

# RFLP – новая культура проектирования в CATIA V6

Дмитрий Крысенков, руководитель службы технической поддержки Dassault Systemes Russia

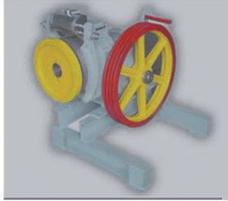
Мы продолжаем цикл статей, рассказывающих о преимуществах платформы V6. Как и было обещано, теперь мы займемся более детальным рассмотрением возможностей CATIA V6 для проектирования высокотехнологичных изделий. Сегодня мы подробно поговорим о новом RFLP-подходе (с базисными принципами которого вы познакомились в *Observer #5/2010*), на примере проектирования подъемного устройства типа лифт.

Первым шагом, является фиксация полученных от заказчика требований, на которые инженеры будут опираться при разработке будущего изделия. Чтобы сделать это, нам необходимо упорядочить имеющуюся текстово-графическую информацию и занести её в базу PLM-системы ENOVIA V6. В нашем случае эту задачу выполняет инженер-системщик, который, используя интернет-браузер (“тонкий клиент”), имеет возможность зайти в защищенное корпоративное пространство системы управления проектами на базе ENOVIA V6 из любой точки земного шара.

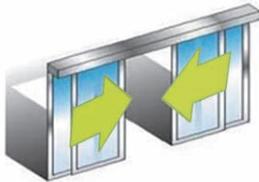
Процесс занесения требования выполняется в специализированном центре – **ENOVIA Requirement Central** – и может проходить как в ручном, так и в автоматизированном (обеспечивается интеграция с решениями MS Office) режимах. Для нашей задачи мы самостоятельно выбираем место хранения информации в соответствии с бизнес-процессом компании и производим декомпозицию требований по различным критериям, например: функциональность, производительность, экономические показатели и т.д. Этот процесс сопровождается вводом пользовательских пояснений.

Таким образом, мы получаем обработанные, структурированные требования, на которые впоследствии будут опираться инженеры разных проектных, технологических, экономических и прочих подразделений. Следует отметить, что исходные данные зачастую подвергаются переработке, что требует внесения изменений в базу PLM-системы. Для этого предусмотрен механизм версионного и конфигурационного контроля в привязке к разрабатываемому изделию, что позволяет избежать

**Constraint about voltage in the building:**  
The building has a specific installation which is **110V DC**.



**Security rule regarding users (floor doors):**  
Prevent users to fall in the elevator hole, doors must be closed when the cabin is not at the floor or when it's moving



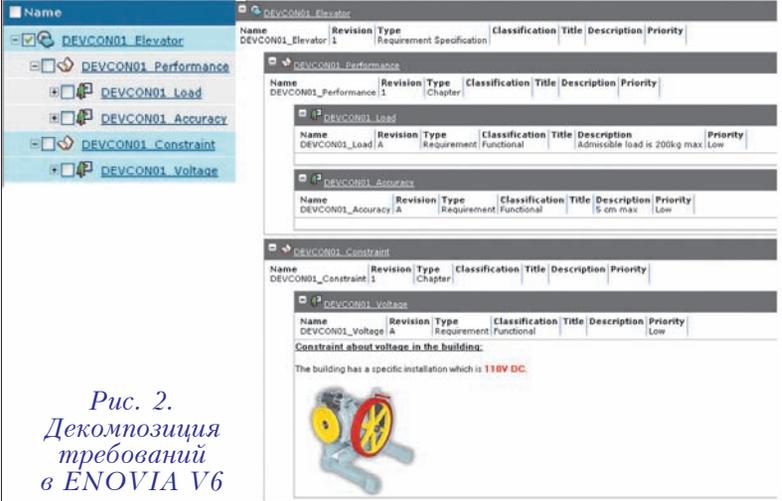
**Security rule regarding gravity effect:**  
In order to avoid any movement down, due to gravity effect, the cabin must be immobilized when the elevator is not moving

Рис. 1. Требования в исходном запросе

дорогостоящих ошибок на стадии планирования и, тем более, производства.

На иллюстрациях представлены исходный запрос заказчика и структурированная информация, занесенная в ENOVIA V6 (рис. 1, 2). Мы разбили требования на две основные группы – требования к лифтовой кабине и к подъемному устройству – и соотнесли их с данными исходного запроса.

Следующим шагом является доведение информации до “рабочей кондиции” и начало работ над



**Figure 2: Decomposition of requirements in ENOVIA V6**

Name	Revision	Type	Classification	Title	Description	Priority
DEVCON01_Elevator	1	Requirement	Specification			
DEVCON01_Performance	1	Requirement	Chapter			
DEVCON01_Load	A	Requirement	Functional		Admissible load is 200kg max	Low
DEVCON01_Accuracy	A	Requirement	Functional		5 cm max	Low
DEVCON01_Constraint	1	Requirement	Chapter			
DEVCON01_Voltage	A	Requirement	Functional			Low

**Constraint about voltage in the building:**  
The building has a specific installation which is **110V DC**.



Рис. 2. Декомпозиция требований в ENOVIA V6

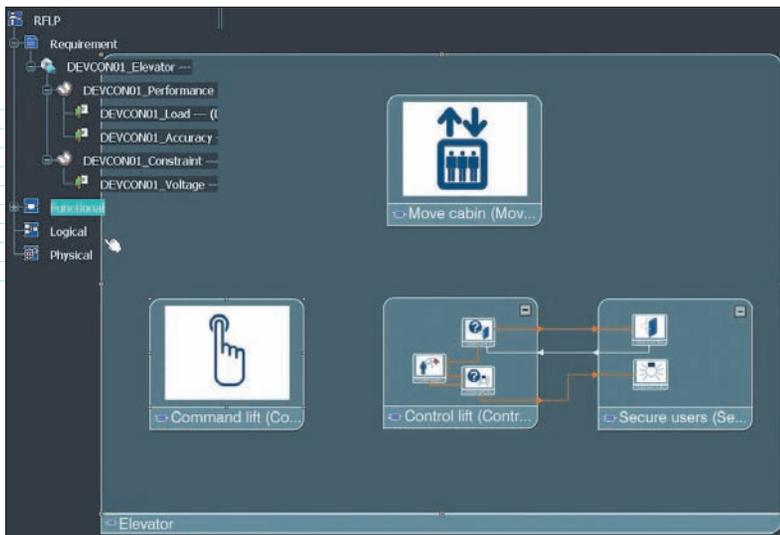


Рис. 3. Функциональная схема. Требования из ENOVIA V6

проектом. Для этого соответствующий инженер из рабочей группы проекта, получив электронное уведомление по электронной почте из *PLM ENOVIA V6*, напрямую открывает пакет требований в программном комплексе *CATIA V6*. Вы скажете, что это невозможно – и ошибетесь. Платформа *V6* позволяет нам сделать это и продолжить работу уже в *CAD*-среде, перейдя на следующий, более высокий уровень описания принципов функционирования проектируемой системы. С помощью модуля **CATIA Systems Architecture Design** мы создаем функциональную схему, внося в нее функции, которые призваны удовлетворить выдвинутые требования. Кроме того, этот модуль позволит нам промоделировать поведение отдельных элементов и системы в целом.

В нашем примере будут фигурировать следующие четыре функции: двигательная, управленческая, контрольная, безопасности. Теперь в каждой функции мы выделим подфункции, чтобы точнее описать работу каждого компонента. Так, контрольная функция охватывает измерение веса поднимаемого груза, закрывание/открывание дверей,

включение сигнального устройства. На соответствующие подфункции раскладывается и функция безопасности. Далее, для завершения функциональной схемы нам предстоит задать связи между подфункциями внутри каждого блока и между блоками – для этого предусмотрен соответствующий инструментарий. К примеру, при входе пассажира в кабину лифта теперь, в соответствии со схемой, происходит измерение веса и включение света (рис. 3).

Но, как вы понимаете, нам еще многое предстоит – например, связать параметр максимальной загрузки кабины с функцией измерения веса поднимаемого груза. Всё это мы благополучно проделываем с помощью данного модуля *CATIA* – то есть, связать параметры отдельных требований и всей функциональной схемы можно в режиме единого окна.

Далее нам предстоит самая трудоемкая описательная часть: подготовить логическую схему работы устройства. Для этого переходим на уровень *Logical* и формируем компонентное представление продукта, создавая соответствующие разным функциям и подфункциям *2D*-объекты (контроллер, электродвигатель с механизмом торможения, кабина, системы крепления и пр.) и, конечно же, связи, отражающие их взаимодействие. Делаем мы это всё в том же модуле, аналогично тому, как и для уровня *Functional* (рис. 4).

В дополнение к логической схеме (так же, как и к функциональной) мы можем задать поведенческую модель работы компонентов в зависимости от устройства каждого отдельного компонента в отдельности.

Выделяют два типа поведенческих моделей: контрольно-логическая и динамическая модели. В качестве иллюстрации опишем поведенческую модель самого сложного компонента логической схемы – контроллера. Для этого создадим контрольно-логическую схему поведения в *2D*-окне редактора модуля, который называется **CATIA Systems Control & Logic Modeling**. Теперь мы можем построить блочную схему, определив входы/выходы, действия и отклики на них со стороны компонента.

Вдобавок к описанному, имеется возможность проводить виртуальное тестирование и отладку работы устройства – например, подав на контроллер сигнал о подъеме кабины на необходимый этаж, проверить, изменится ли её положение, и наоборот (рис. 5).

Логическая поведенческая модель дает обширные возможности для описания принципов работы устройств, но она не способна передать мультифизику процесса, если таковая имеет место быть – то

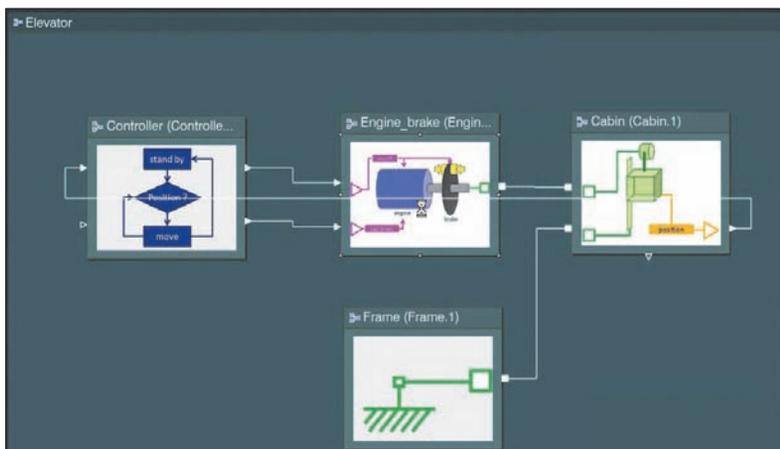


Рис. 4. Логическая схема

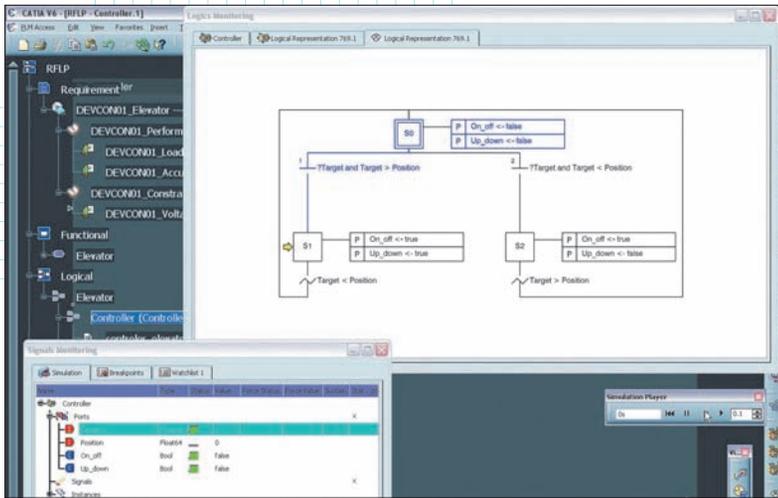


Рис. 5. Логическая поведенческая модель. Виртуальное тестирование

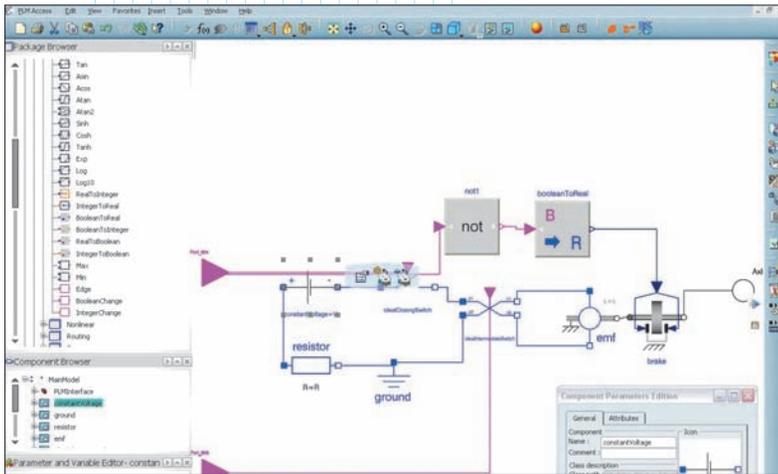


Рис. 6. Динамическая поведенческая модель

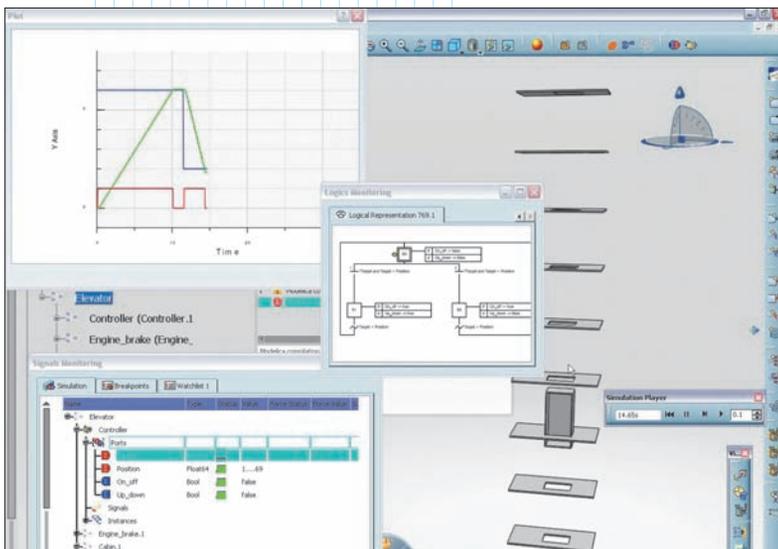


Рис. 7. Общая схема поведения модели с физическим представлением изделия

есть, совместную работу гидравлических, электрических, термодинамических систем. Для этого предусмотрен второй тип поведенческой модели – динамическая модель. Её корни уходят в программный комплекс *Dymola*, который сегодня встроен в *CATIA*. Поведенческая модель в данном случае описывается с помощью блоков, хранящихся в библиотеках различных дисциплин; в основу положен объектно-ориентированный язык *Modelica*.

В нашем примере мы применяем динамическую модель для того, чтобы описать работу электродвигателя с механизмом торможения и позиционирование самой кабины (рис. 6).

После описания логических и динамических поведенческих схем компонентов и отладки логики работы всей структуры, мы можем привязать полученные исходные данные к виртуальной 3D-модели, что является коренным отличием *CATIA V6*.

Вы можете задать вопрос: а зачем нам вообще понадобилось 3D, когда уже и так ясно, как ведет себя модель на логическом уровне? В какой-то мере вы будете правы. Но рассмотренная в нашем примере грузоподъемная система является частью более грандиозного сооружения. Нам необходимо промоделировать варианты столкновений кабины с окружающим объектами – а здесь без визуальной составляющей не обойтись. Поэтому мы можем создать 3D-модель средствами быстрого трехмерного проектирования, не прорисовывая детально, или же импортировать модель (можно взять и детальное представление, если оно имеется).

В нашем примере каждый логический компонент имеет упрощенное 3D-представление, связанное с поведенческой моделью (рис. 7). Далее мы, по сути, проводим то же самое тестирование и отладку, но уже принимая во внимание геометрически описанную физическую оболочку каждого компонента изделия. Причем, даже на этом этапе (который является конечным, с точки зрения системотехника) мы можем изменять требования, логику работы контроллера и остальные входные параметры.

В заключение следует отметить, что платформа *V6* впитала в себя все лучшие решения и подходы современного рынка новейших технологий проектирования в области системной инженерии. Главным успехом, на мой взгляд, можно считать объединение разрозненных направлений в единое целое и успешную их привязку к геометрическому ядру. 😊