

米国における先端的ソフトウェア研究 (SDP計画) 進捗報告

- SDPの研究計画策定のためのワークショップ紹介 -

米国におけるITの中心的研究開発計画であるNetwork and IT R&D (NITRD)計画は順調に発展を続けている。

1999年2月のPITAC勧告によって追加された21世紀における革新的なソフトウェアの設計と製造方法の開発を目指す新研究領域: **Software Design and Productivity (SDP)** については、この領域のビジョンと研究計画策定のための **ブレインストーミング的ワークショップ** が開催された。

NITRD全体、およびSDPの予算は順調に伸びており、米国のITやソフトウェア研究への投資がさらに強化されていることを示している。

Request	NITRD全体	SDP予算
FY2002	1,830 M\$ (2,200億円)	182.1 M\$ (220億円)
FY2003	1,889 M\$ (2,270億円)	196.7 M\$ (236億円)

この種のワークショップは米国の連邦政府支援R&D計画を実施する場合によく行われるもので、衆知を結集し、提案されるプロジェクトの焦点を絞る意味合いがある。この **SDPのPlanning Workshop** も同様のもので、2日間にわたって議論が展開され、その結果がまとめられている。

その方向は、**ユビキタス時代を迎え、ますます複雑化するソフトウェアにいかに対処するか。** 具体的には、その信頼性をいかに確保するか。いかにして生産性をあげるか、そのような要求の基盤となる新しいソフトウェア理論や工学をいかに組み立てるかなど、興味深い課題が議論されている。

このワークショップで投げかけられた課題やそれらについてのスピーカのプレゼンテーションについて、そのスライド資料のみが公開されている。ワークショップは4つのパネルに分けられ、それぞれパネルではその分野の代表的研究者4 - 5名がおのおのの考えを発表し、それらを各パネルごとにまとめている。

AITECのHigh-End Computing and Communication Working Group (HECC-WG)では、このワークショップの内容をWG委員の一人である近山隆氏に手短かに紹介・解説してもらった。なにぶんにも、**もとの資料が謎めいた今後のソフトウェア研究の課題を短く表現したものである。** 具体的な意味合い不明の部分もあるが全体的には、きわめてオーソドックスな提言が多く、**基礎から積み上げようという意図が感じられる。また、政府もそのような方向に研究を導こうとしている。わが国に欠けている方針である。**

ともあれ、わが国のソフトウェア研究者諸氏が、ユビキタス時代に向けてわが国がソフトウェア研究にいかに取り組むべきかを考える一助になるのではなかろうか。このような意図をもって、本WG資料を公開することにした。パネリストの使用したスライドも公開されているのでぜひご覧ください。(AITEC 内田俊一)

* 参考URL

<http://www.hpcc.gov/iwg/pca/sdp.html>

<http://www.hpcc.gov/iwg/pca/sdp/sdp-workshop/vanderbilt/>

<http://www.hpcc.gov/iwg/pca/sdp/sdp-workshop/planning/>

米国 NSF SDP の目指すもの 2001年4月のWSから

近山 隆

東京大学

新領域創成科学研究科

ソフトウェアはあらゆる活動の源

- 全米でソフトウェア技術者200万人
 - サポートスタッフ35万人
 - まだまだ空きポスト多数
- というわけで、
- ソフトウェア生産性の劇的向上
 - 高いソフトウェア品質
- を実現する研究方向を見定めるために...

Planning Workshop on New Visions for Software Design and Productivity

- 2001年4月18日 ~ 19日 Arlington
- NSF主催
- イントロと四パネルで構成
 1. ソフトウェアの将来とソフトウェア研究
 2. ソフトウェア開発の新パラダイム
 3. 実世界のためのソフトウェア
 4. ネットワーク中心の分散ソフトウェア

参加者

Frank Anger, NSF; Barry Boehm, USC; Grady Booch, Rational Software; Doris Carver, Louisiana State Univ.; Paul Domich, Office of Science and Technology Policy; Martin Feather, Jet Propulsion lab.; Steve Fine, EPA; Ian Foster, Argonne National lab.; James Fowler, NIST; Cita Furlani, NCO/ITRD; Helen M. Gigley, NCO/ITRD; Helen Gill, NSF/DARPA; John Grosh, Office of the Deputy Under Secretary of Defense for Science and Technology; Bonnie John, Carnegie Mellon Univ.; Ralph Johnson, Univ. of Illinois; Gabor Karsai, Vanderbilt Univ./ISIS; Gregor Kiczales, Univ. of British Columbia; Norman Kreisman, DOE; Patrick Lardieri, Lockheed Martin; Doug Lea, State Univ. of New York at Oswego; Edward Lee, UC Berkeley; Ernest Lucier, FAA; William Mark, SRI International; Cleve Moler, MathWorks, Inc.; Benjamin Pierce, Univ. of Pennsylvania; Adam Porter, Univ. of Maryland; Marshall Potter, FAA; Steven Ray, NIST; Larry Reeker, NIST; Shankar Sastry, UC Berkeley; Rick Schantz, BBN Technologies; Douglas Schmidt, DARPA; Charles Simonyi, Microsoft; Frank Sledge, NCO/ITRD; Kevin Sullivan, Univ. of Virginia; Janos Sztipanovits, DARPA; John Vlissides, IBM; Ralph Wachter, ONR; Jim Waldo, Sun Microsystems, Inc; Don Winter, Boeing Phantom Works

Agenda

- **第1日**

- 08:30-09:15 イントロ
- 09:15-17:15 よっつのパネル
- 17:15-17:30 まとめ

- **第2日**

- 08:00-08:15 各パネルへの課題
- 08:15-11:15 パネルごとの検討
- 11:15-13:00 検討結果の報告

初日のイントロ (Frank Anger, NSF) まず PITAC について

President's Info. Tech. Advisory Committee
(1991 ~)

- 大統領・国会・連邦政府機関に対しITにおける米国の優越を保つ方法を答申する独立機関
- “連邦政府はソフトウェアの基礎研究にabsolute priorityを与うべし!” (1999年2月)

PITACのメンバー (1997-2001)

Eric A. Benhamou, 3Com; Vinton Cerf, WorldCom;
Ching-chih Chen, Simmons College;
David M. Cooper, Lawrence Livermore;
Steven D. Dorfman Hughes Electronics; David W. Dorman, AT&T;
Robert J. Ewald, Learn 2 Corp; David Farber, Univ. of Pennsylvania;
Sherrilynne S. Fuller, Univ. of Washington;
Hector Garcia-Molina, Stanford Univ; Susan L. Graham, UC-Berkeley;
James N. Gray, Microsoft Research; W. Daniel Hillis, Applied Minds, Inc.;
William Joy, Sun Microsystems; Robert E. Kahn, CNRI;
Ken Kennedy, Rice Univ; John P. Miller, Montana State Univ.;
David C. Nagel, Palm, Inc.; Raj Reddy, CMU;
Edward H. Shortliffe, Columbia Univ.;
Larry Smarr, UCSD; Joe F. Thompson, Mississippi State Univ.;
Leslie Vadasz Intel; Andrew J. Viterbi, QUALC• OMM Inc.;
Steven J. Wallach, Chiaro Networks; Irving Wladawsky-Berger, IBM

PITACの調査結果

- ソフトウェア需要は生産能力を上回る
- 国家は脆いソフトウェアに依存している
- 信頼できる安全なソフトを作る技術が不十分
- ソフトウェア**基礎研究への国家投資**が不足

PITACの推奨した政策

- ソフトウェア開発メソッド・要素技術の
基礎研究に投資すべし
- 諸領域においてSWコンポーネントの
国立ライブラリを提供すべし
- あらゆるIT研究イニシアチブで
SW研究を重要な要素とすべし
- ヒューマン・コンピュータI/Fの
基礎研究に投資すべし

主要な研究領域

- ソフトウェア
- スケーラブルな情報インフラ
- ハイエンドコンピューティング
(SW R&D含む)
- 社会経済的および総労働力上の影響

でも、どうすればよい？

- 解くべき**基礎的な問題**は何？
Javaをセキュアにすることとかじゃないよね...
- 真の**障壁と挑戦**は何？
絶対の安全とか保証とか、無理だよな...
- どの**方向**がもっとも有望？
どうも究極の形式仕様記述とかじゃなさそうだし...
- 何が**理想的成果**？
無限の生産性、でも最低の品質、なんてのじゃだめだよな...
- **どのように研究を進めるべき**？
「理論的力技」とか「永遠に作っては壊し」じゃないよね...

真の障壁と挑戦は...

- 無限の複雑性を知的に制御すること
- システムは:
 - 分散し、
 - 不均一で、
 - ハイブリッドで、
 - 信頼できないのを覚悟しないと

SDP の研究計画を作ろう

- **真の問題に挑戦**しよう
- 技術的進歩をもたらす環境を作ろう
「発明は必要の母」
- 明日の問題を解こう
 - ITは急速な変化をもたらしたが、その犠牲者でもある
 - 技術の収斂と応用の拡大を
- **あくまで工学的問題**だということを忘れずに

で、お金の話

- NCO (National Coordination Office for IT R&D) はいろいろな機関の研究を調整
 - NASA, NIST, DARPA, NSF, ARO, ONR, DOD, FAA, DOE, NOAA, NSA, NIH, ...
- 研究投資を引き出すには魅力的計画を!
- NSF/CISEは5億ドル持っている
最低その1割ぐらい獲得したい!
- NSFはごく基礎的な研究に関心

2日目: アジ演説は続く...

- ソフトウェアプロセス・ソフトウェア作成の基盤になる科学って何?
- サイズ、複雑さ、合成可能性、試験可能性などの根本的限界はある?
- 生産性向上のために評価すべき研究項目は?
要素技術? 高度な抽象化? 利用者教育?
オープンSW? 開発プロセスの自動化?
新たな開発アプローチ?

さらに...

- どうすれば変化に対応するように作れる?
- 大量の既存コードを(非難するのではなく)うまく利用するには?
- サイバーワールドの「都市計画」は?
スラムや歓楽街ばかりには、したくない
- これまでの基本仮定を崩すような新技術は?
- 技術注入の障壁を低くするには?

というわけで、各グループに宿題

- 昼までにPowerPointスライド+ノート
方向性、挑戦課題、心配事、など
- 研究テーマにつながる具体的アイデアを
- 連邦研究予算はどんな研究でも支援できる
- 5億ドルあったら何をしたいか考えて欲しい

Panel 1: ソフトウェア研究の将来

Barry Boehm, Benjamin Pierce

Doris Carver, Shankar Sastry

Bonnie John, Kevin Sullivan

Bill Mark

仮説

- 現在のインフラはまったく不適切
 - 低い信頼性、安全性、可用性
- 大規模SWプロジェクトの半数は失敗
- 応用SW同士が情報をやりとりできない
- 過去のソフトウェア資産が1兆ドルもある

目標

- **各領域の専門家**がSWを使いこなせるように
- 既製品ソフト業界の共通基盤となるアーキテクチャを開発
- SW業界に対し、鍵となる概念・技術を提供
- プログラムの**意図**を機械が理解できる形で記録する方法を開発
- SW開発の現状をしっかりと認識

特に注目すべき領域は

- ネットワーク埋め込み型
- 対人インタラクション
- グループインタラクションの計算機支援
- 領域専門家のための環境
教員; エンジニア; 金融; 科学者

基礎となる科学は...

- ソフトウェア物理
 - 並行処理のモデル
- ソフトウェア経済学
 - 柔軟性の価値を定量化
- ソフトウェア行動科学
 - 人間の認知・記憶に関する知見をSW設計に応用
 - ソフトウェアと人間の役割分担

鍵となるポイントは

- SW開発のプレイグラウンドを変えよう
 - 「プログラム」の意味の変更: 天動説? Simulink?
 - 「プログラマ」の意味の変更:
practitioners (開業医などの専門家)
- 鍵となる進歩の提供
 - 軽い形式的枠組
 - 並行性・耐故障性・安全性などのモデル
 - 「価値」に基礎を置いたデザイン
 - 分散オープンソーステスト方式
- SW 経済学を変えよう
- HW を定義しよう(HWがSWを、ではなく)

Panel 2: 新SW開発パラダイム

複雑性さは指数的な増加

- 組込型、モバイル、浸透性、大域性
- 長寿命、複数バージョン混在
- 共同開発
- 利用者ごとのカスタマイズ
- ...

ゴルディオスの結び目は切らねば解けない!!

国立ソフトウェア考古学センター

- **ソフトウェアの古典を公開** (Webで)
- 何をすればよいかということ
 - ソフトウェア古典の蒐集と保存
 - 芸術的本質をリバースエンジニアリング
- 何に役立つかということ
 - 芸術的表現を奨励、既存技術の特許化、
将来の研究の基礎、SWの基本構成の宝庫
- 「SWゲノムプロジェクト」と呼ぶべきかも

多面的ソフトウェア

- SWは様々な異なる意図を表現したもの
- SWのこれまでの進歩の多くは
 - ある側面をよりよく理解した、または
 - ある側面をうまく表現した、というもの
- 新たなシステムは共存する異なる視点が互いに影響しあって生まれる
 - 従来はこのことについて認識不足

多面的ソフトウェア: 関連研究

- デザインパターン
- 協調開発環境
- アスペクト指向プログラミング
- 特定領域用言語

多面的ソフトウェア: すべきこと

- どの側面が肝要か
性能、価格、使い勝手、過去からの連続性...
- 各側面をどう表現するか
- 諸側面をどう変形・統合しシステムとするか
- 特定領域用言語のためのツール整備の加速

多面的ソフトウェア: 役立つこと

- 設計上の選択とトレードオフの明確化
 - 開発プロセスのメリットはそのままに
 - システム設計を考えるようにしむける
- さまざまな要請に対し
 - 他の要請と独立に考えることを容易にしながら、
 - 他の側面との折り合いを明示的につけられる

協調開発環境

- Webを仮想会議場として、SW開発
- なすべきこと
 - ソフトウェア開発の社会学
 - ソフトウェアプロセスの利用
 - 考古学センターや多面的ソフトウェアを利用するプラットフォームの創造

研究計画例

- 側面中間形式
 - 特定領域用言語のための汎用ツール
 - Weavers?
- 対応代数 (Correspondence Calculus)
- デモプロジェクト (Demonstration Projects)
- 領域回復 (Domain Recovery)
- ソフトウェア保存 (Preservation)
- ソフトウェア開発民俗学 (Ethnography)

Panel 3: 実世界ソフトウェア

Don Winter

Martin Feather, Gabor Karsai, Patrick Lardieri,
Cleve Moler, Edward Lee

現状

- **今日のソフトウェアの持つ問題点**
偶然的複雑性、予測不能性、合成不能性、脆弱性、実世界との相互作用に向かない
- **ソフトウェアが「コードの集まり」という考え方にハマっている**
 - 大規模にするには人力を注ぎ込む
 - システム全体像の理解の喪失
 - 仕様とその実現の分離

問題は...

- 10年で世界はまったく変わるだろう
どこにでもコンピュータだらけになる
- 今日のようなソフトの作り方をしていたら、
世界は危険極まりない場所に...
- さもなければ、技術の実施が遅くなる
自動車はコンピュータ制御にはならず、
あなたはまだ交通渋滞の中に...

モデリング言語

- 人間が以下を表現するモデリング言語がない
 - 複雑な機能の実現
 - 設計の理解
 - 質問の定式化
 - 振舞いの予測
- 問題はプログラムコードではない
「モデル」あるいは「設計」の問題
「仕様」の問題ではない

モデリング言語: 研究テーマ

- システムのモデリング「言語」
- 有用な抽象化を見つけること
- 計算的システム理論
- 組合わせて使える抽象化
- 時間・並行・能力などの表現

システムの合成

システムを組合せて作る組織的方法の欠如

- 要素の枠組
- 組合せの意味論
- その場での (on-the-fly) 合成
- 既成のソフトウェア要素との統合

システムの合成: 研究テーマ

- 意味の枠組と理論
- 方法論 (methods) とツール
- 実験台と挑戦的問題
- 参照実装 (reference implementation)
- アーキテクチャ枠組の定義
- 分散・分割の戦略
- 粒度とモジュラー性の制御戦略

変換 (Transformation)

異なる抽象間の変換についての理論の欠如

- 異なる抽象の間関係
- 生成器 (変換器)
- 複数の視点を許す抽象化
(multi-view abstraction)
- 物理的環境の抽象化
- HCSSとの関係: 証明つきの変換

変換: 研究テーマ

- オープンな生成器インフラ
(方法論、ライブラリ)
- 生成器の理論
- Methods for correct by construction
transformers (?意味不明?)
Methods for correct transformer construction?
- 変換器のパラメタ化

過去のソフトウェア遺産

- 過去の遺産を扱う方法論の欠如
 - システムを徐々に現代化する方法
 - 新しいものを古いものと統合する方法論の欠如
- 研究テーマ
 - 遺産コードの部品化
 - 遺産からの抽象化の抽出
 - 漸次現代化、リバースエンジニアリング
 - 再設計を安価に ソフトウェア遺産からの進化

Panel 4: ネットワーク中心システム

Ian Foster
Jim Fowler
Doug Lea
Adam Porter
Steve Ray
Rick Schantz
Jim Waldo

状況

- すべてがコンピュータ
- すべてがネットワークコンピュータ
- すべてが相互依存する可能性
- 実世界と接続
- ますますヘテロに
- 接続はますます多様で不安定に

応用の例

- 交通制御: 自動車数千台からセンサーデータ
- 戦域管理: 敵対環境で多様粒度の調整・任務
- サプライチェーン管理
- 科学データの共同解析
 - 数千ユーザ、数千資源に対し、ソフトリアルタイムで問合せ最適化の要請
- 家庭内電力管理

技術的問題

- 資源を意識したプログラミング
QoS, 時間・相互依存・エネルギー消費・大きさ
 - 工学的トレードオフを考慮する計算モデルの欠如
 - 複雑なシステムでは端点間の特性が不明
- 動的な資源の振舞い: 故障・不均一性 等
 - 計算モデルや端点間特性の欠如
- ソフトウェア遺産
設計時に想定しなかった利用形態

技術的問題 (続き)

- 危機・信用・制御の管理
 - ポリシー・保障・管理のドメイン
 - 極度に動的なシステムの安全性確保と確認手法
 - プライバシ
- スケールの問題
 - 構成要素数; 構成要素の大きさ
 - 単一の計算に関する要素数
 - ネットワーク中心システムは長時間無停止動作

研究方向とアプローチ

1. 「問題」を扱うプログラミングモデル

- 工学的トレードオフ
- 大規模システムの運用
- 動的な資源の振舞い
- 危機・信頼・安全の管理

2. 「問題」の基本機構

- 資源のトレードオフ: QoS, 実時間性、等
- 適応的振舞い
- さまざまな次元への拡大
- 分散管理

研究方向とアプローチ(続き)

3. 「問題」の全体を扱える、多レベルの資源管理
4. ネットワーク中心の(小規模な)要素を、動的に
大規模システムに合成する技術
interoperability

他の必要な研究テーマ

- 新たな計算モデルを使うためのツール
- 高度に複雑化したシステムの検証・シミュレーション・理解
- 大規模分散応用システムの (自動) 構成・管理

効果

- 社会的には
 - 現状では作れないものを作るように
 - ネットワーク化システムの信頼性・制御性向上
工学的設計によって実現 デバッグ
 - ソフトウェア開発コストの抑制
- 教育
 - 新たなソフトウェアの作成・システムの設計にあたる人材の供給

アプローチについて

- チームワークは必須
- 本当の問題を見つけ、中間成果を検証するには、大規模プロジェクトが必要
- 国際協力が望ましい

全体を通して(近山の感想)

- 基本的には予算取り
- あまり新しいことは言っていない
- まじめな議論は、された様子
 - 象牙の塔にこもるようなテーマは提案していない
 - 単なるソフトウェア開発も提案していない
- 抽象化を武器に複雑性・規模・不均質等に挑戦
- 過去の遺産が常に問題