته می‌شود. عمده‌ترین مزایای روش OO+FB در این مراحل روشن می‌شود که با بکارگیری امکانات زبان مدل‌سازی یکپارچه، UML، و البته اضافه کردن نمایش ویژگیهای بلادرنگ بودن سیستم به آن، فضای مدل‌سازی سیستمهای کنترل صنعتی را بسیار منعطف می‌سازد. این در حالی است که استفاده از مدلهای مختلف استاندارد IEC 61499 به تنهایی، منجر به پیچیدگیهایی می‌شود که امکان درک عمومی آن را کاهش می‌دهد و در نتیجه پیاده‌سازی و نگهداری، مشکل‌تر می‌شود.

در مدل‌سازی دینامیک، بطور خاص، نمودارهای همکاری و مهم‌تر از آنها، نمودارهای ترتیب بسیار به باز شدن مدل و تفهیم آن کمک می‌کنند.

شکل‌های (۸) و (۹) دو نمودار همکاری مهم مرتبط با سیستم مورد مطالعه را نشان می‌دهند. لازم به ذکر است که اینجا هم طبق اصول شیءگرایی، با وجود امکان وارد شدن به جزئیات زیاد، با انتخاب یک سطح تجرید مناسب (انتخاب سطح مناسبی از وارد شدن به جزئیات)، می‌توان از پرداختن به بعضی جزئیات غیرضروری پرهیز نمود.



شکل (۸): نمودار همکاری برای فرآیند تغذیه قطعات و ارسال فرمان به ایستگاه تست، برای آغاز عمل تست.[۸]



شکل (۹): نمودار همکاری برای فرآیند تست قطعات و سپس هدایت آنها به انبار در صورت موفقیت در تست و یا ارسال آنها به واگن قطعات رد شده، در صورت پذیرفته نشدن در تست.[۸]

شکل (۱۰) نمودار ترتیب کلی فرآیند آغاز به کار، تغذیه قطعات، تست قطعات و انتقال آنها به انبار یا واگن قطعات رد شده را نمایش می‌دهد. نمادهایی که برای بُعد زمان به محور عمودی نمودار ترتیب اضافه شده است، جزئی از UML بصورت استاندارد نیست و در واقع در UML-RT یعنی زبان مدل‌سازی یکپارچه برای سیستمهای بلادرنگ است. این یک نمایش نوعی از این نمودارهاست. با تحلیل ریاضی محدودیتها که ضمیمه‌ی چنین نموداری می‌شود، در واقع مدل نمودار ترتیب کامل می‌شود که مبحث طولانی و قابل بحث بطور جداگانه از این مقاله است.

یک نمونه خلاصه شده از این تحلیل زمانی، در نمودار ترتیب شکل (۱۱) که برای تغذیه قطعات به نقاله رسم شده، به نمایش در آمده است.

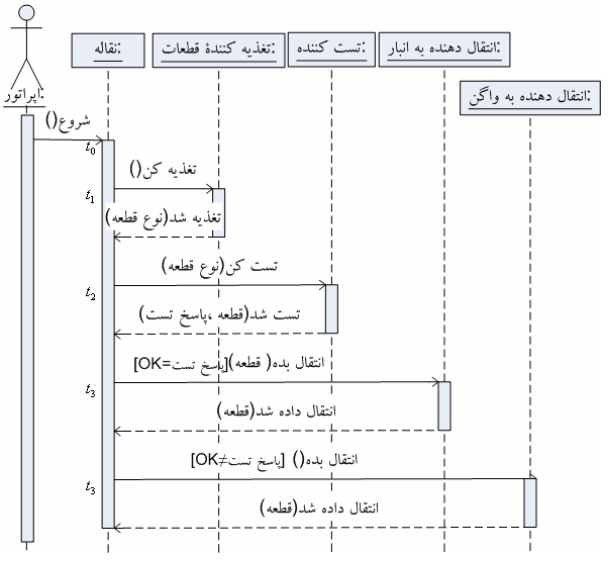
برای مثال یک زمان tstart از لحظه شروع به کار سیستم طول می‌کشد تا سرعت نوار نقاله به مقدار ثابت خود برسد تا بتوان قطعات را روی آن قرار داد. این در نمودار شکل (۱۱) با ذکر یک محدودیت از زمان شروع تا زمان اولین تغذیه نشان داده شده است.

به علاوه در نمودار ترتیب شکل (۱۱) یک محدودیت روی فاصله زمانی تغذیه شدن دو قطعه متوالی روی نوار نقاله، طرح شده است. جزئیات محاسبات این محدودیتهای بلادرنگ، در مرجع [۸] به طور کامل و با استدلالهای ریاضی و فیزیکی مربوطه، ذکر شده است.

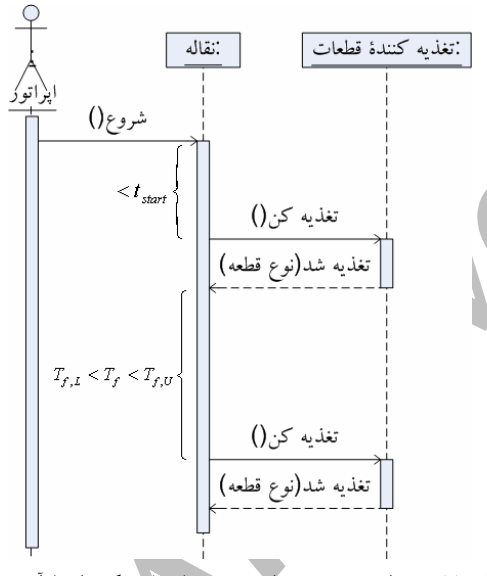
۵-۲- سیستم کنترل دمای مخزن

مثال دیگری که تشریح می‌شود، مربوط به کنترل دمای یک مخزن مایع است که نسبت به مثال قبلی پیچیدگی کمتری دارد. اما در همین سطح هم نشان داده می‌شود که مدل مبتنی بر مفاهیم بلوک تابع، به تنهایی، دچار پیچیدگی‌هایی می‌شود که استفاده از UML به کاهش آنها کمک فراوانی می‌نماید.

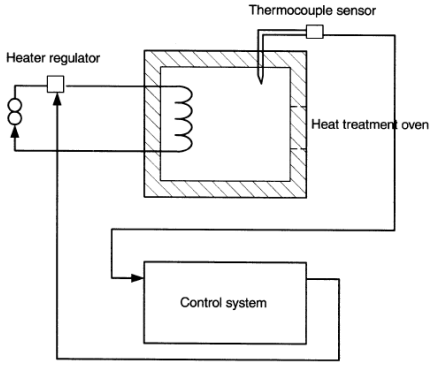
شماتیک این سیستم در شکل (۱۲) نمایش داده شده است. این مخزن در واقع یک اجاق را به کمک دمای مایع داخل مخزن، هدایت می‌کند. اگر فرض کنیم، چنین سیستمی در سطح یک فرآیند صنعتی، بصورت توزیعی بکار گرفته می‌شود، و عوامل مختلفی در بخشهای مختلف سیستم توزیعی، در تعیین دمای اجاق مؤثر هستند، آنگاه طبق مدل‌هایی که استاندارد IEC 61499 در اختیار قرار می‌دهد، مدل‌های سیستم، منبع و کاربرد (از استاندارد فوق) را می‌توان بصورتی خلاصه مطابق شکل (۱۳) نمایش داد. سیستم دارای منابع مختلف است که هر یک کنترل‌کننده‌ای را در اختیار دارد و بخشی از این کنترل‌کننده‌ها مربوط به سیستم کنترل دمای مخزن است. سنسورها در بخشی از سیستم توزیعی قرار گرفته اند و بلوک کنترل‌کننده دما، در بخشی دیگر که از لحاظ جغرافیایی با هم فاصله دارند. از شبکه‌های ارتباطی استانداردی هم برای اتصال منابع استفاده می‌شود.



شکل (۱۰): نمودار ترتیب به همراه محدودیتهای بلادرنگ برای کل فرآیند تغذیه، تست و انتقال به انبار یا واگن، برای مثال سیستم نورد تست قطعات.[۸]

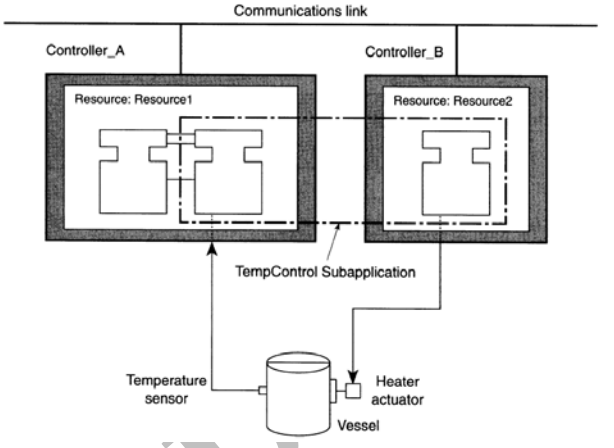


شکل (۱۱): نمودار ترتیب به همراه محدودیتهای بلادرنگ برای فرآیند تغذیه قطعات.[۸]



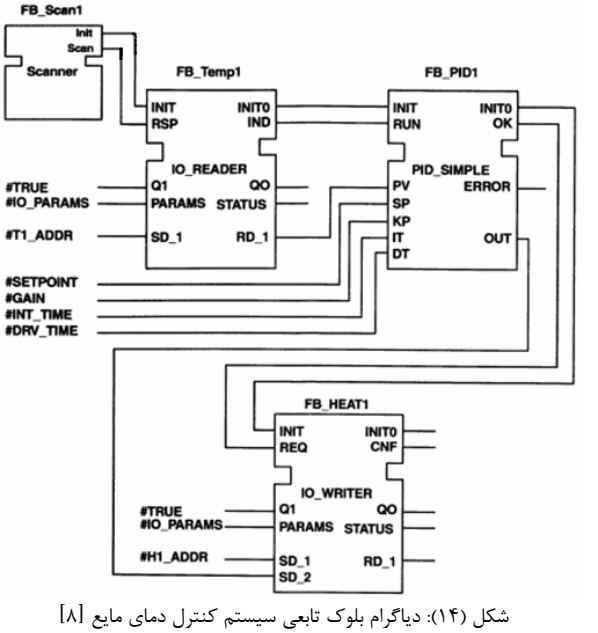
شکل (۱۲): شماتیک سیستم کنترل دمای مخزن مایع.[۸]

Archive of SID



شکل (۱۳): مدل‌های سیستم، منبع و کاربرد طبق استاندارد IEC 61499 در قالب یک نمودار، برای سیستم کنترل توزیعی دمای مخزن مایع.[۸]

یک دیاگرام بلوک تابعی برای سیستم مورد مطالعه، در شکل (۱۴) مشاهده می‌شود. همانگونه که مشاهده می‌شود این مدل و نیز مدل قبلی که در شکل (۱۳) نمایش داده شد، دارای پیچیدگی‌هایی هستند که ویژگیهای سیستم را به صراحت بیان نمی‌کنند. در حالی‌که با بازگشت به مثال قبلی، یعنی سیستم نورد تست قطعات مشاهده شد که چگونه ابزار UML-RT به درک مفهومی‌تر از سیستم کمک می‌نماید.

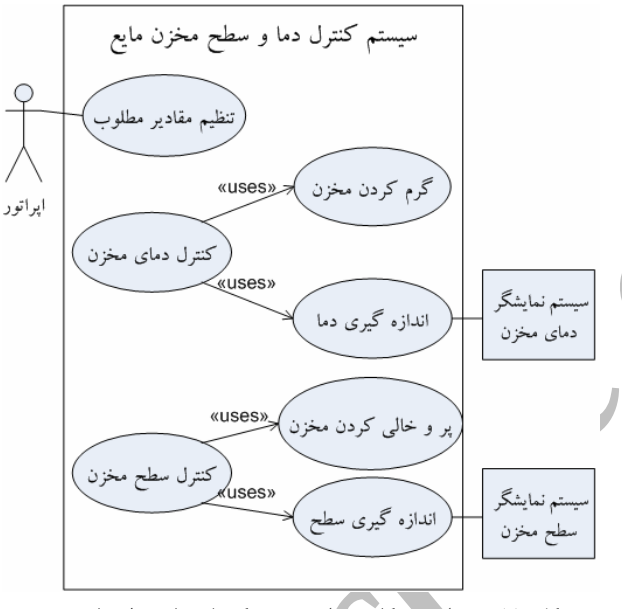


شکل (۱۴): دیاگرام بلوک تابعی سیستم کنترل دمای مایع [۸]

حال اگر بخواهیم سیستم را گسترش دهیم و کنترل سطح مایع و نیز ویژگیهای بلادرنگ را نیز به آن اضافه کنیم (شکل ۱۵)، بدون شک، مدلی که استاندارد IEC 61499 در اختیار می‌گذارد، از آنچه در فوق مشاهده شد، پیچیده‌تر و نامفهوم‌تر نیز می‌شود.

اگر چه، لازم به ذکر است که در نهایت، برای تطابق با استانداردهای کنترل صنعتی، استفاده از بلوکهای تابع گریز ناپذیر و در برخی موارد مفیدتر هم هستند ولی، استفاده همزمان با شیءگرایی، مدل قابل‌فهم‌تری در اختیار قرار می‌دهد. این موضوع، یعنی ارزیابی مزایا و معایب دو روش به تنهایی، و ویژگیهای ترکیب آنها، در قسمت بعدی بررسی می‌شود.

شکل (۱۵) نمودار مورد کاربرد را برای توضیح خواسته‌های سیستم در صورت نیاز به کنترل سطح مایع مخزن، علاوه بر کنترل دمای آن نمایش می‌دهد. با تعریف محدودیتهای زمانی که بصورت عملی و در کاربردها برای سیستمهای کنترل صنعتی مشخص می‌شود، می‌توان به خوبی با کمک نمودارهای ترتیب با پارامتر زمانی، ویژگیهای بلادرنگ را نیز مدل نمود که مشابه مثال قبل است.



شکل (۱۵): نمودار مورد کاربرد برای سیستم کنترل دما و سطح مایع مخزن

۶– ارزیابی و مقایسه

هر یک از روشهای مدل‌سازی بلوک تابعی و شئ‌گرا، نقاط ضعفی را در برخورد با سیستمهای کنترل بلادرنگ توزیعی از خود بروز می‌دهند. مهم‌ترین معایب روش بلوک تابع استاندارد IEC 61499، عبارتند از:

- عدم پشتیبانی از وراثت و وراثت چندگانه؛
- عدم نمایش ارتباط ساختاری بلوکهای تابع؛
- محدود بودن به یک زبان خاص توصیف بلوکهای تابع؛
- نداشتن زبانی برای بیان الگوریتمها و متکی بودن به سایر استانداردها در جهت توصیف این الگوریتمها؛
- بلادرنگ نبودن یا به عبارتی نمایش ندادن ویژگیهای زمانی الگوریتمهای داخلی بلوکها و نیز نداشتن مدلی برای تابع زمان‌بندی اجرای بلوکهای تابع[۸]؛

Archive of SID

- نمایش نه چندان ساده‌ی توزیع، با وجود تمرکز اصلی این مدل روی توزیع بلوکها بین منابع.

نقایص عمده‌ی مدل شئ در توصیف سیستمهای کنترل صنعتی بلادرنگ توزیعی را می‌توان بدین گونه برشمرد:

- عدم آشنایی برای مهندسان کنترل؛
- وابستگی زیاد سطح تجرید به دید ناظر؛
- به عقیده برخی، ممکن است کارایی کمتر شود؛
- در سیستمهای کاربردی کوچک، این روش سربار دارد؛
- مرز تعریف شده‌ای بین فازهای طراحی، پیاده سازی و نگهداری نداریم (گرچه همین می‌تواند مثبت باشد).

در مقابل، متدولوژی OO+FB با بهره‌گیری از خصوصیات مثبت هریک از دو روش اولیه، سعی در رفع نقاط ضعف دیگری نموده و مزایای متعددی در اختیار می‌گذارد:

- آشنا و قابل درک برای مهندسان کنترل؛
- کاهش زمان و هزینه‌های تولید؛
- افزایش قابلیت اطمینان و قابلیت نگهداری؛
- مدل‌سازی استاتیک بلوکهای تابع؛
- دارا بودن مکانیزمی برای ارثبری؛
- رفع نیاز به زبان متنی خاص؛
- بلادرنگ کردن IEC 61499؛
- مدل‌سازی تابع زمان‌بندی؛
- نمایش ساده‌ی توزیع با نمودار استقرار[۱۸]؛
- قابل استفاده بودن سابقه‌ی عظیم متدولوژی شئ‌گرا؛
- پذیرش آسان UML به عنوان یک زبان مشترک؛
- رهیافت معتبر برای آزمایش مدل.

جداول (۱) و (۲)، خلاصه‌ای از نحوه‌ی برطرف شدن نقایص هر یک از مدل های بلوک تابع و شئ را در روش مدل‌سازی OO+FB، بیان می‌نمایند.

جدول (۱): برطرف شدن نقایص مدل بلوک تابع در OO+FB

معایب مدل بلوک تابع	نحوه‌ی برطرف شدن در OO+FB
عدم پشتیبانی از وراثت	پشتیبانی از وراثت ببرای کلاس و شئ
عدم نمایش روابط استاتیک	نمایش روابط ساختاری کلاسها
محدود بودن به زبان خاص	وجود زبان های شئ‌گرای متعدد
نمایش ندادن زمان	استفاده از UML-RT برای مدل زمان
مدل نکردن تابع زمان‌بندی	مدل تابع زمان‌بندی با نمودار ترتیب
نمایش نامناسب توزیع	نمایش ساده توزیع با نمودار استقرار

جدول (۲): برطرف شدن نقایص مدل شیء در OO+FB

معایب مدل شیء	نحوه‌ی برطرف شدن در OO+FB
عدم آشنایی مهندسان کنترل	انطباق با استانداردهای کنترل صنعتی
وابستگی زیاد سطح تجرید به دید ناظر	کاهش وابستگی به کمک بلوک‌های تابع استاندارد از پیش آماده
کاهش کارایی	بهبود کارایی با مدل ساده ولی دقیق
سربار در سیستمهای کوچک	قابل اعمال به سیستمهای کوچک

۷- نتیجه‌گیری

در این مقاله، یک متدولوژی جدید برای مدل‌سازی سیستمهای بلادرنگ توزیعی، با نام OO+FB ارائه شد. این روش، از یک طرف بر اساس مفهوم بلوک تابع که در استانداردهای کنترل صنعتی استفاده می‌شود و از طرفی بر پایه‌ی مدل شئ برای تولید نرم‌افزار، بنا نهاده شد.

با توجه به مزایای فراوانی که از روش مدل‌سازی OO+FB مطرح شد و با نظر به پوشش اکثر نقاط ضعف هر یک از دو روش پایه توسط متدولوژی OO+FB، این روش می‌تواند به عنوان یک چهارچوب مناسب و جامع، جهت مدل‌سازی سیستمهای بلادرنگ توزیعی به‌کار برود.

۸- مراجع

[۱] International Electrotechnical Commission, **"Function Blocks for Industrial-Process Measurement and Control Systems Part 1: Architecture"** *(IEC 61499-1)*, IEC, 2005.

[۲] Lewis, R., "**Modelling Control Systems Using IEC 61499: Applying Function Blocks to Distributed Systems"** , 1st Edition, Springer, Berlin, IEE, London, 2001.

[۳] Grabmair, G., Froschauer, R., Strasser, T., Zoitl, A., "**Modelling Execution Order and Real-time Constraints in IEC 61499 Control Applications**", IEEE Workshop on Distributed Intelligent Systems: Collective Intelligence and Its Applications (DIS'06), 15-16 June, pp. 115 – 120, 2006.

[۴] Thramboulidis K. C., "**Unified Modeling Language: The lndustry Standard for Object-Oriented Development"** , in Industrial Information Technology Handbook, CRC Press, April 2005.

[۵] Thramboulidis K., "**Using UML in Control and Automation: a Model Driven Approach**", Proceedings of the Second IEEE International Conference on Industrial Informatics (INDIN'04), Berlin, Germany, June 2004.

[۶] Douglas, B. P., **"Real Time UML: Advances in the UML for Real-Time Systems"** , 3rd Edition, Addison Wesley, USA, 2004.

[۷] Thramboulidis, K., "**Design Alternatives in the IEC 61499 Function Block Model**", IEEE International Conference on Emerging Technologies and Factory Automation (ETFA'06), September, Prague, 2006.

Archive of SID

[۸] واحدی‌پور، روزبه، **مدل‌سازی شئ‌گرای سیستمهای کنترل بلادرنگ توزیعی**، پایان‌نامه‌ی کارشناسی ارشد مهندسی کنترل، دانشگاه صنعتی اصفهان، دانشکده‌ی برق و کامپیوتر، ۱۳۸۵.

[۹] Laplante, P. A., "**Real-Time Systems Design and AnalysisX"** , 3rd Edition, Wiley Publishing, IEEE Press, USA, 2004.

[۱۰] صدیقی مشکنانی، محسن، **مهندسی نرم‌افزار**، ویرایش اول، دانشگاه صنعتی اصفهان، (در دست چاپ).

[۱۱] Zhang W. Halang W. A., and Diedrich C., "**Specification of Function Block Applications with UML**", Proceedings of the 2005 IEEE International Conference on Robotics and Automation, (ICRA'05), 18-22 April, pp. 4002-4007, 2005.

[۱۲] Heck B. S., Willis L. M., and Vachtsevanos G. J., "**Software Technology for Implementing Reusable, Distributed Control Systems**", IEEE Control Systems Magazine, pp. 21-35, February 2003.

[۱۳] Object Management Group, "**UML Profile for Schedulability, Performance, and Time Specification**", OMG Adopted Specification, www.omg.org, January 2005.

[۱۴] Sampaio A., Mota A., and Ramos R., "**Class and Capsule Refinement in UML for Real-Time**", Electronic Notes in Theoretical Computer Science, Vol. 95, pp. 23–51, 2004.

[۱۵] Bichler L., Radermacher A., and Schürr A., "**Evaluating UML Extensions for Modeling Real-time Systems**", Proceedings of the Seventh International Workshop on Object-Oriented Real-Time Dependable Systems (WORDS 2002), IEEE, 2002.

[۱۶] Gao Q. and Brown L. J. and Capretz L. F., "**Extending UML-RT for Control System Modeling**", American Journal of Applied Sciences, Vol. 1, No. 4, pp. 338-347, 2004.

[۱۷] Unhelkar B., **"Verification and Validation for Quality of UML 2.0 Models"** , Wiley Interscience, New Jersey, USA, 2005.

[۱۸] Dong Y., Li M., and Wang Q., "**A UML Extension for Distributed System**", Proceedings of the First International Conference on Machine Learning and Cybernetics, November 2002.

Archive of SID