Proprietà intellettuale del software

Proprietà intellettuale nel mondo dell'automazione significa strategie e metodologie per preservare il 'know how' di chi realizza software per i sistemi di controllo

Nel campo dell'automazione si pone spesso la problematica della protezione della proprietà intellettuale, soprattutto per quanto riguarda la realizzazione del software di controllo. Il presente articolo analizza questo aspetto, dando particolare risalto al lato tecnico della protezione e lasciando ad altra sede l'analisi dell'aspetto legale. Gli integratori di sistema, o anche le aziende che utilizzano sistemi di automazione, implementano spesso all'interno dei loro controllori degli algoritmi di controllo derivanti da un'esperienza nel loro settore o da una specifica competenza che non intendono fornire a terzi liberamente. Questo, infatti, implicherebbe la perdita di un vantaggio competitivo frutto di lavoro e dispendio non indifferente di risorse. La strategia di protezione è quindi di vitale importanza ed è legata al tipo di hardware su cui si intende operare: essa, infatti, si differenzia notevolmente se si tratta di un software per PLC o di un software per PC.

Software per PLC

Nel caso di un PLC si riscontrano poi notevoli differenze se si utilizzi un hardware che lavora su un codice oggetto compilato oppure su un codice interpretato, intendendo per codice oggetto compilato un codice interpretabile direttamente dalla macchina e derivato dall'elaborazione di un compilatore del codice sorgente scritto dal programmatore. Con codice interpretato, invece, si intende un sorgente scritto dal programmatore in un apposito linguaggio direttamente elaborato dalla macchina tramite un interprete che lavora a tempo di esecuzione e che non necessita di precompilazioni.

L'utilizzo di PLC che lavorano su codice oggetto preserva in maniera consistente la proprietà intellettuale del codice, dato che, una volta recuperato dalla memoria di esecuzione del controllore, richiede un'operazione di 'retro engineering' abbastanza pesante per risalire a un listato interpretabile da terzi. Inoltre, in ogni caso, questi controllori hardware mettono a disposizione del programmatore una protezione tramite password per le operazioni di upload del codice dal PLC, in maniera tale da bloccare a terzi la possibilità di rileggere il codice dalla macchina funzionante in campo.

Nel caso si operi, invece, con un codice interpretato, ad esempio AWL di Siemens, il problema si pone in maniera pesante, dato che una rilettura da parte di terzi consente una ricostruzione relativamente semplice della logica realizzata, anche in assenza di qualsiasi commento del programmatore che, di norma, non sono scaricati nel controllore. In questi casi la protezione tramite pass-word fornita dal PLC è l'unico strumento disponibile per impedire la lettura del programma. Prendendo come esempio le CPU S7 di Siemens, tale operazione viene eseguita definendo il livello di protezione e la password nel riquadro di parametrizzazione dell'unità, come visibile in figura 1, e ricaricando la parametrizzazione modificata nell'unità stessa. Da questo momento in poi la protezione è attiva e qualsiasi attività eseguita sulla CPU richiede il livello di pass-word impostato per poter operare; nel caso non si possegga la password, l'unica modalità per poter riprogrammare il PLC è cancellare il programma attualmente in esecuzione, con le ovvie conseguenze.

Spesso questo tipo di protezione si scontra però con le esigenze dell'utilizzatore finale, specialmente nel caso di integratori di sistema; l'utilizzatore, infatti, chiede che gli venga lasciata la possibilità di agire sul codice di controllo, magari per modificare delle operazioni logiche semplici o legate a eventuali riconfigurazione del sistema, operazioni eseguite ad esempio dai manutentori dell'impianto. In questo caso, l'integratore può fornire al cliente il programma libero nel suo corpo principale e proteggere il proprio 'know how' incapsulando il codice in blocchi funzione o funzioni coperti da opportuna password. I linguaggi di programmazione conformi alla specifica IEC 1131 3, relativa ai linguaggi di programmazione in ambito industriale, consentono anche la protezione di intere librerie di blocchi funzione e funzioni, lasciando all'utilizzatore la possibilità di vedere unicamente le interfacce di scambio dei dati con le quali deve interagire per poter utilizzarli nel suo codice. In figura 2 è visibile, a titolo esemplificativo, l'ambiente Codesys che permette addirittura di definire fino a sette livelli di protezione, ciascuno con diversi diritti, ovvero 'nessun diritto', 'lettura' e

‘lettura/scrittura’, sull’oggetto. Questa politica di protezione segmentata è lo strumento ideale per soddisfare le diverse esigenze del programmatore e dell’utilizzatore.

Software per PC

La scrittura di programmi su PC consente diverse strategie di protezione, che si differenziano a seconda dello scopo per cui il software viene fornito. Nel caso di HMI o Scada, la protezione è legata agli strumenti messi a disposizione da parte del fornitore dell’ambiente di sviluppo e ricalcano in qualche modo quanto già visto relativamente ai PLC. Se invece si tratta di applicazioni sviluppate con ambienti standard, ad esempio Microsoft .NET, il range di possibili soluzioni si amplia: oltre alla fornitura di eseguibili in codice oggetto, è possibile fornire sorgenti con una parte delle funzionalità che si vogliono proteggere inglobate in librerie compilate.
In questo caso, l’utilizzatore può usare le diverse funzioni messe a disposizione dalla libreria, ma non è in grado di risalire al codice sorgente realizzato da chi ha implementato la libreria. Per potersi proteggere da azioni più mirate, quali, ad esempio, la decompilazione dei sorgenti da parte di appositi ‘tool’, è possibile anche ricorrere a software che modificano il codice oggetto ristrutturandolo secondo algoritmi che ne complicano l’interpretazione pur mantenendone il funzionamento invariato.

Lo stato dell’arte

Da quanto visto emerge chiaramente che la protezione intellettuale nell’ambito dell’automazione rispecchia sostanzialmente le tecniche relative ad altri ambiti legati all’utilizzo del software. L’unica differenza è la forte dipendenza dagli ambienti di sviluppo forniti dai produttori dell’hardware e a questi vanno riportate, da parte degli utilizzatori, richieste e osservazioni per un’implementazione sempre migliore e vicina alle esigenze dei fruitori.

Di Luca Marani