Optimal integration and test planning for lithographic systems

Chapter 6

Optimal integration and test planning for lithographic systems

Authors: R. Boumen, I.S.M. de Jong, J.M. van de Mortel-Fronczak, J.E. Rooda

6.1 Introduction

In today's industry, time-to-market of systems is becoming increasingly important. The integration and test phase of a complex system typically takes more than 45% of the total development time [41]. Reducing this time shortens the time-to-market of a new system.

In the integration and test phase of system development, components which were concurrently developed are integrated into a subsystem. Subsequently, the subsystems are integrated into a system. In between these integration actions, tests are applied to check the 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 result in finding faults 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 early and in parallel, as much 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 or decrease the system quality but increases the efficiency of working such that cost and/or time are minimized. Creating good or even optimal integration and test plans is getting 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 in Chapter 7).

Maintaining an integration and test plan is also getting 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 diagnosis and repair 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 chapter, we introduce such a method. This method is called the integration and test planning method and consists of the following steps. First, a model of the integration and test problem is created 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 chapter is structured as follows. Section 6.2 describes the integration and test phases of lithographic machines. Section 6.3 describes the proposed integration and test planning method. Section 6.4 describes two case studies where this method has been applied to the integration and test phases of lithographic machines. The last section gives conclusions. This chapter is based on the paper titled 'Optimal integration and test planning applied to lithographic systems' [20] presented at the International Council of Systems Engineering (INCOSE) 2007 Symposium.

6.2 Integration and testing of lithographic systems

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, whereas the second prototype is used to test all performance requirements. Normally, for each of these prototypes an integration and test plan is manually created by an integration engineer using 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 specific modules with the new modules, upgrading the software and performing tests to check the system requirements. This approach reduces the risk of 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 Project then automatically delays all tasks that depend on this development task. However, the sequence of tasks is not changed by Microsoft Project, which may result in suboptimal plans. Therefore, the sequence of tasks needs to be updated manually which increases the effort to update a plan.

6.3 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 [83, 82] and object-oriented integration strategies as described by [55]. In [24] 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 6.1: 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 implementation duration of that module,
- a set I of interfaces that each connect exactly two modules with each other,
- 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,

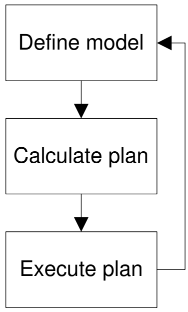


Figure 6.1: Integration and test planning method.

- 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 has already been done during the design phase. Furthermore, the set of modules may consist of the component models that can be used as replacements for other modules, see for more information Chapter 7. The implementation duration of a module denotes the time between the start of the implementation of the module and the end of the implementation 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. We assume that each test needs to be performed exactly once. The duration of each test must be known in advance. The selection of tests that need to be performed is not considered part of this method. In Chapter 5 a test selection and sequencing method has been developed that can be used to determine this sequence of tests. 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.

T	Essential sets of modules	Duration
T1-T6	Reticle handler	1
T7, T8	Reticle handler and stage	2
T9-T11	Wafer stage	1
T12, T13	Wafer handler and stage	2
T14-T17	Lens, laser, illuminator	3
T18-T20	Wafer and reticle stage, lens, laser, illuminator	3
T21-T25	-all modules-	5

Table 6.1: Illustration model.

The assumptions on the integration and test model 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. This also holds for an assembly.
- 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 when one of the essential sets of modules of this test has been 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 have been 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, see Chapter 2, must be integrated and tested. In Figure 6.2, the different modules, interfaces and their development and creation durations (denoted as t) are shown. In Table 6.1, 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.

The optimal plan can be calculated using the algorithm described in [24]. The basic idea behind this algorithm is that the plan is constructed according to the 'assembly by

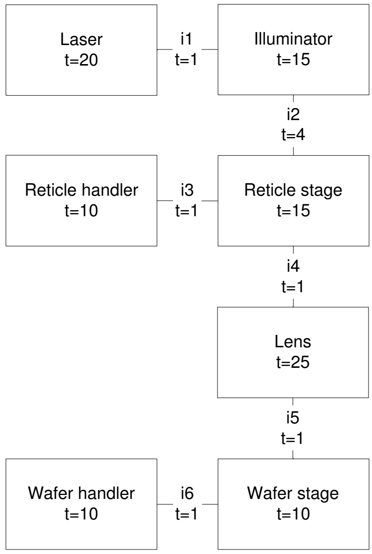


Figure 6.2: Illustration model.

disassembly' approach using an AND/OR graph search, as was suggested by [83, 82] 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 example model, the optimal solution is shown in Figure 6.3 as a 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 6.4. In this chart, the critical paths of the lens and illuminator modules are depicted in light grey.

6.4 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

Figure 6.3: Illustration solution represented as a tree.

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,

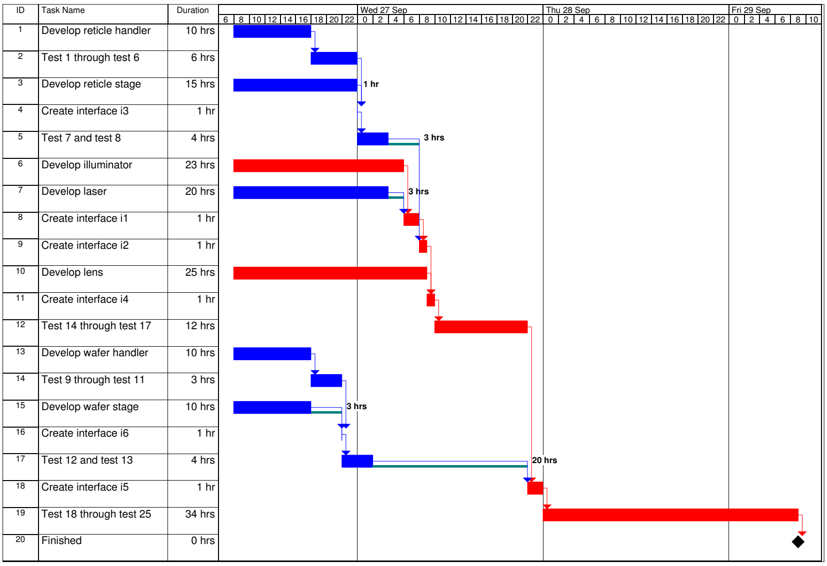


Figure 6.4: Illustration solution represented as an MS project Gantt chart.

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 6.5 and Table 6.2. All modules are connected to the old system (M1), while the three lasers (M10, M11, M12) are also connected to M9.

The integration plan for this model is shown in Figure 6.6(a). The total duration of the plan is 1469 hours. The critical path is the path that module M2 follows and is shown in light grey.

At a certain point in time during the execution of this plan the delivery times of some modules have been changed. In Table 6.3, 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 6.6(b) shows the new critical path of module M14 in light grey. The light grey vertical line in the figure shows the time at which the plan was updated. For this case study we have not made a comparison with a manually created plan.

T	Essential sets of modules	Time [hour]	*T*	Essential sets of modules	Time [hour]
T0	M1	96	T8	M1,M2,M3,M15	10
T1	M1,M2	165	T9	M1,M2,M3,M9	29
T2	M1,M3	68	T10	M1,M2,M3,M17	6.5
T3	M1,M2,M4	5	T11	M1,M2,M3,M16	12
T4	M1,M2,M3	278.5	T12	M1,M2,M3,M6, M9,M11	82
T5	M1,M2,M3,M9, M10	100	T13	M1,M2,M3,M5,M6,M8, M9,(M10 or M11 or M12),M13,M14,M15,M16	212
T6	M1,M2,M3,M13	10	T14	M1,M2,M3,M6,M9,M12	82
T7	M1,M2,M3,M14	10	T15	M1,M2,M3,M9,(M10 or M11 or M12)	10
T8	M1,M2,M3,M15	10	T16	M1,M2,M3,M7	120

Table 6.2: Case study 1 model.

M	Old development duration	New development duration
M7	904	1288
M8	688	1048
M11	688	1216
M12	888	664
M14	536	1416
M15	552	Removed

Table 6.3: Changed development times for case study 1.

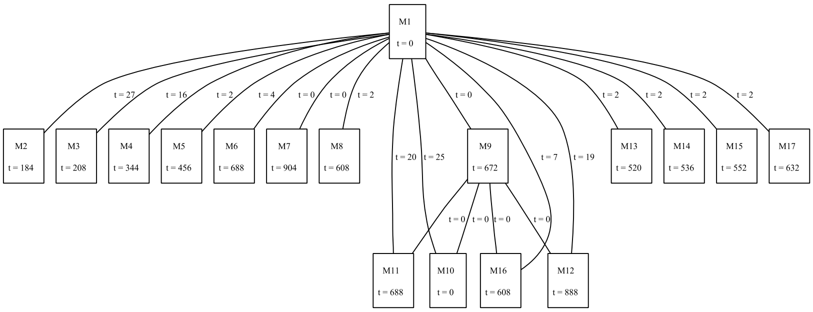


Figure 6.5: Case study 1 model.

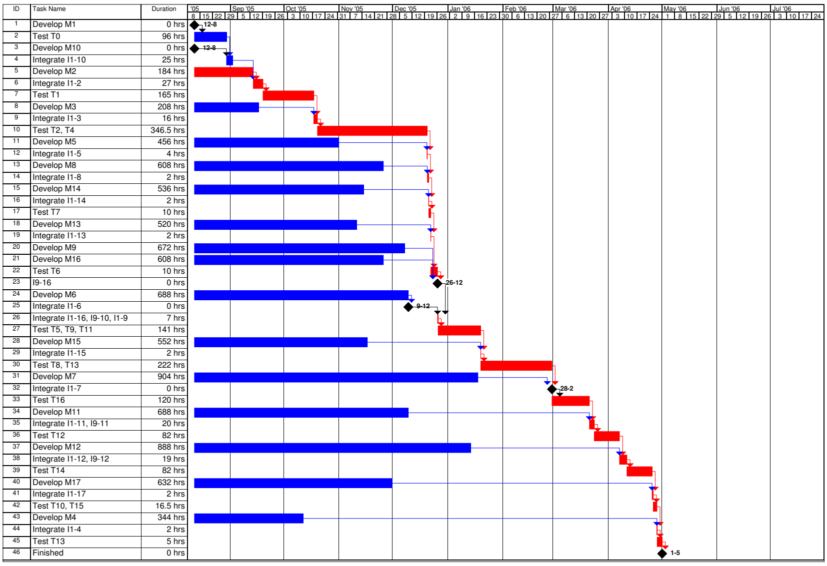
Element	Number	Min and max times
Modules	94	0 to 880 hour
Interfaces	113	0 to 40 hour
Tests	66	4 to 80 hour

Table 6.4: Case study 2 model properties.

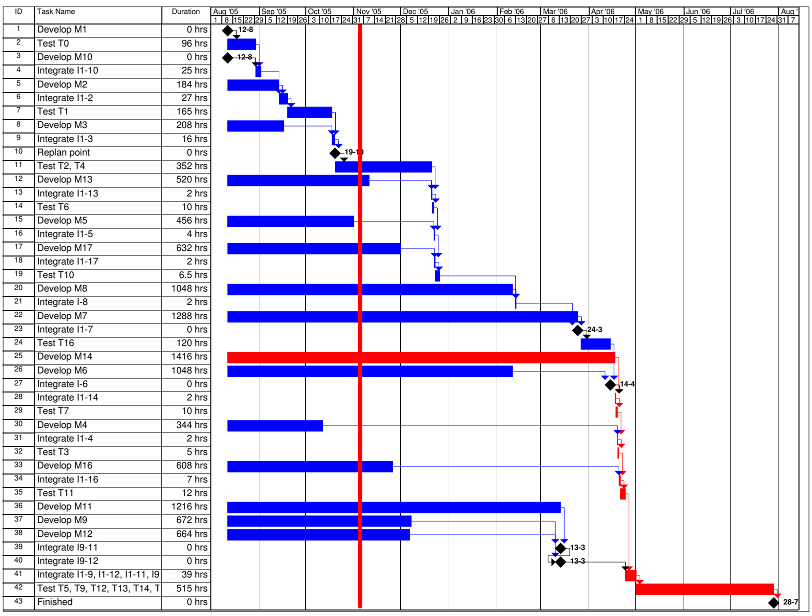
Case study 2

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. An 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 6.4.

The solution to this problem is unfortunately too large to be shown. 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 actually 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).



(a) Case study 1 solution represented as an MS project Gantt chart



(b) Case study 1 replan solution represented as an MS project Gantt chart

Figure 6.6: Case study 1 solutions.

6.5 Conclusions

In this chapter, 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 from [21] where we used the presented method in combination with a test selection method to determine the optimal integration and test plan.

In this chapter, 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 and the optimization of integration and test plans for the manufacturing of multiple systems. Of course, 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 components developed in parallel should be integrated and where time to market is crucial.