

PRIMENA INDUSTRIJSKOG STANDARDA ISA-95**APPLICATION OF INDUSTRY STANDARD ISA-95**Nenad Zelenović, *Fakultet tehničkih nauka, Novi Sad***Oblast – ELEKTROTEHNIKA I RAČUNARSTVO**

Kratak sadržaj – U ovom radu predstavljeno je programsko rešenje koje se bavi komandovanjem nad opremom u polju pomoću simulatora ili preko klijentske aplikacije, primenjujući principe i smernice ISA-95 standarda.

Ključne reči: ISA-95 standard, SCADA komponente, komunikacija

Abstract – This paper presents an application that deals with the execution of the sets of command for the equipment in the field using a simulator or a client application and applying the principles and guidelines of the ISA-95 standard.

Keywords: ISA-95 standard, SCADA components, communication

1. UVOD

Za vođenje uspešne kompanije potrebno je obezbediti odgovarajući sistem kojim je moguće deliti informacije kroz slojeve i delove kompanije, a da sistem bude pouzdan i sinhronizovan. Takođe te informacije moraju biti razumljive i čitljive kroz sve delove kompanije, od pogona i kontrolnih soba do kancelarija gde se razvijaju biznis planovi. Vrlo često, zaposleni iz proizvodnih i poslovnih slojeva u međusobnoj komunikaciji koristili različite nazive za iste pojmove.

U poslednjih par decenija, industrijske kompanije su investirale u sisteme poslovne logistike (ERP - *Enterprise Resource Planning*). Takođe, veliki deo novca i vremena je potrošen na investiranje u automatizaciju industrijskih pogona pomoću upravljačkih sistema kao što je SCADA (*Supervisory Control And Data Acquisition*). Zbog toga se ukazala potreba približavanja i uvezivanja ova dva sistema u jednu celinu. ERP sistemi su najkorisniji kada se hrane trenutnim i validnim informacijama, onim koje se po pravilu nalaze u kontrolnim sistemima [1].

U projekte koji su usmereni na integraciji između ERP i upravljačkih sistema, obično je uključen veliki broj odseka, kao što su proizvodnja, održavanje, laboratorija i kancelarija. Navedeno ukazuje da su preduzeća suočena sa velikom prazninom između poslovnih i kontrolnih sistema. Stoga je bilo potrebno da se razvije standard koji će uvesti odgovarajuće principe, terminologiju i modele za uspešno poslovanje kompanije, kao i mogućnosti integrisanja slojeva u jednu sinhronizovanu celinu. Kao rešenje tog problema uvodi se ISA-95 standard.

NAPOMENA:

Ovaj rad proistekao je iz master rada čiji mentor je bio dr Branislav Atlagić, docent.

Predmet rada jeste istraživanje koncepta koji je izložen u standardu ISA-95, kao i implementacija jedne sprege ovog tipa na primeru školskog akviziciono upravljačkog sistema. Iako daleko od kompleksnosti realnih industrijskih sistema, ovaj rad treba primerom da ukaže na ključne komponente koje su neophodne pri sprezanju poslovnih i proizvodnih segmenata neke kompanije. Naravno, kako je poslovni segment podržan ERP softverom, a proizvodni svojim SCADA sistemom, u radu su u rudimentarnoj formi realizovane obe komponente, kao i sprema između njih u duhu standarda ISA-95.

2. TEORIJSKE OSNOVE**2.1 Istorija ISA-95 standarda**

ISA-95 predstavlja internacionalni standard koji služi za integrisanje biznis i kontrolnih sistema u cilju smanjenja rizika, troškova i stvaranja grešaka koje idu paralelno sa implementacijom interfejsa između takvih sistema. ISA je globalna neprofitna organizacija. Prvobitno ISA je označavala *Instrument Society of America*, ali je ovaj naziv kasnije preimenovan u *Instrumentation, Systems and Automation Society*. Ovaj naziv je promenjen 2000 god. ISA definiše svoje ključne aktivnosti kao što su: standardizacija, sertifikacija, obrazovanje i obuka, publikacije i izložbe iz oblasti industrijske automatizacije [2].

2.2 Svrha ISA-95 standarda

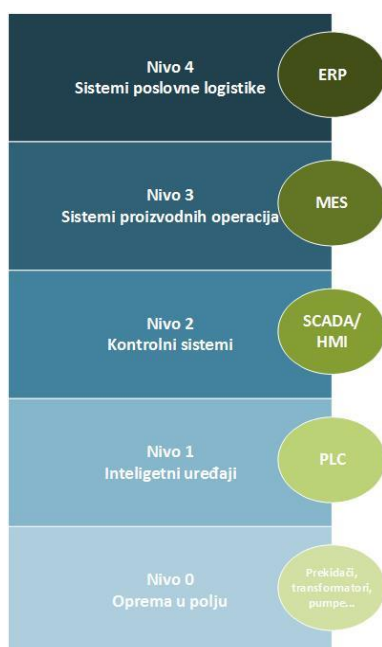
Komuniciranje o sistemu može biti teško, jer različiti ljudi u isti razgovor često koriste različita imena opštih termina. ISA-95 definiše reči koje se odnose na sisteme poslovne logistike (ERP) i na kontrolne sisteme. ISA-95 stavlja ovu terminologiju u modele koji jasno pokazuju vezu između različitih pojmova. Ovaj princip možemo uporediti sa nacrtima za kuću. Reči *prozor*, *vrata*, *zid* i *krov* su svima nama poznati i koristimo ih za međusobno razgovaranje kada se spominje kuća. Svaka kuća je drugačija, ali i dalje možemo opisati svaku kuću sa istim simbolima i rečima za vrata, krovove, zidove i prozore. Isto se odnosi i na ISA-95. Ne postoje dve slične kompanije i ipak možemo koristiti ISA-95 modele i terminologiju za razgovor sa drugim kompanijama o aktivnostima, funkcijama, tokovima informacija unutar tih kompanija... Kao rezultat, postalo je lakše ne samo na nivou ljudske komunikacije, već i na tehničkom nivou, za integraciju različitih sistema.

Cilj ISA-95 standarda je da smanji troškove, rizike i greške povezane sa implementacijom interfejsa između ERP i kontrolnih sistema. Standard se može koristiti za pojednostavljenje implementacije novih softverskih proizvoda i da na kraju se stvori laka interoperabilnost između ERP i kontrolnih sistema.

2.3 Nivoi ISA-95 standarda

ISA-95 standard deli postojenja, opremu i imovinu na 5 nivoa [3]:

- **Nivo 0 (multi)** – predstavlja samu opremu u polju (prekidači, transformatori, pumpe...)
- **Nivo 1** – predstavlja opremu koja očitava stanja sa polja tj iz nultog nivoa, kao i manipulaciju nad njima.
- **Nivo 2** – omogućava monitoring, nadgledanje, manuelnu i automatsku kontrolu nad opremom u polju. Ovaj nivo sadrži kompletan uvid u infrastrukturu celog sistema, kao i uvid u svaki vid promene vrednosti opreme u polju.
- **Nivo 3** – ovaj nivo definiše aktivnosti radnog procesa za proizvodnju željenih krajnjih proizvoda, sadrži uvid u detaljan raspored proizvodnje, kao i kompletne istorijske podatke.
- **Nivo 4** – u ovom nivou se obavljaju funkcije kao što su operacioni menadžment, planiranje proizvodnje i logistika. Utvrđuju se dugoročni, srednjoročni ili kratkoročni planovi koji će doneti profit.



Slika 1 - Nivoi ISA-95 standarda

2.4 Razlike između sistema proizvodnih operacija i poslovne logistike

2.4.1 Fokus i svrha

ERP funkcioniše kao sredstvo za razmenu informacija unutar organizacije [4]. Sistem povezuje svaki deo posla i omogućava nesmetan protok informacija. Delujući kao sveobuhvatni put podataka čini ERP dragocenim alatom za upravljanjem, jer donosi odluka daje mogućnost da duboko zarone u svaki deo poslovanja i na smislene načine povežu ranije udaljene podatke.

Razlog za dodavanje MES-a jeste, s druge strane, pružanje sredstava za preciznu kontrolu proizvodnog procesa. MES preduzima korake ili generiše izveštaje na osnovu onoga što se trenutno događa kako bi nagledale promenljive koje utiču na efikasnost proizvodnje.

2.4.2 Sredstva za prikupljanje podataka

Ljudi obično većinu informacija prosleđuju u ERP. Ovo ručno prikupljanje podataka dobro funkcioniše, jer se sistem prvenstveno bavi prikupljanjem, organizovanjem i razmenjivanjem informacija širom organizacije radi planiranja i vođenja. Savremeni ERP sistemi imaju jednu bazu podataka za celu organizaciju, tako da se suvišni unosi smanjuju ili uklanjaju.

MES se koristi za pokretanje proizvodne operacije, upravljanjem i izveštavanjem o aktivnostima postrojenja u slučaju događaja, u realnom vremenu. Održava se evidencija istorije podataka i ističu se izuzetci. Ove informacije pružaju tačne i pravovremene informacije ERP-u, omogućavajući kompaniji da reaguje dovoljno brzo kako bi mogla pratiti korake koji se brzo menjaju.

2.4.3 Šta pokreće akcije

ERP izvršava instrukcije na osnovu finansijskih transakcija. Kada kupci izvrše narudžbine, dobavljači šalju te narudžbine. ERP je multifunkcionalan, ali ga programeri grade prvenstveno oko ekonomske podstrukture.

S druge strane, MES je vođenjem dešavanjem događaja. Ovaj sistem je dizajniran za nagledanje događaja specifičnih za proizvodno okruženje. Nagledanje podataka u realnom vremenu omogućava MES-u da obavlja zadatke kao što su obezbeđivanje usaglašenosti za proizvodnim procesom, praćenje potrošnje zaliha, zakazivanje održavanja mašina na osnovu performansi i preuređivanje postupaka radi efikasnijeg korišćenja raspoloživih resursa.

2.5 Prednosti korišćenja ISA-95 standarda

ISA-95 standard se može koristiti kao metoda za definisanje interfejsa između sistema poslovne logistike i kontrolnih sistema. Pomaže pri unapređenju komunikacije između različitih kompanija. Svaka proizvodna kompanija koristi sopstvenu terminologiju za opisivanje funkcija, aktivnosti i odeljenja u kompaniji. Kada morate da radite sa spoljnim konsultantima, komunikacija će biti teška. Velika je šansa da ćete te pričati o različitim stvarima kada koristite iste izraze ili obrnutno.

Integracija između sistema poslovne logistike i kontrolnih sistema pre ISA-95 za vreme pravljenja nekog projekta je trajala između 1-2 godine sa procentom od 50% i manje uspešnosti. Nakon korišćenja ISA-95 standarda vremenski period se smanjio na 2-4 meseca sa procentom od 90% i više uspešnosti. Na osnovu ove statistike možemo da utvrdimo da je ISA-95 standard bio veliki uspeh [5].

3. TEHNOLOGIJE I ALATI

3.1 .NET Framework

Okruženje za razvoj softvera, razvijano od strane Microsoft-a za Windows platforme. Uključuje veliku biblioteku klasa (*Framework Class Library*). Microsoft je sa razvojem .NET-a počeo ranih 1990-tih, pod nazivom *Next Generation Windows Services* [6].

3.2 Microsoft Visual Studio

Predstavlja integrisano okruženje za razvoj računarskih programa za Windows, web-stranica, aplikacija i usluga. Koristi Microsoft-ove platforme za razvoj raznih API-ja za Windows, Windows Forms, WPF [7].

3.3 C#

Objektno orijentisan programski jezik koji je razvio Microsoft početkom 21-og veka. Reč je o jeziku opšte namene koji služi za pravljenje aplikacija u okviru .NET okruženja [8].

3.4 Windows Presentation Form (WPF)

Grafički podsistem za renderovanje korisničkog interfejsa u aplikacijama zasnovanim na Windows operativnim sistemima [9]. Razvijen je od strane Microsoft-a. WPF koristi XAML, izveden od XML-a da definiše i poveže različite UI elemente.

3.5 Microsoft SQL Server

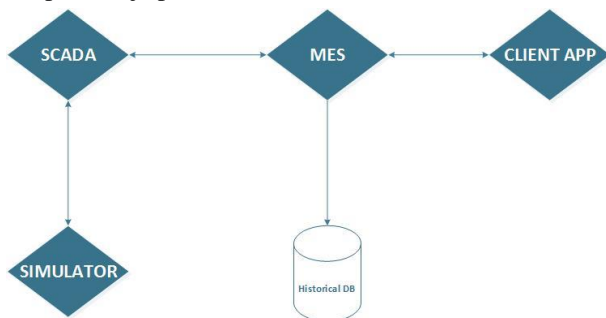
Predstavlja relacijsku bazu podataka kojoj je primarni jezik za upite *Transact SQL* (T-SQL), što znači da osim osnovnih i klasičnih (SELECT tipa) SQL upita dozvoljava i složenije stvari poput *if* naredni ili *while* petlji. SQL server je baza podataka koja je namenjena manjim i srednjim bazama [10].

3.6 Entity Framework

Entity Framework predstavlja *ORM Framework* otvorenog koda za .NET koje podržava Microsoft. Omogućava programerima da rade sa podacima koristeći objekte klase bez fokusiranja na osnovne tabele i kolone baze podataka u kojima se ti podaci čuvaju [11].

4. ARHITEKTURA SISTEMA

Aplikacija sistema je podeljena na nekoliko komponenti. Svaka komponenta ima neka svoja zaduženja i procese koje obavlja nezavisno od ostatka sistema i drugih komponenti. Arhitektura sistema i komunikacija između komponenti je prikazana na Slika 2.



Slika 2 - Arhitektura sistema

5. IMPLEMENTACIJA APLIKACIJE

5.1 SCADA

5.1.1 Inicijalno pokretanje

Da bi se ostvarile visoke performanse SCADA komponente, potrebno je ovaj servis što više olakšati kad su u pitanju procesi koji se dešavaju u njemu. Želimo da SCADA servis ima što manje poslova koje treba da obavlja. Nakon prvog pokretanja SCADA servisa, potrebno je prethodno sačuvano stanje prekidača upisati u simulator. Pošto SCADA ne sadrži te vrednosti, ona mora da ih zatraži od servisa koji poseduje te vrednosti tj. od MES-a. Otvara se komunikacioni kanal prema MES-u i traži se trenutno stanje prekidača. Nakon prijema vrednosti, prvo se ažuriraju vrednosti u SCADA memoriji, a zatim se i ažurira stanje u simulatoru.

5.1.2 Akvizicija podataka sa simulatora

Pošto se stanje prekidača u polju tj. simulatoru može konstantno menjati, mora se periodično proveravati njihovo stanje. SCADA periodično, koristeći MODBUS protokol, proziva simulator i „pita“ kakvo je trenutno stanje u polju. Od simulatora dobije informaciju o novom stanju prekidačke opreme. Po prijemu novog stanja primljena je informacija o stanju svakog prekidača u polju.

5.1.3 Komandovanje

Kada je reč o komandovanju SCADA radi samo ono što joj je zadano. Tu ne postoji nikakva logika. Od MES dobija instrukcije i te instrukcije prosleđuje ka simulatoru. Te instrukcije predstavljaju otvaranje/zatvaranje prekidača.

5.2 Simulator

Za simulaciju sistema koristimo Easy Modbus Server Simulator, sa kojim vršimo komandovanje opreme u polju. Sve izmene nastale u simulatoru, šalju se na SCADA servis preko Modbus TCP protokola. Simulator iza sebe nema nikakvu logiku, njegov posao je da „simulira“ tj. samo vrši otvaranje/zatvaranje opreme u polju. On nema uvid o kakvoj je to opremi reč, kakve su specifikacije opreme, naziv, primena...

5.3 MES

5.3.1 Inicijalno pokretanje

Nakon inicijalnog pokretanja MES servisa, pokreće se inicijalizacija modela tako što se čitaju podaci iz baze podataka i na osnovu podataka kreiraju se objekti koji se smeštaju u model. U sistemu objekti koji se mogu inicijalizovati su prekidači i alarmi.

5.3.2 Obrada akvizicije

5.3.2.1 Čuvanje stanja pristiglih vrednosti

Od SCADA servisa stiže informacija na kom ID-u prekidača je koje novo stanje. Za dobijeni ID se iz MES modela uzima konkretan objekat prekidača u kom se nalaze podaci potrebni za obradu. Nakon dobijanja konkretnog prekidača njegovo stanje se ažurira. Treba primetiti da je ovo samo u radnom modelu i da još nije završilo u istorijskoj bazi podataka.

5.3.2.2 Upisivanje stanja u bazu podataka

Obradeni podaci se moraju čuvati u istorijskoj bazi iz razloga što se istorija koristi za:

- Inicijalizaciju MES servisa
- Kreiranje izveštaja

Zbog performansi podaci ne završavaju odmah u bazi podataka, nego se upisuju u liste. Količina podataka u jednoj listi zavisi od pristiglih promena. Kad se obrade sve pristigle promene, vrši se upis u istorijsku bazu podataka i liste se nakon toga očiste da bi se nove promene mogle primenjivati.

5.3.2.3 Obaveštavanje klijenta o promenama

Odmah nakon obrade podataka, promene se „pakuju“ u format podataka poznat klijentu tj. u objekat **UIModelObject**. U taj objekat se pakuju svi afektovani prekidači, kao i generisani alarmi vezani za njih. Slanje je asinhrono.

5.4 Klijentska aplikacija

5.4.1 Inicijalno pokretanje

Nakon prvog pokretanja, klijentska aplikacija traži od MES-a svu postojeću opremu u sistemu sa njegovom trenutnom vrednošću. Što bi značilo da trebaju prekidači sa vrednostima da li je otvoren/zatvoren dati prekidač. Nakon prijema podataka, klijentska aplikacija vrši ažuriranje glavnog prozora. Prave se objekti u obliku dugmeta za svaki prekidač, gleda se koja mu je trenutna vrednost.

5.4.2 Komandovanje

Komandovanje je akcija koja prolazi kroz sve komponente sistema. Akcija počinje na klijentskoj aplikaciji kliktanje miša na prekidač. Klikom na željeni prekidač šalje se na MES, id prekidača koji želimo da komandujemo. Na osnovu njegovog trenutnog stanja tj. da li je otvoren ili zatvoren pozivamo odgovarajuće funkcije. Zatim MES informacije prosleđuje na SCADA servis. SCADA zatim na osnovu id-a iz modela traži adresu prekidača i tu adresu zajedno sa komandom šalje na simulator.

5.4.3 Alarmi

Alarmi u implementaciji aplikacije predstavljaju odziv na događaje u sistemu. Pod tim podrazumevamo na svaku akciju komandovanja prekidača, bilo to sa klijentske aplikacije ili sa simulatora, se generiše novi alarm. Time želimo da obavestimo krajnjeg korisnika o promenama u sistemu. Izgled alarma može imati dva slučaja, kada je prekidač prešao u otvoreno kolo tj. kada je isključen i kada je u zatvorenom kolu tj. kada je uključen.

5.4.4 Izveštaji

Report ili izveštaj je dokument koji predstavlja informacije koje su organizovane u formatu za određenu svrhu. Iako se sažeci izveštaja mogu dostavljati usmeno, potpuni izveštaji su gotovo uvek u obliku pisanih dokumenata. Način na koji se izveštaji generišu jeste taj da korisnik preko klijentske aplikacije unese odgovarajuće parametre za generisanje izveštaja. Parametri se šalju na MES servis koji zatim na osnovu željenog izveštaja, izvlači iz baze podataka odgovarajuće podatke. Zatim te podatke „pakuje” u **ReportModel** objekat i vraća ih nazad klijentskoj aplikaciji. Kad ti podaci pristignu na klijentsku aplikaciju vrši se ispis na glavni prozor.

6. ZAKLJUČAK

Aplikacija koja je opisana i testirana u ovom radu predstavlja simulaciju komandovanja opreme u polju pomoću simulatora ili preko implementirane klijentske aplikacije, primenjujući principe i smernice ISA-95 standarda. Objasnjen je način rada aplikacije, kao i uloga svih komponenti i delova sistema.

Neke od prednosti korišćenja ovakvog standarda tokom implementacije neke aplikacije predstavlja **nezavisna implementacija svake komponente sistema**, gde svaka komponente sistema je implementirana nezavisno od ostalih komponenti. U principu, rad jedne komponente ne utiče na rad druge komponente. **Apstrakcija** predstavlja još jednu vrstu prednosti ovakvog sistema. Kada vršimo komandovanje nas, kao krajnjeg korisnika ne zanima cela logika iza te akcije. Nas samo zanima krajnji rezultat. Pomoću apstrakcije to je moguće.

Takođe uvek postoji prostora za napredak i usavršavanje što je upravo slučaj i sa ovim radom. Određeni segmenti aplikacije bi mogli biti rešeni na bolji i efikasniji način. Jedan od mogućih unapređenja ili izmena je da klijenta aplikacija bude urađena u nekoj drugoj tehnologiji koja nije WPF npr. Windows Form ili ASP.NET. Još jedna vrsta unapređenja aplikacije je mogla biti uvođenje opreme sa analognim signalima. Pod tim se podrazumeva uvođenje neke vrste brojača ili merila koja može da prati vreme rada prekidača i te vrednosti da ispisuje u simulator i na klijentsku aplikaciju.

7. LITERATURA

- [1] Bianca Scholten, *Road to Integration*, ISBN-13: 978-0979234385
- [2] TechTarget, *Definition, ANSI/ISA-95*, datum pristupa: 25/03/2021 <<https://searcherp.techtarget.com/definition/ANSI-ISA-95>>
- [3] Siemens, *ISA 95 Framework & Layers*, datum pristupa: 26/03/2021 <<https://www.plm.automation.siemens.com/global/en/our-story/glossary/isa-95-framework-and-layers/53244>>
- [4] Intraratio, *3 Key Differences Between ERP and MES Systems* datum pristupa: 26/03/2021 <<https://intraratio.com/post/key-differences-between-erp-and-mes>>
- [5] ISA-95, *Overview of advantages*, datum pristupa: 27/03/2021 <<https://isa-95.com/advantages>>
- [6] Guru99, *What is Microsoft .Net Framework?*, datum pristupa: 27/03/2021 <<https://www.guru99.com/net-framework.html>>
- [7] GeeksforGeeks, *Introduction to Visual Studio*, datum pristupa: 27/03/2021 <<https://www.geeksforgeeks.org/introduction-to-visual-studio>>
- [8] Jon Skeet, *C# in Depth, 3rd Edition*, ISBN-13: 978-1617291340
- [9] WPF Tutorial, *What is WPF?*, datum pristupa: 27/03/2021 <<https://www.wpf-tutorial.com/aboutwpf/what-is-wpf>>
- [10] Wikipedia, *Microsoft SQL Server*, datum pristupa: 28/03/2021 <https://wikipedia.org/wiki/Microsoft_SQL_Server>
- [11] Entity Framework Tutorial, *What is Entity Framework?*, datum pristupa: 28/03/2021 <<https://www.entityframeworktutorial.net/what-is-entityframework.aspx>>

Kratka biografija:

Nenad Zelenović je rođen 01.04.1995. godine u Somboru. Osnovnu školu "Nikola Tesla" u Bačkom Brestovcu završio je 2010. godine. Završio je srednju ekonomsku školu u Somboru 2014. godine. Fakultet Tehničkih Nauka u Novom Sadu je upisao 2014. godine, smer Elektroenergetski Softverski Inženjering. 2019. godine završava osnovne studije i upisuje master akademske studije na Fakultetu Tehničkih Nauka, smer Primenjeno Softversko Inženjerstvo. Ispunio je sve obaveze i položio sve ispite predviđene studijskim programom.