

سخنرانی: 3 الزامات CPS

سید حسین عطارزاده نیاکی

چند اسلاید از پیتر مارودل و فیلیپ کوپمن

سیستم های زمان واقعی جاسازی شده

1

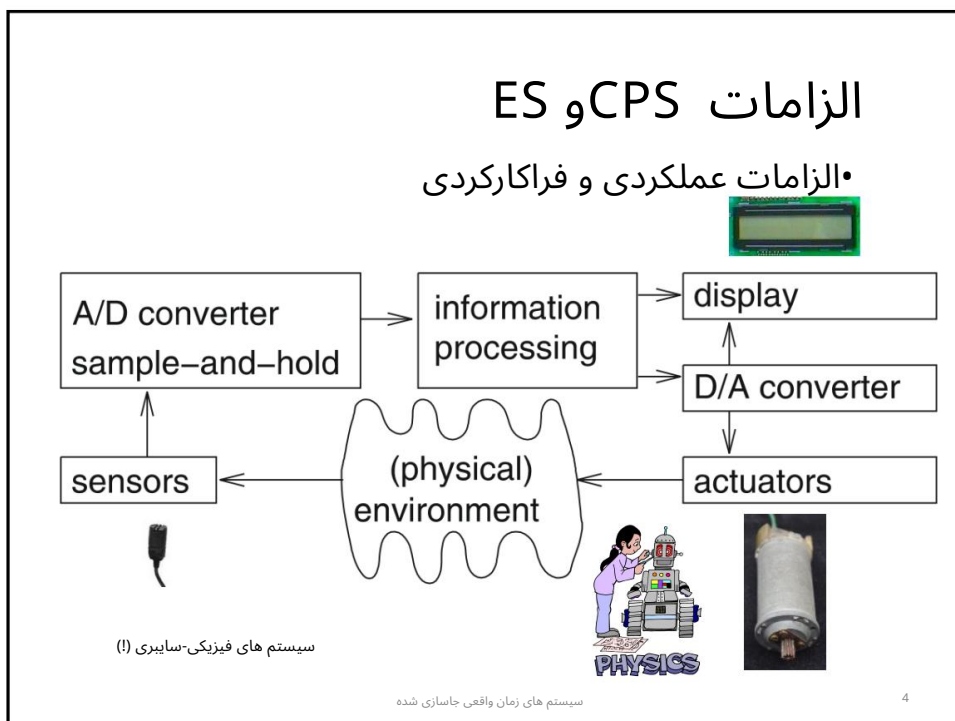
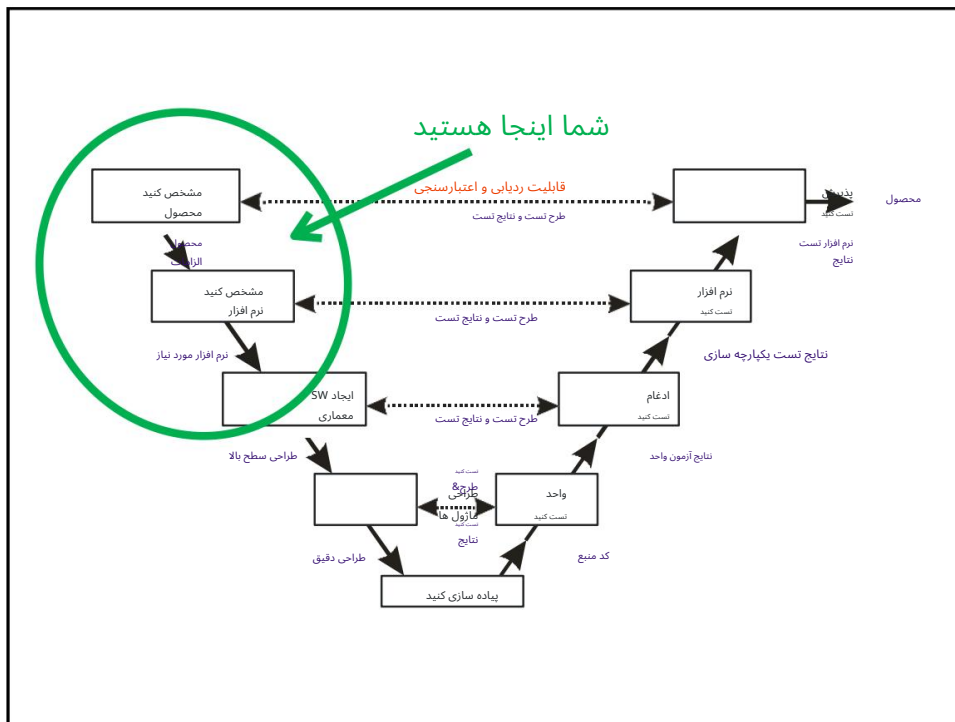
بررسی کنید

• طراحی CPS: سخت افزار و نرم افزار • فرآیند توسعه

• نرم افزار • CPS طراحی مبتنی بر مدل

سیستم های زمان واقعی جاسازی شده

2



الزامات عملکردی

الزامات جمع آوری داده ها

• مشاهده موجودیت های RT در یک خوشه کنترل شده و مجموعه ای از این مشاهدات

الزامات کنترل دیجیتال مستقیم

• محاسبه متغیرهای محرک برای محرک ها به منظور کنترل شیء کنترل شده به طور مستقیم

الزامات تعامل انسان و ماشین

• یک سیستم کامپیوتری بلادرنگ باید اپراتور را از وضعیت فعلی شیء کنترل شده مطلع کنید.
• به اپراتور در کنترل دستگاه یا شیء کارخانه کمک کنید.

سیستم های زمان واقعی جاسازی شده

5

الزامات زمان واقعی

• چه زمانی یک سیستم کامپیوتری بلادرنگ است؟

• هنگامی که صحت رفتار سیستم

نه تنها به نتایج منطقی محاسبات، بلکه به زمان فیزیکی تولید این نتایج نیز بستگی دارد.

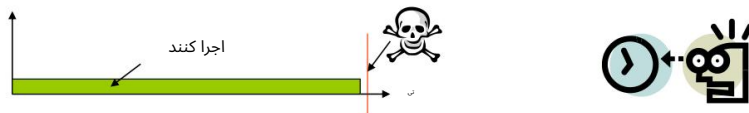
-منظور از رفتار سیستم، توالی خروجی ها در زمان یک سیستم است.

سیستم های زمان واقعی جاسازی شده

6

محدودیت های زمان واقعی

- CPS باید محدودیت‌های هم‌زمان را رعایت کند (مهلت‌ها)
- یک سیستم بلادرنگ باید به محرک‌های جسم کنترل شده (یا اپراتور) در بازه زمانی دیکته شده توسط محیط واکنش نشان دهد.



• اگر نتیجه‌ای حتی پس از سپری شدن ضرب الاجل مفید باشد، مهلت آن است

طبقه بندی شده به عنوان نرم؛

• در غیر این صورت محکم است.

• اگر در صورت از دست رفتن یک مهلت قطعی، عواقب شدیدی ایجاد شود،

مهلت سخت نامیده می‌شود.

□ یک پاسخ سیستم تضمین شده باید بدون استدلال‌های آماری توضیح داده شود

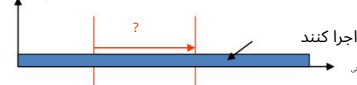
[کوپتزر، 1997]

سیستم‌های زمان واقعی جاسازی شده

7

چهار نوع مشخصات زمان بندی مورد نیاز است

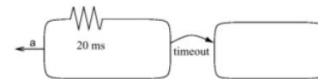
1. زمان سپری شده را اندازه‌گیری کنید
بررسی کنید، چقدر از آخرین تماس گذشته است



2. وسیله‌ای برای به تأخیر انداختن فرآیندها



3. امکان تعیین تایم‌اوت‌ها
حداکثر زمان در یک حالت خاص بمانید.



4. روش‌های تعیین ضرب‌الاجل
موجود نیست یا در فایل کنترل جداگانه است.



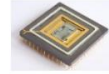
سیستم‌های زمان واقعی جاسازی شده

8

الزامات کارایی

CPS و ES باید کارآمد باشد

-کارآمد در اندازه کد
(به خصوص برای سیستم های روی تراشه)



-کارآمد در زمان اجرا



-کارآمد وزن



-مقرون به صرفه



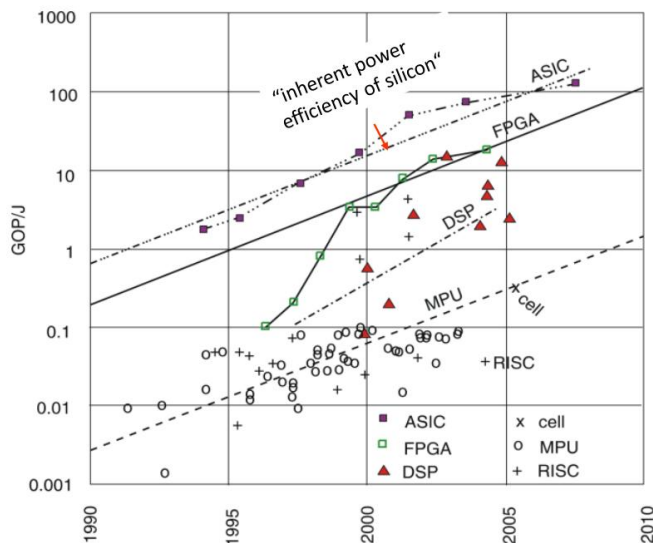
-کارآمد انرژی



سیستم های زمان واقعی جاسازی شده

9

اهمیت بهره وری انرژی



کاملاً غیرقابل اعتماد و نامطمئن به نظر می آید
لایه پردازنده در درجه اول به دلیل مصرف انرژی
افزایش قابل توجهی نیست.

سیستم های زمان واقعی جاسازی شده

© هوگو دی من (IMEC)
فیلپس، 2007

10

الزامات قابلیت اطمینان

CPS/ES □ باید به این معنا قابل اعتماد باشد

تیل باق اطمینان $R(t)$ احتمال کارکرد صحیح سیستم به شرطی که در $t=0$ کار کرده باشد



قابلیت نگهداری $M(d)$ احتمال کارکرد صحیح سیستم d واحدهای زمانی پس از وقوع خطا.



رد دسترس بودن: کسری از زمانی که سیستم کار می کند



نی‌می‌هیچ آسیبی □ شود



تی‌م‌ارتباط محرمانه و معتبر



سیستم های زمان واقعی جاسازی شده

11

الزامات امنیتی

دفاع در برابر

جرایم سایبری ("برآورد هزینه های سالانه جرایم سایبری ایالات متحده 100 میلیارد دلار؛ ... [وال استریت ژورنال، (2013/7/22)]

حملات سایبری □ (استاکس نت)

ترویسیم سایبری

جنگ سایبری



اتصال تهدیدها را افزایش می دهد

کل زنجیره های تولید می تواند تحت تاثیر قرار گیرد

جزایر محلی مقداری محصور شدن را فراهم می کنند، اما با ایده ارتباط جهانی در تناقض هستند

□ امروزه ممکن است حملات فیزیکی سایبری داشته باشیم .

□ ادغام با حوزه فیزیکی امنیت را پیچیده می کند

سیستم های زمان واقعی جاسازی شده

12

خلاصه از الزامات فوق عملکردی

• ویژگی های اضطراری (چیزهایی که نمی توان به یک جزء نسبت داد)

- عملکرد، مهلت های زمان واقعی
- امنیت، ایمنی، قابلیت اطمینان به طور کلی
- اندازه، وزن و مصرف انرژی ("SWaP")
- غالباً با بودجه تخصیصی در بین اجزا انجام می شود
- رفتارهای ممنوع ("نباید X را انجام داد")
- اغلب در زمینه الزامات ایمنی
- "عملکرد ایمنی" راهی برای اطمینان از یک رفتار منفی است، اما برخی از آنها رفتارها ظهور می کنند

• محدودیت های طراحی

- باید مجموعه خاصی از استانداردها را رعایت کند
- باید از تکنولوژی خاصی استفاده کرد
- هزینه سیستم، مهلت پروژه، کارکنان پروژه

سیستم های زمان واقعی جاسازی شده

13

ردیف های مورد نیاز CPS



سیستم های زمان واقعی جاسازی شده

14

قوانین لازم برای نرم افزار CPS خوب

دقیق و حداقل محدود

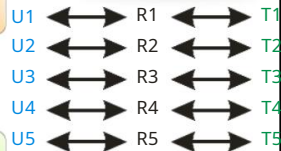
• توضیح می دهد که سیستم چه کاری را باید انجام دهد، نه اینکه چگونه آن را انجام می دهد
• در صورت امکان به جای عبارت کیفی، یک هدف عددی دارد • دارای تحمل (مثلاً 500 میلی ثانیه $\pm 10\%$ ، کمتر از "X")

قابل ردیابی و آزمایش

• هر نیاز دارای یک برچسب منحصر به فرد است (به عنوان مثال، "R-7.3")

در چارچوب سیستم پشتیبانی می شود

• پشتیبانی شده توسط منطق یا تفسیر • از اصطلاحات منسجم استفاده می کند • هر گونه الزامات متناقض حل شده یا اولویت بندی شده است



سیستم های زمان واقعی جاسازی شده

15

نمونه هایی از الزامات مشکل ساز

هنگامی که E-STOP فعال شود، سیستم باید خاموش شود.

غیر قابل ردیابی (بدون برچسب)

غیر قابل آزمایش

R-1.1: سیستم هرگز خراب نمی شود

نادقیق

R-1.7: سیستم بازخورد سریعی را به کاربر ارائه می دهد.

بدون تحمل اندازه گیری

LED: R-2.3: باید با دوره 500 میلی ثانیه چشمک بزند

پیش از حد پیچیده

R-7.3: باید فشار دادن دکمه قرمز، ویجت X فعال می شود، در حالی که با فشار دادن دکمه آبی باید Z LED به جای روشن شدن پیوسته Y، چشمک بزند، که از طریق دکمه زرد انجام می شود.

پیاده سازی را شرح می دهد

R-8.3: باید فشار دادن دکمه W باید دو مقدار صحیح 16 بیتی اضافه شود، سپس...



سیستم های زمان واقعی جاسازی شده

16

نمونه ای از الزامات چالش برانگیز: چراغ های جلو روشن است؟

یک بخش بزرگراه در حال آزمایش یک پیشنهاد ایمنی جدید بود. آنها از رانندگان خواستند که هنگام عبور از تونل، چراغ های جلوی خود را روشن کنند. با این حال، مدت کوتاهی پس از خروج از تونل، رانندگان با منظره ای از منظره روبرو شدند. بسیاری از آنها از جاده خارج شدند تا به انعکاس گل های وحشی در جویبارهای کوهستانی بکر و قله های کوه پوشیده از برف در 50 مایلی دورتر نگاه کنند. هنگامی که رانندگان به سمت خودروهای خود بازگشتند، متوجه شدند که باتری های خودروی آنها مرده است، زیرا چراغ های جلو را روشن گذاشته بودند.

-بنابراین، اداره بزرگراه تصمیم گرفت تابلوهایی نصب کند تا رانندگان چراغ های جلو خود را خاموش کنند.
-ابتدا سعی کردند: چراغ های خود را خاموش کنید. اما یک نفر گفت که همه به این درخواست توجه نمی کنند
چراغ های جلوشان را روشن کنند و نمی توانستند چراغ های جلوشان را خاموش کنند.
-بنابراین آنها سعی کردند: اگر چراغ های جلوی شما روشن است آنها را خاموش کنید. اما کسی مخالفت کرد که چنین خواهد شد اگر شب بود نامناسب
-بنابراین آنها سعی کردند: اگر روز است و چراغ های جلوی شما روشن است، آنها را خاموش کنید. اما شخصی مخالفت کرد که اگر هوا ابری بود و دید بسیار کاهش می یافت، نامناسب بود.
-بنابراین آنها سعی کردند: اگر چراغ های جلو شما روشن هستند و برای دید مورد نیاز نیستند، پس آنها را خاموش کنید. اما یک نفر مخالفت کرد که بسیاری از ماشین های جدید طوری ساخته می شوند که هر زمان که موتور کار می کند چراغ های جلوشان روشن باشد، بنابراین نمی توان آنها را خاموش کرد.
-بنابراین آنها سعی کردند: اگر چراغ های جلو شما روشن است و برای دید لازم نیست، و شما می توان آنها را خاموش کرد، سپس آنها را خاموش کرد. اما یکی مخالفت کرد....
-آنها تصمیم گرفتند تلاش برای شناسایی ایالت های قابل اجرا را متوقف کنند -فقط به رانندگان هشدار داده و به آنها اجازه دهید اقدامات لازم را انجام دهند. آیا چراغ های شما روشن است؟
* [بر اساس گاوز و واینبرگ، 1990]

سیستم های زمان واقعی جاسازی شده

17

فرآیند مثال برای

ایجاد نیازهای جاسازی شده

• استخراج الزامات کسب و کار/سیستم را شناسایی کنید

-مشتتری در صورت درخواست قیمت (RFQ) شرایط لازم را فراهم می کند

-ممکن است فروشنده نیاز به مصاحبه با مشتری داشته باشد

-قضاوت مهندسی ("حدس زدن")

• ایجاد معماری: تخصیص توابع به زیر سیستم ها

-نمودار کلاس / جزء

-و رابط ها

• ایجاد سناریو / موارد استفاده

-جریان های سطح بالا از طریق سیستم (نمودار جریان)

-موارد استفاده از "ساختمان" که بخش هایی از عملکرد را می گیرد

• الزامات نرم افزاری دقیق را ایجاد کنید

-الزامات رفتاری

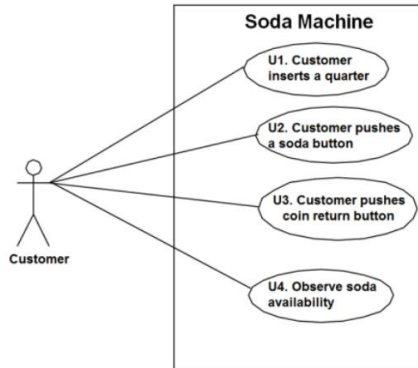
-محدودیت ها

سیستم های زمان واقعی جاسازی شده

18

رویکردهای مورد نیاز

نمودار مورد استفاده UML



سند متنی با لیست الزامات

• اگر کارشناسان دامنه قبلاً requirements را بداند بهترین کار را دارد. • با گذشت زمان، این می تواند به نیازهای بهتر همگرا شود.

موارد استفاده UML

• فعالیت های مختلف انجام شده توسط بازیگران • الزامات سناریوهای هستند که به هر مورد استفاده متصل می شوند

داستان های کاربر چابک

• هر داستان یک تعامل سیستمی را توصیف می کند

نمونه سازی

• مشتریان با دیدن آن متوجه می شوند • گاهی اوقات یک ماکت کاغذی کافی است

تجزیه عملکردی

• با توابع اولیه سیستم شروع کنید • لیست های بیشتر و دقیق تری از توابع فرعی تهیه کنید

(یک "معماری کاربردی" ایجاد می کند)

سیستم های زمان واقعی جاسازی شده

19

نمونه مورد نیاز نرم افزار



National Aeronautics and Space Administration

Communications, Navigation, and Networking reconfigurable Testbed (CoNNeT) Project			
Title: Software Requirements Specification	Document No.: GRC-CONN-REQ-0084	Revision: -	Page 18 of 61
	Effective Date: 02/23/2010		

Parent Req	ReqID	Requirement Text and Rationale	Prior-ity	Allocated To
FSRD-3714	SRS-3.2.6.12	The Software shall send data at a user data rate from zero up to and including 100 Mbps. <i>Rationale: The maximum data rate the Payload Avionics Software must send is 100 Mbps. Lower rates must also be handled.</i>	P2	PAS
FSRD-3133	SRS-3.2.6.13	The software shall send and receive data on two SpaceWire channels simultaneously at up to the maximum SDR interface data rate (full duplex) that can be sustained by both SDRs. <i>Rationale: When communicating with multiple radios, the Software will need to sustain an achievable data rate. In this requirement, it is defined as the minimum data rate of the two (or three, if possible) SDRs involved in the experiment. For instance, this data could be provided in the routing table. If two other radios are involved, then the data rate may change, based on the capability of those two radios (i.e. a new minimum interface data rate). This value should not be hard coded, but should have the capability for change, once on-orbit.</i>	P2	PAS

<https://goo.gl/qct5tL>

سیستم های زمان واقعی جاسازی شده

20



GRC-CONN-REQ-0084
EFFECTIVE DATE: 02/23/2010

استاندارد IEEE 830-1998

«عملیات توصیه شده IEEE برای مشخصات مورد نیاز نرم افزار»

• طرح کلی

«SRS» = - مشخصات مورد نیاز نرم افزار

- در دنیای تعبیه شده، باید "مشخصات مورد نیاز سیستم" باشد!

• مناطق مورد بررسی

- عملکرد (چه کاری انجام می دهد)

- رابط های خارجی (این واقعاً است)

معماری، اما داشتن آن در SRS مهم است)

- عملکرد (سرعت، مسائل بلادرنگ)

- ویژگی ها (الزامات غیر کاربردی مانند قابلیت نگهداری، قابلیت اطمینان)

- محدودیت های طراحی (استانداردهای

قابل اجرا، سیاست ها و غیره)

Table of Contents

1. Introduction
1.1 Purpose
1.2 Scope
1.3 Definitions, acronyms, and abbreviations
1.4 References
1.5 Overview
2. Overall description
2.1 Product perspective
2.2 Product functions
2.3 User characteristics
2.4 Constraints
2.5 Assumptions and dependencies
3. Specific requirements (See 5.3.1 through 5.3.8 for explanations of possible specific requirements. See also Annex A for several different ways of organizing this section of the SRS.)
Appendixes
Index

سیستم های زمان واقعی جاسازی شده

21

سیستم نمونه: دستگاه فروش نوشابه

• الزامات سطح بالا: کاری کنید که مانند یک

ماشین فروش واقعی کار کند

• ساده سازی:

-نوشابه ها چند ربع قیمت دارند

-تمام سکه های دیگر رد می شوند (برای سیستم کنترل شما نامرئی)

• یک سیستم توزیع شده در نمودار داده شده را

فرض کنید

-پردازنده برای هر دکمه، سکه

کنترل کننده برگشت، کنترل کننده فروش

-فرهنگ لغت پیام و بیشتر مشخصات مورد نیاز را دریافت می

کنید («معماری»)

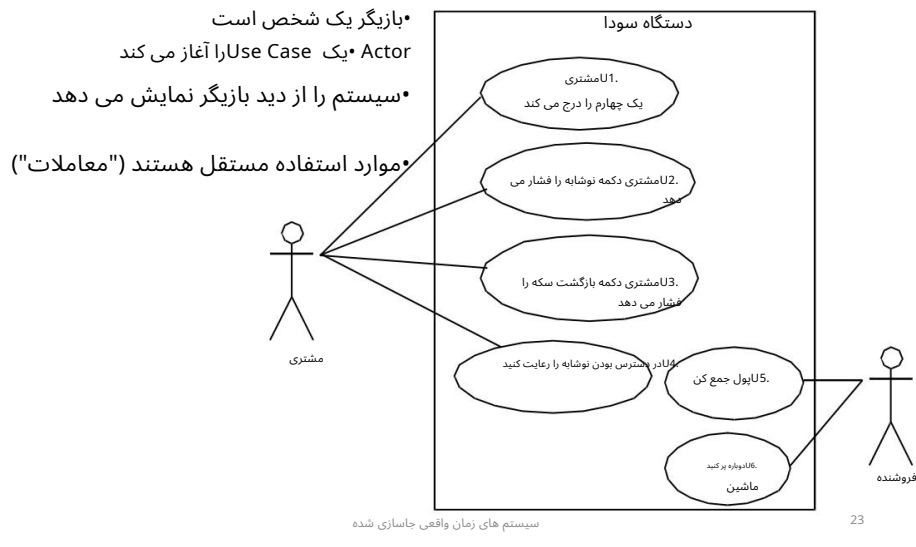


سیستم های زمان واقعی جاسازی شده

22

موارد استفاده UML

برای توسعه نیازمندی ها



الزامات متن در سطح سیستم

• هدف: اجرای یک دستگاه فروش نوشابه

- R1- فشار دادن یک دکمه باید نوشابه ای از نوع مربوط به آن را بفروشد آن دکمه
- R2- دستگاه باید دقیقاً سکه های SODACOST را برای همیشه نگه دارد هر قوطی نوشابه فروخته شد
- R3- بازگشت سکه باید تمام سکه های سپرده شده از آخرین چرخه فروش را برگرداند.
- R4- ماشین باید تمام پول واریز شده بیش از آن را برگرداند SODACOST سکه قبل از چرخه فروش.
- R5- ماشین باید چراغ را برای یک آیتم انتخاب شده در حین فروش در حال انجام است تا نشان دهنده پذیرش انتخاب برای خریدار باشد.
- R6- دستگاه باید چراغ را برای هر اقلامی که در انبار موجود نیست روشن کند

• سیستم کاملاً توزیع شده را فرض کنید

-پردازنده برای هر دکمه، کنترل کننده بازگشت سکه، کنترل کننده فروش

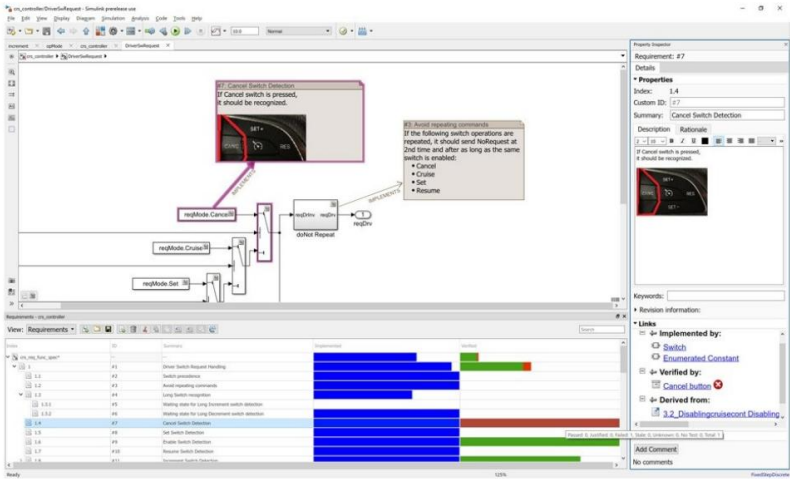
قابلیت ردیابی: UML و الزامات متن

	الزامات متن					
موارد استفاده	R1	R2		R3	R5	R6
U1. مشتری یک چهارم درج می کند				X		
U2. مشتری دکمه نوشتابه را فشار می دهد	X				X	
U3. مشتری دکمه بازگشت سکه را فشار می دهد			X			
U4. دسترس بودن نوشتابه را رعایت کنید						X
U5. پول جمع کن		X				
U6. دستگاه شارژ مجدد		X				X

سیستم های زمان واقعی جاسازی شده

25

الزامات سیمولینک



سیستم های زمان واقعی جاسازی شده

26

بهترین روش ها برای نیازمندی ها

• شش اصطلاح C برای شرایط خوب
- واضح، مختصر، صحیح، منسجم، کامل و قابل تایید • همچنین

- با مسائل غیر عملکردی مقابله کنید
- الزامات را به جریان طراحی مرتبط کنید
- به عنوان مثال، ارتباط با داستان های کاربر یا موارد استفاده • مشکلات الزامات -
- اجتناب از جزئیات غیرضروری و پیاده سازی - اگر در الزامات وجود نداشته باشد، انجام نمی شود
- اگر قابل آزمایش نباشد، نمی دانید که آیا انجام شده است

سیستم های زمان واقعی جاسازی شده

27

سخنرانی بعدی

- مدل های محاسبات
- مدل سازی مبتنی بر بازیگر • پویایی مستمر
- سیستم های علی
- سیستم های بدون حافظه
- سیستم های خطی زمان ناپذیر • فصل 2 LeeSshia را بخوانید • دانش
- خود را از "سیگنال ها و" مرور کنید

دوره سیستم ها

سیستم های زمان واقعی جاسازی شده

28