

[らぼなび！]

らぼなび！

第 3 回 大阪大学大学院 情報科学研究科 井上研究室 の巻

井上 克郎教授，肥後 芳樹さん (D3)，市井 誠さん (D1)，森岡 佑さん (M2)



第 3 回目の「らぼなび！」は、大阪大学大学院 情報科学研究科 の 井上研究室 にお邪魔しました。井上研はソフトウェア工学を専攻分野としておられる研究室ですが、再利用や保守などのソフトウェア開発の後工程に焦点を当てて研究活動をされています。

井上研の研究の特徴は、コードクローンと呼ばれるコードの類似部分を発見するツールや、コードを解析して求められる部品の重要度に基づく部品検索ツールなど、大量のデータを処理可能なコンピュータの特性を生かしたものづくりをされていることでした。またその研究内容

は、ソフトウェアを開発している現場が常々抱えている問題を解決するためのものでした。

ソフトウェア開発の現場をより良くしたいと願う井上研究室の井上教授と学生 3 名へのインタビューです。

井上克郎教授



ものを作ってみて、使ってみて、初めて始まる

-- まず始めに、井上先生の研究の概要を教えてくださいか？

ソフトウェア工学一般、その中でも分析・評価といった開発の後工程、下流や保守と言われる部分を中心に行っています。また、奈良先端科学技術大学院大学が中心となって活動している EASE プロジェクトにも 参加 しています。具体的な研究テーマとしては、以下のようなものがありますね。

1. コードクローン [[CCFinder/Gemeni](#)]
2. プログラム部品の検索と再利用 [[SPARS](#)]
3. プログラムスライシング
4. ソフトウェアの自動分類 [[MUDABlue](#)]
5. アスペクト指向
6. ソースコードの構造を考慮した版管理システム
7. 実行履歴の可視化
8. ソフトウェアの開発履歴の理解促進
9. 開発履歴情報を利用したソフトウェアアーキテクチャ理解支援
10. メタモデルを用いた成果物の一貫性管理
11. ソフトウェア理解支援を目的としたブラウザ

-- ソフトウェア工学の研究を大学で行おうとしたときに、難しい面はありますか？

一般的には難しいと言われてはいますが、私自身は難しいとは感じていません。他の研究を経験していないからかもしれませんが。ただ、ソフトウェア工学は理論だけではうまくいかず、社会のニーズがないと成立しない研究分野だと思っています。ですので、大学に閉じこもっていても、インパクトを与えられる研究成果は出ない。社会と接点を持って、そこでヒントを貰い、それを元に大学で理論的な背景をはっきりさせて、社会に返す。そういう部分が大切だと思っています。

-- 社会と接点を持つことは難しいと思うのですが、どうやって克服されているのですか？

いろいろやっていますよ。例えば、いろいろな人とお話ししたり、会合に出てお酒を飲んだり(笑)。

-- この研究室では、ものづくりをたくさんされている印象があるのですが、それも社会との接点を持つためでしょうか？

そうですね、やはり形にしないと、世の中に影響を与えられないと考えています。基本的にプログラムを作ることができる学生には、卒業研究などで使えるツールを作ってもらうようお願いしています。さらに、オープンソースとして提供をお願いするものもあります。

この業界は、ものを作ってみて、使ってみて、初めて始まるという部分が多くて、「こんなアイデアがあるよ。」や「こんなやり方があるよ。」と言うだけでは伝わらない。ツールを作って持っていても伝わらないのですが、そもそもツールがなかったら話にも拳がらない、そんな雰囲気がこの業界にはあると思っています。これまでに作ったものだと、コードクローンの分析は、[CCFinder](#) というツールが世間では結構使われていますし、[SPARS](#) という部品検索システムも、オープンソースとして提供しています。

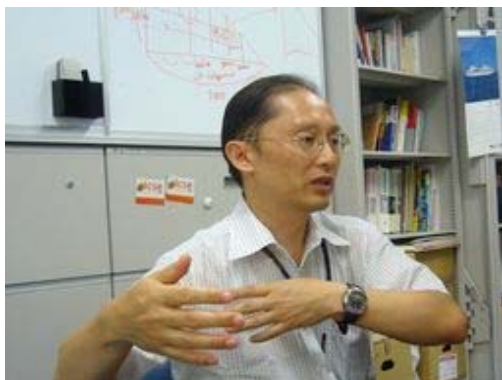
.....

-- これらの研究のきっかけについて教えていただけますか？

コードクローンの研究は、ある企業の方からうかがった話がきっかけで始めました。その方がおっしゃるには、「ソースコードの保守のために、どこで同じコードを使っているかを調べてドキュメントにしているが、そのドキュメントがだんだん散逸してきて困っている」ということだったんです。その話を学生とした結果、ツールを作ればよいのでは？という話になり、始めたのがきっかけです。ニーズがあって研究を始めた例ですね。

一方、プログラム部品の検索・再利用の研究は、コンピュータにパワーが出てきて、メモリもディスクも豊富になったのに、ソフトウェアエンジニアリングはその恩恵をこうむっていない。高性能になった計算機をぶん回して(*1)、なんとかソフトウェアエンジニアリングをできないかと考えていたんですね。そのときに、[Google](#) で行っているような検索というアイデアを、ソフトウェアエンジニアリングでも使えないかと考え、ソフトウェア部品を検索する研究を始めました。こちらはシーズから研究を始めた例ですね。

ものづくりも大事だけど、学生は論文や学会で、もっと自分をアピールしてほしい



-- 研究していると、ものづくりだけでは評価されない部分がありますよね。でも、プログラミングにはまると、文献調査や論文書きなどが疎かになってしまったりしそうです。なにか特別な工夫はされていますか？

工夫はないですよ。苦労はしていますが（笑）。両方できる学生はなかなかいないですね。コーディングが得意な学生は、文献を読んだり結果をまとめたりするのは苦手。逆にそれができる学生は、コーディングが苦手です。ですので、結果をまとめるのが得意な学生とコーディングが得意な学生をペアにして、1つのことをしてもら

うようにしています。

しかし、論文を書いて研究を発表することは全員にやってもらいたいです。コーディングが得意な学生は、やったことの詳細を書くことはできるんですが、それを読みやすくなるように抽象化する部分が苦手で論文書きに苦労します。博士課程まで進むと論文が重要なので大変です。

-- ソフトウェアを作っていると、自分の世界にとじこもりがちになると思います。他の人に文章を書いて分かってもらえるようにする重要性を理解してもらうのは難しいですね。

理解しているのかもしれませんが、文章を磨くとかモデリングを一生懸命やるとか、そういう部分に時間割いてもらいにくいです。やっぱり動くコードを書く方が面白いですからね。でも、そのバランスを取るのには教員側の仕事だと思っています。

-- 大学生は、大学時代に何を一番勉強すべきだと思いますか？

英語だと思います。学生の頃にもっと英語を勉強して欲しい。国際会議に行っても、やはり日本人が一番英語が下手で、もったいないですね。韓国や中国の大学院生は皆、自分の研究を英語で紹介してくれます。日本の学生もレベルは高いと思うのですが、英語でものすごく損していると感じます。あと、国際的な場で積極的に行動するという習慣があまりないですね。これももったいないです。



-- 国際交流できる場所に連れて行くなど、英語に触れるように工夫はされていますか？

留学生を研究室に入れるようにしています。でも、あまり効果はないですね。留学生は、すぐ日本語がうまくなるし（笑）。できるだけ国際会議には連れて行くようにしていますが、連れて行くと隅の方で日本人同士で話したりしています。難しいですね。

-- 先生の研究室はツールを作られています、役立つツールを作った後に、メンテナンスするようなベンチャーを立ち上げるといった話はないのでしょうか？

ないですよ。学生に「やってくれへんか？ コードクローンなんて、分析サービスとツールとを一緒にしたら商売になるで。」と言っているのですが、なかなか話に乗ってくれません。大学を卒業した後の選択肢には、“大学に残る”、“企業に就職する”はあるようですが、“ベンチャーを興す”という選択はないようですね。若い時はもう少し冒険をしてもよいと思うのですが。皆非常にオーソドックスですね。

-- ベンチャー意欲が少ないのは、日本の IT 業界が抱えている問題だと思います。

そうですね。私はもう少し怪しい会社があってもいいと思いますが(笑)。日本でも技術 1 本だけ持っていて、そのコンサルだけで生きているような会社があってもいいと思います。

-- そういう会社ができほしいなという気がしますね。大学のシステムの中だと、ポスドク(*2)の人たちが、どこかから一定の予算を貰い、給料を貰いという体制になれば、改善するのでしょうか？

日本の社会では、何年かすると落とし所を求められるようになって、「就職するの？」とか「どこかの大学に行くの？」「次どうするの？」という心配や議論になってしまうんですね。そういった社会の中で、一定の予算をもらったプロジェクトがあったとして、そこに積極的に引っ張るのもどうかなーと思いますね。「いい就職口があるなら、そっちを選んだら？」と言いたくなるときもありますし。難しい問題です。

*2 ポスドクター。博士号をとった後の任期付きの職のこと。

コンピュータを使って、力技でソフトウェアを作ってみたい

-- では最後に、研究の今後の方向性について教えてくださいいただけますか？

大きくは 2 つあります。1 つは、EASE ですね。収集したデータに基づいて、実証的にソフトウェア工学をサイエンスにするというものです。いい加減な評価ではなく、ちゃんとデータに基づいてやるとことを今後もやっていこうと思っています。



もう 1 つは、計算機をぶん回して “力技でソフトウェアを作る” というようなことができないかと考えています。ソフトウェア開発はまだ人手の積み上げでしかできていないですよ。

一方で Google など検索の世界は、文章の対応を完全に記憶して、検索語に対応した文章を出すという、データベースと CPU パワーの力技でやっていますよね。それと同じように、力技でソフトウェアを作る、発見するというようなことをできないかなと思っています。僕はそれをメガソフトエンジニアリングと呼んでいます。

-- それは MDA(*3) というレベルを越えてもっと未来的なものですか？

そうですね。でも、何十年も先の未来ではないとは思っています。SPARS も、「手で 1 個 1 個作るよりは探して来た方が早い」という考えの部分が、それに少し関連しています。

-- 何年か前に雑誌で読んだ話ですが、研究分野が 2 つあって、同じような分野を研究しているけど

用語などは異なる。その 2 つの分野の研究論文の文章を、スーパーコンピュータで解析し、用語などをマッピングして、概念の類似性を求める研究をされているそうです。

面白いですね。“ソフトウェア”の近似性を定義するのは面白いかもしれないですね。今は“ソースコード”の類似度を計算させているんですけども。

-- 今日は貴重なお話をありがとうございました。

*3 モデル駆動型アーキテクチャ。モデル主導のシステム開発のこと。

井上研究室の学生紹介

井上研究室で研究に取り組んでいらっしゃる、博士課程 3 年目の肥後さん、博士課程 1 年目の市井さん、修士課程 2 年目の森岡さんの 3 名にお話を伺いました。

ものづくりベースの研究



肥後さん

僕は修士課程の頃から約 5 年間、[コードクローン](#)に関する研究を行っています。コードクローンとは、ソースコード中の類似している部分のことです。それは一般的にソフトウェアの保守・開発を妨げるものですので、改善したいと考えています。現在は、CCFinder というソースコード中からコードクローンを高速に検出するツールをベースに、いかにソフトウェアの保守・開発に支援を行えるかということを念頭に研究を行っています。[\[1\]](#)[\[2\]](#)[\[3\]](#)

また、研究の方針として、なるべく企業の方の意見を伺って進めています。実際の開発現場に使える技術として確立することを目指していますので、研究室内で留めるのではなく、開発現場の意見を聞いて、そこで役に立つように作っていきたいと考えています。

-- 肥後さんご自身でもコードクローンツールを使われているんですか？

はい。僕も使っています。

-- 使ってみてどうですか？

もう欠かせないものになっていますね。例えば、CCFinder の GUI である [Gemeni](#) を使用すれば、ファイル間・ディレクトリ間で、何パーセントクローンが存在しているか、このクローンが他に何個存在しているか、などの情報を簡単に得ることができます。

ただ、まだ完璧なわけではなく、ユーザが興味がないクローンも検出してしまいうんです。例えば、コピー&ペーストで作成したクローンは検出してほしいですが、それ以外の言語依存でどうしてもクローンになってしまう部分、例えば、ファイルを開いたときのエラー処理など、誰が書いても定型処理になってしまう部分はできれば検出したくない。これは、他のコードクローン検出ツールでも共通の課題です。

-- Gemini/CCFinder はどのくらいのユーザさんが使われているんですか？

正確な人数は把握していませんが、ソフトは国内外合わせて100組織以上に配布されています。[メーリングリスト](#)には、270人ほど登録されています。



市井さん

僕は、[ソフトウェア部品検索](#)に関する研究を行っています。ソフトウェアは、部品 --- 例えば、Java ならばクラスを組み上げて作りますね。しかし、ソフトウェアを作るときに、いちいち部品を作っていくとコストがかかってしまいますので、再利用するものはないかと考えるのですが、探すのはなかなか難しい。そこで、その問題を解決するために検索システムの研究をしています。単純な全文検索システムでは、合致したものが検索されるだけですが、[SPARS](#) というシステムには、再利用度や重要性を

考慮したアルゴリズムを実装しています。[\[4\]](#)[\[5\]](#)[\[6\]](#)

僕はこのシステムがある程度完成した段階から参加していて、これまで実際に企業などで利用されるためのカスタマイズ作業を行っていました。例えば、ある企業から自社の部品管理に使いたいという要望を受けて、パッケージ単位での行数などの情報の表示や、同じ名前のファイルがあれば、その情報も表示するなどの細かいカスタマイズを行いました。

-- 企業での評価はどうでしたか？

好評でした。1つの部品に対して、どれくらい使われているかが分かることと、不具合があったときに、不具合を修正した場合の影響範囲が分かるところがよいそうです。

-- そのソフトウェアはオープンソースで公開されているのですか？

一部のアルゴリズムが特許の都合公開できないこともあり、一般には公開していません。セミナーなどで配布する際には、問題の無い部分のみオープンソースとしています。ソフトウェア自体は、[SPARS-J](#) としてインターネット上のサービスとして公開されています。

森岡さん

僕は、[ファンクションポイント \(*4\)](#) の値を自動的に計測しようという目的で開発を行っています[\[7\]](#)。研究の動機は2つあります。1つは、現在のファンクションポイントの値は、設計仕様書からしか求められないですよ。ですので、設計仕様書がない場合、プログラムソースコード自体から算出できればよいなと思ったこと。もう1つは、現在のファンクションポイントは、人でしか測れないため、どうしても人の主観が入ってしまい、値によってぶれが生じてしまう問題があります。この主観部分をできるだけ排除して、誰が測っても同じ値が出るようにしたいと思ったことです。



現在は、ファンクションポイントの算出にはまだ道のりがあり、まだ研究の初期段階、10% ~ 20% くらいの進捗だと考えています。現在は、ファンクションポイントの抽出のために必要なトランザクションファンクションという、ソフトウェアの機能を抽出する部分を、ソフトウェアを実際に回した時の実行履歴から取ってくる部分を開発しています。ファンクションポイントの算出には、データファンクションの値も必要なんですけど、その値も、現在は、人間が外部から与えるという形で行っています。

-- いつごろ完成する予定ですか？

僕が学生でいる間には無理だと思っています(笑)。おそらく、後の者に引き継ぐ形になると思います。僕は、トランザクションファンクションを取ってきて、できれば、データファンクションもソフトウェアから取ってきて、仮の値でもよいから、ファンクションポイントが出すところまで作りたいですね。できれば、自動的に出した値と、専門家の出した値が近ければよいなと思っています。

**4 ソフトウェアの開発規模を見積もるための手法の1つ。*

ソフトウェア開発に携わる人が楽になるために

-- ご自身の研究のメリット、使う人にとってどのように役立つかを教えていただけますか？

肥後さん

クローンを提示して、その部分を修正してもらうことで、将来のコスト削減に向けて改善できることですね。ただ、何度か商用のソースコードのクローン分析を行う機会があったのですが、実際にクローン見つけて、「このクローンと、このクローンは1つにした方がいいですよ。」と提示をしたことがあるんですが、実際は「いやそのコードは動いているからいいですよ。」と言われることが多いですね。今後は、クローンを提示するだけでなく、改善のメリットやリファクタリングの指針も提示することができれば、もっと役立つのではと考えています。



市井さん

SPARS では、オープンソースのソフトウェアを集めてきて、データベース化し、検索できるようにしています。ですので、ユーザは、ほしい部品をそこで検索し、使いたいものを使えるので、ネット上を渡り歩く手間が省けます。

-- オープンソースのライブラリを使って、さらにオープンソースを作る、再帰的な関係ができていますので、そこでどの程度使われているかを見ることができるのは結構面白いですね。現状、いくつくらいのオープンソースが登録されているのですか？

18万クラスあります。ただ、どうしてもオープンソースのプロジェクトには、レポジトリからコードを機械的に集める手段を提供してくれないものもあって、いろいろ工夫をしています。結局、現在は手作業で追加しています。

-- 世の中に存在をもうちょっとアピールできると、変わってくるかもしれないですね。

森岡さん

ファンクションポイントは工数の見積もりとかでよく利用されます。工数の見積もりには、過去に開発されたデータが必要になります。しかし、開発が過去になればなるほど、設計仕様書が残っていないですね。ですので、設計仕様書なしで、ソフトウェア自体から計測できるのが一番嬉しいんじゃないかなと思います。

-- 例えば、ソフトウェアのソースコードを考えると、ビジネス系のアプリケーション、エンジニアリング系のアプリケーション、ソフトウェア製品などで、全然状況が異なると思いますが、ターゲットはどこですか？

現在はビジネス系のアプリケーションを考えています。実際も、共同研究先のアプリケーションを貸していただいて、使わせていただいています。

英語と論文は大変ですね

— 大学で研究されていると、英語と論文書きは克服すべき課題だと思いますが、どうやって克服されていますか？

肥後さん

英語に関しては、読む方はどうにかなるんですが、話す方はものすごく苦手ですね。国際会議で発表する時の原稿をどうしよう。外人との話はどうしよう。と考えてしまいます。最初の頃は、発表を文章に書いて、丸暗記していたんですよ。今は、丸暗記はしてませんが。



論文も、構成も悩みですが、論文を出すと、通るが、通らまいが、「お前の英語はおかしい。」という回答が返ってくるんですね。悩んではいるのですが、何もしていないのは問題ですね。ただ、研究室にいる留学生が「論文の英語は硬い英語だから、ある程度覚えてしまえば大丈夫だよ」と言っているの、じゃあ頑張ろうかなと思っています（笑）。

市井さん

英語に関しては、論文を読むときに引っかかることはありますが、他のことで何か努力をしていることはありません。読んでいけばちよとずつ読めるようになっていくなあとと思っています。ただ、日本語の論文を書くときに、構成や言い回しで困りますね。技術論文の書き方など参考にしながら頑張っています。

森岡さん

僕は修士なので、海外での発表もなく、英語に触れる機会は、論文を読むときくらいです。それほど難点はありません。ただ、自分で書くとなると、つまずくと思います。論文も、卒論しかないのですが、そのときも構成が難しかったことを覚えています。初めてだったので、何をどう書いていいかわかりませんでした。大幅に修正をくらって、大変でした。

ものづくりと研究

— ものづくりと研究を両立するのはむずかしいですよね。どちらが好きですか？また、両立させるために何か工夫はしていますか？

肥後さん

そうですね。僕はどちらかといえば、プログラムを書く方が好きだし、全然苦じゃないですね。論文に関しては、先生が「はよ書け、はよ書け」とハツパをかけてくるんですね（笑）。で、「すいません、すいません、書きます」と。ただ、学生でプログラムを書いているだけなんで、実際に企業に入って、実際にお客さんに使ってもらえるプログラムを書いたりすると、また違った喜びがあるのかもしれないですね。その辺は未知数です。

市井さん

僕も、コードを書いている方が楽しいですね。修論の研究では、プログラムを書くよりも、データを取って研究をしている方が多かったので、個人的にはとても苦痛でした。

森岡さん

僕もものづくりの方が好きなタイプですね。論文に関しては、研究の業績も博士の二人よりは要求されていないですし、業績も、共同研究先との研究の中間結果を報告しているので、特に問題は起こっていません。やはり、基礎研究は大事だと思いますが、僕は、相手のお客さんに使ってもらうことを考えてしまいます。それが、博士課程に進まず、就職する理由だとも思います。

将来について

-- 最後の質問ですが、今後目指す姿や、成し遂げたいことを教えてください。



肥後さん

来年で博士を修了するのですが、来年以降のことは決まっていません。今、この研究室でやっていることがとても楽しいので、どこかで続けたいと思っています。

市井さん

僕は、開発現場に近いところに行きたいなと思っています。どちらかといえば、新しい技術を見つけてきては、つまみ食いできるような、新しいことができるところにいきたいです。研究というよりも、実際に使われるものを作っていきたいですね。

-- 新しいもの好きなんですか？

そうですね。でも今は新しいものに手を出せていないんです。

森岡さん

僕は、来年から SE として働くことになると思うのですが、そこで、お客様の意見を聞いて、それに見合ったシステムを 100 %に近い形で提供できるように、まずはそこからがんばっていきたいと思います。ゆくゆくは、研究も視野に入れていきますし、畑違いのところ、例えば、経営などでも、社会を見るという目で、いろいろ見て行きたいと思っています。夢は去年まではレクサスに乗ることだったんですが（笑）。今はお客さんあつての会社ということが分かっているので、そこをないがしろにしない、お客さんのニーズにこたえられるようになりたいですね。

-- プロとして生きていくんですね。

森岡さん

はい。

-- 本日は貴重なお話ありがとうございました。

皆さん

ありがとうございました。

デモ

インタビュー中、研究室で開発しているシステムをデモンストレーションしていただきました。

SPARS-J



<http://demo.spars.info/> がトップページです。

検索ボックスにキーワードを入力して「Search」ボタンを押すことで検索を行うことができます。今回は、XMLを検索します。

キーワードでの検索結果です。ソースコード中にキーワードを含むクラスの一覧が表示されます。この一覧は、キーワードへの適合度が高く、かつ、よく利用されるクラスが上位になるようにソートされています。また、類似性の高いクラスはグループ化されて表示されます。この場合だと、6位: `org.xml.sax.XMLReader` などがそうですね。

クラス名をクリックすると個別のクラス表示画面に移ります。

検索結果の1位、[org.w3c.dom.Document](#)の表示画面です。左上のフレームにはパッケージ階層、左下フレームにはフィールド及びメソッドの一覧が表示されます。右フレームにはクラスのソースコード、上部には、クラスに関する様々な情報のリンクが並んでいます。

-- よく検索されているのはありますか？

画像周りの検索がよく検索されていますね。この検索システム自体がまだマイナーなので、ユーザが何をほしがっているかがまだ分かりませんね。

-- 一番使われるコード例など、教育的な面があると、もっと使われるようになるかもしれませんね。

よく使われているものが出てくるから、それがよく使われるコード例になっているのかもしれないですね。今はまだ、どこを見て、どこを使えばよいという部分は示せてないので、その部分を今後研究していきたいですね。

インタビューを終えて

「ものづくりを大事にされている研究室だな」とインタビューを終えて強く感じました。ただ作るだけでなく、研究室から出て一般に使うことのできるものを目指しておられる研究室でした。実際に多数のソフトウェアが企業等で使用されているとのこと。開発したソフトウェアが研究室レベルを越えるためには、使い勝手の向上や安定性を高めることなど、多くの作業が必要になることと思います。インタビューさせて頂いた学生の皆さんは、「コードを書くのが好きだから...」とさらりと言っておられましたが、そう簡単なことではないと推測致します。

「ものをつくってなんぼ」の井上先生のご精神は、学生の皆さんにもしっかりと根付いているようで、「ものを作って、使ってみて、初めて始まる」というメッセージを、学生の皆さんからもインタビューを通じて受けました。斬新なアイデアや素晴らしい理論であっても、使える形になっていないと現実社会では存在しないことと同じ。また、作ってみることで、さらに新しいアイデアも生まれる。そういうことだと感じました。

さて、多少泥臭くてもいいから CPU のパワーを存分に使ってコンピュータに何かをさせてみたいという考えは、個人的にはとても好きです。理屈が素晴らしいが実装するときに難のある理論よりは、力技だとしても動くものの方が見ていてワクワクしますし、先にも言いましたが、逆にそこから新しい理論もうまれる気がします。「力技でソフトウェアを作る」---とても好奇心をくすぐられました。

研究室情報

研究室名	大阪大学大学院情報科学研究科 井上研究室
教授	井上 克郎
主な研究内容	<ul style="list-style-type: none">• プログラム部品の検索と再利用• コードクローン• プログラムスライシング• ソフトウェアの自動分類• アスペクト指向• ソースコードの構造を考慮した版管理システム• 実行履歴の可視化• ソフトウェアの開発履歴の理解促進• 開発履歴情報を利用したソフトウェアアーキテクチャ理解支援• メタモデルを用いた成果物の一貫性管理• ソフトウェア理解支援を目的としたブラウザ
Website	http://sel.ist.osaka-u.ac.jp/

リンク・参考資料

- [大阪大学大学院 情報科学研究科](#)
 - [ICAA](#)
 - [SPARS](#)
 - [EASE](#)
- [1] 肥後芳樹, 神谷年洋, 楠本真二, 井上克郎, “コードクローンを対象としたリファクタリング支援環境”, 電子情報通信学会, Vol.J88-D-I, No.2, pp.186-195, 2005年2月.
 - [2] 肥後芳樹, 楠本真二, 井上克郎, “コードクローン分析ツールGeminiを用いたコードクローン分析手法”, 電子情報通信学会技術研究報告, SS2005-29, Vol.105, No.228, pp.37-42, 2005年8月.
 - [3] 佐々木亨, 肥後芳樹, 神谷年洋, 楠本真二, 井上克郎, “プログラム変更支援を目的としたコードクローン情報付加ツールの実装と評価”, 電子情報通信学会論文誌 D-I, Vol.J87-D-I, No.9, pp.868-870, 2004年9月
 - [4] 市井 誠, 松下 誠, 井上 克郎, “ソフトウェア部品グラフの次数分布におけるべき乗則の調査”, ソフトウェア信頼性研究会 第3回ワークショップ論文集, pp.87-96, 2006年7月
 - [5] 市井 誠, 松下 誠, 井上 克郎, “ソフトウェア部品の利用関係におけるスケールフリー性の調査”, 電子情報通信学会技術研究報告, SS2005-69, Vol.105, No.491, pp.37-42, 2005年12月
 - [6] 市井 誠, 山本 哲男, 横森 励士, 井上 克郎, “ソフトウェア部品推薦のための協調フィルタリング手法の提案と実現”, 電子情報通信学会技術研究報告, SS2004-15, Vol.104, No.243, pp.7-12, 2004年8月
 - [7] 森岡 佑, 谷口 考治, 楠本 真二, 井上 克郎, 英 繁雄, 前田 憲一, 津田 道夫, “プログラム実行時情報を用いたトランザクションファンクション抽出手法”, 情報処理学会第68回全国大会講演論文集(1), pp.303-304, 2006年3月



藤井 拓(ふじい たく):オーグス総研 技術部ソフトウェア工学センター 兼 大阪大学大学院 工学研究科 電気電子情報工学科専攻(招へい教授)。1990 年頃に C++ と OMT を勉強して以来、オブジェクト指向三昧な日々を送ってきたおじさん。最近は、アジャイルモデリングと反復的な開発アプローチをメインに研究、執筆、翻訳、開発プロジェクトの支援、大学での講義などに従事。大阪大学で SOA 関係の研究も行っている。

(編集:角内 里江, 水野 正隆)

記事の内容を 5 点満点で評価してください。
○ 1 点 ○ 2 点 ● 3 点 ○ 4 点 ○ 5 点
記事に関するコメントがあれば併せてご記入ください。

オブジェクトの広場では、「らぼなび！」で紹介させて頂ける研究室を募集しています。「是非、我々の研究室紹介をして欲しい！」という研究室は、オーグス総研オブジェクトの広場編集部 (oosquare-editor@ogis-ri.co.jp) までご連絡下さい。