

Literaturübersicht zur kollaborativen Modellierung

Daniel Dietrich

01. September 2020

Abstract

Das Interesse am kollaborativen Modellieren ist groß und gewinnt zunehmend an Bedeutung. In diesem Paper werden die Resultate einer Meta-Studie von 2017 als Basis genommen, um sich mit einem der vielversprechendsten Ansätze für große Projekte mit unterschiedlichen Akteuren genauer zu beschäftigen. Dafür wird eine Vergleichsstudie von 2020 herangezogen, die einen speziell entwickelten Chatbot für derartige Aufgaben mit einem typischen, klassischen Online-Werkzeug vergleicht. Die Resultate der Vergleichsstudie ermutigen, die Forschung auf dem Feld der natürlichsprachlichen Assistenten zur Unterstützung bei kollaborativen Modellierungsaufgaben weiter auszubauen.

Kategorien

- CCS Concepts
 - Software and its engineering
 - * Programming teams
 - * Specification languages
 - * System modeling languages
 - Open source model
 - * Unified Modeling Language (UML)

Schlüsselwörter

Collaborative Software Engineering, Model Driven Software Engineering, Collaborative Modeling, Natural Language Processing, Chatbot, SOCIO

1 Einleitung

Die kollaborative Modellierung ist eine Teamaktivität, an der mehrere Ingenieure und Stakeholder aktiv an der Erstellung eines gemeinsamen Modells beteiligt sind (45). Sie ist somit Teil des Collaborative Software Engineering (CoSE), welches sich mit Methoden, Prozessen und Werkzeugen zur Verbesserung der Zusammenarbeit, Kommunikation und Koordination (3C)¹ zwischen Team-Mitgliedern befasst (31, 14). Die kollaborative Modellierung geht über eine kooperative Zusammenarbeit hinaus, bei der einzelne Teile durch starke Arbeitsteilung separat bearbeitet und später zu

einem Ergebnis zusammengefügt werden. Sie zeichnet sich stattdessen durch einen hohen Grad an Zusammenarbeit und der gleichzeitigen Arbeit an Teilaufgaben aus.

Zur Kollaboration werden Mitten benötigt, um den beteiligten Stakeholdern das gemeinsame Arbeiten an den Modellierungsartefakten zu ermöglichen und sich als Team zu koordinieren. Dafür werden unter anderem Systeme zur Versionsverwaltung, Versionsmanagement, Konfliktmanagement- und Visualisierung sowie Vergleichswerkzeuge für das Modell verwendet.

Eine Reihe von Kommunikationsmitteln ermöglicht es den beteiligten Stakeholdern, Informationen über die Aktivitäten der anderen Stakeholder zu erlangen und Informationen innerhalb des Teams auszutauschen. Diese Aktivitäten können durchgeführt werden, indem (Entwurfs-) Entscheidungen ausgetauscht, Diskussionen zwischen den Stakeholdern hinsichtlich der Modellierung von Artefakten verfolgt und relevante Informationen zu verschiedenen Aspekten des Projekts gesammelt werden. Hierfür können beispielsweise Übersichten zu Aktivitäten, allgemeine Benachrichtigungen über Aktivitäten, Aufgaben-Verwaltung, asynchrone Nachrichtensysteme wie Chat, soziale Netzwerke oder auch Wikis eingesetzt werden.

Für die Koordination ist eine Infrastruktur zur Verwaltung des Lebenszyklus der Modelle notwendig. Diese kann ein (möglicherweise verteiltes) Repository zum Verwalten der Persistenz der Modelle und der zugehörigen Metadaten, ein Modellierungswerkzeug zum Erstellen, Bearbeiten oder Löschen von Modellen sowie zum Austausch von Formaten für die gemeinsame Nutzung von Modellen zwischen Projekten und Organisationen enthalten.

Allgemein kann gemeinsame Modellierung synchron oder asynchron, lokal oder räumlich getrennt, sowie in unterschiedlicher Intensität bezüglich der Mitarbeit erfolgen. Gemeinsame Modellierung ist aus einer Vielzahl von Gründen sinnvoll und teils notwendig. Gerade bei größeren Projekten sind oftmals verschiedene Expertisen fürs Modell notwendig, sodass mehrere Personen wie Mitglieder eines Softwareentwicklungsteams, aber auch externe und nichttechnische Stakeholder wie Kunden und Endbenutzer an der Modellierung beteiligt sein können (37, 44, 14). Die Bedeutung von Collaborative Software Engineering ist daher groß (30, 23, 2) und wird durch die wachsende Bedeutung (31, 14) agiler Methoden, von Open-Source Softwareprojekten und globaler Softwareentwicklung gestärkt.(29)

¹Collaboration, Communication, Coordination

2 Überblick bis 2017

In einer Meta-Studie (14) werden 48 Paper aus 19 Jahren zum Thema kollaborativer modellgetriebener Softwareentwicklung² zusammengefasst und analysiert. Dabei konzentriert man sich auf diejenigen Ansätze, bei denen mehrere Stakeholder über eine gemeinsame Plattform zusammenarbeiten können, um Modelle eines Softwaresystems zu erstellen und zu verwalten. Die Stakeholder werden dabei in technische und nicht-technische Stakeholder unterteilt.

Das Interesse am kollaborativen Modellieren ist groß, sodass sich eine Reihe von Forschungsprojekten (23) und Produkten entwickelt haben, um großen Teams das gemeinsame Modellieren zu ermöglichen. Im Folgenden eine Übersicht von ausgewählten Projekten und Produkten.(27, 41, 10, 42, 7, 3, 25, 22)

Dawn basiert auf Connected Data Objects (CDO) (18) des gleichnamigen Eclipse Projektes und schafft kollaborative Netzwerklösungen für Benutzeroberflächen. Neben der gemeinsamen Bearbeitung in Echtzeit bietet Dawn Konfliktvisualisierung und -behandlung, sowie weitere nützliche Funktionen für die kollaborative Modellierung. Außerdem stellt Dawn eine Ansicht zur Verfolgung von Diagrammänderungen bereit.(11)

emfCollab ist eine leichtgewichtige Lösung, mit der mehrere Benutzer gleichzeitig ein einzelnes Modell des Eclipse Modeling Frameworks (EMF) (16) bearbeiten können. emfCollab verfügt über eine Client-Server-Architektur, bei der der Client mit einigen Anpassungen in jeden EMF-basierten Eclipse-Editor integriert werden kann.(34)

MTF steht für das Modeling Team Framework (MTF), einem Open-Source Projekt im Rahmen des Eclipse Modeling Framework Technology (EMFT) Projektes (39). Es ist ein Entwurf eines Mechanismus, der über ein Meta-Repository die Versionsverwaltung des Quellcodes mit Werkzeugen wie git oder svn mit einem weiteren Repository wie CDO (18) zur Verwaltung der EMF Blobs verbindet.(40)

GenMyModel unterstützt die Erstellung und gemeinsame Bearbeitung von Diagrammen in Echtzeit. Das System ist so designt, dass die Konformität der Diagramme immer gewährleistet und somit keine Konfliktlösung für Endbenutzer notwendig ist.(1)

UNICASE ist ein Open-Source Projekt auf Basis der Eclipse-Plattform einschließlich EMF und GMF (Graphical Modeling Framework) (17). Es verbindet Elemente aus

dem Systemmodell (funktionale Anforderungen, Anwendungsfälle, Komponenten oder UML) mit jenen aus dem Projektmodell (Aufgaben, Fehlerberichte, Organisationsstruktur, Iterationen oder Besprechungen), die in einem einheitlichen Repository auf einem Server gespeichert werden.(43)

Collaboro ist ein Open-Source Ansatz, um Sprachentwicklungsprozesse partizipativer zu gestalten. Sowohl Entwickler als auch Benutzer der domänenspezifischen Sprache (DSL) können zusammenarbeiten, um diese zu erstellen und weiterzuentwickeln. Collaboro unterstützt sowohl die kollaborative Definition der abstrakten als auch der konkreten Syntax eine DSL, indem es eine Umgebung für Diskussionen, Vorschläge und Abstimmungen bereitstellt. Die Diskussionen bereichert die Sprachdefinition erheblich und stellt sicher, dass das Endergebnis den Erwartungen der Endbenutzer so weit wie möglich entspricht.(19, 20, 21, 4, 6, 5)

MDEFForge ist eine webbasierte Modellierungsplattform, welche die Entwicklung, Analyse und Wiederverwendung von Modellierungsartefakten ermöglicht und ein erweiterbares und Community-basiertes Repository für Modellierungsartefakten als Software-as-a-Service (SaaS) bietet.(36, 28)

2.1 Resultate der Meta-Studie

Die Meta-Studie liefert unter anderem folgende Resultate.

- Das wissenschaftliche Interesse an kollaborativer, modellgetriebener Softwareentwicklung³ (MDSE) ist im Laufe der Zeit gestiegen.
- Die meisten Ansätze sind sprachspezifisch, wobei UML-basierte Ansätze dominieren.
- Nur wenige Ansätze unterstützen bisher sowohl synchrone als auch asynchrone Zusammenarbeit.
- Die Aspekte Multi-View-Modellierung, Validierungsunterstützung, Wiederverwendung und Verzweigung werden seltener betrachtet.

Dabei zeigte sich, dass die Mehrheit der Systeme (22/27)⁴ für synchrone Zusammenarbeit keine Versionskontrolle bietet, wohingegen die Mehrheit der Systeme (18/24) für asynchrone Zusammenarbeit sehr wohl über eine Versionskontrolle verfügt. Bezüglich der Konfliktlösung bei den synchronen Systemen bieten 9 keine Konfliktlösung, 9 vermeiden Konflikte durchs

²Collaborative Model Driven Software Engineering

³Model Driven Software Engineering

⁴Im Original steht „22/28“, was wohl ein Tippfehler ist.

Design (Sperren) und 9 verfügen über eine 3-Wege-Zusammenführung als Konfliktlösung. Dabei zeigte sich, dass erhöhte Kommunikation bei synchroner Zusammenarbeit die Konflikte reduziert. Insbesondere erwies sich Chat als der klare Gewinner zur Konfliktvermeidung bei Ansätzen mit synchroner Zusammenarbeit. Bei den asynchronen Systemen bieten 21 Systeme eine Konfliktverwaltung und 3 vermeiden Konflikte durchs Design.

Bei den Editoren von Drittanbietern unterstützen 5 von 6 nur Single-View-Modellierung und nur einer die Multi-View-Modellierung. Bei den meisten UML-basierten Ansätzen (17/22) besteht keine Möglichkeit der kollaborativen Modellierung mit mehreren Ansichten. Hier ist stets nur die Bearbeitung eines UML-Diagramms zur selben Zeit möglich. Die anderen 5 von 22 Ansätze basieren auf Projektionsansichten. Nur ein System (47) unterstützt Echtzeit-Modellaktualisierungen.

In 5 von 6 Ansätzen, in denen technische und nicht-technische Stakeholder zusammenarbeiten, ist der grafische Editor das einzige Mittel zur Bearbeitung von Modellen. Nur bei einem Ansatz (13) wird der grafische Editor durch einen Texteditor ergänzt. Diese Entscheidung kann auf Aspekten der Benutzerfreundlichkeit und der Verständlichkeit der Editoren für nicht-technische Benutzer beruhen. Da bekannt ist, dass grafische Editoren für große oder komplexe Modelle nicht gut geeignet sind (23), können technische und nicht-technische Stakeholdern bisher nicht gut, gemeinsam an größeren Modellen arbeiten. Angesichts des Trends zu immer größeren Modellen (23), sollen künftige kollaborative MDSE-Ansätze effiziente und elegante Lösungen für diese Lücke bieten.

In den 41 Ansätzen nur für technische Stakeholder gibt es eine Vielzahl an Kombinationen von Modellierungseeditoren, wobei grafische, baumbasierte und textuelle Editoren am meisten verbreitet sind. Ein Ansatz (46) für nur nicht-technische Interessengruppen stellt einen skizzenbasierten Modellierungseeditor bereit.

Die Softwareproduktion unterliegt immer mehr der Globalisierung. Die Teams müssen verteilt und schnell arbeiten, und die Stakeholder haben unterschiedliche, potenziell widersprüchliche Ansprüche. Diese Anforderungen erfordern einen höheren Automatisierungsgrad, eine bessere Kommunikation, sowie eine verteilte und schnelle Entscheidungsfindung (14).

Die Resultate der Meta-Studie ermutigen, sich insbesondere mehr mit Chat-basierten Ansätzen für die synchrone Zusammenarbeit zu beschäftigen. Diese weisen ein hohes Potential an Konfliktvermeidung auf und erleichtern nicht-technischen Stakeholder wegen der Nutzung eines natürlichsprachlichen Interfaces den Zugang. Da ein solches System keinen grafischen Editor zur Erstellung verwendet, könnte es sich auch für nicht-technische Stakeholder zur Erstellung größerer und komplexerer Modelle eignen.

3 Natürlichsprachliche Assistenten

Der Fortschritt bei der Verarbeitung natürlicher Sprache (NLP)⁵ führte zur Entwicklung von Chatbots und Sprachassistenten. Nahezu jede Branche schlägt ein natürlichsprachliches System für einen flexibleren Zugang zu ihren Diensten vor. So können Chatbots im Gesundheitswesen helfen, Krankheiten wie Diabetes selbst besser zu kontrollieren und bei Fragen oder Problemen als zusätzlicher Ansprechpartner zur Seite stehen. (8, 38, 12)

Viele Software-Engineering Aufgaben können ebenfalls ganz oder teilweise von Chatbots übernommen werden. Dazu zählt die Automatisierung von Bereitstellungsaufgaben, die Zuweisung von Aufgaben, die Reparatur von Buildfehlern, die Planung von Aufgaben wie dem Senden von Erinnerungen, die Integration von Kommunikationskanälen oder die Unterstützung beim Kundensupport (24). Zu den Vorteilen von Chatbots und Sprachassistenten zählt die Senkung der Eintrittsbarriere für nichttechnische Experten durch die Verwendung natürlicher Sprache. Dieser Umstand fördert auch eine aktivere Beteiligung aller Beteiligten. Außerdem kann die Arbeit über einen, von der sonstigen Kommunikation über Kurznachrichten vertrauten Kanal stattfinden (35).

3.1 Vergleich von SOCIO und Creately

Eine aktuelle Studie (35) vergleicht SOCIO (32, 33) mit klassischen Online-Werkzeugen zur kollaborativen Modellierung. Dabei liegt der Fokus der Studie auf Modellierung mit synchronen Werkzeugen.

SOCIO ist ein Chatbot für die kollaborative Modellierung, der derzeit nur für die Sprache Englisch verfügbar ist und über Telegram und Twitter (@ModellingBot) genutzt werden kann. Dafür sendet ein Benutzer eine Nachricht an SOCIO, welcher diese interpretiert und sowohl bei der Erstellung von Modellen helfen als auch Statistiken über Urheberschaften und Nachrichten ausgeben kann. Als Antwort sendet SOCIO ein Bild des aktuellen Modellplans zurück, wobei veränderte Teile farbig gekennzeichnet sind. Allgemein kann SOCIO über zwei Arten von Befehlen angesprochen werden.

Wird SOCIO über einen natürlichsprachlichen Befehl wie the house contains rooms angesprochen, werden zunächst die relevanten Teile wie Substantive, Verben und Adjektive einer Phrase identifiziert, um zu entscheiden, welche Aktionen ausgeführt werden sollen. Bei den Aktionen kann es sich um die Erstellung oder Aktualisierung einer Klasse, eines Attributs oder einer Beziehung handeln. In der Beispielphrase werden die beiden Substantive Haus und Räume identifiziert, für die jeweils eine Klasse erstellt wird. Das Verb enthält wird einer Ist-Teil-von-Referenz zugeordnet, während Räumen als Pluralform viele Kardinalitäten suggeriert. Alternativ kann SOCIO auch über Kommandos wie add class X oder set attribute size to int gesteuert werden.

⁵Natural Language Processing

Für den Vergleich wurden GenMyModel (1), Lucid-Chart (26), Gliffy (15) und Creately (9) als Online-Werkzeuge in Betracht gezogen und Creately als repräsentatives Online-Modellierungswerkzeug ausgewählt. Creately ist eines der am meisten genutzten Online-Werkzeuge mit benutzerfreundlicher, grafischer Benutzeroberfläche, was es zu einem guten Vergleichskandidaten macht.(35)

3.2 Durchführung der Studie

Um die beiden Modellierungswerkzeuge miteinander zu vergleichen, wurde eine Studie mit 54 Teilnehmern durchgeführt, die alle einen Abschluss in Informatik oder etwas Vergleichbarem besitzen, über das notwendige Fachwissen zur Erstellung solcher Modelle verfügen und ein ausreichend hohes englisches Sprachniveau haben, um mit dem Chatbot interagieren zu können. Die Teilnehmer wurden in zwei Gruppen mit jeweils 27 Teilnehmern aufgeteilt, wobei jede Gruppe aus 9 Teams mit jeweils 3 Teilnehmern bestand. Jedes Team musste zwei Modellierungsaufgaben bearbeiten und hatte 30 Min. je Modellierung zur Verfügung. Außerdem durften Teilnehmer nur über die Telegram-Gruppe miteinander kommunizieren.

Die Studie wurde als Cross-over-Studie durchgeführt, sodass Gruppe 1 zunächst die erste Aufgabe mit SOCIO und danach die zweite Aufgabe mit Creately bearbeitet hat und Gruppe 2 genau umgekehrt, also die erste Aufgabe mit Creately und danach die zweite Aufgabe mit SOCIO. Die Durchführung der Modellierungsaufgaben wurde bezüglich Effizienz, Effektivität, Zufriedenheit und Qualität der Ergebnisse bewertet.(35)

3.3 Resultate

Die Auswertung der Vergleichsstudie zeigt, dass die Modellierungsaufgaben mit SOCIO im Durchschnitt weniger Zeit- und Kommunikationsaufwand kosten und trotzdem zu einer höheren Benutzerzufriedenheit und einem höheren Detailgrad der Diagramme führt. Nutzer erstellten zwar mit Creately mehr Klassen und somit vollständigere Diagramme, dafür waren die mit SOCIO erstellten Klassen vollständiger. Bezüglich der Effektivität, Genauigkeit und Fehler ist SOCIO mit Creately vergleichbar. Außerdem scheint die Benutzeroberfläche von Creately nicht so ansprechend zu sein wie jene von SOCIO. Obwohl die Resultate des Chatbots recht gut waren, besteht die Notwendigkeit, die Genauigkeit zu verbessern. Insgesamt scheint SOCIO in Bezug auf Effizienz und Zufriedenheit gegenüber Creately überlegen zu sein, während sie in Bezug auf Effektivität ähnlich sind.(35)

4 Fazit

Die Vergleichsstudie (35) hat gezeigt, dass der Einsatz von natürlichsprachlichen Assistenten in der kollaborativen Modellierung einige Vorteile hinsichtlich Effizienz und Benutzerfreundlichkeit, insbesondere für nicht-technische Akteure bringt. Die gelieferten Resultate

zeigen aber auch die Notwendigkeit auf, an der Verbesserung der Genauigkeit zu arbeiten.

Literatur

- [1] Axellience. The leading online modeling platform, 2020. 2020-09-22.
- [2] Marco Brambilla, Jordi Cabot, and Manuel Wimmer. Model-Driven Software Engineering in Practice, Second Edition. Synthesis Lectures on Software Engineering. Morgan & Claypool Publishers, 2017.
- [3] Bernd Bruegge, Oliver Creighton, Jonas Helming, and Maximilian Kögel. Unicae—an ecosystem for unified software engineering research tools. In Third IEEE International Conference on Global Software Engineering, ICGSE, volume 2008. Cite-seer, 2008.
- [4] Jordi Cabot, Javier Luis Cánovas Izquierdo, and Valerio Cosentino. Community-based software development for MDE tools. In Francis Bordeleau, Jean-Michel Bruel, Jürgen Dingel, Sébastien Gérard, Henry Muccini, Gunter Mussbacher, and Sebastian Voss, editors, Joint Proceedings of the 12th Educators Symposium (EduSymp 2016) and 3rd International Workshop on Open Source Software for Model Driven Engineering (OSS4MDE 2016) co-located with the ACM/IEEE 19th International Conference on Model Driven Engineering Languages and Systems (MODELS 2016), Saint Malo, France, October 3, 2016, volume 1835 of CEUR Workshop Proceedings, pages 45–50. CEUR-WS.org, 2016.
- [5] Javier Canovas and Jordi Cabot. Create your domain-specific language collaboratively. GitHub.io, October 2017. 2020-09-23.
- [6] Javier Canovas and Juan Villa. Collaboro. GitHub, 2017. 2020-09-23.
- [7] Marcelo Cataldo, Charles P. Shelton, Yongjoon Choi, Yun-Yin Huang, Vytresh Ramesh, Darpan Saini, and Liang-Yun Wang. CAMEL: A tool for collaborative distributed software design. In 4th IEEE International Conference on Global Software Engineering, ICGSE 2009, Limerick, Ireland, 13-16 July, 2009, pages 83–92. IEEE Computer Society, 2009.
- [8] Amy Cheng, Vaishnavi Raghavaraju, Jayanth Kanugo, Yohanes P. Handrianto, and Yi Shang. Development and evaluation of a healthy coping voice interface application using the google home for elderly patients with type 2 diabetes. In 15th IEEE Annual Consumer Communications & Networking Conference, CCNC 2018, Las Vegas, NV, USA, January 12-15, 2018, pages 1–5. IEEE, 2018.
- [9] creately.com/. The visualworkspace forteam collaboration. creately.com, 2020. 2020-09-23.

- [10] Matthias Farwick, Berthold Agreiter, Jules White, Simon Forster, Norbert Lanzanasto, and Ruth Breu. A web-based collaborative metamodelling environment with secure remote model access. In Boualem Benatallah, Fabio Casati, Gerti Kappel, and Gustavo Rossi, editors, *Web Engineering, 10th International Conference, ICWE 2010, Vienna, Austria, July 5-9, 2010. Proceedings*, volume 6189 of *Lecture Notes in Computer Science*, pages 278–291. Springer, 2010.
- [11] Martin Fluegge. Dawn. Eclipse Wiki, October 2011. 2020-09-22.
- [12] Center for Advanced Hindsight. Chatbots for diabetes self-management: Diabetes coaching at scale. Technical report, Center for Advanced Hindsight, 2018.
- [13] Chiara Di Francescomarino, Chiara Ghidini, Marco Rospocher, Luciano Serafini, and Paolo Tonella. A framework for the collaborative specification of semantically annotated business processes. *J. Softw. Maintenance Res. Pract.*, 23(4):261–295, 2011.
- [14] Mirco Franzago, Davide Di Ruscio, Ivano Malavolta, and Henry Muccini. Collaborative model-driven software engineering: a classification framework and a research map. *IEEE Transactions on Software Engineering*, 44(12):1146–1175, 2017.
- [15] gliffy.com. Diagramming software built for any team. gliffy.com, 2020. 2020-09-23.
- [16] Richard Gronback. Eclipse modeling framework (emf). Eclipse Modeling, 2020. 2020-09-23.
- [17] Richard C. Gronback and Nick Boldt. Graphical modeling project (gmp). Eclipse Modeling, 2020. 2020-09-23.
- [18] Hendy Irawan and Ibrahim Sallam. Cdo. Eclipse Wiki, November 2016. 2020-09-22.
- [19] Javier Luis Cánovas Izquierdo and Jordi Cabot. Community-driven language development. In Joanne M. Atlee, Robert Baillargeon, Robert B. France, Geri Georg, Ana Moreira, Bernhard Rumpe, and Steffen Zschaler, editors, *Proceedings of the 4th International Workshop on Modeling in Software Engineering, MiSE 2012, Zurich, Switzerland, June 2-3, 2012*, pages 29–35. IEEE Computer Society, 2012.
- [20] Javier Luis Cánovas Izquierdo and Jordi Cabot. Collaboro: a collaborative (meta) modeling tool. *PeerJ Comput. Sci.*, 2:e84, 2016.
- [21] Javier Luis Cánovas Izquierdo and Jordi Cabot. Collaboro: A collaborative (meta) modeling tool. *PeerJ Prepr.*, 4:e2046, 2016.
- [22] Steven Kelly, Kalle Lyytinen, and Matti Rossi. Metaedit+: A fully configurable multi-user and multi-tool CASE and CAME environment. In Panos Constantopoulos, John Mylopoulos, and Yannis Vassiliou, editors, *Advances Information System Engineering, 8th International Conference, CAISE'96, Heraklion, Crete, Greece, May 20-24, 1996, Proceedings*, volume 1080 of *Lecture Notes in Computer Science*, pages 1–21. Springer, 1996.
- [23] Dimitrios S. Kolovos, Louis M. Rose, Nicholas Drivalos Matragkas, Richard F. Paige, Esther Guerra, Jesús Sánchez Cuadrado, Juan de Lara, István Ráth, Dániel Varró, Massimo Tisi, and Jordi Cabot. A research roadmap towards achieving scalability in model driven engineering. In Davide Di Ruscio, Dimitris S. Kolovos, and Nicholas Matragkas, editors, *Proceedings of the Workshop on Scalability in Model Driven Engineering, Budapest, Hungary, June 17, 2013*, page 2. ACM, 2013.
- [24] Carlene Lebeuf, Margaret-Anne D. Storey, and Alexey Zagalsky. Software bots. *IEEE Softw.*, 35(1):18–23, 2018.
- [25] Andrea De Lucia, Fausto Fasano, Giuseppe Scanniello, and Genny Tortora. Enhancing collaborative synchronous UML modelling with fine-grained versioning of software artefacts. *J. Vis. Lang. Comput.*, 18(5):492–503, 2007.
- [26] lucidchart.com. Der visuelle arbeitsbereich für remote-teams. lucidchart.com, 2020. 2020-09-23.
- [27] Miklós Maróti, Tamás Kecskés, Róbert Kerekényi, Brian Broll, Péter Völgyesi, László Jurácz, Tihamer Levendovszky, and Ákos Lédeczi. Next generation (meta)modeling: Web- and cloud-based collaborative tool infrastructure. In Daniel Balasubramanian, Christophe Jacquet, Pieter Van Gorp, Sahar Kokaly, and Tamás Mészáros, editors, *Proceedings of the 8th Workshop on Multi-Paradigm Modeling co-located with the 17th International Conference on Model Driven Engineering Languages and Systems, MPM@MODELS 2014, Valencia, Spain, September 30, 2014*, volume 1237 of *CEUR Workshop Proceedings*, pages 41–60. CEUR-WS.org, 2014.
- [28] mdeforge.org. Mdeforge software-as-a-service modeling platform. mdeforge.org, 2020. 2020-09-22.
- [29] Frank Mertz. *Kollaborative Modellierung. Aktueller Forschungsstand, Ansätze und Methoden*. München, GRIN Verlag, 2013.
- [30] Ivan Mistrík, André van der Hoek, John Grundy, and Jim Whitehead, editors. *Collaborative Software Engineering*. Springer, 2010.
- [31] Ivan Mistrík, John Grundy, André van der Hoek, and Jim Whitehead. Collaborative software engineering: Challenges and prospects, 2010.

- [32] Sara Pérez-Soler, Esther Guerra, and Juan de Lara. Collaborative modeling and group decision making using chatbots in social networks. IEEE Softw., 35(6):48–54, 2018.
- [33] Sara Pérez-Soler. Modellingbot by saraperezsoler. Github.io, 2020. 2020-09-24.
- [34] QGears. emfcollab. collaborative editing for emf models, 2020. 2020-09-22.
- [35] Ranci Ren, John W. Castro, Adrián Santos, Sara Pérez-Soler, Silvia Teresita Acuña, and Juan de Lara. Collaborative modelling: Chatbots or on-line tools? an experimental study. In Jingyue Li, Letizia Jaccheri, Torgeir Dingsøy, and Ruzana Chitchyan, editors, EASE '20: Evaluation and Assessment in Software Engineering, Trondheim, Norway, April 15-17, 2020, pages 260–269. ACM, 2020.
- [36] Juri Di Rocco, Davide Di Ruscio, Ludovico Iovino, and Alfonso Pierantonio. Collaborative repositories in model-driven engineering. IEEE Softw., 32(3):28–34, 2015.
- [37] Douglas Schuler and Aki Namioka. Participatory design: Principles and practices. CRC Press, 1993.
- [38] Claudia Sinoo, Sylvia van der Pal, Olivier A. Blanson Henkemans, Anouk Keizer, Bert P. B. Bierman, Rosemarijn Looije, and Mark A. Neerincx. Friendship with a robot: Children’s perception of similarity between a robot’s physical and virtual embodiment that supports diabetes self-management. 101:1248–1255, 2018.
- [39] Neil Skrypuch and Nick Boldt. Eclipse modeling framework technology (emft). Eclipse Modeling, 2020. 2020-09-22.
- [40] Steffen Stundzig. Modeling team framework (mtf). Eclipse Proposals, 2020. 2020-09-22.
- [41] Eugene Syriani, Hans Vangheluwe, Raphael Mannadiar, Conner Hansen, Simon Van Mierlo, and Hüseyin Ergin. Atompm: A web-based modeling environment. In Yan Liu, Steffen Zschaler, Benoit Baudry, Sudipto Ghosh, Davide Di Ruscio, Ethan K. Jackson, and Manuel Wimmer, editors, Joint Proceedings of MODELS’13 Invited Talks, Demonstration Session, Poster Session, and ACM Student Research Competition co-located with the 16th International Conference on Model Driven Engineering Languages and Systems (MODELS 2013), Miami, USA, September 29 - October 4, 2013, volume 1115 of CEUR Workshop Proceedings, pages 21–25. CEUR-WS.org, 2013.
- [42] Christian Thum, Michael Schwind, and Martin Schader. SLIM - A lightweight environment for synchronous collaborative modeling. In Andy Schürr and Bran Selic, editors, Model Driven Engineering Languages and Systems, 12th International Conference, MODELS 2009, Denver, CO, USA, October 4-9, 2009. Proceedings, volume 5795 of Lecture Notes in Computer Science, pages 137–151. Springer, 2009.
- [43] unicasel1. Unicasel. case tool. github.io, December 2015. 2020-09-22.
- [44] Karel Vredenburg, Ji-Ye Mao, Paul W. Smith, and Tom Carey. A survey of user-centered design practice. In Dennis R. Wixon, editor, Proceedings of the CHI 2002 Conference on Human Factors in Computing Systems: Changing our World, Changing ourselves, Minneapolis, Minnesota, USA, April 20-25, 2002, pages 471–478. ACM, 2002.
- [45] Jim Whitehead, Ivan Mistrík, John Grundy, and André van der Hoek. Collaborative software engineering: Concepts and techniques. In Ivan Mistrík, André van der Hoek, John Grundy, and Jim Whitehead, editors, Collaborative Software Engineering, pages 1–30. Springer, 2010.
- [46] Dustin Wüest, Norbert Seyff, and Martin Glinz. Flexisketch team: Collaborative sketching and notation creation on the fly. In 2015 IEEE/ACM 37th IEEE International Conference on Software Engineering, volume 2, pages 685–688. IEEE, 2015.
- [47] Nianping Zhu, John C. Grundy, John G. Hosking, Na Liu, Shuping Cao, and Akhil Mehra. Pounamu: A meta-tool for exploratory domain-specific visual language tool development. J. Syst. Softw., 80(8):1390–1407, 2007.