

Interview Prof. Dr. Óscar Pastor

c//mag: Könnten Sie unseren Lesern in einem einfachen Satz erklären, was sich hinter „OlivaNova Model Execution“ verbirgt?

Óscar Pastor: OlivaNova Model Execution (ONME) ist ein modellbasiertes Toolset zur Codegenerierung, das aus einem Conceptual Modeler sowie einem Set von Model Compilern besteht, die die Generierung von Modellcode automatisieren. Somit wird das Modell zur wichtigsten Größe im Software-Entwicklungsprozess.

c//mag: Worin bestanden die Ziele bei der Entwicklung von OlivaNova, und wo sehen Sie den Haupteinsatzbereich für dieses Tool?

Óscar Pastor: Viele Softwaresysteme und -anwendungen werden nicht den Anforderungen der Kunden gerecht. Deshalb muss der Software-Entwicklungsprozess aus verschiedenen Perspektiven betrachtet werden. Mit ONME möchten wir zeigen, dass es für die Entwicklung eines Informationssystems nicht nur ausreichend, sondern auch notwendig ist, dessen konzeptuelles Schema zu definieren.

c//mag: Welche Position und Aufgaben übernimmt der Mensch mit XP, und wie hoch sind die Anforderungen?

Óscar Pastor: Bei der Arbeit mit XP ist der Code das Modell. Die Menschen müssen programmieren, das ist ihre grundlegende Aufgabe. Bei der Arbeit mit ONME ist das Modell der Code, und die Menschen müssen modellieren, was den natürlichen kognitiven Fähigkeiten des Menschen weit mehr entspricht. XP bietet uns wertvolle Methoden zur Verbesserung der Softwareentwicklung. Warum also nicht diese Philosophie auf die Entwicklung von Software übertragen und das Modellieren gegenüber dem reinen Programmieren in den Vordergrund rücken?

c//mag: Wie kann ein Software-Ingenieur, der neues Softwareverhalten entwickelt oder bestehendes Softwareverhalten erfasst, durch Tools (statisch und dynamisch) unterstützt werden?

Óscar Pastor: Tools können die Entwicklung von konzeptuellen Modellen vielfach vereinfachen. Diese „Entwürfe“ zukünftiger Softwaresysteme repräsentieren alle im Modell enthaltenen Systemeigenschaften. Mit gut durchdachten Tools, die auf konzeptuellen Modellen basieren, können die verschiedenen Systemansichten korrekt dargestellt, gespeichert und integriert werden, darunter Statik, Dynamik,

Systemfunktionalität und Mensch-Computer-Interaktion. Ein entsprechendes Tool erlaubt die Schaffung einheitlicher Ansichten und deren Darstellung als Gesamt-Systemansicht, die das konzeptuelle Modell repräsentiert.

c//mag: Wie wird die Implementierung von Änderungen innerhalb von OlivaNova unterstützt, d. h. wie werden Nebenerscheinungen von Änderungen an anderen Funktionalitäten bestimmt?

Óscar Pastor: Bei gängigen Entwicklungsverfahren wird die Pflege hauptsächlich auf Codeebene durchgeführt, was eine Änderung des Codes und erneutes Testen bedeutet. Bei OlivaNova erfolgt die Pflege auf Modellebene, was eine Änderung des Modells und eine Umgestaltung der Anwendung auf Basis des Modells bedeutet. Code ist viel komplexer und größer als ein konzeptuelles Modell, daher kann die Pflege auf Modellebene viel einfacher ausgeführt werden.

c//mag: Wie verfährt XP bei Code-Duplizierung, und gibt es vergleichbare Probleme beim MDA-Ansatz (Model Driven Architecture) von OlivaNova?

Óscar Pastor: Der Ansatz von ONME basiert auf der Idee, dass hinter jeder menschlichen Entscheidung ein Konzept steht, das in einem konzeptuellen Modell korrekt dargestellt werden kann. Wenn wir diese konzeptuellen Grundfunktionen erfassen und ihre Gegenstücke in der Software beschreiben, haben wir den Kern eines Model Compilers. Wenn diese Art von „Programmiermaschine“ für das Schreiben des Codes verantwortlich ist, kann man davon ausgehen, dass durch den vom Model Compiler durchgeführten Modell-Transformationsprozess die Code-Replikation minimiert und die Qualität der Software sichergestellt wird.

Aus Valencia in Spanien kommen nicht nur gute Fußballteams, sondern auch der geistige Vater der „**Programmiermaschine**“ OlivaNova, **Professor Dr. Óscar Pastor**. OlivaNova steht für ein neues Verfahren in der Software-Entwicklung, bei dem Modelle automatisch in Programmcode transformiert werden. Der Modell-Compiler soll vor allem die Geschäftsprozesse von Organisationseinheiten abbilden können. Laut Experten könnte OlivaNova (www.programmiermaschine.de) eine neue Ära der Software-Entwicklung einläuten. In der nächsten Ausgabe des c//mag wird Kent Beck, der „Papst“ des „Extreme Programming“ (XP) zu denselben Fragen Stellung nehmen, die das c//mag hier an Professor Dr. Óscar Pastor gestellt hat.

c//mag: Für welche Art von Projekten ist XP geeignet und welche Einschränkungen gibt es in der Praxis?

Óscar Pastor: Unabhängig von der Größe oder Art des Projekts hat die Erfahrung gezeigt, dass die Entwicklung eines Informationssystems auf der Basis von Programmierungsverfahren kostspielig, zu langsam, schwierig zu steuern und äußerst unsicher ist.



Der auf Programmierung basierende Entwicklungsprozess hat sich in den letzten 40 Jahren nicht viel verändert – lohnt es sich da nicht, nach einem besseren Weg zu suchen?

Und dieser bessere Weg ist meiner Meinung nach ein Software-Entwicklungsprozess auf Grundlage eines zentralen konzeptuellen Schemas.

c//mag: Wie bewerten Sie den transformationsbasierten Ansatz von MDA im Vergleich zum inkrementellen Ansatz von XP?

Óscar Pastor: Bei der Nutzung von auf Modelltransformation basierenden Verfahren wird aus dem Modell anschließend das Programm. Die Verbesserungsrate ist sehr beachtlich, denn anstelle eines Softwareprozesses, der auf der traditionellen Reihenfolge Analyse – Entwicklung - Implementierung basiert, reden wir hier nur über die Analyse (konzeptuelles Modellieren). Wir sparen uns die Phasen der Entwicklung und Implementierung, die zudem am kostspieligsten sind. Auf der anderen Seite bewahren wir auch den inkrementellen Aspekt: Der Prozess der konzeptuellen Modell-Erstellung eignet sich hervorragend für einen inkrementellen Ansatz. In diesem Sinne sind transformationsbasiert und inkrementell keine unvereinbaren Begriffe.

c//mag: Können Sie sich einen gemeinsamen Entwicklungsprozess von OlivaNova und XP vorstellen, der ein Testen des Softwareverhaltens bereits auf Modellebene erlaubt?

Óscar Pastor: Ja. In der Tat können einige bekannte Verfahren von XP mit Blick auf die endgültige Produkterstellung auf den Erstellungsprozess eines konzeptuellen Modells angewendet werden. In diesem Fall sprechen wir von Extreme Conceptual Modeling (XCM!).

c//mag: Wenn Sie OlivaNova mit einem kleinen Entwicklerteam in einen XP-Prozess implementieren möchten, welche Vorteile kommen Ihnen da als Erstes in den Sinn?

Óscar Pastor: Die Schaffung eines Softwareprodukts direkt von einem konzeptuellen Schema ausgehend ist immer effizienter als mithilfe eines herkömmlichen Software-Erstellungsprozesses, basierend auf manueller Programmierung. In jedem Fall können diese erwiesenermaßen effektiven XP-Verfahren erfolgreich für die konzeptuelle Modellerstellung angewendet werden. Ein weiterer Vorteil bezieht sich auf die Handhabung der Komplexität. Da sich konzeptuelle Modelle durch eine höhere Abstraktionsebene auszeichnen als Modelle mit Quellcode, können Entwickler mithilfe des modellbasierten Ansatzes komplexere und größere Probleme bewältigen. Dieses Phänomen trat mit den ersten Compilern auf: Die Entwickler waren produktiver, weil sie auf einer höheren Abstraktionsebene arbeiten konnten.

c//mag: Welche Entwicklungen oder Konzepte im Bereich der Softwareentwicklung halten Sie mittelfristig für viel versprechend?

Óscar Pastor: Aus der Sicht der Modelltransformation sind die Anforderungen die nächste Herausforderung, die es zu bewältigen gilt. Wenn wir in der Lage sind, ein konzeptuelles Schema in ein entsprechendes endgültiges Softwareprodukt zu konvertieren, warum dann nicht dasselbe Muster anwenden, um von den Anforderungen zum konzeptuellen Modell zu gelangen? Die Transformation des konzeptuellen Modells in den Endcode reicht uns nicht aus... Wir möchten unseren Softwareprozess bereichern, indem wir einen vollständigen Anwendungsbereich zur Verfügung stellen, angefangen bei einem Anforderungsmodell bis hin zur Transformation dieses Modells in ein konzeptuelles Schema, aus dem wir mit ONME die Endanwendung entwickeln könnten. Unsere nächste Herausforderung ist die Transformation der Anforderungen in Code, und zwar mit dem höchstmöglichen Automatisierungsgrad. —

Die Fragen wurden von Rodger Burmeister und Jan Wloka zusammengestellt, wissenschaftliche Mitarbeiter am Fraunhofer Institut für Rechnerarchitektur und Softwaretechnik (FIRST) in Berlin.