Chapter 1
Introduction

The aim of this book is to provide researchers in the area of automated software composition with (i) a complete and comprehensive guide that helps understand the field and easily relate new approaches to existing ones and (ii) literature recommendations for potentially relevant related work. In this book, term "automated software composition" refers to the process of automatically assembling a new software artifact using existing ones.

Automated software composition has been tackled by many people in one way or the other, and it is hard to keep track of the approaches developed so far and to understand important differences among them. For example, in 2009 two algorithms claiming to tackle "the" service composition problem were published with somewhat contradictory evaluation results [14, 23]. On one hand, Bertoli et al. propose an algorithm technique that needs about 70 s to find a composition out of a repository of 18 services [23]. On the other hand, Bartalos et al. present a mechanism that finds a composition in only 5 ms using a repository of 100.000 services [14]. Clearly, the approaches cannot really address the same task, which rises the question of which exactly are the differences between them. Another example is the different understanding of the composition process itself, which is sometimes interleaved with the execution of services and sometimes not. Understanding the differences and advantages of the different approaches is far from trivial, and judging their suitability or relevance for a particular task is just as hard.

There are already dozens of survey papers [17, 18, 21, 46, 48, 49, 80, 96, 101, 114, 120, 130, 142, 147], but these contain merely neutral paper descriptions instead of helpful discussions. Indeed, some of these surveys are worth being read carefully, because they contain a lot of valuable information. My objection is, however, that the reader does not learn anything about the appropriateness of assumptions made by the described approaches, potential use cases, and their scientific quality (formal soundness, evaluation, etc.). For example, several of the above surveys pose Petri nets as a possible model for services. While one *can* model services with Petri nets, other techniques are much more appropriate (cf. Sect. 4.1.2.3); putting the technique on one level with others is irritating to the reader. *Judging* the approaches, which is the actual challenge, is only ever left to the reader; of course, this is usually impossible without

reading the papers one by one. Also, none of the above surveys can be considered a systematic literature review. That is, the choice of discussed approaches is arbitrary and it is not at all clear why an approach is included or excluded from the overview.

1.1   Contribution and Scope

This book is by far the most exhaustive and systematic review that has been carried out on the field of automated software composition. In order to create this survey, I analyzed many dozens of papers with respect to the concrete problem they tackle and the proposed solutions. This book gives both an overview and a *qualitative comparison* of the approaches.

More precisely, it is a literature review answering three research questions:

1. *Which types of automated software composition problems exist?*
   This question aims at *classifying* the variants of automated software composition problems using the most distinguishing features. It also asks for the *goals and capabilities* inherent to these classes.
2. *Which are the typical use cases where these problems occur?*
   This asks for *situations* in which we would apply the different approaches.
3. *Which are the most prominent solution paradigms for the different types?*
   Here we examine the *solution techniques* used to address the problems.

The first and the second questions are partially answered in Chap. 2. The field of automated software composition can be divided into two areas. Approaches in the first area assume that the behavior of the target software artifact is described by a template that must be instantiated; the main use case is to find an admissible and possibly optimal *refinement of an abstract workflow for an individual context of usage*. Approaches in the second area assume that the behavior is described in terms of logical preconditions and postconditions; the main use case is that we want to *convert a declarative programming statement into imperative code*. Chapter 2 explains why this high-level classification is a good choice and gives answers to the question of use cases for the two classes.

However, the classification system I apply is rather distributed over the three chapters. Chapter 2 explains the two high-level classes, their use cases and differences between them but does not provide a discussion on their respective subclasses. These discussions are part of the introduction of Chap. 3 (for approaches that assume a template given) and Chap. 4 (for approaches that create compositions from scratch) respectively. The reason is that the these classifications are very specific and can be better explained in the respective context. The big picture can be found in Chap. 5; Fig. 5.1 merges these distributed class descriptions into one single classification scheme.

Hence, Chap. 2 should be seen as a general introduction into the field of software composition but without the claim to provide a complete classification framework. The detailed discussion of the two main classes that also contains the answers to the

research questions takes place in Chaps. 3 and 4 respectively. The implied merged classification tree can be found in the conclusion in Chap. 5.

For every approach, there is a detailed discussion and a summarizing evaluation comprising strengths and weaknesses. The detail of discussion depends on several factors such as the novelty, quality of the presentation and the used formal model.

However, the reader may also miss two aspects of discussion:

- *Comparison of performance.* I claim that it is not possible to give a comparison of the performance of composition approaches without a centralized challenge. They cannot be compared merely by the results claimed to have been obtained in the respective papers. However, implementations are often not available, and there is no standardized benchmark set for software composition, yet. Hence, a quantitative comparison of approaches would be desirable but is beyond the scope of this book.
- *Comparison of tool support.* The availability of tools is of tremendous importance for the practical relevance of an approach. However, tools with roots in the scientific community often tend to expire. In fact, some of the approaches discussed here such as OWLS-XPlan once came with tools that are not available anymore or only work on outdated platforms. In order to keep the content of this book independent from changes that tend to occur over time, tool support is not part of the comparative discussion.

In the following, I describe how the approaches discussed in this book were determined. That is, the methodology under which the systematic literature review was carried out.

1.2 Method for Selection of Approaches

This section describes how the approaches discussed in this book were identified. Sect. 1.2.1 describes how a basis of approaches was created, and Sect. 1.2.2 describes how the final set was achieved out of these.

1.2.1 Creating a Basis for Selection

1.2.1.1 Initial Set of Potentially Relevant Publications

First, I created an initial set of publications systematically using the scientific search engines Google Scholar, Citeseer, and Science Direct. The search terms used for this process consisted of two words that must be contained in the title of the publications. The first keyword indicates a composition *activity* and the second keyword indicates a *subject* of composition.

The considered keywords for the activity were: composition, compose, composing, composer, synthesis, synthesize, synthesizing, synthesizer, configuration, configure, configuring, configurator, coordination, coordinate, coordinating, coordinator, orchestration, orchestrate, orchestrating, orchestrator, plan, planning, replanning,

planner, adapter, adapting, adaption, adapt, connector, connect, connecting, connection, mediator, mediating, mediation, mediate, and choreography.

The considered keywords for the subject were: service, services, component, components, software, program, programs, module, modules, operation, operations, workflow, workflows, process, and processes.

I performed a search for each such combination of keywords. A publication is included if, for at least one search term, it contains all words of the search term in its *title*; this resulted in a basic set of 118.530 publications.

1.2.1.2   Removing Topically False Positives and Manually Excluded

Since keywords may be used in different semantic contexts, there are many approaches with titles that seem relevant to the topic of service composition but which are not. So at this step, I removed approaches that are in no way related to the field of service composition, e.g., biological processes, etc.

This removal was done semiautomatically using stoplists with black listed words that clearly indicate an off-topic publication. The blacklisted words are: biol, biodiv, chemi, molec, toxi, amino, diox, silic, medic, lipid, fischer, family, nano, psychiatric, psycho, physio, lympho, human, pharma, pheromone, cataly, oil, child, adult, hydro, thermal, zeolit, liquid, food, milk, nitro, organ, education, kine, fusion, cultur, acetyl, choline, brain, nerve, magnet, spectr, geom, chlor, amphenicol, dna, gluco, stereos, tumor, cancer, infect, protein, lactam, bacillus, depress, gas, fpga, micro, macro, ethyl, ramoplanin, alamethicin, cedrene, cedrol, ferro, peptide, ligand, pyridin, pyrrolo, mannosid, drug, galact, ribosomal, proteolysis, school, hospital, music, channel, nucl, nickel, crystal, heat, lumber, combustion, octanol, fuel, methan, bismuth, sol–gel, mineral, oxi, polyol, morph, cell, liver, surgery, teeth, tooth, bone, carbid, metabolic, membrane, cardiac, halogen, electr, smok, water, drink, weight, jogg, body, life, clinic, genes, condens, ionic, photo, energ, atmosph, synops, distill, ecosystem, deposition, public, student, pupil, classroom, lecture, freshman, statewide, institutional, writers, grade, demogr, transport, rhetoric, enterprises, glyce, soybean, larva, ß, anoid, legal, judicial, justi, logistics, osmo, schedul, supply, volcan, magma, melt, cognitive, teach, facilit, laser, patient, spatial, qfd, Arabidopsis, economic, business, product, resource, group, team, age, aging, robot, and mechanic.

This list seems quite restrictive and to potentially exclude papers that actually do have to do something with software synthesis. While this objection is generally true, we must keep in mind that an approach is only excluded this way if *all* the related papers contain a blacklisted word. I admit this problem, but a manual revision of over 100.000 papers would have simply not been practical. For the next time, one could apply some machine learning classifier in order to carry out a more sophisticated detection of false positives.

In addition, I manually created a second blacklist of roughly 150 irrelevant publications that are not related to the topic of interest. This step removed a huge set of publications; 56.891 remained in the pool.

1.2.1.3 Merging Publications to Approaches

Since many authors publish multiple papers on the same or very similar approach, I merged publications to *approaches*. In this paper, an approach is simply a set of publications, all of which have the same first author. This way, the 56.891 publications were merged to 42.808 approaches. Like for publications, I created a blacklist of 34 approaches, which were associated with 167 publications. Hence, after this step, there were 56.724 publications defining 42.774 approaches under consideration. A complete version of this set, later denoted as M_0 can found at http://felixmohr.eu/research/crc901/survey.

1.2.1.4 Computing the Citation Graph

The huge set of publications makes it impossible to review each of them, so the only viable strategy is to use an evaluable criterion for automated processing. Even though not perfect, a good criterion for filtering is the number of citations made and obtained by approaches with respect to other approaches in the considered set. To this end, I created a citation graph for approaches. In this graph, there is one node for every approach (56.724 nodes) and one edge between node n and n' if any publication of approach n *is cited by* any publication of approach n'. If there is an edge from n and n' as well as from n' to n, I only selected the citation link where the later cites the earlier approach. Even though this is not a sufficient criterion for acyclicity in general, the resulting citation graph is acyclic with 18.438 links.

1.2.2 Determining the Considered Approaches

1.2.2.1 Determine Recent Relevant Work

Removing approaches that do not cite enough others of the area: Considering the huge number of publications, it is reasonable to first outsort approaches that do not relate themselves to other approaches in the field. In particular, I require that every approach cites at least five other approaches in the set; this means, every approach with input degree at least five in the citation graph. Of course, this also eliminates important early approaches that could not cite five other approaches; I reinclude them in the following step. Quite amazingly, this step reduced the number of approaches by 98 % to a rather manageable number of 733. Note that the high ratio of outsorted papers is not only caused by flawed related work of papers but also by the fact that the set still contained many approaches from foreign topics; since those approaches do not cite software composition approaches, they do not achieve the required number of made citations and are eliminated in this step.

Removing non-recent approaches: For now, we are interested in rather *recent* approaches, which I define as approaches from the last 5 years. Hence, from the remaining 733 approaches, I removed approaches older than 2010, which resulted in another 77 % reduction and a total number of remaining approaches of 168.

Removing recent but not brand new approaches without impact: Somewhat moderately, I required approaches from 2010, 2011, and 2012 to have obtained at least 3, 2, and 1 citations respectively. The computational base here is the output degree in the original citation graph. That is, a link from n to approach n' also counts for n even if n' was removed in the last step. Approaches from 2013, 2014 and 2015 are not excluded. Formally,

$$M_1^0 = \left\{ x \in M_0 \middle| \begin{array}{l} -x \text{ cites at least 5 other approaches from } M_0 \text{ and} \\ - x \text{ published in 2010 and has at least 3 citations, or} \\ - x \text{ published in 2011 and has at least 2 citations, or} \\ - x \text{ published in 2012 and has at least 1 citation, or} \\ - x \text{ published in 2013, 2014, or 2015} \end{array} \right\}$$

In our case, this yielded a set with $|M_1^0| = 87$.

1.2.2.2   Determine Very Influential Approaches

Computing most influential approaches. I define the (citation-based) relevance of an approach as a basic (unconditional) value of 1 that is increased by the relevance of approaches that cite it. For the computation of relevance values, we used the formula $f(n) = 1 + \sum_{n'} 5 \cdot \sqrt{f(n')}$, where n' are successors of n in the original citation graph. Using this function f to determine the relevance, I found the intuitively most influential approaches (based on my own research and on the results of other surveys) having the best values in an appropriate order. For the following, I used the 300 approaches with the highest such values. Formally, I define

$$M_2^0 = \{x \in M_0 \mid \text{there are at most 299 other } x' \in M_0 \text{ with } f(x') > f(x)\}$$

1.2.2.3   Reject Approaches that Ignore Very Influential Works

Based on the recent approaches on one hand, and most influential approaches on the other hand, I update the set of recent approaches that do relate themselves to the most influential papers sufficiently. More precisely, I required that a recent approach cites at least 3 of the 200 most influential approaches.

$$M_1^1 = \{x \in M_1^0 \mid x \text{ cites at least 3 elements of } M_2^0\}$$

This step reduced the recent approaches from $|M_0^1| = 87$ to $|M_1^1| = 52$. This tells a lot about the quality of related work of these publications.

1.2.2.4   Determine Somewhat Relevant Approaches

There are a lot of approaches that are important to track the development of an area but that are neither heavily influential nor very recent. In order to include these, I include approaches with at least one citation *obtained* from and five citations *made* on currently considered approaches.

The definition of the set of somewhat relevant approaches is recursive. Let M_3^0 be the label for the set of somewhat relevant approaches and let $M = M_1^1 \cup M_2^0 \cup M_3^0$. Every approach with a publication with at least one citation obtained from approaches in M and 5 citations made on approaches in M is also in M_3^0 (and hence in M). The obtained citation reflects (some) relevance, and the made citations are a necessary condition for reasonable discussion of related work. Formally,

$$M_3^0 = \left\{ x \in M_0 \,\middle|\, \begin{array}{l} -\text{at least one } y \in M_1^1 \cup M_2^0 \cup M_3^0 \text{ cites } x \text{ and} \\ -x \text{ cites five distinct } y_1,\, y_2,\, y_3,\, y_4,\, y_5 \in M_1^1 \cup M_2^0 \cup M_3^0 \end{array} \right\}$$

We obtain a set of size $|M_3^0| = 172$. Note that M_1^1, M_2^0, and M_3^0 are not generally disjoint. http://felixmohr.eu/research/crc901/survey contains an overview of which approach is contained in which of the sets.

1.2.2.5   The Final Set of Considered Approaches

First, not all of the 300 most influential approaches are really relevant for the discourse, so I only consider those approaches that are cited by at least two other approaches in the set. Most influential approaches not satisfying this condition may have been important but not for the actual discourse of the topic of automated software composition. Formally,

$$M_2^1 = \left\{ x \in M_2^0 \,|\, \text{at least 2 other approaches from } M_1^1 \cup M_2^0 \cup M_3^0 \text{ cite } x \right\}$$

Of the initially 300 approaches, only 135 satisfy this criterion.

Second, I update the set of approaches in M_1^1 and M_3^0 with respect to the related work. Due to the incredible amount of approaches in the area of automated service composition, every "non-ancient" approach in this domain must relate itself to (and therefore cite) at least 5 other (relevant) approaches.

Formally, this yields the following final recursively defined set:

$$M = \left\{ x \mid x \in M_2^1 \text{ or } (x \in M_1^1 \cup M_3^0 \text{ and } x \text{ cites at least 5 items of } M) \right\}$$

The final set M contains 211 approaches, which I examined manually.

1.2.2.6   Individual Revision of Remaining Approaches

In a very laborious revision process, I then outsorted another 105 of the 218 approaches. There were three main reasons for being outsorted manually. First, an approach was outsorted if its publications did not contain any concrete composition technique; these were basically surveys and roadmap papers and papers dealing with nonautomated techniques. Second, an approach was outsorted if it does not discuss related work at all (but merely lists other papers) or does not discuss very relevant related work in sufficient detail; the latter was the case when an approach extends an existing one but does not explain the difference. Third, flaws with respect to the content also led to exclusion; the most frequent cases were the lack of a clear contribution statement or unacceptably heavy formal flaws. There is no point in discussing this in more detail within this paper, but I provide a justification for the exclusion of any manually excluded approach elsewhere.

   I acknowledge that this last criterion is, in parts, subjective, but it is still better than previously published surveys. Not only is every survey published so far *completely* based on subjective selection criteria, but these criteria are even nontransparent. The reader has no chance to reconstruct the results and must blindly trust in the quality of research done by the respective authors.