Optimal integration and test planning applied to 
lithographic systems

R. Boumen, I.S.M. de Jong, J.M. van de Mortel-Fronczak and J.E. Rooda, 
Systems Engineering Group, Department of Mechanical Engineering, 

Abstract. In the current integration and test phase of the development of a complex system, 
(the “right side” of the V-model) planning is becoming more and more difficult because of: 1) 
variability in delivery times of components, 2) failing tests and subsequent repairs, 3) resource 
changes and the use of component models, and 4) the growing system complexity and growing 
number of components and tests. Manually created integration and test plans are often not 
optimal regarding time-to-market. Furthermore, creating and maintaining these plans costs a lot 
of effort. In this paper, we introduce a method that allows to automatically create optimal 
integration and test plans. This method can be used by intelligent enterprises to shorten the time-
to-market of a system and to reduce the effort needed to create and maintain integration and test 
plans. We illustrate this method with two cases studies related to the development of ASML 
lithographic machines (ASML 2006).

Introduction

In today’s industry, time-to-market of systems is becoming more and more important. The 
integration and test phase of a complex system typically takes more than 45% of the total 
development time (Engel et al. 2004). Reducing this time shortens the time-to-market of a new 
system.

In the integration and test phase of system development, components are concurrently developed 
and integrated into a subsystem. Subsequently, the subsystems are integrated into a system. In 
between these integration actions, tests are applied to check system requirements. An integration 
plan describes the sequence of integration actions and tests that need to be performed. For new 
ASML lithographic systems, integration and test plans are currently created manually.

An inefficient integration and test plan may cause that faults are found late in the integration and 
test process, because tests are performed late in the process. The rework caused by these faults 
increases the integration and test phase duration. Furthermore, not keeping a plan up to date 
causes an inefficient way of working that increases the duration of the complete phase. A good 
integration and test plan performs tests as early and as parallel as possible such that faults are 
found early in the process. Furthermore, when a plan is kept up to date, it is easier to make the 
correct decisions during the often chaotic integration and test phase. An optimal integration and 
test plan generally does not increase the system quality but increases the efficiency of working

---

1 This work has been carried out as part of the Tangram project (www.esi.nl/tangram) under the 
responsibility of the Embedded Systems Institute Eindhoven, the Netherlands, and in cooperation with 
several academic and industrial partners. This project is partially supported by the Netherlands Ministry of 
Economic Affairs under grant TSIT2026.
such that cost and/or time are minimized. Creating good or even optimal integration and test plans is becoming more and more difficult because of:

- The growing number of components in today’s systems. This results in numerous possible integration and test plans.
- The parallelism in the plan. Subsystems or modules should be tested in parallel as much as possible. Also, component models can be used to perform certain tests before actual components are delivered (see model-based integration by Braspennin et al. 2006)

Maintaining an integration and test plan is also becoming more and more difficult because of:

- The variability in delivery times of components. If a component arrives later than planned, the plan should be updated.
- The variability in test phase duration. Failing tests initiate a repair and diagnosis action and may increase the test phase duration.
- Varying number of components. During integration, it is possible that more components are needed than originally planned, such as software patches that were not included in the original system design.

Due to these difficulties, a method is needed that allows for automatic creation of integration and test plans that are optimal for the time-to-market of a system. This method should also minimize the effort needed to keep a plan up to date. In this paper, we introduce such a method. This method is called the integration and test planning method and consists of the following steps. First, a model is created of the integration and test problem that describes the problem mathematically. Second, an algorithm is used to automatically calculate the optimal integration and test plan. Finally, the plan is executed. A new plan can be calculated automatically after updating the model if a plan update is needed during the execution of the original plan.

The paper is structured as follows. Section two describes the integration and test phases of lithographic machines. Section three describes the proposed integration and test planning method. Section four describes two case studies where this method is applied to the integration and test phases of lithographic machines. The last section gives conclusions.

**Integration and test of lithographic systems**

To explain the integration and test phases of a lithographic machine, we first describe shortly how such a system works. A lithographic machine (or wafer scanner) performs the lithographic step within the manufacturing of a semiconductor or IC. Two items are cycling through a scanner: a wafer that is the basis of the IC and contains a photo resistant coating, and a reticle that contains (a part of) the negative of the image that needs to be placed on the wafer using laser light. The system consists of 7 basic subsystems: a reticle handler that brings and takes reticles to the reticle stage which holds the reticle during the lithographic process, a wafer handler that brings and takes wafers to the wafer stage which holds the wafer during the lithographic process, a laser that produces the light needed for the lithographic process, an illuminator that uniform the light produced by the laser and a lens that shrinks and images the pattern from the reticle on the wafer. Figure 1 shows a picture of an ASML lithographic machine with its main subsystems.
To reduce time-to-market of a new type of lithographic system, often multiple prototypes are created to perform tests in parallel. Normally, each of these prototypes is used for a specific goal; for example, the first prototype is used to test all functional requirements, while the second prototype is used to test all performance requirements. At this moment, for each of these prototypes an integration and test plan is manually created by an integration engineer in Microsoft Project.

The integration and test phase of these systems is characterized by a large time-to-market pressure and a huge number of multidisciplinary components (mechanical, electrical, optical, software) that are developed in parallel and should be integrated. During such an integration phase, first an old type lithographic system is manufactured and qualified. This system is then upgraded to the new type system by replacing certain modules with the new modules, upgrading the software and performing tests to check the system requirements. This approach reduces the risk on possible faults because a complete working machine is taken as starting point. Often, models or ‘dummy’ components are used during the integration phases to perform tests earlier in the integration phases, before the actual modules are delivered.

During the execution of an integration and test plan, the plan often needs to be updated. If a module arrives later than planned, the duration of the module development is updated in the plan. Microsoft (MS) Project then automatically delays all tasks that depend on this development task. However, the sequence of tasks is not changed by MS project, which results in suboptimal plans. Therefore, the sequence of tasks needs to be updated manually which increases the effort to update a plan.
Integration and test planning method

In this section, we introduce the integration and test planning method that allows to automatically create an optimal integration and test plan. The method originates from assembly sequencing methods as described by (de Mello et al. 1991a, 1991b) and object-oriented integration strategies as described by (Hanh et al. 2001). In (Boumen et al. 2007) the assembly sequencing method and the object-oriented integration strategy method are combined into a method to solve integration and test planning problems. The method consists of three steps as shown in Figure 2: define the integration and test model, calculate the integration plan, and execute the plan. During execution it is possible that the model needs to be adjusted (for example because of delays in delivery times) and the plan needs to be updated. In the remainder of this section, we describe each step in more detail.

To calculate an optimal plan for a certain problem, the problem is defined in a mathematical way as an integration and test model. This model consists of:

- a set $M$ of modules,
- for each module in $M$, the associated development duration of that module,
- a set $I$ of interfaces,
- for each interface in $I$, the associated construction duration of that interface,
- for each interface in $I$, the two modules of $M$ associated with it,
- a set $T$ of tests,
- for each test in $T$, the associated duration of performing that test,
- for each test in $T$, its essential sets of modules; that is the sets of modules from $M$ that must be integrated before the test can be performed.

This model needs to be defined by an engineer and contains all information needed to create an integration and test plan. The set $M$ of modules can be obtained by decomposing the system into subsystems or components that are implemented or developed in parallel. Normally, this is already done during the design phase. Furthermore, the set of modules consists of the component models that can be used as replacements for other modules. The development duration of a
module denotes the time between the start of the project and the delivery of the module.

The set $I$ of interfaces between modules denotes how the modules can be integrated with each other. Every interface is created between exactly two modules. If two modules have an interface, they can be integrated with each other. Examples of interfaces are mechanical interfaces such as bolts and screws, but also software interfaces. The construction of an interface may take some time, for example a mechanical interface may take a few hours to construct.

The set $T$ consists of the tests that need to be performed to check system requirements. Each test is performed exactly once. The duration of each test must be denoted on beforehand. The selection of tests that need to be performed is not considered part of this method. In (Boumen et al. 2006) a test selection and sequencing method has been developed that can be used to determine this sequence of tests. In (Boumen et al. 2007) we performed a case study for the integration and test phase of software using a combination of the integration and test planning method introduced in this paper and the test selection and sequencing method. Furthermore, for each test, the essential sets of modules must be defined. An essential module set denotes the minimal set of modules that need to be integrated before that test can be performed. If component models are used to replace certain modules, two essential sets of modules can be created to denote that either the real component or the component model should be integrated before the test may be performed.

The assumptions of the integration and test planning method are:

- All modules in $M$ must be connected with each other, so there exists a path of interfaces that connects every module in $M$ with every other module in $M$.
- For every test in $T$, there exists at least one module that is present in all essential sets of modules belonging to this test, to make sure that each test is performed exactly once.
- Each test is performed exactly once at the moment that one of the essential sets of modules of this test is integrated.
- The durations of the tests and the durations of constructing the interfaces are independent of the current assembly of modules.

We define that an assembly consists of one or more modules that are integrated. An integration action is defined as creating all interfaces between exactly two assemblies sequentially. A test phase consists of the set of tests that are performed on an assembly.

We illustrate the integration and test model with a small example. In this example, all subsystems of a simple lithographic machine must be integrated and tested. In Figure 3, the different modules, interfaces and their development and creation durations (denoted as $t$) are shown. In Table 1, the essential sets of modules per test and the test durations per test are shown.

After the model has been defined, the optimal plan can be calculated. The optimal plan is the plan that integrates all modules into one system and performs each test exactly once in the shortest possible integration and test time. Note that no tests are removed or skipped and that the total test duration is therefore always the same. The optimal plan is the most efficient plan because the tests and integration actions are performed in parallel as much as possible.
<table>
<thead>
<tr>
<th>$T$</th>
<th>Essential sets of modules</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1-T6</td>
<td>reticle handler</td>
<td>1</td>
</tr>
<tr>
<td>T7, T8</td>
<td>reticle handler and stage</td>
<td>2</td>
</tr>
<tr>
<td>T9-T11</td>
<td>wafer stage</td>
<td>1</td>
</tr>
<tr>
<td>T12, T13</td>
<td>wafer handler and stage</td>
<td>2</td>
</tr>
<tr>
<td>T14-T17</td>
<td>lens, laser, illuminator</td>
<td>3</td>
</tr>
<tr>
<td>T18-T20</td>
<td>wafer and reticle stage, lens, laser, illuminator</td>
<td>3</td>
</tr>
<tr>
<td>T21-T25</td>
<td>all modules</td>
<td>5</td>
</tr>
</tbody>
</table>

Table 1: Illustration model

![Diagram](image)

Figure 3. Illustration model

The optimal plan can be calculated using the algorithm as described in (Boumen et al. 2007). The basic idea behind this algorithm is that the plan is constructed according to the ‘assembly by disassembly’ approach using an AND/OR graph search, as was suggested by (de Mello et al. 1991a, 1991b) to create assembly plans. This approach starts with the complete system and investigates all possible ways in which the system can be disassembled into two separate assemblies, which can again be disassembled into two separate assemblies, and so on until the single modules remain.

For the model illustrated in the previous subsection, the optimal solution is shown in Figure 4 as a (reversed) tree. In this tree, the development of a module is shown as a square node, the construction of a set of interfaces is shown as a hexagonal node, the execution of a test phase is shown as an oval node and the sequence of actions is denoted by the edges between the nodes. The critical paths of this plan are the path of the lens and the path of the illuminator that both take 73 time units. The cost of the total integration and test plan is therefore also 73 time units. Another representation of the solution is the Microsoft Project Gantt chart in Figure 5. In this chart, the critical paths of the lens and illuminator modules are depicted in red.
Figure 4. Illustration solution represented as a tree

Figure 5. Illustration solution represented as an MS project Gantt chart
Case studies

This section shows the results of two case studies that were performed during the integration and test phase of the development of two new ASML systems. The first case study shows the optimization of the integration and test plan of a new lithographic prototype and shows a plan update that was performed when the deliveries of certain modules were delayed. The second case study shows the optimization of the integration and test plan of two prototypes of a completely new type of system where some tests must be performed on one specific prototype and other tests may be performed on either the first or the second prototype.

Case study 1. This new lithographic machine is constructed based on an old type system. Only the upgrade of certain modules is considered and not the integration of the old type system. Therefore, the old system is modeled as one assembly (M1) that is completely present at the start of the project. There are 16 other modules (M2 through M17) that are integrated in the old system to upgrade this system to the new system. Modules M10, M11 and M12 are different laser system types. Some tests require one of these modules to be integrated before they can be performed while other tests require one specific laser to be integrated. The complete model of this system is shown in Figure 6 and Table 2. All modules are connected to the old system (M1), while the three lasers (M10, M11, M12) are also connected to M9.

<table>
<thead>
<tr>
<th>T</th>
<th>Essential sets of modules</th>
<th>Time [hour]</th>
</tr>
</thead>
<tbody>
<tr>
<td>T0</td>
<td>M1</td>
<td>96</td>
</tr>
<tr>
<td>T1</td>
<td>M1,M2</td>
<td>165</td>
</tr>
<tr>
<td>T2</td>
<td>M1,M3</td>
<td>68</td>
</tr>
<tr>
<td>T3</td>
<td>M1,M2,M4</td>
<td>5</td>
</tr>
<tr>
<td>T4</td>
<td>M1,M2,M3</td>
<td>278.5</td>
</tr>
<tr>
<td>T5</td>
<td>M1,M2,M3,M9,M10</td>
<td>100</td>
</tr>
<tr>
<td>T6</td>
<td>M1,M2,M3,M13</td>
<td>10</td>
</tr>
<tr>
<td>T7</td>
<td>M1,M2,M3,M14</td>
<td>10</td>
</tr>
<tr>
<td>T8</td>
<td>M1,M2,M3,M15</td>
<td>10</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>T</th>
<th>Essential sets of modules</th>
<th>Time [hour]</th>
</tr>
</thead>
<tbody>
<tr>
<td>T8</td>
<td>M1,M2,M3,M15</td>
<td>10</td>
</tr>
<tr>
<td>T9</td>
<td>M1,M2,M3,M17</td>
<td>29</td>
</tr>
<tr>
<td>T10</td>
<td>M1,M2,M3,M17</td>
<td>6.5</td>
</tr>
<tr>
<td>T11</td>
<td>M1,M2,M3,M16</td>
<td>12</td>
</tr>
<tr>
<td>T12</td>
<td>M1,M2,M3,M6,M9,M11</td>
<td>82</td>
</tr>
<tr>
<td>T13</td>
<td>M1,M2,M3,M5,M6,M8,M9,(M10 or M11 or M12),M13,M14,M15,M16</td>
<td>212</td>
</tr>
<tr>
<td>T14</td>
<td>M1,M2,M3,M6,M9,M12</td>
<td>82</td>
</tr>
<tr>
<td>T15</td>
<td>M1,M2,M3,M9,(M10 or M11 or M12)</td>
<td>10</td>
</tr>
<tr>
<td>T16</td>
<td>M1,M2,M3,M7</td>
<td>120</td>
</tr>
</tbody>
</table>

Table 2: Case study 1 model

Figure 6. Case study 1 model
The integration plan for this model is shown in Figure 7. The total duration of the plan is 1469 hours. The critical path is the path that module M2 follows and is shown in red.

At a certain point in time during the execution of this plan the delivery times of some modules have been changed. In Table 3, the new development durations of these modules are shown. Furthermore, module M15 is removed in the new plan. After a simple update of the model, a new plan has been calculated automatically. This new plan shown in Figure 8 shows the new critical path of module M14 in red. The red line in the figure shows the time at which the plan is updated.

<table>
<thead>
<tr>
<th></th>
<th>Old development duration</th>
<th>New development duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>M7</td>
<td>904</td>
<td>1288</td>
</tr>
<tr>
<td>M8</td>
<td>688</td>
<td>1048</td>
</tr>
<tr>
<td>M11</td>
<td>688</td>
<td>1216</td>
</tr>
<tr>
<td>M12</td>
<td>888</td>
<td>664</td>
</tr>
<tr>
<td>M14</td>
<td>536</td>
<td>1416</td>
</tr>
<tr>
<td>M15</td>
<td>552</td>
<td>Removed</td>
</tr>
</tbody>
</table>

**Table 3: Changed development times for case study 1**

**Case study 2.** In this case study, two prototypes that have been developed in parallel are used to test some specific requirements of a new type of system. These two prototypes have been built from scratch, so no old system type is upgraded. Important detail of the problem is that 80% of the 66 tests are required to be performed on a specific system while 20% of the tests can be performed on either the first or the second prototype. Therefore, the two prototypes cannot be considered separately but have to be considered as one system. This means that both prototypes are defined in one model to create the optimal combined integration and test plan. Afterwards, the individual integration plans for both prototypes can be retrieved from the combined plan. The properties of the combined model are shown in Table 4.

<table>
<thead>
<tr>
<th>Element</th>
<th>Number</th>
<th>Min and max times</th>
</tr>
</thead>
<tbody>
<tr>
<td>Modules</td>
<td>94</td>
<td>0 – 880 hour</td>
</tr>
<tr>
<td>Interfaces</td>
<td>113</td>
<td>0 – 40 hour</td>
</tr>
<tr>
<td>Tests</td>
<td>66</td>
<td>4 – 80 hour</td>
</tr>
</tbody>
</table>

**Table 4: Case study 2 model properties**

The solution to this problem is shown in Figure 9. In this combined plan, the two individual sequences for each prototype can be distinguished. Both prototypes are on the critical path which is denoted in red. The total duration of this plan is 1346 hours. The plan that was created manually by an engineer without using this method takes 1536 hours to perform. This is mainly due to the fact that tests are scheduled less efficiently over the two prototypes compared to the optimal plan. The optimal plan is therefore more than 10% shorter than the plan created manually. Note that we compare two initial plans with each other and not the actual executed plans. These initial plans do not contain the disturbances that may occur during the integration and test phase (although new plans can be created automatically as shown in the previous case study).
Figure 7. Case study 1 solution represented as an MS project Gantt chart

Figure 8. Case study 1 replan solution represented as an MS project Gantt chart
Figure 9. Case study 2 solution represented in an MS project Gantt chart
Conclusions

In this paper, we have introduced a method that allows to create optimal integration and test plans for the integration and test phase during the development of a complex system. This method consists of: 1) defining a model of the problem, 2) creating a plan and 3) executing the plan. Two case studies within the development of new ASML lithographic systems showed the benefits of the method, which are:

- The integration and test plans created with the proposed method are optimal and may therefore be shorter than manually created plans. The second case study shows that the optimal plan is more than 10% shorter than a manually generated plan.

- Planning and re-planning effort can be reduced. The first case study shows that it is very easy to re-plan when certain modules arrive later than planned. The only step that has to be performed is updating the model with the new times. The plan can then be updated automatically. Unfortunately, we cannot give any hard numbers on the actual effort reduction because the method has not been used on a large scale yet.

Another benefit of this method is the actual model. The model can be used as a knowledge container and denotes how the integration and test problem is defined in a very simple and precise way. This makes it easy to review the model with peer engineers.

The planning method does not influence the quality of the system because the selection of tests is not taken into account. This is different in (Boumen et. al 2007) where we used the presented method in combination with a test selection method to determine the optimal integration and test plan.

In this paper, we have shown that the integration and test planning method can be used to optimize an integration and test plan for the development of a new system. However, this method is used to solve other problems, such as the optimization of integration and test plans for (evolutionary) software releases (see Boumen et. al 2007) and the optimization of integration and test plans for the manufacturing of multiple systems. In (Braspenning et al. 2007), the method has been used to determine which component models should be developed to perform tests earlier when the realizations of components are not yet ready.

The presented method can also be used to optimize integration and test plans of complex systems other than lithographic systems. In the case studies we did not use lithographic system specific properties. Although we did not perform actual studies with other types of systems, the method may also be suitable for systems that have integration and test phases where large numbers of parallel developed components should be integrated and where time to market is crucial.

References


**Biography**

**R. Boumen** received his M.Sc. degree in Mechanical Engineering from the Eindhoven University of Technology, the Netherlands, in 2004. During his work as a master student he worked in the field of supervisory machine control of lithographic machines. Since 2004 he has been a Ph.D. student at the Eindhoven University of Technology. His research concerns test strategy within the Tangram project.

**I.S.M. de Jong** has a B.Sc. in Laboratory Informatics and Automation from Breda Polytechnic. He has been a software engineer in various companies in the USA and The Netherlands. Since 1996 he has been working with ASML in systems testing, integration, release, and reliability projects. His specialization is in the field of test strategy. Since 2003 he has been a Ph.D. student of the Eindhoven University of Technology and an active member in the Tangram project.

**J.M. van de Mortel-Fronczak** graduated in computer science at the AGH University of Science and Technology of Cracow, Poland, in 1982. In 1993, she received the Ph.D. degree in computer science from the Eindhoven University of Technology, the Netherlands. Since 1997 she works as an assistant professor at the Department of Mechanical Engineering, Eindhoven University of Technology. Her research interests include specification, design, analysis and verification of supervisory machine control systems.

**J.E. Rooda** received the M.S. degree from Wageningen University of Agriculture Engineering and the Ph.D. degree from Twente University of Technology, The Netherlands. Since 1985 he is Professor of (Manufacturing) Systems Engineering at the Department of Mechanical Engineering of Eindhoven University of Technology, The Netherlands. His research fields of interest are modeling and analysis of manufacturing systems. His interest is especially in control of manufacturing lines and in supervisory control of manufacturing machines.