یک زبان مدل­سازی مبتنی بر اکتور برای سیستم­های سایبر-فیزیکی

ایمان جهاندیده

*چکیده*- یکی از چالش­های مدل­سازی سیستم­های سایبر-فیزیکی[[1]](#footnote-1)، 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 فراهم می­کنند. در این پروژه 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.

*کلمات کلیدی– سیستم­های سایبر-فیزیکی، مدل­سازی، مدل­سازی مبتنی بر اکتور*

# مقدمه

سیستم­های سایبر-فیزیکی از ادغام فرایندهای محاسباتی و فیزیکی ایجاد می­شوند. در این سیستم­ها، رایانه­های نهفته[[2]](#footnote-2) از طریق شبکه­، فرآیندهای فیزیکی را کنترل می­کنند که معمولا همراه با حلقه­ی بازخورد[[3]](#footnote-3) 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[[4]](#footnote-4) ،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[[5]](#footnote-5) 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 در این زبان سخت است.

در مقابل Timed Rebeca [1] 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 برداشتن 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[[6]](#footnote-6)

یک سیستم گذار یک چندتایی به صروت است که:

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

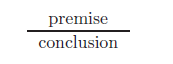
### خودکاره­ی ترکیبی

یک خودکاره­ی ترکیبی [1] 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.

### قوانین SOS

این قوانین از دو بخش فرض و نتیجه تشکیل شده­اند که در شکل 1 ساختار نمایشی این قانون آورده شده است. این قانون به معنی این است که در صورت برقرار بودن فرض، نتیجه نیز برقرار است.



شکل 1: ساختار قانون SOS

# مسئله و کارهای مشابه

زبان­های مختلفی برای مدل­سازی و تحلیل و درستی­یابی سیستم­های سایبر-فیزیکی ارائه شده است. این زبان­ها در صوری بودن مدل معنایی و نوع تحلیل و ساختار مولفه­ها متفاوت­اند.

Ptolemy [2] 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 بر اجرای همروند مولفه­ها و طریقه­ی ارتباط بین آن­ها است. در Ptolemy مدل­های محاسباتی مختلفی از جمله ماشین­ حالت متناهی[[7]](#footnote-7)، زمان پیوسته، واکنشی-سکنرون[[8]](#footnote-8)، گسسته رخداد[[9]](#footnote-9) و جریان داده­ای[[10]](#footnote-10) گنجانده شده است. برای ایجاد مدل­های ناهمگون، از اکتورهای سلسله مراتبی استفاده می­شود که هر اکتور مرکب شامل یک کارگردان[[11]](#footnote-11) 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 [3] یک زبان مدل­سازی برای سیستم­های ترکیبی است. یک مدل در این زبان دارای چند خودکاره و گروه است. خودکاره­ها رفتار یک سیستم را تعریف می­کنند و گروه­ها مجموعه­ای از خودکاره­ها و اعلان­ها هستند. گرامر این زبان نزدیک به خودکاره­های ترکیبی است. یک خودکاره­ی ترکیبی در این زبان از تعریف چند location و edgeهای مربوط به آن، توصیف می­گردد.

زبان Acumen [4] یک زبان مدل­سازی و شبیه سازی برای سیستم­های ترکیبی است. برای ساخت مدل در این زبان، از مفهوم "مدل" استفاده می­شود. هر مدل در این زبان حداقل دارای یک تعریف مدل Main است که شروع مدل را مشخص می­کند و تمام بخش­های دیگر سیستم باید در این قسمت تعریف یا ساخته شوند. رفتار­های فیزیکی هر مدل در بلاک always توصیف می­شوند. در این بلاک می­تواند از دستورات شرطی برای تعیین مدل­های مختلف سیستم استفاده کرد. رفتارهای فیزکی به صورت مشتق متغییرها تعریف می­شوند.

ابزار SpaceEx [5] 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.

در طراحی زبان HPalang مدل­سازی بخش­های مختلف سیستم­های سایبر-فیزیکی و ویرایش و بهبود کم هزینه­ی مدل تمرکز اصلی بوده است. مدل­سازی مبتنی بر اکتور یکی از روش­های موفق در زمینه­ی مدل­سازی سیستم­ها در انجمن مهندسی نرم­افزار است. انگیزه­ی این زبان، ارائه­ی زبان مدل­سازی مناسب مبتنی بر اکتور برای سیستم­های سایبر-فیزیکی است. همچنین با توجه به ساختار مبتنی بر اکتور، این زبان یک روش طراحی پیمانه­ای را ارائه می­دهد. این زبان اجازه تعریف رفتار­های سایبری به صورت دستوری فراهم می­کند که یکی از روش­های معمول برای بیان رفتار­های سیستمی است.

هرچند روش­های درستی­یابی در زمینه­ی سیستم­های سایبر-فیزیکی در ابتدای راه هستند ولی روش­های تقریبی و ایمن برای درستی­یابی زیرمجموعه­ای از خودکاره­های ترکیبی وجود دارد. مدل معنایی زبان HPalang مبتنی بر خودکاره­های ترکیبی بوده و علاوه بر شبیه­سازی، اجازه­ی درستی­یابی مدل­های این زبان را ارائه می­دهد.

# تعریف زبان

## نشانه­گذاری­ها

در تعریف زبان HPalang دنباله خالی با نماد نمایش داده شده است و دنباله­ای است که المان ابتدا آن و دنباله­ی باقی مانده است که یک مجموعه است. برای دو دنباله­ی و تعریف شده روی ، دنباله­ی حاصل از اضافه کردن به انتهای است.

برای تابع از نماد به معنی تابع استفاده شده است.

## نحو[[12]](#footnote-12)

مدل HPalang شامل تعریف چند اکتور و یک بلوک main برای پیام­های اولیه است. در شکل 2 گرامر زبان آمده است.

### اکتور

یک اکتور به صورت یک چندتایی به فرم تعریف می شود که شناسه، مجموعه­ای متغییرهای گسسته و پیوسته و مجموعه متد­های اکتور است. هر متد به صورت یک چندتایی تعریف می­شود که نام متد و بدنه­ی متد شامل دنباله­ای از دستورات است. پارامتر­های ورودی متد در این نسخه در نظر گرفته نشده است.

### دستورات

در این قسمت دستورات این زبان تعریف شده­اند. و به ترتیب مجموعه­ی متغییر­های گسسته و پیوسته هستند و و به ترتیب عبارات ریاضی شامل متغییر­های گسسته و عبارات ریاضی شامل متغییر­های پیوسته است. وجود پیشوند قبل از به معنی عبارات با مقادیر بولی است.

عملیات انتصاب یک مقدار گسسته به یک متغییر گسسته است. از نماد برای نمایش استفاده شده است.

عملیات انتصاب یک مقدار پیوسته به یک متغییر پیوسته است. از نماد برای نمایش استفاده شده است.

دستور شرطی است 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 می­شود. از نماد برای نمایش استفاده شده است.

*دستور ارسال است که یک پیام به یک اکتور است.* از نماد برای نمایش استفاده شده است.

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

*دستوری است اکتور را برای واحد زمانی مشغول نگه می­دارد.* از نماد برای نمایش استفاده شده است.

*دستور ادامه­ی اجرای دستورات است که پس از پایان اعمال می­گردد.* از نماد برای نمایش استفاده شده است.



*شکل 2*: گرامر زبان HPalang به فرم EBNF

### نحو 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.*
  + *گیرنده­ی دارای متدی با نام پیغام ارسال شده است.*

## معنای اجرای

*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.*

### توابع کمکی

*از توابع کمکی زیر را برای تعریف معنایی صوری در بخش­های بعد، استفاده شده است.*

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

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

### حالات

*اکتور­ها توسط ارسال پیام ارتباط برقرار می­کنند و پیام­های دریافتی را در صف ذخیره می­کنند. نوع یک پیام به صورت تعریف می­شود. اکتور دارای دو صف پیام و است که به ترتیب صف مربوط به پیام­های 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 گسسته­ی اکتور، و صف پیام­ها، دنباله­ی دستورات اجرایی و دنباله­ای از رفتار­های پیوسته­ی اکتور است.*

### گذار

*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

## Discrete Variable Assignment

## Conitional True

## Conitional False

## Resume Statement

## Continuous Behavior Statement

## Delay Statement

## Message Send

### سیستم گذار میانی

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 مقدارهی اولیه شده­اند و پیغام­های درون بلوک main در صف اولیت پایین اکتور­های مشخص شده گذاشته شده است.
* برچسب­هایی است که مطابق قوانین بالا مشخص 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، گذار­هایی شرط­دار این حالت حذف می­شوند.
* ادغام و حذف گذارهای : گذار­های ادغام شده تا در سیستم گذار نهایی تنها گذارهای شرط دار باقی بمانند.

خودکاره­ی ترکیبی حاصل از سیستم گذار به دست آمده به صورت زیر تعریف می­شود.

* مجموعه­ی متغییرهای پیوسته­ی مدل
* مقداردهی اولیه به متغییرهای پیوسته
* عطف منطقی ثبات­های تعریف شده در مکان
* مجموعه­ی عبارات دیفرانسیلی در مکان

# مثال­­ها

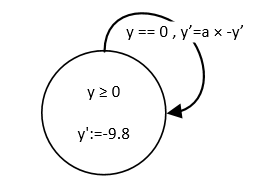
## سقوط توپ

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



شکل 3: مدل توپ

مدل تعریف شده این سیستم در شکل 3 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 مقدار اولیه آن h است. متد Fall، 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 تغییر می­کند. زمانی که y 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 بلوک 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 4 آمده است.



شکل 4: خودکاره­ی ترکیبی مدل توپ

## ماشین 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 سپس لیوان 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.

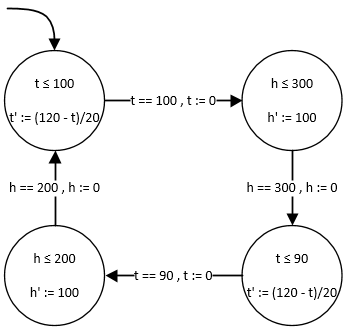


شکل 5: مدل اکتور Machine

در این مدل چهار اکتور User، Machine، Heater و Filler تعریف شده است. اکتور User رفتار کاربر را مدل می­کند که در این مثال خواص، کاربر 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 ارسال 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. در اشکل 5و 6مدل اکتور Machine و Heater آمده است. همچنین در شکل 7 خودکاره­ی ترکیبی معادل مدل نشان داده شده است.



شکل 6: مدل اکتور Heater



شکل 7: خودکاره­ی ترکیبی مدل ماشین نوشیدنی

## 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 پشت 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.

در این مدل دو اکتور تعریف CarDispatcher و DrawBridge تعریف شده است. CarDispatcher وظیفه­ی مدل کردن ورود ماشین­ها را دارد. این اکتور به صورت دوره­ای یک ماشین به DrawBridge اضافه می­کند. 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.



شکل 8: متدهای EnqueueCar و StartLowering

در متد EnqueueCar به ماشین­های پشت پل یک واحد اضافه می­شود و در صورت بالا بودن پل، اکتور پیام StartLowering 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. متد­های EnqueueCar و StartLowering در شکل 8 آورده شده است.

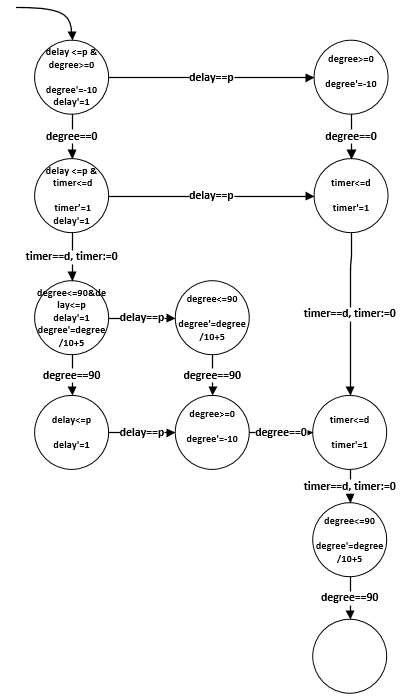


شکل 9: متدهای PassACar و StartRaisng

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 StarRaising را ارسال می­کند. 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. متد­های PassACar و StartRaising در شکل 9 نشان داده شده است. خودکاره­ی ترکیبی معادل برای دو ماشین در شکل 10 آمده است. متغییر delay برای رفتار دوره­ای CarDispatcher است.

# جمع‌بندی و کارهای آینده

مدل­سازی مبتنی بر اکتور به دلیل ساختار پیمانه­ای، اجازه­ی تعریف مدل­های پیچیده را از ترکیب مولفه­های ساده­تر، می­دهد. در زبان HPalang اکتور­های تعریف شده به طور مستقل و هم­روند اجرا شده و با استفاده از ارسال­ پیام­های غیرسنکرون با یکدیگر ارتباط برقرار می­کنند. همچنین هر اکتور می­تواند چندین رفتار پیوسته برای مدل­سازی رفتار فیزیکی تعریف کند. این امر باعث می­شود به راحتی بتوان رفتار­های پیچیده­ی سایبری را در کنار رفتار­های فیزیکی، مدل­سازی کرد.



شکل 10: خودکاره ترکیبی مدل DrawBridge

مدل معنایی تعریف شده برپایه خودکاره­ی ترکیبی، تنها نشان دهنده­ی رفتار فیزیکی سیستم در نتیجه­ی رفتار­های سایبری است ولی خود رفتار­های سایبری در خودکاره­ی نهایی وجود ندارند. یکی از کار­های آینده در این رابطه­، ایجاد یک چارچوب تحلیل برای این زبان است، تا بتوان رفتار­های فیزیکی و سایبری را در کنار یکدیگر تحلیل کرد. همچنین با توجه به تعریف مدل معنایی بدون در نظر گرفتن رفتار فیزیکی واقعی سیستم، حجم مدل خودکاره­ی ترکیبی با افزایش تعداد رفتار­های پیوسته، به طور نمایی افزایش پیدا می­کند، ولی بیشتر این حالات با توجه به مشخصات رفتار فیزیکی سیستم، غیر قابل درسترس هستند. با تحلیل تقریبی رفتار فیزیکی مدل، می­توان حجم مدل نهایی را به شدت کاهش داد.

در این نسخه­ی زبان تنها بخش ارسال پیام غیرسکنرون برای مدل­سازی رفتار شبکه­ تعریف شده است. یکی دیگر از کار­های آینده تعریف مفاهیمی چون تاخیر شبکه و مهلت پردازش پیام­ها است تا با توجه به نقش مهم شبکه در سیستم­های سایبر-فیزیکی، مدل دقیق­تری را بتوان تعریف کرد.

استفاده­ی ترکیبی از متغییر­های پیوسته و گسسته نیز یکی دیگر از زمینه­های تحقیق در این زبان است. در این نسخه امکان استفاده ترکیبی از متغییر­های پیوسته و گسسته در دستورات چون دستور شرطی و دستور رفتار پیوسته وجود ندارد.

# مراجع

|  |  |
| --- | --- |
| [1] | A. H. Reynisson, M. Sirjani, L. Aceto, M. Cimini, A. Jafari, A. Ingólfsdóttir and S. H. Sigurdarson, "Modelling and simulation of asynchronous real-time systems using Timed Rebeca," *Science of Computer Programming,* vol. 89, pp. 41-68, 2014. |
| [2] | R. Goebel, R. G. Sanfelice and A. R. Teel, "Hybrid dynamical systems," *IEEE Control Systems,* pp. 28-93, 2009. |
| [3] | C. Ptolemaeus, System Design, Modeling, and Simulation: Using Ptolemy II, Ptolemy. org Berkeley, 2014. |
| [4] | D. A. van Beek, W. Fokkink, D. Hendriks, A. Hofkamp, J. Markovski, J. M. van de Mortel-Fronczak and M. A. Reniers, "CIF 3: Model-Based Engineering of Supervisory Controllers," in *Tools and Algorithms for the Construction and Analysis of Systems - 20th International Conference, {TACAS} 2014, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2014, Grenoble, France, April 5-13, 2014. Proceedings*, 2014. |
| [5] | W. Taha, "Acumen 2014 Reference Manual," Halmstad University, Department of Computer, 2014. [Online]. Available: http://bit.ly/Acumen-manual-2014. [Accessed 23 1 2017]. |
| [6] | G. Frehse, C. L. Guernic, A. Donzé, S. Cotton, R. Ray, OLebeltel, R. Ripado, A. Girard, T. Dang and O. Maler, "SpaceEx: Scalable Verification of Hybrid Systems," in *Computer Aided Verification - 23rd International Conference, CAV 2011, Snowbird, UT, USA, July 14-20, 2011. Proceedings*, 2011. |

1. Cyber-Physical [↑](#footnote-ref-1)
2. Embedded [↑](#footnote-ref-2)
3. Feedback Loop [↑](#footnote-ref-3)
4. Safety-Critical [↑](#footnote-ref-4)
5. Hybrid Automata [↑](#footnote-ref-5)
6. Transition System [↑](#footnote-ref-6)
7. Finite Transition System [↑](#footnote-ref-7)
8. Synchronous-Reactive [↑](#footnote-ref-8)
9. Discrete Event [↑](#footnote-ref-9)
10. Data Flow [↑](#footnote-ref-10)
11. Director [↑](#footnote-ref-11)
12. Syntax [↑](#footnote-ref-12)