

# ソフトウェア工学における設計根拠研究の Systematic Literature Review

樋山 淳雄<sup>†1</sup>

**概要:** ソフトウェア開発において設計根拠は重要な情報である。それにも関わらず、ソフトウェア工学における設計根拠研究の体系的な文献レビューは行われていない。そこで本論文ではソフトウェア工学における設計根拠研究の体系的なレビューを Systematic Literature Review により実施した。その結果、次のことが明らかになった：(1)設計根拠研究は ICSE, ASE, TOSEM, JSS 等の難関国際会議や国際ジャーナルに数多く発表されている、(2)設計根拠研究が対象としている工程は、設計、アーキテクチャ(設計)、要求といった上流工程が過半数を占めている、(3)研究の問題認識として「追跡可能性」、「支援が不十分」、「記録コストの問題」、「効果が不明」、「設計根拠が抽出されていない・抽出の困難さ」が上位を占めており、この5項目で全体の約75%を占めた、(4)研究のアプローチとして、理論・枠組開発が最も多く、次いでツール開発であった。この2項目で全体の約80%を占めた、(5)設計根拠を表現する代表的なモデルである IBIS, gIBIS, QOC, DRL, PHI はあまり使われていない。

**キーワード:** 設計根拠, Systematic Literature Review, ソフトウェア工学

## Systematic Literature Review on Design Rationale Studies in Software Engineering

ATSUO HAZEYAMA<sup>†1</sup>

**Abstract:** Design rationale is important information in software development. In spite of it, systematic literature review on design rationale studies in software engineering has not been conducted. This study conducts systematic literature review on design rationale studies in software engineering. As the result, we clarify the followings: (1) papers regarding design rationale research have been published in high-level international conferences/journals such as ICSE, ASE, TOSEM, JSS, (2) more than half studies deal with the upstream processes such as design, architectural design, or requirement analysis, (3) top five of research issues are “traceability,” “insufficient support,” “cost for capturing,” “little knowledge for its effectiveness,” “shortage or difficulty of rationale capturing.” These occupied around 75% of the total, (4) as approach of research, “theory or framework development” and “tool development” are the top two, and they occupied around 80% of the total, and (5) IBIS, gIBIS, QOC, DRL, and PHI, which are representative rationale representation models, are not used frequently.

**Keywords:** Design rationale, Systematic Literature Review, Software engineering

### 1. はじめに

ソフトウェア開発では様々な意志決定がなされる。意志決定の結果は文書やソースコードとして表現される。一方で意志決定の背後にはその決定に至った意図や根拠がある。それら根拠は設計根拠 (design rationale) と呼ばれ、その起源は1970年に W. R. Kunz と H. Rittel により開発された Issue-Based Information System (IBIS) とされている[23]。

設計根拠はソフトウェア開発において非常に重要な情報である。設計根拠には様々な定義がなされているが[11]、その中で Dutoit らは「成果物の設計の決定に至る議論の筋道 (reasoning)」と定義している[3]。設計根拠を残す意義について、Alkadhi らは「開発知識を文書化し再利用することにより、変更への対応や効果的な保守が可能になる」[1]と述べている。

設計根拠の研究は Computer-Supported Cooperative Work (CSCW) 等の分野で活発に行われてきたが、ソフトウェア工

学分野における設計根拠研究を体系的に文献調査した研究は著者が調べた限りでは存在しない。

そこで、本論文では Systematic Literature Review (以下 SLR と記す) の手法を用いてソフトウェア工学分野における設計根拠研究を体系的に示す。

### 2. 関連研究

設計根拠を対象としたサーベイ論文はいくつかある。

Sagoo らは、設計根拠の定義、設計根拠の表現、設計根拠が行うことのできる能力 (capability) について、SLR を行っている[11]。ソフトウェア工学に限定したのではなく、本研究とは対象が異なる。

文献[10]は設計根拠システムに関するサーベイ論文である。設計根拠の表現方式、設計根拠の記録機能、検索機能について述べている。文献[10]は2000年の出版であり、設計根拠研究初期の頃のサーベイ論文である。

<sup>†1</sup> 東京学芸大学  
Tokyo Gakuhei University

### 3. SLR の手順

SLR とは特定の研究課題 (research question), トピックの領域, 関心の現象に関するすべての利用可能な研究を識別, 評価, 解釈する手段である[7]. 手順は, 研究課題の設定, データベースの選定, データベースを検索, 検索結果を吟味し調査対象の文献を決定, 研究課題に回答する.

以下に, 本研究で扱うテーマについて, 上述の各項目について述べる.

#### 3.1 Research Question (RQ)

RQ として以下を設定した.

RQ1. いつからどの程度設計根拠に関する研究が行われているのか

RQ2. どこで発表されているのか

RQ3. ソフトウェア開発のどの工程を対象にした研究であるのか

RQ4. どのような問題認識で研究がなされたか

RQ5. 根拠研究において, どのようなアプローチをとっているのか

RQ6. 根拠を表現するモデルに何を採用しているのか

RQ7. 根拠の情報源に特徴はあるのか

RQ1 と RQ2 で, いつ, どこで, どの程度研究されてきたのかを把握する. RQ3 で, 開発プロセスのどこを対象に設計根拠の研究がなされてきたのかを把握する. RQ4 で, 各研究の問題認識を明らかにする. RQ5 で, 各研究がとったアプローチ (新たな理論や枠組の提案なのか? ツール開発なのか? 実証実験なのか?) を明らかにする. 設計根拠を表現するモデルはこれまで数多く提案されてきた. 文献[11]では根拠を表現するモデルとして, Issue-Based Information System (IBIS), questions, options and criteria (QOC), decision representation language (DRL), procedural hierarchy of issues (PHI)をあげている. これらの代表的なモデルがどの程度用いられているのかを RQ6 で明らかにする. RQ7 で根拠の情報源に特徴があるかを明らかにする.

#### 3.2 データベースの選定

文献データベースとして, Elsevier の SCOPUS[22]を使用した.

#### 3.3 検索式

検索式として ("software engineering" AND ("design rationale"))を設定し, 2018 年 10 月 2 日に検索を実施した.

その結果, 178 件の検索結果が得られた.

#### 3.4 適合基準/除外基準の設定

調査対象の文献を決定するために, 本研究で設定した適合基準を示す.

(1) 英語で書かれた国際ジャーナル, 国際会議, 国際ワークショップ掲載論文並びにソフトウェア工学における根拠を扱った書籍に収録された記事

(2) 概要を読んでソフトウェア工学における設計根拠を扱

っている論文

(3) 被引用数 5 件以上の論文

(4) ページ数が 4 ページ以上の論文

除外基準は上述の(1)~(4)に該当しない論文に加え, サーベイ論文も除外とする.

その結果 43 件を SLR の対象とした.

### 4. RQ への回答

本節では, RQ に対する結果を述べる.

#### 4.1 RQ1: 年別の掲載論文数

年別の掲載論文数の結果を図 1 に示す. 適合基準を満たす論文は 1992 年に初めて掲載された. 年ごとの掲載論文数では, 2006 年に 6 件の掲載があったのが最多であった. 2000 年代初頭には適合基準を満たした論文が発表されていない年もあるが, 全体的にはコンスタントに研究成果が発表されている. 2017 年以降が 0 件なのは今回被引用数を除外基準に設けたので, ここ数年の論文は十分に引用されるまで時間が経過していない可能性もある.

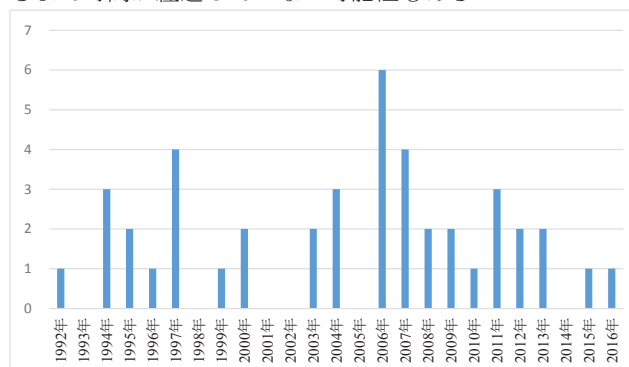


図 1 適合基準を満たした年別の掲載論文数  
Figure 1 Number of publication per year.

#### 4.2 RQ2: 掲載誌

掲載誌の分布に関する結果を図 2 に示す.

国際会議 ICSE (International Conference on Software Engineering) が最多の 7 編であった. 次に Rationale management に関する書籍に掲載された記事, 国際会議 ASE (Automated Software Engineering), JSS (Journal of Systems and Software), TOSEM (ACM Transactions on Software Engineering and Methodology), TSE (IEEE Transactions on Software Engineering), 国際会議 RE (IEEE International Requirements Engineering Conference)と続いた. ソフトウェア工学における難関の国際会議や国際ジャーナルが続き, これらが全体の約半数を占めた. このことから当該分野への関心が難関の国際会議や国際ジャーナルにあることがわかる.

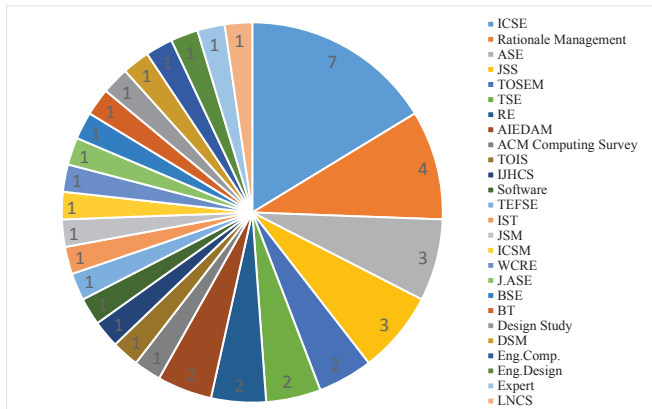


図 2 掲載誌の分布  
Figure 2 Distribution of publication.

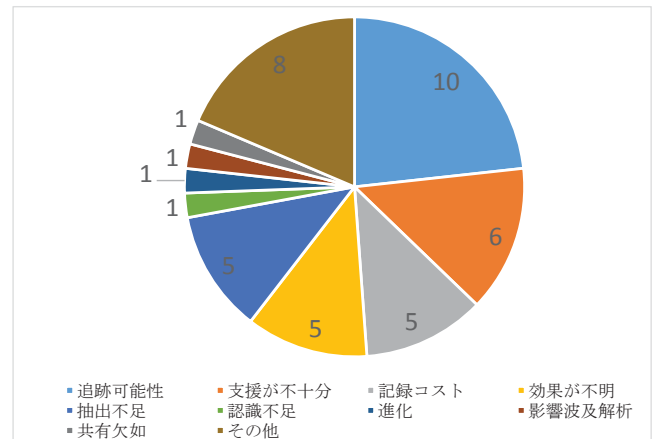


図 4 研究の問題認識  
Figure 4 Issues raised by studies.

#### 4.3 RQ3: 対象工程

研究対象の工程の結果を図 3 に示す。

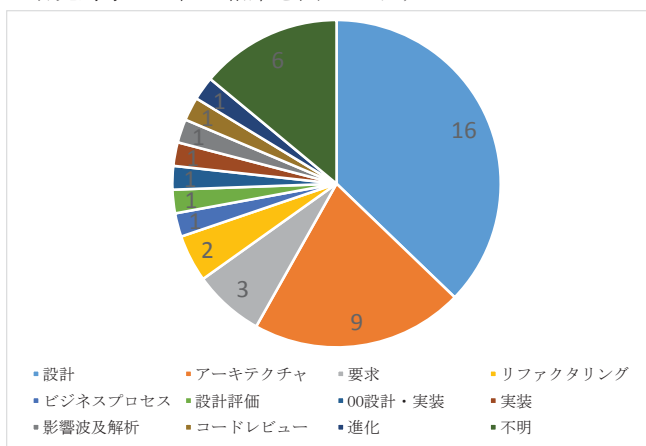


図 3 研究対象の工程  
Figure 3 Target phase.

研究対象の工程としては設計が最も多く、次いで、特にアーキテクチャ(設計)であった。その次に多かったのが要求である。上流工程を対象としたものが過半数を占めていることがわかる。一方、件数は少ないが、リファクタリングやコードレビューのようなコードを対象とする工程に関する研究があることも分かった。

#### 4.4 RQ4: 問題認識

研究の問題認識として「追跡可能性」、「支援が不十分」、「記録コストの問題」、「効果が不明」、「設計根拠が抽出されていない・抽出の困難さ」、「設計根拠に対する認識の低さ」、「設計根拠の進化を扱っていない」、「影響波及解析が行えない」、「履歴データの共有の欠如」、「その他」を識別した。それらを分類した結果を図 4 に示す。

問題認識として「追跡可能性」が最も多く、「支援が不十分」が続いた。さらにその次に同数で「記録コストの問題」、「効果が不明」、「設計根拠が抽出されていない・抽出の困難さ」が続いた。この 5 項目で全体の約 75%を占めた。

「追跡可能性」について、成果物と設計根拠の関連がない[6]、トレーサビリティリンク増大への対応[2]等が問題と認識されている。

「支援が不十分」について、文献[14]ではコードレビュー時には設計根拠が重要な役割を担うが、この情報の獲得や発見の支援が不十分であると指摘している。文献[21]では、設計者は最適な意志決定をしていないことを理由に、ツールによる意志決定支援の必要性を述べている。文献[19]では、リファクタリングにおいて設計根拠の理解は重要にもかかわらず、ほとんど支援がなされていないと問題指摘している。

「効果が不明」については、実際に論証 (reasoning)がどの程度なされているのかが明らかになっていない[18]、これまで提案されてきた手法が有用であるのかという問題認識[13][17]が示されている。

「設計根拠が抽出されていない・抽出の困難さ」については、文献[16]でアーキテクチャ設計の論証が未成熟なことを、文献[15]で設計根拠は重要だが、文書化されておらず、このことが矛盾、抜けの検出を困難にすると問題指摘している。

「設計根拠に対する認識の低さ」については、設計活動の文書化に力点が置かれ、その背後にある知識が過小評価されたり、設計根拠について無視される傾向にあることを指摘している。

#### 4.5 RQ5: 研究のアプローチ

研究のアプローチを「理論・枠組開発」、「ツール開発」、「実験」、「例示」、「評価」の 5 つに分類した。分類結果を図 5 に示す。

「理論・枠組開発」の提案が半数で、次いで、「ツール開発」であった。

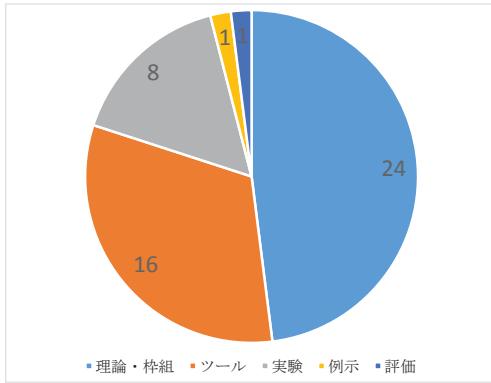


図 5 研究のアプローチ  
Figure 5 Approach for resolution of study.

問題認識とその解決のためのアプローチの対応を表 1 に示す。

表 1 研究の問題認識とアプローチの関係  
Table 1 Relationship between raised issues and approach for resolution.

	理論・枠組	ツール開発	実験	例示	評価
追跡可能性	8	4			
支援不十分	3	1	2		
記録コスト	2	1	3		
効果不明	1	1	2		1
抽出困難	2	4		1	
認識低		1			
進化	1				
共有欠如	1	1			
影響波及解析	1				

1 つの論文で、1 つの問題認識に対して複数の解決策のアプローチを記述した場合、記述されたアプローチすべてにカウントした。

表 1 から、理論・枠組開発並びにツール開発は多くの問題認識に対応していることがわかる。

「実験」は「支援が不十分」、「記録コストの問題」、「効果が不明」という問題認識に対して実施されていた。例えば、「記録コストの問題」について、実験結果に基づいて記録項目のカスタマイズの改善提案[4][5]や、近年では、マイニングにより根拠を取得する試み[9]も行われている。マイニングについては、今回の SLR では 1 件[6]の抽出にとどまっているが、文献[8]ではユーザレビューから根拠を抽出する試みも行われており、自動化による根拠の抽出はこれからの 1 つの潮流になる可能性がある。

例示は 1 件[16]で、アーキテクチャ設計への論証の適用を例示している。

評価も 1 件[13]で、設計根拠の効果が明らかでないという問題認識に対して、文献による評価を試みている。

次に、問題認識から研究アプローチとの関係を見つめる。追跡可能性の問題については、理論・枠組開発とツール

開発で解決を図っていることがわかる。

「支援が不十分」、「記録コストの問題」に対しては、理論・枠組開発、ツール開発、実験で解決を図っていることがわかる。

「効果が不明」の問題に対しては、理論・枠組開発、ツール開発、実験に加えて、文献による評価も行われている。

「設計根拠が抽出されていない・抽出の困難さ」に対して、理論・枠組開発、ツール開発、例示がなされている。文献[20]はリファクタリングにおける設計根拠抽出を支援するシステムを開発している。具体的にはバージョン間での設計レベルの変更の抽出、変更の可視化、分析等の機能を提供している。

#### 4.6 RQ6: 設計根拠表現モデル

採用された表現モデルの結果を図 6 に示す。

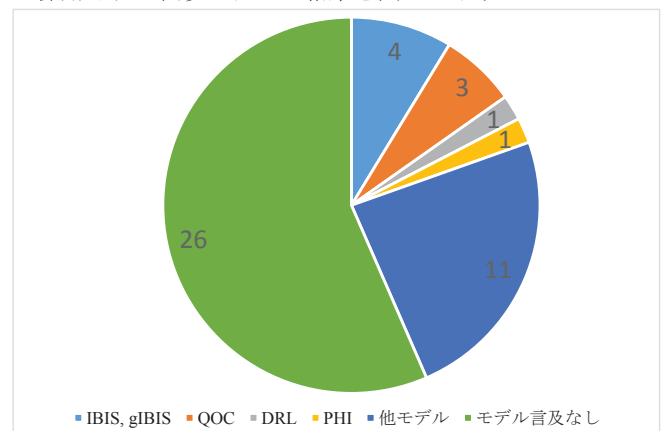


図 6 採用された設計根拠表現モデル  
Figure 6 Design rationale models adopted.

根拠を表現するモデルとして IBIS, gIBIS, QOC, DRL, PHI が知られているので、それらが多く用いられているという予想を持っていたが、半分以上はモデルに関する言及はなかった。また、モデルの言及があった研究においても、上述以外の他のモデルを採用しているものが多く、IBIS, gIBIS, QOC, DRL, PHI を言及していたのは合計しても全体の約 20%であった。上述以外の他のモデルでは AREL を言及しているものが多かった。これは AREL を開発した Tang の論文が多かったことによる。

#### 4.7 RQ7: 根拠の情報源

根拠の情報源の分類結果を図 7 に示す。

多くの論文では根拠の情報源を明示していなかった。情報源を明記したもののうちで最も多かったのは「対話」であり、「シナリオ」、「パターン」がそれに続いた。

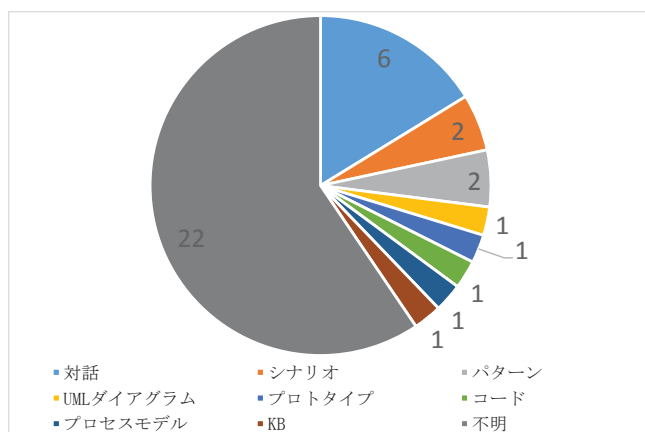


図 7 根拠の情報源  
Figure 7 Information sources for rationale.

#### 4.8 結果のまとめ

SLRにより明らかになったことを列挙する。

- RQ1 に対する回答：本研究で設定した適合基準に基づいた論文は 1992 年に初めて掲載され、以降コンスタントに当該分野の研究論文は発表され続けている。
- RQ2 に対する回答：設計根拠研究は ICSE, ASE, TOSEM, JSS 等の難関国際会議や国際ジャーナルが主たる発表の場である。
- RQ3 に対する回答：設計根拠研究が対象としている工程は、設計、アーキテクチャ(設計)、要求といった上流工程が過半数を占めている。
- RQ4 に対する回答：研究の問題認識として「追跡可能性」、「支援が不十分」、「記録コストの問題」、「効果が不明」、「設計根拠が抽出されていない・抽出の困難さ」が上位を占めており、この 5 項目で全体の約 75%を占めた。
- RQ5 に対する回答：研究のアプローチとして、理論・枠組開発が最も多く、次いでツール開発であった。この 2 項目で全体の約 80%を占めた。
- RQ6 に対する回答：設計根拠を表現する代表的なモデルである IBIS, gIBIS, QOC, DRL, PHI はあまり使われておらず調査対象全体の約 20%であった。
- RQ7 に対する回答：論文の半数以上は根拠の情報源を明記していなかった。情報源を明記した中では対話が最も多く用いられた情報源であった。

#### 4.9 妥当性の脅威

本 SLR には以下にあげる脅威を認識している。

- 論文の網羅性の問題：本論文で文献検索に採用した検索式は 1 つのパターンにとどまっている。関連する検索語を含む検索式を試して網羅性を高める必要がある。
- 各 RQ に対する分類は著者一人で行った。そのために解釈の誤りがある可能性がある。そのため、分類結果を複数人によりチェックする必要性もある。

## 5. おわりに

本論文では、ソフトウェア工学における設計根拠研究の体系的な文献レビューは行われていないという問題認識の下、SLR 手法に基づいてソフトウェア工学における設計根拠研究の体系的な文献レビューを実施、7つの RQ を設定し、その結果について述べた。

この分野は継続的に研究が進められており、追跡可能性、支援が不十分、記録コストの問題、効果が不明、設計根拠が抽出されていない、抽出の困難さという主要課題への挑戦が続いている。そのような中で、近年、マイニング技術等を導入した新たな方向性が示されつつある。また、ユーザレビュー等のインターネット上に公開されている情報源から根拠を抽出する試みもなされている。本 SLR ではそれらを十分に調査の対象に取り込めていない。これらの新たな試みも対象とするべく体系的調査を進めていくことが今後の課題である。

### 謝辞

本研究は科学研究費補助金基盤研究(C)課題番号 17K00475 の助成の下で行われた。記して謝意を表す。

### 参考文献

- [1] R. Alkadhi, T. Lata, E. Guzman, and B. Bruegge, Rationale in development chat messages: an exploratory study, IEEE/ACM 14th International Conference on Mining Software Repositories (MSR), pp. 436-446, IEEE, 2017.
- [2] J. Cleland-Huang, M. Mirakhorli, A. Czauderna, and M. Wieloch, Decision-centric traceability of architectural concerns, Proceedings of the 7th International Workshop on Traceability in Emerging Forms of Software Engineering (TEFSE2013), pp. 5-11, IEEE, 2013.
- [3] A. H. Dutoit, R. McCall, I. Mistrík, B. Paech, Rationale management in software engineering, Rationale Management in Software Engineering, Springer, 2006.
- [4] D. Falessi, R. Capilla, and G. Cantone, A value-based approach for documenting design decisions rationale: a replicated experiment, Proceedings of the 3rd International Workshop on Sharing and reusing architectural knowledge (SHARK2008), pp. 63-70, ACM, 2008.
- [5] D. Falessi, L. C. Briand, G. Cantone, R. Capilla, and P. Kruchten, The value of design rationale information, ACM Transactions on Software Engineering and Methodology (TOSEM), 22(3), 32 pages, 2013.
- [6] S. Gill, and E. V. Munson, A version-aware tool for design rationale, Proceedings of the 12th Brazilian Symposium on Multimedia and the web, pp. 20-26, ACM, 2006.
- [7] B. Kitchenham, Procedures for performing systematic reviews, Keele, UK, Keele University 33.2004, pp. 1-26 (2004).
- [8] Z. Kurtanović, and W. Maalej, Mining user rationale from software reviews, IEEE 25th International Requirements Engineering Conference (RE), pp. 61-70, 2017.
- [9] M. Lippi, and P. Torroni, Argumentation mining: State of the art and emerging trends, ACM Transactions on Internet Technology (TOIT), 16(2), 25 pages, 2016.
- [10] W. C. Regli, X. Hu, M. Atwood, and W. Sun, A survey of design rationale systems: approaches, representation, capture and retrieval, Engineering with computers, 16(3-4), pp. 209-235, 2000.

- [11] J. Sagoo, A. Tiwari, and J. Alcock, Reviewing the state-of-the-art design rationale definitions, representations and capabilities, *International Journal of Design Engineering*, 5(3), pp. 211-231, 2014.
- [12] F. Servant, and J. A. Jones, History slicing, *Proceedings of the 2011 26th IEEE/ACM International Conference on Automated Software Engineering*, pp. 452-455, IEEE Computer Society, 2011.
- [13] S. B. Shum, and N. Hammond, Argumentation-based design rationale: what use at what cost?, *International Journal of Human-Computer Studies*, 40(4), pp. 603-652, 1994.
- [14] A. Sutherland, and G. Venolia, Can peer code reviews be exploited for later information needs?, *Proceedings of the 31st International Conference on Software Engineering - Companion Volume*, pp. 259-262, IEEE, 2009.
- [15] A. Tang, Y. Jin, and J. Han, A rationale-based architecture model for design traceability and reasoning, *Journal of Systems and Software*, 80(6), pp. 918-934, 2007.
- [16] A. Tang, J. Han, and R. Vasa, Software architecture design reasoning: a case for improved methodology support, *IEEE software*, 26(2), pp. 43-49, 2009.
- [17] A. Tang, Software designers, are you biased?, *Proceedings of the 6th International Workshop on SHaring and Reusing Architectural Knowledge (SHARK2011)*, pp. 1-8, ACM, 2011.
- [18] A. Tang, and H. van Vliet, Software designers satisfice, *Proceedings of the European Conference on Software Architecture*, pp. 105-120, Springer, 2015.
- [19] Z. Xing, and E. Stroulia, Refactoring detection based on umldiff change-facts queries, *Proceedings of the 13th Working Conference on Reverse Engineering*, pp. 263-274, IEEE, 2006.
- [20] Z. Xing, and E. Stroulia, API-evolution support with Diff-CatchUp, *IEEE Transactions on Software Engineering*, 33(12), pp. 818-836, 2007.
- [21] C. Zannier, and F. Maurer, Social factors relevant to capturing design decisions, *Proceedings of the Second Workshop on SHaring and Reusing architectural Knowledge Architecture, Rationale, and Design Intent (SHARK2007)*, 4 pages, IEEE Computer Society, 2007.
- [22] <https://www.elsevier.com/ja-jp/solutions/scopus>
- [23] [https://en.wikipedia.org/wiki/Design\\_rationale](https://en.wikipedia.org/wiki/Design_rationale)