چکیده

یکی از چالش­های مدل­سازی سیستم­های سایبر-فیزیکی، ادغام رفتارهای فیزیکی با رفتارهای محاسباتی و شبکه است و با توجه به این که سیستم­های سایبر­-فیزیکی معمولا دارای مولفه­ی توزیع شده هستند، مدل­های مبتنی بر اکتور، چارچوب مناسبی را برای مدل­سازی چنین سیستم­هایی ارائه می­دهند. در این یک زبان مدل­سازی صوری مبتنی بر­ اکتور برای سیستم­های سایبر-فیزیکی ارائه شده است، که توانایی توصیف رفتار­های فیزیکی، محاسباتی و شبکه را فراهم می­کند.

مقدمه

سیستم­های سایبر-فیزیکی از ادغام فرایندهای محاسباتی و فیزیکی ایجاد می­شوند. در این سیستم­ها، رایانه­های نهفته[[1]](#footnote-1) از طریق شبکه­، فرآیندهای فیزیکی را کنترل می­کنند که معمولا همراه با حلقه­ی بازخورد[[2]](#footnote-2) است، به این معنی که فرآیندهای فیزیکی بر فرآیند­های محاسباتی تاثیر می­گذارند و بلعکس. به همین خاطر برای تحلیل و توسعه­ی این سیستم­ها نیاز به آگاهی از تعامل و اشتراک بین سیستم­ رایانه­ای، نرم­افزار­، شبکه و فرایند­های فیزیکی است.

سیستم­های سایبر-فیزیکی چند تفاوت کلیدی با سیستم­های نرم­افزاری دارند. در سیستم­های نرم­افزاری زمان اجرای یک دستور تنها مرتبط با کارایی سیستم است و نه درستی آن ولی در مقابل در سیستم­های سایبر-فیزیکی زمان اجرای یک دستور می­تواند برای درستی سیستم حیاتی باشد. همچنین فرآیند­های فیزیکی ترکیب رخداد چندین رویداد همزمان هستند، بر خلاف فرایندهای نرم­افزاری که به صورت گام­ها­ی متوالی اجرا می­شوند.

با توجه به کاربرد سیستم­های سایبر-فیزیکی در زمینه­های ایمنی-­مهم[[3]](#footnote-3) ،ارزیابی و بررسی این سیستم­ها به شدت حیاتی است. روش­های مبتنی بر مدل، یکی از روش­های اصلی در طراحی چنین سیستم­هایی است. این مدل­ها علاوه­ بر تعیین مشخصات سیستم، توانایی شبیه­سازی و بررسی آن را فراهم می­کنند.

یکی از پایه­ای­ترین زبان­های مدل­سازی برای سیستم­های سایبر-فیزیکی، خودکاره­های ترکیبی[[4]](#footnote-4) است. این مدل می­تواند رفتارهای پیوسته (فیزیکی) و رفتارهای گسسته (سایبری) را توصیف کند. به طور خلاصه یک خودکاره­ی ترکیبی دارای چندین مکان برای مدل­کردن حالات گسسته سیستم و دارای چندین متغیر پیوسته و معادلات زمانی­ آن­ها برای بیان رفتار پیوسته­ی سیستم است. با این همه مدل سازی رفتار­های پیچیده سایبری و مدل­سازی رفتار­های شبکه سخت است.

در مقابل Timed Rebeca [13] یک زبان برای مدل­سازی سیستم­های غیرسنکرون بی­درنگ گسسته است. این زبان مبتنی بر اکتور است و دارای مفاهیمی همچون زمان پردازش، ارسال و دریافت پیام به صورت غیرسنکرون، تاخیر شبکه و مهلت اجرای پیام است. با این حال این زبان برای مدل­سازی رفتار پیوسته، مفهومی ارائه نمی­دهد.

در این پروژه زبان مبتنی بر اکتور HPalang ارائه شده است که به طور خلاصه ویژگی­های زیر را دارد:

* مدل معنایی صوری مبتنی بر خودکاره­ی ترکیبی
* مدل­سازی ارسال و دریافت پیام به صورت غیرسنکرون
* مدل­سازی زمان پردازشی بخش سایبر
* مدل­سازی رفتار­های فیزیکی و پیوسته

تعاریف اولیه

در این بخش بعضی از مفاهیمی که در ادامه وجود دارند، تعریف شده­اند.

سیستم گذار

یک سیستم گذار یک چندتایی به صروت است که:

* مجموعه­ی حالات است.
* مجموعه­ی برچسب­ها است.
* رابطه­ی گذارها است.
* حالت شروع است.

خودکاره­ی ترکیبی

یک خودکاره­ی ترکیبی به صورت یک چندتایی تعریف می­شود که:

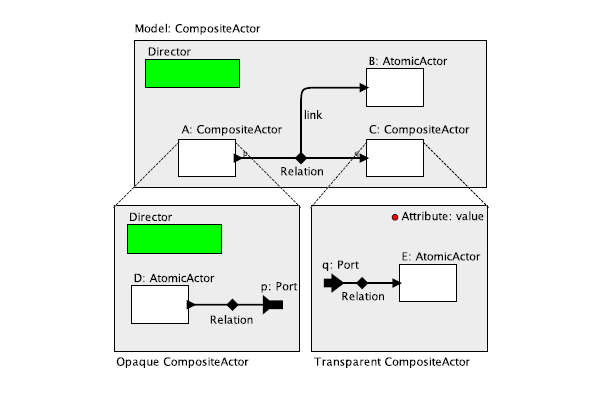
* مجموعه محدود مکان­ها است.
* مجموعه­ای از متغییرهای پیوسته است.
* مکان شروع و مقادیر اولیه متغییرهای است.
* رابطه­ی پرش­ها است که:
  + شروط ارضا شدن پرش و
  + مجموعه­ای از مقداردهی به متغییرهای پس از پرش است.
* مقادیر مجاز متغییرها را در هر مکان مشخص می­کند.
* محدودیت­ها روی متغییرها و مشتق آن­ها توصیف می­کند و رفتار پیوسته در هر مکان را مشخص می­کند.

کارهای مشابه

در این بخش به بررسی بعضی از زبان­ها و ابزارهای مدل­سازی برای سیستم­های سایبر-­فیزکی پرداخته می­شود.

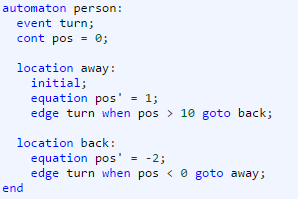
Ptolemy

Ptolemy یک چارچوب مدل­سازی، شبیه­سازی و طراحی سیستم­ها بی­درنگ و هم­روند مبتنی بر اکتور است. تمرکز این چارچوب بر ادغام مدل­های محاسباتی نا­همگون است. مدل محاسباتی مجموعه قوانین مربوط روی اجرای همروند مولفه­ها و طریقه­ی ارتباط بین آن­ها است. در Ptolemy مدل­های محاسباتی مختلفی از جمله ماشین­ حالت متناهی[[5]](#footnote-5)، زمان پیوسته، واکنشی-سکنرون[[6]](#footnote-6)، گسسته رخداد[[7]](#footnote-7) و جریان داده­ای[[8]](#footnote-8) گنجانده شده است. برای ایجاد مدل­های ناهمگون، از اکتورهای سلسله مراتبی استفاده می­شود که هر اکتور مرکب شامل یک کارگردان[[9]](#footnote-9) و چندین اکتور دیگر است. کارگردان در مدل، درواقع همان مفهوم مدل محاسباتی است که با تغییر کارگردان می­تواند رفتار مدل را تغییر داد. با توجه به تمرکز این چارچوب بر شبیه­سازی، مدل معنایی صوری برای Ptolemy تعریف نشده است.



CIF

CIF یک زبان مدل­سازی برای سیستم­های ترکیبی است. یک مدل در این زبان دارای چند خودکار و گروه است. خودکارها رفتار یک سیستم را تعریف می­کنند و گروه­ها مجموعه­ای از خودکاره­ها و اعلان­ها هستند. مثال یک خودکاره ترکیبی در این زبان است. این خودکاره دارای یک متغییر پیوسته pos است که مقدار اولیه آن صفر است. دو مکان برای این خودکاره تعریف شده است. مکان away که مکان اولیه است دارای یک رابطه برای تغییر متغییر pos بر حسب زمان و یک گذار با رخداد turn و شرط pos>10 به مکان back است.



Acumen

زبان Acument یک زبان مدل­سازی و شبیه سازی برای سیستم­های ترکیبی است. برای ساخت مدل در این زبان، از مفهوم "مدل" استفاده می شود. هر مدل در این زبان حداقل دارای یک تعریف مدل Main است که شروع مدل از همین مدل است و تمام بخش­های دیگر سیستم باید در این قسمت تعریف یا ساخته شوند. شکل یک مدل ترمستات در این زبان است. از کلیدواژه­ی initially برای توصیف حالت اولیه سیستم استفاده می­شود و از کلیدواژه­ی always برای توصیف رفتار پیوسته­ی سیستم. در این زبان برخلاف زبان CIF مکان­ها به صورت مستقیم تعریف نمی­شوند بلکه با استفاده از ترکیبی از match و if می­توان مد­های مختلف سیستم را بیان کرد. به عنوان مثال در مدل داده شده، زمانی که متغییر مد برابر یک باشد و مقدار t کمتر از 25 باشد

model thermostat\_model() =

initially

//SYSTEM variables [BEGIN]

t=0.0,t'=0.0,mode=1,tmin=17,tmax=28

//SYSTEM variables [END]

always

match mode with [

1 ->

if t >= 25 then

mode+ = 2

else

t' = -t+tmax

| 2 ->

if t <= 20 then

mode+ = 1

else

t' = -t+tmin

]

SpaceEx

ابزار SpaceEx [6] یک ابزار درستی­یابی برای سیستم­های ترکیبی است. هدف در این ابزار اثبات یک ویژگی ایمنی در مدل سیستم است. SpaceEx دارای سه بخش ویرایشگر مدل برای ویرایش مدل­ سیستم به صورت بصری، هسته­ی آنالیز برای بررسی مدل با توجه به پارامتر­های ورودی و رابط وب، یک رابط گرافیکی برای استفاده آسان از هسته­ی آنالیز و مشاهده نتیجه خروجی است. مدل پایه­ی این ابزار خودکاره­ی ترکیبی است و برای ماژولار کردن مدل، مفاهیم مولفه­ی پایه و مولفه­ی شبکه در این ابزار وجود دارد. مولفه­ی پایه در واقع یک خودکاره­ی ترکیبی است و مولفه­ی شبکه، از اتصال چند مولفه­ (پایه یا شبکه) ایجاد می­شود. ارتباط بین مولفه­ها به صورت برچسب­های همگام­سازی روی گذارهای مولفه­ی پایه بیان می­شود. برای درستی­یابی مدل چند الگوریتم دسترسی­یابی در ابزار تعبیه شده است.

Rebeca

Timed Rebeca [13] یک نسخه­ی گسترش­یافه­ی [14] برای مدل­سازی و شبیه­سازی سیستم­های غیرسکنرون بی­درنگ گسسته است. این زبان مبتنی بر اکتور است و علاوه بر مفاهیم ایجاد اکتور، ارسال پیغام و پردازش پیغام دارای مفاهیم زمانی چون زمان پردازش،تاخیر در دریافت پیغام و زمان انقضای پیغام می­باشد.در این زبان هر پیغام دارای دو برچسب زمان و ضرب­الاجل است. برچسب زمان، مقدار زمان ارسال پیغام بعلاوه­ی مقدار تاخیر پیغام (در صورت وجود) است. برچسب ضرب­الاجل نیز مشخص­کننده­ی حداکثر زمان معتبر بودن پیغام است. این زبان مدل مناسبی برای توصیف بخش­های محاسباتی و شبکه­ی سیستم­های سایبر-فیزیکی ارائه می­کند، به خصوص مبتنی بر اکتور بودن این زبان توانایی بازاستفاده­ی بالایی به مدل ساخته شده می­دهد. با این حال این زبان برای مدل­سازی رفتار پیوسته، مفهومی ارائه نمی­دهد.

مسئله

در این پروژه یک زبان مدل­سازی مبتنی بر اکتور برای سیستم­های سایبر-فیزیکی ارائه شده است.

مدل

Syntax

مدل HPalang شامل تعریف چند اکتور و یک main block برای پیغام­های اولیه است.

Actors

یک اکتور به صورت یک چندتایی به فرم (id, vars, mthds) تعریف می شود که id شناسه، vars مجموعه­ای متغییرهای گسسته و پیوسته و mtds مجموعه متد­های اکتور است. هر متد به صورت یک چندتایی تعریف می­شود که نام متد و دنباله­ی دستورات این متد است.

Statements

Discrete Assignment

عملیات انتصاب یک مقدار گسسته به یک متغییر گسسته است.

عملیات انتصاب یک مقدار پیوسته به یک متغییر پیوسته است.

دستور شرطی که دارای یک عبارت بولی گسسته و دو دنباله دستور است. در صورت ارضا شدن شرط دنباله اول و در غیر این صورت دنباله دوم اجرا می شود.

*دستور ارسال است که پیغام به اکتور با شناسه­ی یا خود اکتور کنونی ارسال می شود.*

*دستور یک رفتار پیوسته است. یک رفتار پیوسته­ی همانند خودکاره­ی ترکیبی تعریف شده است. برای سادگی تنها یک گذار تعریف شده است. دنباله­ی دستوراتی است که پس از انجام گذار اجرا می­شوند.*

*دستوری است اکتور را برای واحد زمانی مشغول نگه می­دارد.*

*دستور ادامه­ی اجرای دستورات است که پس از پایان اعمال می­گردد.*

*توابع کمکی*

*از توابع کمکی زیر را برای تعریف معنایی صوری در بخش بعد، تعریف می­کنیم.*

*، که بدنه­ی متد از کتور با شناسه­ی را بر­می­گرداند.*

که شرط ثابت رفتار پیوسته­ی را بر می­گرداند.

که معادلات دیفرانسیلی رفتار پیوسته­ی را بر می­گرداند.

که شرط گذار رفتار پیوسته­ی را بر می­گرداند.

که دستورات گذار رفتار پیوسته­ی را بر می­گرداند.

که متغییر پیوسته­ی تخصیص داده شده برای تاخیر اکتور از اکتور را برمی­گرداند.

*قوانین معنایی ایستا*

*قوانین زیر، قوانین مربوط به مناسب بودن ساختار یک مدل در این زبان هستند که به سادگی در گرامر زبان قابل تعریف نیستند ولی به صورت ایستا قابل بررسی هستند.*

* *اکتور­ها دارای شناسه­های یکتا هستند.*
* *نام متغییرها در یک اکتور یکتا است.*
* *نام متدها در یک اکتور یکتا است.*
* *مدل نوع-ایمن است. یعنی:*
  + *عبارات نوع-ایمن هستند.*
  + *هر دو طرف یک انتصاب هم نوع هستند.*
  + *شروط از نوع بولی هستند.*
  + *گیرنده­ی دارای متدی با نام پیغام ارسال شده است.*

*معنای اجرایی*

*در این بخش معنای صوری این زبان مبتنی بر خودکاره­ی ترکیبی بیان می­شود. تعریف این معنا در دو مرحله صورت می­گیرد. در مرحله­ی اول یک سیستم گذار تعریف می­شود و در مرحله­ی دوم این سیستم­گذار به یک خودکاره­ی ترکیبی تبدیل می­شود.*

*حالات*

*اکتور­ها توسط ارسال پیام ارتباط برقرار می­کنند و پیغام­های دریافتی را در یک صف ذخیره می­کنند. نوع یک پیغام به صورت تعریف می­شود. اکتور دارای دو صف پیام و است که به ترتیب صف مربوط به پیام­ها با اولیت بالا و صف مربوط به پیام­های با اولیت پایین است. نوع این دو صف به صورت تعریف می­شود.*

*حالات سرارسی یک مدل به صورت یک تابع نمایش داده می­شود که و . تابع شناسه­ی یک اکتور را به حالت محلی آن نگاشت می­کند. حالت محلی یک اکتور یک چندتایی به صورت است که مقداردهی به متغییر­های اکتور و صف پیام­ها، دنباله­ی دستورات اجرایی و یک چندمجموعه از رفتار­های پیوسته­ی اکتور است.*

*گذارها*

*در این قسمت گذار­های بین حالات به صورت* SOS*تعریف شده­اند.*

## Low Priority Message Take

## High Priority Message Take

## Continuous Behavior Expiration

## Continuous Variable Assignment

## Resume Statement

## Continuous Behavior Statement

## Delay Statement

## Message Send

سیستم گذار میانی یک مدل M به صورت تعریف می­شود که:

* مجموعه حالات سراسری است (مجموعه­ی تمام توابع از شناسه­ی اکتورها به حالات محلی)
* کوچکترین رابط­ه­ای است که توسط قوانین SOS بالا تعریف شده است
* حالات اولیه است که متغییر­های گسسته صفر مقدارهی شده­اند و پیغام­های درون Main Block در صف اولیت پایین اکتور­های مشخص شده گذاشته شده است.

خودکاره­ی ترکیبی

برای تبدیل به یک خودکاره­ی ابتدا دو عمل زیر روی سیستم گذار اعمال می­شود.

* اولیت­دهی به گذارهای : اگر در حالتی حداقل یک گذار وجود داشته باشد، گذار­های g این حالت حذف می­شوند.
* حذف گذارهای :

خودکاره­ی ترکیبی به صورت زیر تعریف می­شود.

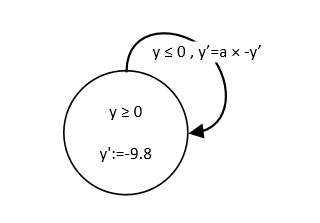
مثال­ها

توپ

در این بخش سقوط آزاد یک توپ مدل می­شود. در این سیستم، توپ از ارتفاع اولیه رها شده و با شتاب –g در جهت عمودی حرکت می­کند. زمانی که ارتفاع توپ از سطح زمین صفر شود، سرعت توپ برعکس شده و با ضریف کاهش می­یابد. این کاهش سرعت، از دست رفتن قسمتی از انرژی توپ با برخود به زمین را بیان می­کند.

مدل تعریف شده این سیستم در شکل زیر آمده است. در این مدل یک اکتور Ball وجود دارد که رفتار توپ را مدل می­کند. این اکتور دارای یک متغییر پیوسته­ی y است که ارتفاع توپ از سطح زمین است و با مقدار اولیه آن h است. متد Fall در، سقوط توپ را مدل می­کند. در این متد تنها رفتار پیوسته­ی سقوط تعریف شده است. در این رفتار تا زمانی که ارتقاع بیشتر از صفر است، مشتق دوم y با نرخ –g تغییر می­کند. زمانی که y کمتر مساوی صفر شود، مشتق اول y، که سرعت توپ است، عکس شده و با ضریف a کاهش می­یابد و یک پیام Fall به خود توپ ارسال می­شود تا عمل سقوط دوباره صورت گیرد. در بلاک Main این مدل نیز پیام Fall به اکتور Ball ارسال می­شود تا توپ شروع به سقوط کند. ترجمه­ی این مدل به خودکاره ترکیبی در شکل زیر آمده است.





دستگاه نوشیدنی

در این سیستم، دستگاه دارای دو نوع نوشیدنی چای و قهوه است. با توجه به درخواست کاربر، دستگاه نوشیدنی مورد نظر را ابتدا تا دمای مشخص شده گرم می­کند و لیوانی را با نوشیدنی انتخاب شده پر می­کند. دما و حجم نوشیدنی چای به ترتیب 100 درجه­ی سانتی گراد و 300 سی سی و دما و حجم نوشیدنی قهوه به ترتیب 90 درجه­ی سانتی­گراد و 200 سی سی است.

در شکل مدل این سیستم را مشاهده می­کنید. در این مدل چهار اکتور User، Machine، Heater و Filler وجود دارد. اکتور User رفتار کاربر را مدل می­کند که در این مثال خواص، کار نوشیدنی خود را به طور یکی در میان، تغییر می­دهد. User با ارسال پیام PrepareTea یا PrepareCoffee به Machine نوع نوشیدنی درخواستی خود را مشخص می­کند. Machine با دریافت یکی از این دو پیام، ابتدا نوع نوشیدنی را در متغییر گسسته­ی orderType ذخیره می­کند و سپس پیام HeatUp100 یا HeatUp90 را با توجه به نوع درخواست به Heater ارسال می­کند. Heater با دریافت یکی از این دو پیام، یک رفتار پیوسته برای گرم شدن تعریف می­کند که در این رفتار پس از رسیدن به دمای مطلوب، متغییر t صفر می­شود و پیام Heated به Machine فرستاده می­شود. در متد Heated، Machine با توجه به نوع نوشیدنی درخواستی، پیام Fill200 یا Fill300 را به Filler رسال می­کند. Filler با دریافت یکی از این دو پیام، یک رفتار پیوسته برای مدل کردن پرشدن لیوان تعریف می­کند. در این رفتار پس از رسیدن به حجم مناسب، پیام Filled به Machine ارسال می­شود. در متد Filled، Machine پیام RecieveOrder را به User می­فرستاد و User در این متد دوباره یک Order جدید می­دهد. دلیل استفاده از دو متد در اکتورهای Filler و Heater، عدم تعریف شدن پارامتر ورودی برای متد­ها، در این نسخه­ است.

پل متحرک

در این مثال یک پل متحرک در نظر گرفته شده است که به طور پیش فرض در حالت بالا آمده قرار دارد. در صورت وجود ماشین در پش پل، پل شروع به پایین آمدن می­کند و زمانی که پل به طور کامل پایین آمده باشد، ماشین­ها شروع به گذر می­کنند. در صورت رد شدن تمام ماشین­ها، پل دوباره شروع با بالا رفتن می­کند و با آمدن ماشین­های جدید این رویه دوباره تکرار می­شود.

مدل تعریف شده برای این پل در شکل زیر آمده است. برای نشان دادن قابلیت غیرسنکرون بودن زبان HPalang دو اکتور CarDispatcher1 و CarDispatcher2 تعریف شده است که با یک دوره­ی مشخص، به اکتور DrawBridge پیام EnqueueCar را ارسال می­کنند. این دو اکتور یک مدل ساده از ورود ماشین­ها به پل هستند. اکتور DrawBridge وظیفه­ی کنترل پل را به عهده دارد. این اکتور دارای دو متغییر گسسته­ی cars و bridgeStatus است. متغییر cars تعداد ماشین­های پشت پل است و متغییر bridgeStatus حالت پل را مشخص می­کند که در این مدل، پل دارای سه حالت بالاآمده، درحال تغییر و پایین آمده است که به ترتیب با مقادیر 0، 1 و 2 مشخص شده است. این اکتور همچنین دارای دو متغییر پیوسته­ی degree و timer است. متغییر degree نشان دهنده­ی زاویه­ی دسته­ی پل است که بین اعداد 90 و 0 متغییر است. متغییر timer یک متغییر کمکی برای مدل سازی خروج ماشین­ها به صورت دوره­ای است.

در متد EnqueueCar به ماشین­های پشت پل یک واحد اضافه می­شود و در صورت بالا بودن پل، اکتور پیام StartLoweringBridge را به خودش ارسال می­کند. در این پیام حالت پل به درحال تغییر، تغییر می­کند. و یک رفتار فیزیکی برای مدل کردن پایین آمدن پل اجرا می­شود که در پایان این رفتار، حالت پل به پایین آمده تغییر می­کند و پیام PassACar به خود اکتور ارسال می­شود. متد PassACar به صورت دوره­ای با استفاده از متغییر timer یک ماشین را از پل خارج می­کند و پس از اتمام خروج در صورت وجود ماشین دیگری در صف، دوباره پیام PassACar را ارسال می­کند و در صورت اتمام ماشین­ها پیام StarRaisingBridge را ارسال می­کند که این متد بالا آمدن پل را مدل می­کند. پس از بالا آمدن پل، در صورت وجود ماشین، پل دوباره به شروع به پایین آمدن می­کند.



جایگاه پروژه نسبت به موارد مشابه

اهمیت و نوآوری پروژه ( در صورت لزوم )

شبیه‌سازی و آزمایش‌ها

جمع‌بندی و نتیجه‌گیری

1. Embedded [↑](#footnote-ref-1)
2. Feedback Loop [↑](#footnote-ref-2)
3. Safety-Critical [↑](#footnote-ref-3)
4. Hybrid Automata [↑](#footnote-ref-4)
5. Finite Transition System [↑](#footnote-ref-5)
6. Synchronous-Reactive [↑](#footnote-ref-6)
7. Discrete Event [↑](#footnote-ref-7)
8. Data Flow [↑](#footnote-ref-8)
9. Director [↑](#footnote-ref-9)