様　式　Ｃ－１９、Ｆ－１９－１、Ｚ－１９　（共通）

科学研究費助成事業　　研究成果報告書

平成　３０　年　　６　月　２０　日現在

機関番号：　３２６８９
研究種目：　挑戦的萌芽研究
研究期間：　2015～2017
課題番号：　１５Ｋ１２０１０

研究課題名（和文）実数と時間の概念を備えた汎用高水準プログラミング言語

研究課題名（英文）General-purpose high-level language with the notion of real numbers and time

研究代表者
上田　和紀（Kazunori, Ueda）

早稲田大学・理工学術院・教授

研究者番号：１０２５７２０６

交付決定額（研究期間全体）：（直接経費）　　2,700,000円

研究成果の概要（和文）：サイバーフィジカルシステムにおける計算とプログラミングの基盤の確立に向けて，連続量と時間を的確に扱うことのでき汎用の高水準プログラミング言語が備えるべき言語要素と意味論の詳細検討を推進した．目標とする言語は並行処理や通信機能を有する並行プログラミング言語であり，かつ連続量とその不確定性を扱うことのできる制約プログラミング言語でもあるという作業仮説に立ち，既存の言語に対する詳細検討を通じて，(1) 制約階層概念とその意味論，(2) 時間概念，(3) データ領域，(4) 動的に進化する並行系の記述，の各側面を中心に，目標とする言語の理論基盤と備えるべき機能の多くを明らかにした．

研究成果の概要（英文）：Towards the foundations for computing and programming of cyber-physical systems, we studied language constructs and semantics of general-purpose high-level programming languages that allow us to represent and handle continuous quantities and the notion of time. Our working hypothesis was that those languages should be concurrent programming languages featuring concurrency and communication and, at the same time, be constraint programming languages featuring continuous quantities and uncertainties. Through detailed study of existing languages, we clarified many of theoretical foundations and necessary language constructs, including (i) constraint hierarchies and its semantics, (ii) the notion of time, (iii) data domain, and (iv) description of dynamically evolving concurrent systems.

研究分野：　プログラミング言語

キーワード：　プログラミング言語　サイバーフィジカルシステム　ハイブリッドシステム　制約プログラミング　並行プログラミング

１．研究開始当初の背景

制約プログラミング (constraint programming)とは，変数値や変数値間の関係を，方程式や不等式を組み合わせた論理式（これを制約と呼ぶ）によって表現するプログラミングパラダイムである．研究代表者らは1980年代に，制約概念が並行計算の同期・通信機構の表現に使えることに着目して，並行制約プログラミングパラダイムの開拓に貢献した．並行制約プログラミングは離散的な計算の定式化から始まったが，1990年代に入り，時間の経過に伴う系の挙動が連続変化と離散変化の両方を示すハイブリッドシステムのモデリングへと拡張された．研究代表者らは，ハイブリッド並行制約言語として提案されたHybrid CC の使用経験に基づいて同言語の言語機能と実装法の再検討を進め，制約概念に基づく新たなハイブリッドシステムモデリング言語 HydLa を提案し，記号実行に基づく厳密なシミュレーションを行う処理系を構築してきた．

物理系と計算系とが相互作用をもつ体系を総称するサイバーフィジカルシステムは，米国を中心に社会，工学，科学面からの戦略研究が唱えられている．この中の科学面には，連続量とその観測制御機構からなるハイブリッドシステムのモデリングと，プログラミング言語における時間概念の扱いが，計算モデルとプログラミング言語の研究における喫緊の課題として含まれている (E. A. Lee: Cyber-Physical Systems—Are Computing Foundations Adequate? NSF Workshop On Cyber-Physical Systems, 2006).

シミュレーションや検証を目的とするモデリング言語においては連続量と時間の扱いが第一義的に重要であるが，モデリング言語は基本的に領域特化型言語であるため，一般的設計論が十分に開拓されてこなかった．一方，設計論や意味論の研究が盛んな汎用プログラミング言語においては，連続量やその一つである時間を厳密に扱う機能や方法の研究が課題となっている．

２．研究の目的

本研究は，サイバーフィジカルシステムにおける計算とプログラミングの基盤を与える高水準プログラミング言語の基本概念の確立を目標とする．

研究の推進にあたって，連続量の時間変化によって記述される物理系とのインタラクションを記述するプログラミング言語は，入出力や通信機能を有する並行プログラミング言語であり，かつ連続量やその不確定性を扱うことのできる制約プログラミング言語でもあるという作業仮説を設定する．この作業仮説に立って，並行と制約の両側面における研究代表者の過去の言語開発の経験を活かしながら，サイバーフィジカルシステム固有の考慮点と，汎用プログラミング言語の設計論を統合してゆく．物理系と計算系との統合においては物理系が設計の境界条件を与えるという観点から，HydLa を中心とするハイブリッドシステムモデリング言語の意味論と表現能力の詳細検討を軸に基本概念の確立を目指す．

３．研究の方法

稠密な時間を含む実数概念をもつ汎用プログラミング言語の確立には考慮すべき多くの技術的ポイントがある．本研究では，目標とする言語が並行プログラミング言語であり制約プログラミング言語でもあるという上述の作業仮説に基づいて，プログラミング言語分野およびハイブリッドシステム分野の知見を導入しつつ，言語要素および意味論の基本概念の詳細検討を推進する．

言語設計の妥当性を，意味論と記述能力にプログラミング言語としての実装可能性を加えた三面から保証すべく，研究代表者のグループで設計と実装（約 3 万行）を推進しているハイブリッド制約言語 HydLa とその処理系を基軸として各機能の検討を行う．さらに，高水準ハイブリッド言語 Acumen の設計と実装を推進している研究協力者 Walid Taha 教授（Halmstad 大学）のグループを年１回訪問し，両言語の相互比較および意味論などに関する集中検討を定期的に行う．

４．研究成果

時間を含む連続量を的確に表現する汎用高水準プログラミング言語の確立に向けて，言語要素と意味論の基本概念についての詳細検討を推進した．特に，HydLa および Acumen という二つの異なるモデリング言語に関する密度の高い研究交流を繰り返すことで，以下の項目について詳細な比較対照を進めることができた：（i）言語の基本設計原理，(ii) 言語が提供する基本概念（時間概念を含む），(iii) 言語の使用経験から得た知見，(iv) 言語が提供するデータ型，(v) 言語が提供する制御構造，(vi) 言語が提供する抽象化機能，(vii)プログラムの基本構成要素とその結合子，(viii) 入出力機能，(ix) 意味論，(x) 実装技術．

これらの擦り合わせを経て，本研究課題が目標とする新たな言語は，時間および連続量についての基本的な枠組において制約プログラミング言語である HydLa の計算モデルの多くの側面を引き継ぎつつ，系の動的進化の記述のためにはプロセスの動的生成と消滅を許す並行プログラミング言語の概念を導入することが適当であるとの知見を得た．制約と並行の両側面の統合を実現する具体的な言語のプロトタイピングを行うには至らなかったが，目標とする言語の理論基盤および備えるべき機能の多くを明らかにすることができた．

以下では，このような検討を通じて明らかになった知見について，四つの側面からその概要を述べる．

```
INIT <=> y = 10 & y' = 0.
FALL <=> [](y'' = -10).
BOUNCE <=> [](y- = 0 => y' = -4/5*y'-).
INIT, (FALL << BOUNCE).
```

図 1　弾む質点の HydLa モデル

(1) 制約階層と極大無矛盾性

目標言語のベースとなる HydLa の宣言的意味論の詳細化を行った．HydLa の特徴は，系の挙動の記述単位である制約モジュールに半順序階層構造を導入し，デフォルトの挙動と例外的挙動を簡潔に記述できるようにした点である．

たとえば，図 1 の HydLa プログラムは弾む質点の挙動を表し，INIT，FALL，BOUNCE の三つの制約モジュールからなる．変数の値は（暗黙的に）時刻の関数で，INIT は時刻 0 における質点の位置 y が 9 以上 11 未満の不確定値をとることを表す．FALL は質点の加速度が常に -10 であることを表し，[] はその制約が発行時点以降常に成り立つことを表す時相演算子である．BOUNCE は質点の位置が 0 に到達しようとするたびに，その速度が到達直前の速度の $-4/5$ 倍になることを表し，y やその微分値 y'の後ろの負号は時刻の関数である y や y'の値の左極限を表す．最後の行は，制約 INIT は有効で，BOUNCE も有効で，FALL は BOUNCE と無矛盾である時刻に有効であることを表明している．このように HydLa は，階層関係を満たす中で（各時刻において）極大無矛盾な制約集合が系の挙動を規定するものとした．

系の挙動を制約の極大無矛盾集合によって与えることの利点は，制約の無矛盾性(consistency)という制約プログラミングにおける最も基本的な概念を言語の制御構造として利用できる点と，制約条件を過不足なく与えることを容易にする点である．他方，この制御構造は，制約充足問題 (constraint satisfaction problems, CSP) の拡張である Max CSP 問題（充足する制約の個数を最大化する問題）をさらに半順序構造上での極大化問題に拡張するものであるため，効率の良い解法の開発，および，効率の良い求解を可能にするための制限条件の同定が課題となる．この観点から，プログラマが与えた制約階層が，制約過多もしくは制約不足に陥ることなく系の挙動を規定するかどうかを静的に解析する技法について，HydLa 処理系の上で設計と試験実装を行った．

系の軌道を規定する制約の極大無矛盾集合は，採用候補となる制約集合の中から一意に定まるとは限らない．ある時点で複数の極大無矛盾集合が存在する場合，その中の一つを非決定的に選択することになるが，これまでの HydLa の宣言的意味論では，複数の極大無矛盾集合がある期間にわたって存在しつづける場合，その期間中の任意の時刻で極大無矛盾集合の選択の変更を行うことを禁止していなかった．しかし，極大無矛盾集合を変更しなければならない時刻以外の時刻における変更を認めると，軌道の計算可能性の観点から困難をもたらす．このことから，「いったん選択した極大無矛盾集合は，変更が必要となる時点まで採用し続ける」という仮定を導入する必要があるとの結論に達した．これは，動的システムにおいては，論理的に帰結される以外の変化は引き起こさないという一種の慣性の法則の具体例と考えることができる．

```
N := {n0 .. n5}.
F := {f0 .. f5}.
[](f0 = 1 & n0 = n & f = f5).
{ [](N[i]> 0 =>
  F[i+1] = F[i]*N[i] & N[i+1] = N[i]-1),
  [](N[i]<=0 =>
  F[i+1] = F[i] & N[i+1] = N[i])
  | i in {1..|F|-1} }.
[](n = 3).
```

図 2　時間進行を伴わない繰返し計算

(2) 時間概念

ハイブリッドシステムのモデリングにおいては，系が離散変化を起こす時刻において，複数回の状態変化が連鎖的かつ時間経過なしに起きるモデルをいかに記述するかが課題の一つとなっていた．サイバーフィジカルシステムにおけるイベント処理の計算過程や，物理系における 3 個以上の物体の同時衝突などを表現することがその動機である．これらの記述を可能にするための時間モデルとして，時刻 t とその時刻 t におけるステップ番号 n とを対にした superdense time の概念が提唱されてきた．一方 HydLa では，系の軌道を通常の実数時刻の関数として扱っている．この時間概念は，一見 superdense time よりも記述力が劣るように見えるが，HydLa のアプローチが以下の二点において十分な記述力を持っていることを明らかにした．

まず，サイバーフィジカルシステムのモデリングでは通常，計算系が無限に高速であると仮定するが，この仮定が現実的であるための要件は計算ステップ数が有界であることである．この要件を満たす場合，各離散変化時刻で行われる計算過程のステップ数の最大値をNとすると，$N+1$ 個の変数 $x_0,\ldots,x_N$ を用意して，x_k が第 k ステップの計算結果を表すように $x_0,\ldots,x_N$ の間に制約ネットワークを構成すればよい．ここで，ある離散変化時刻 t における計算ステップ数が $n\ (\leq N)$ の場合，$x_{n+1},\ldots,x_N$ は x_n と同じ値になるように制約式を立てる．この方式を実際に HydLa でモデリングして動作を確認した．説明のため

の具体例として，$f = n!$ という制約 $(n \leq 5)$ を計算するプログラムを図 2 に示す．最初の 2 行は必要個数の作業用変数を生成するリスト記法，3 行目は作業用変数と n，f とを関連づける初期設定，4〜8 行目は階乗関数計算のための制約ネットワークの構築，9 行目が n の値の指定である．このように，制約プログラミングの枠組では，ステップ数が有界な計算は制約のネットワークにエンコードできる．さらにこの例を拡張して，n の値を時間変化させることもできることを確認した．なおこの例では各制約が優先度をもたないので，制約モジュールの宣言は行っていない．これらの考察と実験を通じて，制約プログラミングの枠組では，時間のモデルを通常のものから拡張することなく superdense time と同等の記述ができることを明らかにした．

次に，静摩擦と動摩擦のように複数の法則をもつ物理系では，多数の離散変化が微小な時間間隔で起きることがあり，それらのモデリングは上述の手法ではできない．このような物理系は，HydLa では微小量を表す記号定数を用いることでモデリングできる．たとえば「ニュートンのゆりかご」のような複数個の金属球の同時衝突を扱う場合，通常の衝突方程式は二体衝突しかモデル化していないため，金属球どうしの間隔を微小値 $\varepsilon\ (> 0)$ にするなどして，問題を微小な時間間隔で発生する複数回の二体衝突に変換する必要がある．HydLa の記号実行シミュレータ HyLaGI は，このようなモデルに対して ε を残したまま記号的に求解を行い，さらに $\varepsilon \to 0$ の極限をとることができる．微小量とその極限を任意に扱うことができれば，制御工学におけるスライディングモード制御のように，離散変化が頻繁かつ長期にわたって繰り返される系のモデリングとシミュレーションも可能になる．このように，記号的微小量や記号摂動の考え方は，領域特化型のモデリング言語だけでなく，より汎用のプログラミング言語においても有用であると期待される．

(3) データ領域

ハイブリッドシステムの解軌道のように，連続変化と離散変化の両方の挙動をもつ量を扱うための宣言的（表示的）意味論について，HydLa と Acumen の意味論の最新の知見の詳細な比較を通じた検討を行った．Acumen のデータ領域は位相空間として定式化され，データ領域自体に連続性の仮定を表現するメカニズムが組み込まれている．これに対して HydLa のデータ領域は単に時刻の関数として定式化している．図 1 の弾む質点のような Zeno 挙動をもつモデルのモデリングを通じて両者の比較検討を行った．その結果，Zeno 挙動をもつモデルの Zeno 時刻（弾む質点が弾まなくなる時刻）以降の挙動の記述にとって連続性の仮定は重要であるが，Acumen はそれをデータ領域の意味論レベルで与えているのに対し，HydLa は言語レベルでのデフォルト連続性の仮定によって同等の表現力を実現していることが判明した．また，どちらの言語でも，Zeno 時刻に何らかのイベントを発生させることができることを確認した．

連続性の仮定は「モデルから帰結できない（＝不必要な）変化は起こさない」という一般指針から来るもので，(1)で論じた慣性とも関連する原理であるため，データ領域には組み込まずに言語設計レベルで明示的に扱うことには合理性があると考えられる．

(4) 動的な並行性

ハイブリッドシステムのモデリングはこれまで，構造が決まっていて変数の個数が固定されている系を対象としており，構成要素の動的生成や消滅はほとんど考慮されてこなかった．これに対し，本研究が目指す高水準プログラミング言語は並行性について高い記述能力を持つことが求められ，それには最近の他の並行プログラミング言語と同様，構成要素の動的な生成や消滅の記述が含まれる．さらに，制約階層を備えた制約言語においては，構成要素の生成に伴って，それが満たすべき制約や制約間の階層も動的に生成する必要がある．

そこでまず，構成要素が動的に増加するハイブリッドシステムの記述を可能にするための変数の動的生成機能と制約階層の動的生成機能について，構文と意味論の設計を HydLa の拡張機能として行い，試験実装を行った．変数の動的生成は，存在量化子 $\exists$ を計算的に解釈することで実現できるが，制約階層を表す演算子 << は，論理和や論理積など他の論理演算子と異なり，その意味を構文主導の考え方でボトムアップに与えることができない点が大きなチャレンジ点となった．

次に，構成要素が動的に消滅するハイブリッドシステムを適切に記述するための枠組の基礎検討を行った．HydLa のように純粋に制約に基づくモデリング言語では変数の動的消滅の表現が困難であるが，並行プロセス計算ではプロセスの動的消滅が自然に記述できていることに着目し，並行プログラミング言語の枠組をハイブリッドシステムに適応させるのが有望な方向であるとの知見を得た．そのために必要な概念の整理を進め，大域的クロックを持たずに非同期的に計算が進む並行プログラミング言語においても，単一のクロックを共有して計算が進む局所的プログラム単位の記述においては，Esterel や Lustre に代表される同期型言語のフェーズの考え方が参考になるという知見を得た．さらに，同期型言語のフェーズの考え方は，ハイブリッド制約言語 HydLa の各フェーズにおける閉包計算と密接に関連することが明らかになってきた．目標とする言語は並行言語でありかつ制約言語であるという作業仮説のもと，上述の各概念，すなわち制約，制約階層，閉包計算などを並行言語に組み入れる作業が次のステップとなる．

もうひとつの重要な側面は，実用言語に必要となる動的データ構造機能である．そこで，制約の概念を実数領域だけでなくデータ構造領域にも活用することによってデータ構造概念を自然に導入する可能性について検討を行った．この方向はすでに並行制約言語を含む制約プログラミング言語で多大な実績があり，大きな技術的困難はなく妥当な方向であるとの結論を得た．

以上のように，本研究ではハイブリッド制約言語 HydLa を軸足としつつ，関連する言語やパラダイムが提供する概念や機能の検討および関係の解明を通じて，実数と時間を備えた高水準言語の基本概念の確立に近づく成果を得ることができた．もちろん，具体的かつ本格的な高水準言語を構築するためには多くの検討課題が残されている．たとえば，モデリングの観点からは有用だが求解の観点からは挑戦的課題の多い微分代数方程式をプログラミング言語においてどのように扱うかについての検討，浮動小数点計算に基づく近似実行系が計算した解軌道が意味論の規定する解軌道とどれくらい適合するかの尺度の検討，などである．多くの関連分野の知見を統合してこれらの問題の解決を図るのは今後のチャレンジである．

５．主な発表論文等
（研究代表者、研究分担者及び連携研究者には下線）

〔雑誌論文〕（計４件）

(1) Kazunori Ueda: Logic/Constraint Programming and Concurrency: The Hard-Won Lessons of the Fifth Generation Computer Project. Science of Computer Programming, 2018, http://dx.doi.org/10.1016/j.scico.2017.06.002. (査読有)

(2) Kenichi Betsuno, Shota Matsumoto, and Kazunori Ueda: Symbolic Analysis of Hybrid Systems Involving Numerous Discrete Changes Using Loop Detection. Sixth International Workshop on Design, Modeling, and Evaluation of Cyber Physical Systems, LNCS 10107, Springer, 2017, pp.17-30. (査読有)
DOI: 10.1007/978-3-319-51738-4_2.

(3) Shota Matsumoto and Kazunori Ueda: Symbolic Simulation of Parametrized Hybrid Systems with Affine Arithmetic. 23rd International Symposium on Temporal Representation and Reasoning, 2016, pp.4-11. (査読有)
DOI: 10.1109/TIME.2016.8

(4) Shota Matsumoto, Fumihiko Kono, Teruya Kobayashi and Kazunori Ueda: HyLaGI: Symbolic Implementation of a Hybrid Constraint Language, Electronic Notes in Theoretical Computer Science, Vol.317, 2015, pp.109-115. (査読有)
DOI: 10.1016/j.entcs.2015.10.011

〔学会発表〕（計１４件）

(1) 佐藤柾史，上田和紀：ハイブリッドシステムモデリング言語 HydLa における変数と制約階層の動的生成記法の設計と実装．2018 年度人工知能学会全国大会，2018.

(2) 増田健太，上田和紀：ハイブリッド制約言語 HydLa における非線形常微分方程式の表現とその記号付き精度保証計算．2018 年度人工知能学会全国大会，2018.

(3) 小山峻平，上田和紀：制約階層に基づくハイブリッドシステムモデリング言語 HydLa の静的誤り検出手法．情報処理学会第 80 回全国大会，2018.

(4) Kazunori Ueda: High-Level Programming Languages and Systems for Cyber-Physical Systems. Halmstad Summer School of Cyber-Physical Systems, Halmstad University, 2017. (招待講演)（国際学会）

(5) 松本翔太，別納健市，増田健太，上田和紀：制約に基づいたパラメトリックハイブリッドシステムの精度保証シミュレーション．2017 年度人工知能学会全国大会，2017.

(6) 小山峻平，松本翔太，上田和紀：ハイブリッドシステムモデリング言語 HydLa におけるモデリングエラーの体系化．2016 年度人工知能学会全国大会，2016.

(7) 別納健市，松本翔太，若槻祐彰，上田和紀：多数の離散変化をともなうハイブリッドシステムに対するループ検出を用いた解析．2016 年度人工知能学会全国大会，2016.

(8) 若槻祐彰，松本翔太，上田和紀：ハイブリッド制約処理系 HyLaGI における LTL モデル検査．2016 年度人工知能学会全国大会，2016.

(9) 松本翔太，上田和紀：パラメタを含むハイブリッドシステムに対するアフィン演算を用いた記号シミュレーション．日本ソフトウェア科学会第 33 回大会，2016.

(10) 松本翔太，上田和紀：ハイブリッドシステムのシミュレーションにおける精度保証数値計算と数式処理との連携．電子情報通信学会ソフトウェアサイエンス研究会 (SS2015-60)，2016.

(11) 若槻祐彰，松本翔太，伊藤剛史，和田

努，上田和紀：ハイブリッド制約処理系HyLaGIによる微小誤差を用いたモデル解析．日本ソフトウェア科学会第 32 回大会，2015.

〔その他〕

ホームページ等

(1) http://www.ueda.info.waseda.ac.jp/hydla/ (HydLa 処理系ポータルサイト)

(2) http://github.com/HydLa/HyLaGI (HyLaGI GitHub レポジトリ)

６．研究組織

(1) 研究代表者

上田 和紀 (UEDA, Kazunori)

早稲田大学・理工学術院・教授

研究者番号： 10257206

(2) 研究協力者

TAHA, Walid

松本 翔太 (MATSUMOTO, Shota)