VLM によるソフトウェア図表の理解に関する予備調査

高橋舞衣 [1]　小原有以 [1]　中澤初穂 [2]　秋信有花 [3]　倉林利行[3]　倉光君郎[2]
[1] 日本女子大学大学院 理学研究科　[2] 日本女子大学 理学部
[3]NTT ソフトウェアイノベーションセンタ
{m1916046tm,m2016026oy}@ug.jwu.ac.jp　kuramitsuk@fc.jwu.ac.jp

概要

近年，ソフトウェア開発において LLM の活用が注目されている．しかし，視覚的な情報を扱えない LLM にテキストのみで開発における複雑な情報を正確に伝えるのは難しい．一方，ソフトウェア開発現場では，要件定義や設計を効率的に行うために，UML ダイアグラムなどの視覚的情報が広く利用されてきた．視覚的情報を活用することで，テキスト主体の LLM の限界を補い，より実用的な支援が期待できる．そのため，視覚的情報とテキストを統合理解する VLM の導入が求められる．しかし，現状の VLM がソフトウェア開発における図表をどの程度理解できるのかは不明である．本研究では，VLM のソフトウェア開発における図表理解能力の調査を目的として，これらの図表に特化したベンチマーク「JSWEMU」を開発した．本論文では，JSWEMU を用いた調査結果について述べる．

1　はじめに

近年，ソフトウェア開発における LLM（Large Language Model）の活用が進んでいる [1, 2]．開発者はテキストで要件や仕様を LLM に伝え，それに基づいて LLM はコード生成 [3] やバグ修正 [4] を行う．しかし，テキストのみで開発における全体の構造や関係性，振る舞いなどを LLM に十分に伝えるのは難しい．

一方で，従来のソフトウェア開発では，設計や仕様伝達，要件やタスクの整理のために視覚的な情報共有が不可欠であり，多様な図表が開発現場で活用されてきた．本論文では，これらの図表を「ソフトウェア図表」と表現する．ソフトウェア図表には，クラス図やシーケンス図など，13 種類の UML ダイアグラムが含まれる．また，UML 以外にも表やデータベース，業務フロー図，アーキテクチャ図といった図表も含まれる．

このような視覚的情報を活用することで，テキスト情報のみを処理する LLM の限界を補い，より実用的な支援を実現できる可能性がある．そのため，視覚的情報とテキスト情報を統合的に理解できる VLM（Vision Language Model）の導入が必要である．しかし，現時点で VLM がソフトウェア図表をどの程度理解し，活用できるのかについては，明らかになっていない．

本研究の目的は，VLM のソフトウェア図表理解能力について調査することである．そのために我々は，ソフトウェア図表理解に特化した日本語のベンチマークである「JSWEMU (SoftWare Engineering Multimodal Understanding benchmark)」を開発した．本論文では，JSWEMU による調査の結果を示す．

2　JSWEMU

2.1　特徴

JSWEMU は，ソフトウェア図表を対象とした画像理解能力を評価するための 4 択式のベンチマークである．JSWEMU の問題例を図 1 に示す．JSWEMU は，16 種類の図表を含む 53 枚の画像と 411 問の設問で構成されている．図表ごとの画像数と設問数については付録 A の表 2 に示す．

JSWEMU は，図表の画像，図表に関する設問（問題と 4 つの選択肢），正解ラベル，メタデータで構成される．評価対象の VLM は，図表の画像とその図表に関する設問が与えられ，選択肢ラベルで回答することが求められる．

また，画像には，異なる描画スタイルに対するモデルの性能や適応力を評価するために，以下の 3 種類の描画スタイルを取り入れた．

- PowerPoint の描画ツールで作成した画像
- 紙に手書きした画像

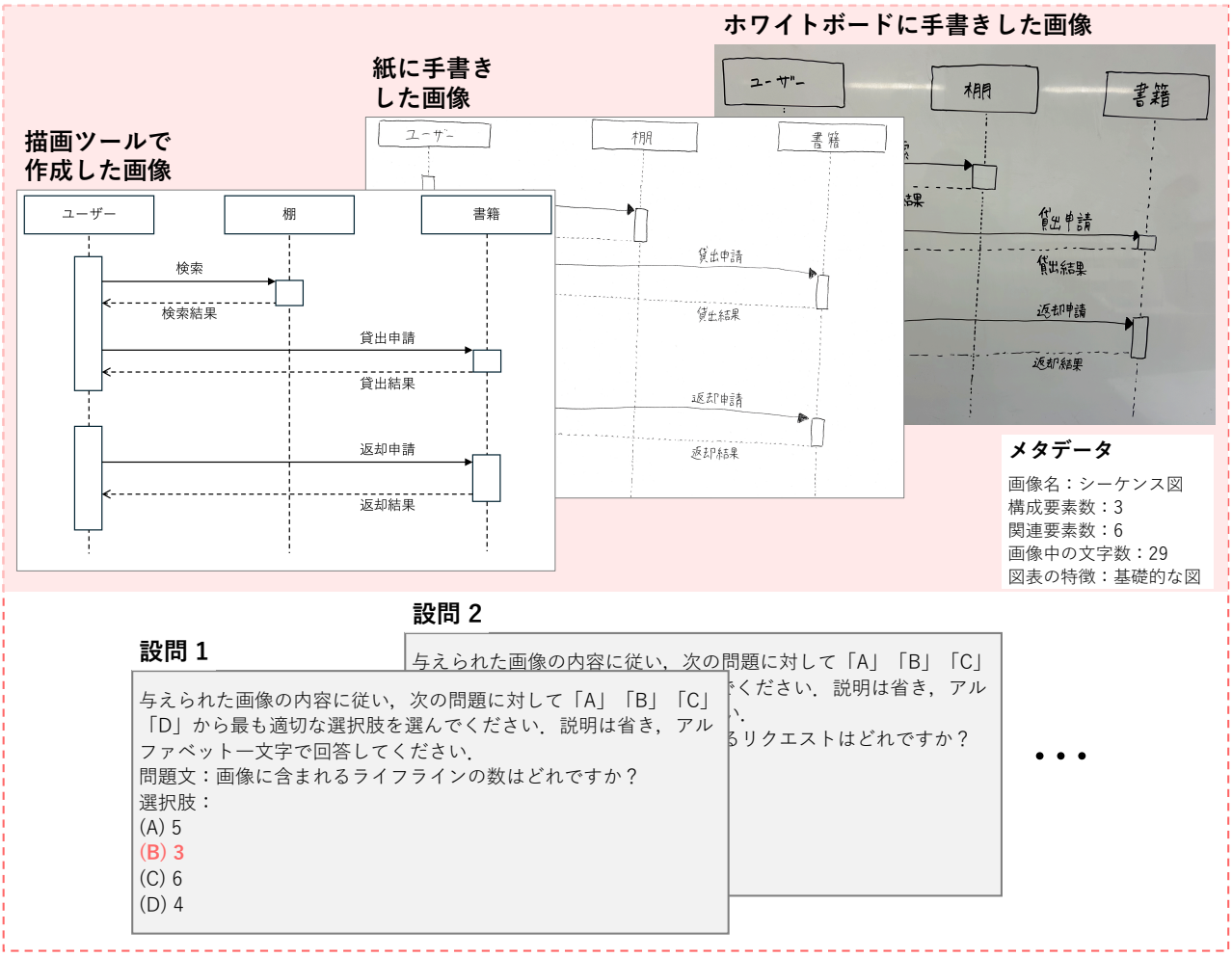

図 1　JSWEMU のデータ例

- ホワイトボードに手書きした画像

これらの描画スタイルは，文書からスキャンされた画像や，開発現場で紙やホワイトボードに描かれた図表を使用する状況を想定して選定したものである．描画ツールで作成された画像は，線や形状が正確でフォーマットが統一されており，モデルが視覚的特徴を認識しやすいため性能が良いと予想される．一方，紙やホワイトボードの画像は，不揃いな線や文字の崩れ，光の反射などのノイズがモデルの性能に影響を与える可能性がある．

2.2　開発

データの収集　本研究では，UML 2.5.1 で定められている 13 種類の図に加えて，表やデータベース，UML 以外の図表（業務フロー図・データフロー図・アーキテクチャ図・ER 図・ガントチャート・業務フロー図・状態遷移表・CRUD 表）を対象とした．我々は，これらの画像をソフトウェア開発に関する参考書やインターネットから収集した．

画像の作成　収集した画像を参考に，描画ツールで作成した画像，紙に手書きした画像，ホワイトボードに手書きした画像，の 3 種類の描画スタイルで画像を作成した．紙に手書きした画像はスキャンにより，ホワイトボードに手書きした画像は撮影によりデータ化した．

設問の作成　図表に基づき，問題文と 4 つの選択肢からなる設問を作成した．最終的に，大学院生 2 名で設問のレビューを行った．

3　実験

本節では，JSWEMU を用いた評価実験について説明する．

3.1 実験概要

モデルや描画スタイルによるソフトウェア図表理解能力の違いを確認するために，JSWEMU を用いて VLM の評価を行う．

モデル　以下の 7 種類の VLM を評価対象とする．

- GPT-4o（gpt-4o-2024-11-20）
- Gemini 1.5 Pro（gemini-1.5-pro）
- Claude 3.5 Sonnet（claude-3-5-sonnet-20241022）
- LLaVA-NeXT（llava-hf/llava-v1.6-mistral-7b-hf）
- Qwen2-VL（Qwen/Qwen2-VL-2B-Instruct）
- Llama 3.2-Vision
  （meta-llama/Llama-3.2-11B-Vision-Instruct）
- Phi-3.5-vision（microsoft/Phi-3.5-vision-instruct）

実験設定　クローズドソースモデルでは temperature を 0 に，オープンソースモデルでは do_sample を False に設定した．実験における他のハイパーパラメータはデフォルトに設定されている．
画像のサイズは 928×928 ピクセルに統一した．

評価指標　正答率により評価する．なお，必要に応じて，VLM の回答から正規表現を用いて選択肢ラベルのアルファベット 1 文字を抽出する処理を行っている．

3.2 実験結果

図表ごとの各モデルの正答率を表 1 に，描画スタイル別のモデルごとの正答率を図 2 に示す．主な結果は以下の通りである．

モデルによる影響　最も正答率が高いのは Claude 3.5 Sonnet で，全ての図表に対する正答率は 46.47% であった．一般的に，オープンソースモデルはクローズドソースモデルに比べて性能が劣る傾向がある [5, 6]．しかし，JSWEMU による評価では，Claude 3.5 Sonnet や GPT-4o が比較的高い正答率を示した一方で，Gemini 1.5 Pro では一部のオープンソースモデルに劣る結果となった．この要因として，JSWEMU が単なる画像理解を超えて，ソフトウェア開発における図表に関する専門的な知識を要求する点が挙げられる．

図表の種類による影響　表 1 の各モデルの正答率の平均より，パッケージ図，オブジェクト図，ステートマシン図，表やデータベースでは正答率が高い傾向が見られ，配置図，コンポーネント図，合成構造図では正答率が低い傾向が見られた．正答率が高い図表は，構造が明確で直感的に理解しやすい特徴がある．一方，正答率が低い図表は直感的な理解が難しく，図表の記法やシステム全体の構造，設計意図など，図表を正確に読み解くための専門知識を必要とすることが要因として考えられる．

また，図表の種類によって正答率の高いモデルが異なることが確認された．図表の種類ごとにモデルの性能が異なるため，これらの図表を扱う際には図表の種類に応じた適切なモデルの選択が重要である．

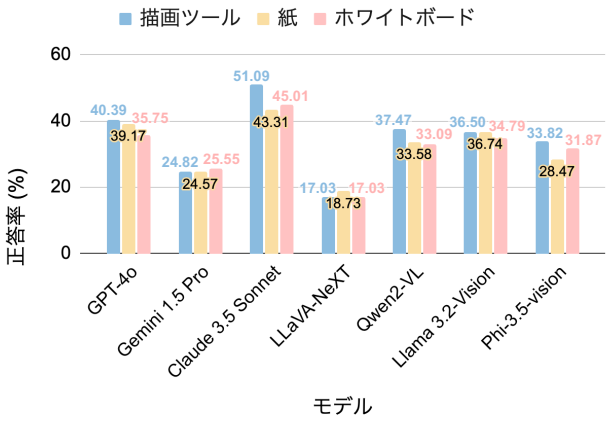

図 2　描画スタイル別のモデルごとの正答率

描画スタイルによる影響　図 2 に示すように，Claude 3.5 Sonnet や Qwen2-VL のような一部のモデルでは，描画ツールで作成した画像の正答率が手書き画像より高い傾向が観察された．一方で，GPT-4o や Phi-3.5-vision のように描画スタイル間の差がわずかなモデルも存在し，さらに Gemini 1.5 Pro，LLaVA-NeXT，Llama 3.2-Vision では，描画ツールで作成した画像よりも手書き画像の方が高い正答率が得られた．この結果から，描画ツールが必ずしも最良の結果を示すわけではないことが分かる．

4 関連研究

本節では画像理解ベンチマークやソフトウェア開発に関するベンチマークの関連研究を紹介する．

4.1 画像理解ベンチマーク

近年，VLM の画像理解能力を評価するために多肢選択形式のベンチマークが用いられている．たと

表 1　JSWEMU による図表ごとの各モデルの正答率（%）

図表＼モデル	クローズドソースモデル			オープンソースモデル				平均
	GPT-4o	Gemini 1.5 Pro	Claude 3.5 Sonnet	LLaVA-NeXT	Qwen2-VL	Llama 3.2-Vision	Phi-3.5-vision	
ユースケース図	43.43	25.25	59.60	12.12	44.44	35.35	30.30	35.78
オブジェクト図	41.18	29.41	54.90	35.29	37.25	60.78	52.94	44.54
クラス図	50.00	22.81	57.02	9.65	42.98	40.35	29.82	36.09
シーケンス図	27.78	21.11	43.33	24.44	25.56	35.56	26.67	29.21
コミュニケーション図	20.83	31.94	31.94	27.78	29.17	27.78	20.83	27.18
ステートマシン図	49.02	33.33	58.82	7.84	54.90	41.18	43.14	41.18
アクティビティ図	50.98	24.51	50.00	6.86	26.47	35.29	24.51	31.23
パッケージ図	46.15	17.95	69.23	53.85	53.85	53.85	53.85	49.82
コンポーネント図	19.44	26.39	18.06	8.33	26.39	22.22	23.61	20.63
配置図	9.72	26.39	16.67	2.78	12.50	22.22	15.28	15.08
合成構造図	24.56	8.77	19.30	12.28	22.81	40.35	17.54	20.80
タイミング図	28.79	27.27	33.33	15.15	25.76	28.79	37.88	28.14
相互作用概要図	25.64	30.77	46.15	20.51	43.59	30.77	30.77	32.60
表	56.67	21.67	51.67	8.33	48.33	51.67	33.33	38.81
データベース	45.00	25.00	56.67	21.67	48.33	33.33	36.67	38.10
その他	49.21	26.46	58.20	26.98	33.33	34.39	38.10	38.10
全ての図表	38.44	24.98	46.47	17.60	34.71	36.01	31.39	32.80

※ 3種の描画スタイルの平均　※ 表中の色は、各図表における最大・最小を示す

えば，Visual Commonsense Reasoning[7]，MMMU[5]，MMBench[8] などがある．

これらのベンチマークは，画像に関する説明や質問への回答を自然言語で生成し，正解と比較して評価する従来のベンチマーク（COCO[9]，VQA[10]，GQA[11]，VizWiz[12]，TouchStone[13]，LLaVA-Bench[14] など）と異なり，モデルの回答を選択肢と直接比較するため，モデルの画像理解能力をより明瞭に判定できる．

また，特定分野に特化した多肢選択形式の画像理解ベンチマークとして，ScienceQA[15]（科学），DocVQA[16]（食品や科学など産業ドキュメント画像），Japanese Heron-Bench[17] や JMMMU[18]（日本語 VLM 評価）がある．しかし，ソフトウェア開発分野に特化した画像理解ベンチマークはいまだ開発されていない．

JSWEMU は，ソフトウェア開発における図表の理解に特化したベンチマークで，多肢選択形式によって明瞭な評価を実現する．

4.2　ソフトウェア開発

ソフトウェア開発分野では，コードの生成によって LLM の能力を評価するベンチマークとして，MBPP[19] や HumanEval[3], SWE-bench[20] などが利用されている．また，多肢選択形式による評価で LLM のソフトウェア開発能力を評価するベンチマークとして，PythonIO[21]，CodeApex[22] などがある．

近年，ソフトウェア開発分野における VLM 用ベンチマークの開発も進んでいる．たとえば，Plot2Code[23] ではグラフ画像からコードを生成する能力を測定し，HumanEval-V[24] や MMCode[6] では画像を補助的に用いてコードを生成する能力を測定する．しかし，これらのベンチマークは主に開発能力の評価に焦点を当てており，ソフトウェア開発特有の図表理解能力を評価する設計になっていない．

JSWEMU は，ソフトウェア開発における多様な図表を対象とし，図表理解の評価に重点を置いたベンチマークである．

5　おわりに

本論文では，VLM のソフトウェア図表理解能力を調査することを目的とし，ソフトウェア図表に特化した日本語ベンチマーク「JSWEMU」を開発した．JSWEMU を用いた評価の結果，図表の種類によって正答率に差が生じ，特に専門的な知識を要する図表では正答率が低下する傾向が確認された．また，描画ツールにより作成した画像の正答率が必ずしも最良の結果を示すわけではないことがわかった．

今後は，ソフトウェア図表に対する VLM の理解能力を高める方法について検討していきたい．

謝辞

本研究は JSPS 科研費 JP23K11374 の助成を一部受けたものです.

参考文献

[1] Albert Ziegler, Eirini Kalliamvakou, X. Alice Li, Andrew Rice, Devon Rifkin, Shawn Simister, Ganesh Sittampalam, and Edward Aftandilian. Productivity assessment of neural code completion. In **Proceedings of the 6th ACM SIGPLAN International Symposium on Machine Programming**, MAPS 2022, p. 21–29, New York, NY, USA, 2022. Association for Computing Machinery.

[2] Angela Fan, Beliz Gokkaya, Mark Harman, Mitya Lyubarskiy, Shubho Sengupta, Shin Yoo, and Jie M. Zhang. Large language models for software engineering: Survey and open problems, 2023.

[3] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code, 2021.

[4] Rahul Gupta, Soham Pal, Aditya Kanade, and Shirish Shevade. Deepfix: Fixing common c language errors by deep learning. **Proceedings of the AAAI Conference on Artificial Intelligence**, Vol. 31, No. 1, Feb. 2017.

[5] Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, Cong Wei, Botao Yu, Ruibin Yuan, Renliang Sun, Ming Yin, Boyuan Zheng, Zhenzhu Yang, Yibo Liu, Wenhao Huang, Huan Sun, Yu Su, and Wenhu Chen. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi, 2024.

[6] Kaixin Li, Yuchen Tian, Qisheng Hu, Ziyang Luo, Zhiyong Huang, and Jing Ma. Mmcode: Benchmarking multimodal large language models for code generation with visually rich programming problems, 2024.

[7] Rowan Zellers, Yonatan Bisk, Ali Farhadi, and Yejin Choi. From recognition to cognition: Visual commonsense reasoning, 2019.

[8] Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, Kai Chen, and Dahua Lin. Mmbench: Is your multi-modal model an all-around player?, 2024.

[9] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context, 2014.

[10] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering, 2017.

[11] Drew A. Hudson and Christopher D. Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering, June 2019.

[12] Danna Gurari, Qing Li, Abigale J. Stangl, Anhong Guo, Chi Lin, Kristen Grauman, Jiebo Luo, and Jeffrey P. Bigham. Vizwiz grand challenge: Answering visual questions from blind people. In **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**, June 2018.

[13] Shuai Bai, Shusheng Yang, Jinze Bai, Peng Wang, Xingxuan Zhang, Junyang Lin, Xinggang Wang, Chang Zhou, and Jingren Zhou. Touchstone: Evaluating vision-language models by language models, 2023.

[14] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning, 2023.

[15] Pan Lu, Swaroop Mishra, Tanglin Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering, 2022.

[16] Minesh Mathew, Dimosthenis Karatzas, and C.V. Jawahar. Docvqa: A dataset for vqa on document images. In **Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)**, pp. 2200–2209, January 2021.

[17] Yuichi Inoue, Kento Sasaki, Yuma Ochi, Kazuki Fujii, Kotaro Tanahashi, and Yu Yamaguchi. Heron-bench: A benchmark for evaluating vision language models in japanese, 2024.

[18] Shota Onohara, Atsuyuki Miyai, Yuki Imajuku, Kazuki Egashira, Jeonghun Baek, Xiang Yue, Graham Neubig, and Kiyoharu Aizawa. Jmmmu: A japanese massive multi-discipline multimodal understanding benchmark for culture-aware evaluation, 2024.

[19] Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. Program synthesis with large language models, 2021.

[20] Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. Swebench: Can language models resolve real-world github issues?, 2024.

[21] Ziyin Zhang, Zhaokun Jiang, Lizhen Xu, Hongkun Hao, and Rui Wang. Multiple-choice questions are efficient and robust llm evaluators, 2024.

[22] Lingyue Fu, Huacan Chai, Shuang Luo, Kounianhua Du, Weiming Zhang, Longteng Fan, Jiayi Lei, Renting Rui, Jianghao Lin, Yuchen Fang, Yifan Liu, Jingkuan Wang, Siyuan Qi, Kangning Zhang, Weinan Zhang, and Yong Yu. Codeapex: A bilingual programming evaluation benchmark for large language models, 2024.

[23] Chengyue Wu, Yixiao Ge, Qiushan Guo, Jiahao Wang, Zhixuan Liang, Zeyu Lu, Ying Shan, and Ping Luo. Plot2code: A comprehensive benchmark for evaluating multi-modal large language models in code generation from scientific plots, 2024.

[24] Fengji Zhang, Linquan Wu, Huiyu Bai, Guancheng Lin, Xiao Li, Xiao Yu, Yue Wang, Bei Chen, and Jacky Keung. Humaneval-v: Evaluating visual understanding and reasoning abilities of large multimodal models through coding tasks, 2024.

A　JSWEMU の画像数と設問数

JSWEMU の図表ごとの画像数と設問数を表 2 に示す.

表 2　JSWEMU の図表ごとの画像数と設問数

図表	画像数（枚）	設問数（問）
ユースケース図	3	33
オブジェクト図	3	17
クラス図	3	38
シーケンス図	3	30
コミュニケーション図	3	24
ステートマシン図	3	17
アクティビティ図	3	34
パッケージ図	3	13
コンポーネント図	3	24
配置図	3	24
合成構造図	3	19
タイミング図	3	22
相互作用概要図	3	13
表	3	20
データベース	2	20
その他	9	63
全ての図表	53	411