

プログラム理解の困難さの脳血流による計測の試み

中川 尊雄 亀井 靖高 上野 秀剛 門田 暁人 松本 健一

本論文では、脳血流を計測する近赤外分光法 (Near InfraRed Spectroscopy; NIRS) を用いて、開発者がプログラム理解時に困難を感じている状態にあるかどうかを定量的に観測することを試みた。10名の被験者に対して、難易度の異なる二種類のプログラムの理解時の脳血流を計測する実験を行い、10名中8名において難度の高いプログラムの理解時に脳活動がより活発化するという結果が得られた。また、被験者ごとに脳活動を Z-score で正規化し、難度ごとに集計した上で t 検定を実施した結果、脳活動の平均値に有意な差 ($p < 0.001$) が見られた。

In this research, we aim to quantify the difficulty of program comprehension during source code reading. We use Near Infra-Red Spectroscopy(NIRS) to measure the activation of brain. As a result of an experiment with 10 subjects, 8 of them showed a strong activation in the brain during reading of strongly obfuscated programs that are extremely difficult to comprehend. We also normalized the data for each participant and aggregated them for statistical testing. As a result of t-test, significant difference ($p < 0.001$) was seen in the mean of the brain blood flow between obfuscated and non-obfuscated programs.

1 はじめに

プログラム理解は、コーディング、テスト、保守など、ソフトウェア開発における幅広い工程で必要とされる重要な活動であり、成果物の品質に強く影響する。例えば、プログラムを理解できている人とそうでない人の間で、バグ発見効率に十倍以上の差があることが示されている [2]。ただし、プログラム理解は、情

報の記憶や想起、認知、注意などの内的な認知プロセスを含む知的活動であり、その計測は容易ではない。

一方、神経科学や認知科学の分野においては、脳波や脳血流といった脳周辺情報から知的活動を計測するアプローチが広く用いられている [1]。これらの研究では、知的活動の種類や量と、脳の特定部位の活動を結び付ける試みがなされている。プログラム理解の分野においても、Siegmond ら [7] が、こうした神経科学分野の知見を応用し、プログラム理解時の脳活動を計測する実験の必要性を主張している。

本論文では、脳活動計測によってプログラム理解行動の定量化を目指す第一歩として、プログラム理解時の脳血流を NIRS(Near-infrared spectroscopy ; 近赤外分光法) で計測し、以下の仮説を検証することを目的とする。

仮説：プログラム理解に困難が生じている状態を、理解時の脳血流によって計測できる。

On Measuring the Difficulty of Program Comprehension based on Cerebral Blood Flow.

Takao Nakagawa, Akito Monden, Ken-ichi Matsumoto, 奈良先端科学技術大学院大学情報科学研究科, Graduate School of Information Science, Nara Institute of Science and Technology.

Yasutaka Kamei, 九州大学大学院システム情報科学研究所, Graduate School and Faculty of Information Science and Electrical Engineering, Kyushu University.

Hidetake Uwano, 奈良工業高等専門学校情報工学科, Department of Information Science, Nara National College of Technology.

コンピュータソフトウェア, Vol.31, No.3 (2014), pp.270-276.

[研究論文 (レター)] 2014年2月10日受付。

本論文は第20回ソフトウェア工学の基礎ワークショップ (FOSE2013) の発表論文を発展させたものである

開発者にとってプログラムの理解難度が高い場合には、理解に必要な知的活動の量が増大し、記憶の操作

に代表される高次の認知機能を担うとされる脳前部(前頭前野)[9]が活発化すると考えられる。このことから、本論文では、前頭前野の脳血流を計測することで、仮説の検証を試みる。

本仮説が支持された場合、プログラム理解に困難さを感じている開発者を判別することが可能となり、将来的には、熟練者やマネージャによる適切なサポートにつなげることが期待される。

2 関連研究

プログラム理解における人的要因(理解度、開発者の行動、理解戦略など)を分析するために、従来研究では一般的にアンケートやインタビュー、実験中に思考内容を発話してもらうシンクアラウド法といった方法が用いられてきた。しかし、これらのアプローチは結果が被験者の主観に依存する、課題と直接関係のない発話や記憶のための負荷がかかる、アンケートは実験終了後の記憶であるため時系列分析が困難である、といった問題を有する。

一方、神経科学の分野では、知的活動を脳がどのように実行しているかを、脳活動に関係する生体情報(脳周辺情報)から分析することが一般的である[1]。脳周辺情報の計測手法・機器には、脳の血流動態を測定するfMRI(functional Magnetic Resonance Imaging; 核磁気共鳴法)、脳表面の電位(脳波)を測定するEEG(Electroencephalogram; 脳電図)、脳血流の酸化傾向を測定するNIRSなどがある。

これまで、課題の難度や種類による脳活動の違いを論じるため、NIRSを利用した研究が複数報告されている。例えば黒田は、繰り返し下がり暗算を行う被験者のNIRS信号値の変化傾向が、繰り返し下がりありの暗算の得意・不得意によって、明確に異なることを報告した[10]。このことは、被験者のプログラム理解に困難が生じている状態をNIRSによって同様に測定できる可能性を示唆する。ただし、著者の知る限りプログラム理解のような長時間の課題を採用した例はなく、手指を使った物体の操作[11]のような1分程度で終了する課題が対象とされている。

Siegmundら[7]はプログラム理解と脳活動の関係を探ることの重要性について言及し、fMRIを用いて

プログラム理解時の脳血流を計測する実験デザインについて報告している。Siegmundらは実験にfMRIを用いることを想定しており、被験者は仰向けに装置に固定された状態で計測されることから、大規模なプログラムはタスクとして採用できないと述べている。また、高橋[8]は簡易脳波計測機を用い、実環境に近い状態でのプログラミング活動のストレス測定を試みた。実験の結果、強いストレスを感じた作業ほど、計測機が出力するMeditationと呼ばれる値が低く、有意な差が観測されている。ただし、この値の計算方法は明らかにされておらず、脳波計測機が原理的にノイズを多く含むことが課題となっている。

また、大木ら[5]は、ソフトウェアのモデリング時の脳血流計測によってモデリングに関わる認知プロセスの種類や出現パターンを分析した。実験の結果、「DFDモデリング」と「パズルの解法探索」などの課題間に強い相関があることが認められ、作業に共通する認知プロセス(仮説生成など)が働いている可能性について論じている。ただし、文中で分析の対象となっている被験者は1名のみであり、分析の継続を今後の課題としている。

これらの報告は、プログラム理解やその関連作業の分析に脳活動計測を用いることが注目されつつあることを示唆している。一方で、耐ノイズ性の高い装置を用いた、実環境に近く、多人数を対象とした実験結果や分析については報告されていない。本論文では、fMRIやMEGに比べて被験者の姿勢・体動に制約が少なく、脳波計測器(EEG)に比べてノイズが少ないとされるNIRSを用いて脳血流を計測し、プログラム理解に困難が生じる状態の定量的測定を試みる。

3 脳血流計測による理解困難状態の測定

3.1 実験のデザイン

実験では、仮説の検証のために、プログラム理解に困難が生じている状態と、そうでない状態を誘発する必要がある。そのために、著しく理解が容易なプログラムと、著しく理解が困難なプログラムの2種類を用意し、それぞれの理解時の脳血流を計測・比較する。理解が困難なプログラムの作成には、プログラム難読化手法を用いる。難読化プロセスには、門田ら

[3] の提案するループ難読化手法を用いる。ループ難読化を施したプログラムは、制御構造が複雑化するとともに変数の種類とその更新パターンが複雑化し、難読化前のプログラムと比べて、理解に平均 18 倍の時間を要することが報告されている [3]。

また、実験では、被験者がとるプログラム理解戦略の違いの影響を排除するため、被験者のプログラム理解戦略を“メンタルシミュレーション” [6] に限定する。メンタルシミュレーションは、プログラム理解手法の 1 つであり、プログラムの実行過程を頭の中でシミュレートすることを指す [6]。コードを読み、頭の中に処理内容の概形を作り上げることから、ボトムアップな理解手法の代表として知られている。メンタルシミュレーションの際には、変数の名前と格納値の対応や変化の順序を記憶し、制御構造に従って何度も記憶内容を更新する必要があるため、特に記憶に関する脳部位が活動することが予測される。

3.2 NIRS による脳血流・脳活動計測の原理

脳血流計測は、脳が活動した際の神経活動に応じて、エネルギーを供給するために脳血流量とその酸化度合が上昇する現象を利用して、脳活動量やその部位を特定する手法である。本研究で用いる NIRS は、血中の酸化ヘモグロビン (oxy-Hb) と脱酸化ヘモグロビン (deoxy-Hb) の吸光特性が異なることを利用し、頭皮表面に照射した近赤外光の反射成分を検知することで脳表層における活動を計測する装置である。fMRI や MEG などの他装置と比較して、計測に必要な準備が少なく、取り付けが容易で、また fMRI や MEG に比べて体動の自由があるなど実験条件の制限も少ないほか、時間分解能が高いという利点がある。こうした特性は、長時間の計測や、現場への適用などを考慮すると、プログラム理解時の脳活動計測に適しているといえる。

4 実験

被験者が難度の異なるプログラムを読む際の脳活動を NIRS で計測した。被験者は奈良先端科学技術大学院大学の学生 10 名で、全員が 22 歳から 26 歳の男性、プログラミング経験が 3 年以上であった。

```
A[0][0][0] = 97, A[0][0][1] = 48, A[0][1][0] = 52
A[0][1][1] = 71, A[1][0][0] = 17, A[1][0][1] = 64
A[1][1][0] = 11, A[1][1][1] = 32, A[2][0][0] = 20
A[2][0][1] = 22, A[2][1][0] = 48, A[2][1][1] = 86
```

```
N = 3, M = 2, L = 2
```

```
int func(int ***A, int N, int M, int L){
  int i,j,k;
  int p;
  p = A[0][0][0]; (1)
  for(i = 0; i < M; i++){
    for(j = 0; j < M; j++){
      for(k = 0; k < L; k++){
        if(A[i][j][k] < p) p = A[i][j][k]; (2)
      }
    }
  }
  return p; (3)
}
```

図 1 実験で用いたソースコードと提示した引数の例

表 1 回答用紙の記入例

N	M	L	i	j	k	p	位置
3	2	2				97	(1)
			0	0	1	47	(2)
			1	1	0	11	(2)
			1	0	0	17	(2)
				1	0	11	(2)
			2	1	1		(3)
			3	2	2		(3)

4.1 タスクと実験手順

実験タスクは、紙に印刷された 1 つの関数と関数呼び出し時の引数の値を元に、被験者が動作をメンタルシミュレーションし、その過程を回答するものである。タスクでは、17 行から 32 行の C 言語で記されたソースコードを用いた。図 1 に、ソースコードと引数の例を示す。

コードには各行の右側に括弧書きの数字で目印を記載している。被験者はメンタルシミュレーションが目印の行に到達するたびに、各変数の値と目印の番号を解答用紙に記入する。この時、変数の値が前回と同じか、値が不定であれば無記入とする。表 1 に解答用紙の記入例を示す。実験者は随時回答を確認し、被験者に正誤を口頭で通知する。値がすべて正しければ「正答」とし、シミュレーションを続けてもらう。値が 1 つでも誤っていれば「誤答」とし、被験者にシミュレーションをやり直し、再度解答用紙に記入してもらう。目印はループ中に含まれることもあるため、同じ目印に対して複数回の回答をすることもある。タ

スクには制限時間を設けず、メンタルシミュレーション中のすべての回答に正答した時点で終了とした。

タスクに用いるソースコードは理解が容易な3つ (easy タスク) と、easy タスクを難読化し、変数の数や処理構造を複雑にした3つ (hard タスク) の6つである。また、練習用タスクとして同様に、p-easy タスクと、それを難読化した p-hard の2つを用意した。本タスクのソースコードは配列内の最小値の判定、配列内の数値の合計計算、配列内に含まれる特定要素の数え上げの3個を用いた。また、練習タスクのプログラムは配列内の最大値の判定を用いた。

それぞれの被験者は、p-easy, p-hard を実施後、easy タスクと hard タスクを1つずつの計4タスクを実施する。学習効果および順序効果による影響を防ぐため、easy タスクと hard タスクで処理内容が同じ (難読化前と難読化後の) ソースコードを提示しないよう配慮し、被験者の半分は easy タスクを先に、残りの半分は hard タスクを先に実施する。以下に実験手順を示す。

- 手順1 実験内容の説明、装置の取り付け
- 手順2 p-easy タスク, p-hard タスクの実施
- 手順3 休憩 (2分)
- 手順4 本タスク1の実施 (easy タスクまたは hard タスク)
- 手順5 休憩 (2分)
- 手順6 本タスク2の実施 (本タスク1と異なる難易度・異なる目的)
- 手順7 装置の取り外し・アンケート

4.2 実験環境

実験には、日立製ウェアラブル光トポグラフィWOT-220を用いた。本装置は、前頭葉の脳血流量を測ることが可能で、また、他のNIRS装置と比較して軽く、装着が容易であるという特徴を持っている。図2に、装置と、装着時の外観を示す。

実験は被験者1名と実験者1名のみが居る静かな部屋において、椅子に座った状態で行われた。姿勢の変化によるノイズを防ぐため、ソースコードを印刷した紙と回答用紙を書見台に置き、楽な姿勢で読めるよう椅子の位置や高さを調整してもらった。また、



図2 装置と装着時の外観

頭部の動きによる血流変化を防ぐため、事前に下を向きすぎないように被験者に伝え、実験中も計測に影響があると考えられた場合は、口頭で頭を上げるように指示した。

4.3 分析方法

計測機器であるNIRSは、脳血流の酸化傾向を測ることで脳活動を測定する装置である。すなわち、血中の酸化ヘモグロビンを表すoxy-Hbが多いほど、その部位での脳活動が活発であることを表す。

実験で得られるデータは200msごとのoxy-Hbの値で、値が大きいかほど脳の活動量が多いことを示す。なお、計測値は計測開始時のoxy-Hbを0[mM*mm]としたときの各時点における相対的な変化量であるため、異なる被験者同士の比較、異なる時期に計測された値を比較することは適切ではない。

そこで本稿では、oxy-Hbの変化量を式(1)に示すように各被験者の平均と分散で正規化(z-score化)し、各被験者のeasyタスクとhardタスクの脳活動を合算した場合と、個別に見た場合の両方で比較する。

$$Norm.oxyHb(s) = \frac{oxyHb - avr(s)}{sd(s)} \quad (1)$$

$Norm.oxyHb(s)$ は、ある被験者 s の正規化oxy-Hbを、 $avr(s)$ は被験者 s のoxy-Hbの平均値を、 $sd(s)$ は被験者 s のoxy-Hbの標準偏差をあらわす。 $Norm.oxyHb$ の値は、被験者ごとに平均0、分散1の形に正規化されたものとなる。

5 結果

5.1 実験結果の概略

すべての被験者について、easyタスクよりhardタスクのほうがプログラムの理解に時間を要し、誤答数も多い傾向が見られた。手順7のアンケートの結果

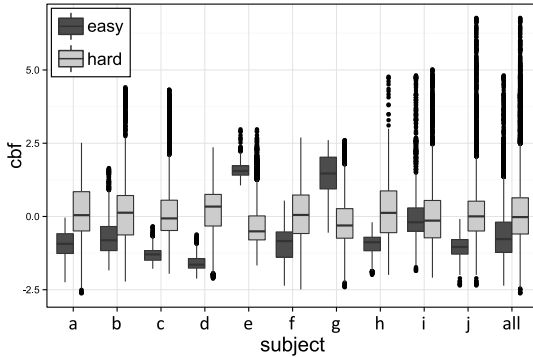


図3 被験者ごとの正規化 oxy-Hb の難易度別分布

からは、すべての被験者が hard タスクの読解のほうが難しく感じたと回答している。

5.2 難易度による脳活動の比較

図3は、被験者ごとに難易度別の正規化 oxy-Hb の分布を表した箱ひげ図である。a から j が個々の被験者を表し、各々の被験者について左が easy タスク、右が hard タスクの分布を表している。縦軸は正規化 oxy-Hb を表し、値が高いほど脳が活動していることを表す。

図3の一番右に示された“ALL”のラベルは被験者すべての正規化 oxy-Hb の分布を集計し、難易度別にあらわしたものである。全被験者のデータからは、hard タスク中の正規化 oxy-Hb が easy タスク中の正規化 oxy-Hb よりも全体的に高い値を取っていることがわかる。両者の差が有意であるかどうかを t 検定で調べた結果、 $p < 0.001$ で有意であった。

個別に見た場合、被験者 10 人中 8 人の正規化 oxy-Hb、ならびに正規化 oxy-Hb の全体の合算において、難易度が高いほど平均値が高い傾向が見られた。このことは仮説 1 を支持し、プログラムの被験者にとっての難度が、前頭前野の活動と、その部位における oxy-Hb の上昇という形で現れることを示唆している。すなわち、前頭前野における血流計測は、開発者にとってのプログラムの理解難度や、ある時点において理解が困難であるかどうかを定量的に計測するための一指標となりうる。

一方で、10 人中 2 人(被験者 e, g) が仮説と逆の結果を示し、1 人(被験者 i) はほとんど難易度による変

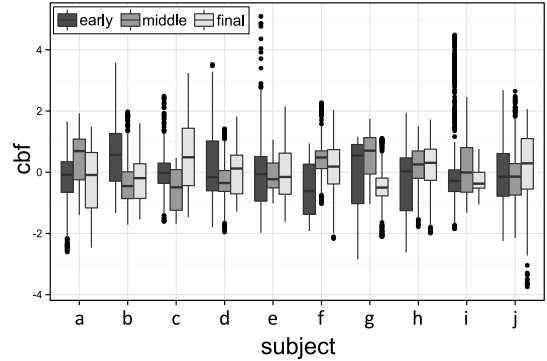


図4 タスク実行中の脳活動量の時系列変化 (easy)

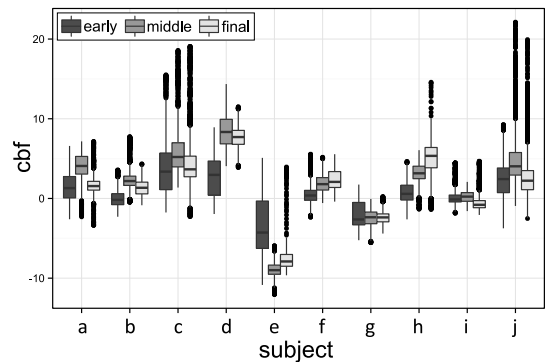


図5 タスク実行中の脳活動量の時系列変化 (hard)

化を見せなかった。過去の NIRS に関する研究では、10 人に 1 人程度の割合で逆傾向の被験者が含まれる可能性が指摘されており [4]、頭皮や頭骨の厚み等の個人差のために oxy-Hb を正しく計測できていない可能性がある。

10 人中 8 人の正規化 oxy-Hb について、hard の中央値が easy の中央値よりも高かった。本結果は変数の数や変数の更新回数が多いプログラムを理解する際に前頭前野の活動が上昇することを示している。

5.3 読解の進捗度による脳活動の比較

難易度別の比較とは別に、プログラム理解時の脳血流計測が、“理解に困難が生じる状態がいつ発生したか”を判別できるか調査するため、タスク実行時における脳活動量の時系列変化を分析した。被験者ごとにタスク時間を序盤、中盤、終盤に三等分した際の脳血流の変化を、図4(easy)、図5(hard)にそれぞれ示す。

図からは、easy タスクと hard タスクに共通する傾向は見られなかったが、hard タスクにおいては、10

人中9人で序盤より中盤の正規化 oxy-Hb が高かった。また、10人中7人で終盤より中盤の正規化 oxy-Hb が高かった。この結果には、1) タスクを開始してすぐの時点から変数を徐々に多く覚えていくこと、2) 終盤ではリハーサル効果や、制御構造の把握によって、記憶を遡って考えなくてはならない部分が減っていること、の二点が影響している可能性がある。

また、実験時における誤答の発生回数が中盤に最も多く、序盤に最も少なかったことから、読解の進行度によって誤答の発生頻度に違いがあることも検証2の結果に影響していると考えられる。また、検証1で逆傾向を示した被験者のうち、被験者 e は検証2でも同様に逆傾向(時間変化に沿って谷型の変化)を示していた。

一方、タスク実行中に正規化 oxy-Hb が最大となった時間は被験者ごとに異なることが確認された。このことは、高い時間分解能で脳活性化のタイミングを定量化できることを示唆しているが、今後、被験者間の時系列変化の違いについてより詳細な分析を行っていくことが課題となる。

6 おわりに

本論文では、脳活動計測によるソフトウェア開発活動の定量化を目指す研究の第一歩として、プログラム理解に困難が生じている状態を定量的に観測することを目的とし、脳血流計測の利用を提案した。提案の妥当性を検証するため、理解が容易なプログラムと、理解に困難が生じるよう難読化されたプログラムの二種類を読解し、その際の脳血流を計測する被験者実験を行った。実験の結果、10人中8人の被験者について、難読化されたプログラムの読解時に、前頭前野の脳血流量が上昇する傾向が確認できた。本実験の結果は、プログラム理解に困難を感じている開発者を定量的・客観的に判別できることを示唆しており、将来的には、熟練者やマネージャによる適切なサポートにつなげることが期待される。

一方で、本提案の信頼性を高め、現実に役立つものとしていくためにはまだ多くの課題がある。まず、他と異なる傾向を示した2名の被験者について、その原因を明らかにしていく必要がある。また、本論文の

実験は、理解手法をボトムアップな理解手法であるメンタルシミュレーションに限定し、小規模なプログラムを対象として行ったが、より実環境に近づけるためには、大規模プログラムや、トップダウンな理解手法についても今後実験していく必要がある。また逆に、何が直接脳活動に作用しているかを明らかにするため、より目的を絞った短いコード片を与える実験を行うことも検討していきたい。

参考文献

- [1] Cabeza, R. and Nyberg, L.: Imaging Cognition II: An Empirical Review of 275 PET and fMRI Studies, *Journal of cognitive neuroscience*, Vol. 12, No. 1(2000), pp. 1-47.
- [2] 栗山進, 大平雅雄, 門田暁人, 松本健一: プログラム理解度がコードレビュー達成度に及ぼす影響の分析, 電子情報通信学会技術研究報告. SS, ソフトウェアサイエンス, Vol. 104, No. 571(2005), pp. 17-22.
- [3] 門田暁人, 高田義広, 鳥居宏次: ループを含むプログラムを難読化する方法の提案, 電子情報通信学会論文誌. D-I, 情報・システム, I-コンピュータ, Vol. 80, No. 7(1997), pp. 644-652.
- [4] 西牧謙吾: 脳科学と障害のある子どもの教育に関する研究, B-215. 課題別研究報告;平成16年度~18年度, 国立特殊教育総合研究所, 2007.
- [5] Ohki, M. and Murase, H.: Brain Physiological Characteristic Analysis for Software Analysis Support Environments, in *Proc. of the 11th Int'l Conf. on Enterprise Information Systems (ICEIS2009)*, Vol. ISAS, 2009, pp. 329-337.
- [6] Pennington, N. and Grabowski, B.: The tasks of programming, *Hoc et al*, Vol. 307(1990), pp. 45-62.
- [7] Siegmund, J., Brechmann, A., Apel, S., Kästner, C., Liebig, J., Leich, T. and Saake, G.: Toward measuring program comprehension with functional magnetic resonance imaging, in *Proc. of the ACM SIGSOFT 20th International Symposium on the Foundations of Software Engineering, FSE '12*, ACM, 2012, pp. 24:1-24:4.
- [8] 高橋圭一: 簡易脳波測定装置を用いたプログラミング活動のストレス測定に関する実験, ソフトウェア工学の基礎 XX (日本ソフトウェア科学会 FOSE2013), 2013, pp. 185-190.
- [9] 山口修平: 前頭葉と記憶, 高次脳機能研究, Vol. 27, No. 3(2007), pp. 222-230.
- [10] 黒田恭史: 計算課題遂行時の脳内ヘモグロビン濃度変化の特徴: 減法課題を用いて, 教育学部論集, Vol. 16(2005), pp. 37-50.
- [11] 内藤幾愛, 大西秀明, 古沢アドリアネ明美: 単純動作と複雑動作時における脳活動の比較: 近赤外分光法(NIRS)による検討, 理学療法学, Vol. 35, No. 2(2008), pp. 50-55.



中川 尊雄

2012年奈良工業高等専門学校専攻科電子情報工学専攻修了。2014年奈良先端科学技術大学院大学博士前期課程修了。現在、同大学大学院博士後期課程に在学中。生ソフトウェア開発者の生体情報計測に関する研究に従事。IEEE 会員。



亀井 靖高

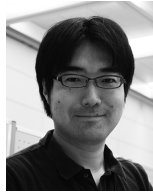
2005年関西大学総合情報学部卒業。2009年奈良先端科学技術大学院大学情報科学研究科博士後期課程修了。同年日本学術振興会特別研究員 (PD)。2010年カナダ Queen's 大学博士研究員。2011年九州大学大学院システム情報科学研究院助教。博士 (工学)。ソフトウェアメトリクス、マイニングソフトウェアリポジトリの研究に従事。情報処理学会、電子情報通信学会、IEEE 各会員。



上野 秀剛

2004年岩手県立大学ソフトウェア情報学部卒業。2009年奈良先端科学技術大学院大学博士課程修了。同年奈良工業高等専門学校情報工学科助教。博士 (工学)。ソフトウェア開発におけるヒューマン

ファクタの研究に従事。電子情報通信学会、ヒューマンインタフェース学会、IEEE、ACM 各会員。



門田 暁人

1994年名古屋大学工学部電気学科卒業。1989年奈良先端科学技術大学院大学情報科学研究科博士後期課程修了。同年同大学同研究科助手。2004年同大学助教授。2007年同大学准教授。2003~2004年 Auckland 大学客員研究員。博士 (工学)。ソフトウェアメトリクス、ソフトウェアプロテクション、ヒューマンファクタ等の研究に従事。日本ソフトウェア科学会、情報処理学会、電子情報通信学会、IEEE、ACM 各会員。



松本 健一

1985年大阪大学基礎工学部情報工学科卒業。1989年同大学大学院博士課程中退。同年同大学基礎工学部情報工学科助手。1993年奈良先端科学技術大学院大学助教授。2001年同大学教授。工学博士。エンピリカルソフトウェア工学、特に、プロジェクトデータ収集/利用支援の研究に従事。日本ソフトウェア科学会、電子情報通信学会、情報処理学会、ACM 各会員、IEEE Senior Member。