© GrammaTech

ZUSAMMENFÜHRUNG VON MISRA C++ UND AUTOSAR C++

Aus zwei mach eins

Die Industrieorganisationen MISRA und AUTOSAR planen ihre jeweiligen Standards zur Software-Entwicklung in C++ zu einem gemeinsamen Werk zu verschmelzen. Davon sollen alle profitieren: Entwickler, Automobilhersteller und nicht zuletzt Tool-Anbieter.

Für die Software-Entwicklung in der Automobilbranche geben zwei Organisationen Regeln aus, wenn es um Embedded-Systeme im sicherheitskritischen Bereich geht: MISRA (Motor Industry Software Reliability Association) und AUTOSAR. Beide verfolgen ein ähnliches Ziel, nämlich die Software in Fahrzeugen besser zu machen.

Für MISRA stehen Zuverlässigkeit und Sicherheit im primären Fokus, für AUTOSAR eine offene und standardisierte Architektur. Beide geben Regelwerke heraus, die Standards definieren, die bei der Entwicklung von Software für Automotive-Systeme eingehalten werden sollten. Für die Entwicklung mit C++ sind MISRA C++ 2008 und AUTOSAR C++ 14 unverzichtbare Hilfestellungen. Das betrifft den größten Teil der Entwickler im Embedded-Umfeld: Laut dem Datenanbieter Statista nutzen 82 Prozent aller Embedded-Systems-Anbieter C++ zumindest als eine von mehreren Sprachen in ihren Projekten. C++ ist weit verbreitet, mit steigender Tendenz.

Beide Richtlinien sind eng verwandt, denn AUTOSAR C++ wurde in weiten Teilen aus MISRA C++ abgeleitet. Zudem flossen wichtige Aspekte anderer Entwicklungsvorgaben mit ein, darunter etwa SEI CERT C++, C++ Core Guidelines, HCI++ oder JSF. Für Entwickler und auch für die Hersteller von Entwickler-Tools bedeutet das allerding nicht zuletzt, dass zwei Regelsätze gepflegt und unterstützt werden müssen, die sich in Zielrichtung und Ausgestaltung sehr ähnlich sind. So ist es durchaus konsequent, dass das MISRA-Konsortium jüngst ankündigte, AUTOSAR C++ in MISRA C++ zu integrieren und so einen gemeinsamen Standard zu schaffen. MISRA möchte damit eine einheitliche Richtlinie für die Entwicklung von sicherheitskritischen Anwendungen mit C++ schaffen. Und auch den eigenen, 2008 zuletzt aktualisierten Standard auf den Stand von C++ 17 – und in der Folge C++ 20 – bringen.

Einheitlicher Branchenstandard

Für Entwickler und Embedded-Systems-Anbieter ist das eine gute Nachricht. Durch das Zusammenlegen der Standards unterschiedlicher Branchengruppierungen wird vor allem die Zertifizierung der sicherheitsrelevanten und sicherheitskritischen ECUs signifikant einfacher, da nur noch die Compliance zu einem Rahmenwerk nachgewiesen werden muss. Die bislang unvermeidlichen doppelten Aufwände können entfallen.
Und auch für die Anbieter von Entwickler-Tools wie **GrammaTech** ist diese Nachricht sehr positiv. Auch hier müssen künftige Neuerungen und Produkte

nur noch auf Grundlage einer einheitlichen Richtlinie entworfen und validiert werden. Zudem verspricht der neue gemeinsame Standard eine noch bessere Konsistenz, da viele Jahre Praxiserfahrung mit dem Einsatz der Richtlinien in der täglichen Entwicklungsarbeit einfließen können. Besonders im Bereich der statischen Code-Analyse können die Einflüsse von AUTOSAR sehr hilfreich sein: AUTOSAR C++ 14 spezifiziert recht genau, welche Regeln durch Automatisierung überwacht werden sollen. Die Regeln werden in „automatisiert", „teilweise automatisiert" und „nicht automatisiert" klassifiziert. AUTOSAR fordert hier explizit: Die meisten Regeln können automatisch durch statische Analyse durchgesetzt werden. Ein Tool zur statischen Code-Analyse, dass die vollständige Übereinstimmung mit diesem Standard für sich reklamiert, muss alle automatisch durchsetzbaren Regeln vollständig überprüfen können. Zudem muss es die Einhaltung der teilweise automatisierbaren Regeln soweit prüfen, wie es möglich oder sinnvoll ist.

Klarheit bei der Konformität

Übernimmt die aktualisierte Version von MISRA C++ diesen Ansatz, ist klar, welcher Regelsatz von den Tool-Herstellern entwickelt und getestet werden muss. Ob ein Werkzeug konform zu der Richtlinie ist, ist dann sowohl für den Anbieter als auch für die Kunden einfach nachzuvollziehen. Denn die Regeln selbst geben unmissverständlich vor, ob eine automatische Überprüfung im Rahmen der statischen Analyse erwartet wird oder nicht. Die Kunden können sich also darauf verlassen, dass ein standardkonformes Werkzeug mindestens die vollständig automatisch prüfbaren Regeln umfänglich beherrscht. Und die Tool-Anbieter können sich am Markt voneinander durch die Überprüfung der teilweise automatisierbaren Regeln voneinander abgrenzen.

Für die Entwickler heißt das: Sie können sich auf die statische Code-Analyse als einen Grundpfeiler der Qualitätssicherung verlassen und die Compliance zum Standard automatisieren. Nicht nur als Audit-Funktion am Ende des Projekts. Sondern als fortlaufenden Prozess in allen Phasen des Software Development Lifecycles. Das betrifft zum einen den Arbeitsplatz der Entwickler, die ihren Beitrag noch vor der Integration in die Mainline des Build-Systems auf Standardkonformität hin überprüfen können. Und andererseits das Build-System, auf dem tiefer gehende Analysen des gesamten Codes automatisch vor jedem Build durchgeführt werden.

Die Integration von AUTOSAR C++ 14 in MISRA C++ 2008 kann so einen wichtigen Beitrag dazu leisten, sicheren und zuverlässigen Code mit weniger Aufwand und Kosten zu erhalten. ■ (oe)

www.grammatech.com

INFO

CodeSonar 5.2

CodeSonar 5.2 von GrammaTech ermöglicht es, die statische Code-Analyse mit einem einheitlichen Tool sowohl bei Embedded-Systemen als auch bei Enterprise-Anwendungen durchzuführen. Unterstützt wird nun auch AUTOSAR C++ 14. Der Support von Compilern und offenen Standards wurde verbessert. So arbeitet CodeSonar 5.2 nun mit den aktuellsten Versionen der Compiler JAR, GNU C und CLANG zusammen. Die Neuerungen der Standards C 17/C++ 17 und C++ 20 wurden auch integriert. Damit beherrscht CodeSonar nun C und C++ von den älteren bis zu den neuesten Sprachmerkmalen. Darüber hinaus wurde CodeSonar for Binaries weiterentwickelt. Neben der bereits vorhandenen Unterstützung für x68- und ARM-Architekturen können nun auch Binärdateien für Power-Architekturen analysiert werden.

Dr. Paul Anderson leitet die Entwicklungsabteilung bei GrammaTech und gehört in dieser Funktion dem MISRA-Komitee an.