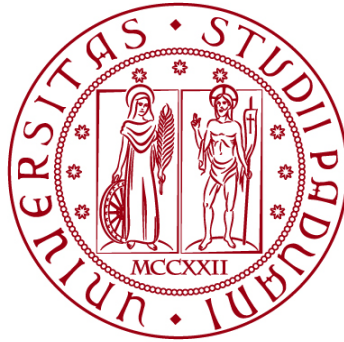


UNIVERSITÀ DEGLI STUDI DI PADOVA
DIPARTIMENTO DI INGEGNERIA CIVILE, EDILE E AMBIENTALE
Department Of Civil, Environmental and Architectural Engineering

Corso di Laurea in Ingegneria Edile-Architettura



TESI DI LAUREA

**ANALISI DEL FLUSSO DI INFORMAZIONI TRA I MODELLI BIM
E I SISTEMI DI AUTOMAZIONE E CONTROLLO DEGLI EDIFICI
(BACS) TRAMITE BACNET E IFC**

Relatore:
prof. CARLO ZANCHETTA

Laureanda:
SANTINATO FEDERICA
1151653

ANNO ACCADEMICO 2022-2023

SOMMARIO

ABSTRACT	1
INTRODUZIONE	3
CAPITOLO PRIMO LA BUILDING AUTOMATION E LA DIGITALIZZAZIONE	7
1 L'EVOLUZIONE DIGITALE	7
1.1 <i>Industry Foundation Classes</i>	9
1.2 <i>L'Internet of Things</i>	18
CAPITOLO SECONDO LA BUILDING AUTOMATION E LE TECNOLOGIE DI COMUNICAZIONE	27
1 CONCETTI GENERALI DEI SISTEMI DI AUTOMAZIONE E CONTROLLO DEGLI EDIFICI	27
1.1 <i>Architettura di sistema</i>	28
1.2 <i>Sensori, attuatori e controller</i>	30
1.3 <i>Modello del dispositivo</i>	32
2 STANDARD PER LA COMUNICAZIONE DEI SISTEMI DI AUTOMAZIONE	33
2.1 <i>BACnet</i>	34
2.2 <i>KNX</i>	35
2.3 <i>LonWorks</i>	38
2.4 <i>ZigBee</i>	41
2.5 <i>Altri standard</i>	44
3 BACNET	47
3.1 <i>Dispositivi BACnet</i>	47
3.2 <i>Oggetti BACnet</i>	52
3.3 <i>Proprietà degli oggetti BACnet</i>	66
3.4 <i>Servizi applicativi BACnet</i>	68
3.5 <i>PICS and BIBB's</i>	73
3.6 <i>Certificazione BTL</i>	78
3.7 <i>Architettura BACnet</i>	80
CAPITOLO TERZO LO SCAMBIO DI INFORMAZIONI TRA BACNET E IFC	85
1.1 <i>Riferimenti BACnet nello standard IFC</i>	87
1.2 <i>Gli oggetti BACnet nello standard IFC</i>	95
1.3 <i>Proposta di un'integration system tra BACnet e IFC mediante una BACnet MVD</i>	114
CAPITOLO QUARTO METODOLOGIA E CASO STUDIO APPLICATIVO	123
1 ECOSTRUXURE™	123
1.1 <i>EcoStruxure™ Building</i>	125
2 ECOSTRUXURE BUILDING OPERATION	126
2.1 <i>Concetto di Binding in EcoStruxure Building Operation</i>	129
2.2 <i>Topologie di rete in EcoStruxure Building Operation</i>	132
2.3 <i>Estensioni file consentite da EcoStruxure Building Operation</i>	134
2.4 <i>BACnet Interface in EcoStruxure Building Operation</i>	135
3 CASO STUDIO APPLICATIVO	140
3.1 <i>Classificazione IFC degli elementi del modello</i>	144
3.2 <i>Mappatura ed esportazione del modello IFC</i>	149
3.3 <i>Creazione del modello in EcoStruxure Building Operation</i>	151

3.4	<i>Export del file XML da EBO</i>	157
3.5	<i>Visual Studio Code per parsing IFC to XML</i>	159
CAPITOLO QUINTO ANALISI CRITICA DEI RISULTATI E SVILUPPI FUTURI		167
1	ANALISI DEI RISULTATI	167
2	SVILUPPI FUTURI.....	170
CAPITOLO SESTO CONCLUSIONI		173
1	SINTESI DELLA RICERCA	173
BIBLIOGRAFIA		175
INDICE DELLE TABELLE		179
INDICE ICONOGRAFICO		181

ABSTRACT

La presente tesi ha come obiettivo quello di analizzare nel dettaglio la relazione che intercorre tra i sistemi informativi digitali (BIM) e i sistemi di controllo e di automazione degli edifici (BACS), dato il loro ruolo sempre più strategico e fondamentale nel settore costruttivo e la rapida evoluzione digitale alla quale si sta assistendo.

Lo studio si concentra prevalentemente sulle affinità tra due standard aperti: l'IFC (Industry Foundation Classes) il formato di interscambio dati tra applicativi BIM più utilizzato, nonché l'unico recepito a partire dal 2013 dall'International Organization for Standardization (ISO), e BACnet (Building Automation and Control Networking protocol), il quale si contraddistingue dalle altre tecnologie di comunicazione per un tipo di organizzazione dei dati orientata agli oggetti, i quali hanno tutti delle determinate proprietà e garantiscono al protocollo lo sviluppo di servizi.

Da questa analisi poi, ne deriva lo sviluppo di un caso studio applicativo con lo scopo di testare l'integrazione sistematica di un unico file completo IFC nell'ambiente BACS, riducendo gli oneri, semplificando le procedure ed eliminando le ambiguità anagrafiche derivabili dall'elaborazione di più modelli in discipline diverse.

Nell'elaborazione dei contenuti si è svolta un'analisi iniziale sullo sviluppo della building automation a partire dal 1980, quando l'automazione era intesa a livello del singolo apparato o dispositivo. Successivamente sono stati accurati grandi progressi nell'integrazione del sistema di intelligenza degli edifici in termini sia di tecnologia che di scala. Non solo, è stata studiata anche la digitalizzazione con la recente affermazione dell'IoT, la quale disegna una convergenza di varie tendenze informatiche e di connettività, evolutesi per decenni. È, infatti, diventato un fulcro tecnologico per il mondo accademico, per i settori industriali (automobilistico, elettronico casalingo e di consumo, sanitario, manifatturiero e tanti altri) e persino per le organizzazioni governative.

Nel secondo capitolo, si è trattato esaurientemente il concetto di building automation, delineando le tecnologie di comunicazione associate al tema. Sono dunque stati presentati i concetti generali dei sistemi di automazione e controllo dell'edificio, descrivendone per prima cosa l'architettura di sistema contraddistinta da stack di livelli gerarchici. Sono poi stati analizzati gli elementi elementari che compongono un BACS e le relative interfacce di comunicazione.

Particolare attenzione è stata dedicata allo studio degli standard per la comunicazione dei sistemi di automazione e controllo degli edifici, notando come la building automation sia stata, fin dal principio, popolata da una pleora di soluzioni proprietarie. Ci si è focalizzati soprattutto nel delineare accuratamente lo standard BACnet, descrivendone tutti i dispositivi, gli oggetti con le relative proprietà e i servizi applicativi, essendo oggetto di sperimentazione.

Nel capitolo terzo vengono indagati in letteratura i riferimenti BACnet all'interno dello standard IFC, notando come questi siano perlopiù assenti o sommari. Si è rivelato perciò necessario procedere ad associare, dove possibile, ad ogni oggetto BACnet un'entità specificata dallo standard

IFC, comprendendo come esista un'analogia da implementare in un percorso di sperimentazione.

Nel capitolo quarto viene proposto il caso studio applicativo citato, il quale si propone di dimostrare come questi standard abbiano un'effettiva affinità e come sia possibile trasferire in modo soddisfacente determinati dati a partire da un modello in formato ifc ad un file in formato xml, al fine di automatizzare il percorso di modellazione degli impianti domotici in software specializzati che applicano lo standard BACnet.

INTRODUZIONE

Si definisce con l'acronimo BACS (Building Automation and Control System) un sistema di automazione e controllo degli edifici generalmente costituito da un apparato installato in un edificio che mira a controllare e monitorare i servizi dello stesso responsabili di comfort termico, qualità dell'aria interna, illuminazione, ombreggiatura, sicurezza della vita e molti altri fattori. Dati questi precedenti, un immobile dotato di BACS è definito *smart building*, grazie alla sua capacità di fornire taluni servizi all'edificio massimizzando l'efficienza e il risparmio dei costi e minimizzando l'impatto ambientale.

È importante porre attenzione alle differenze tra BAC, BACS e TBM. Con Building Automation and Control si includono i prodotti, i software e le prestazioni ingegneristiche volti al controllo automatico, al monitoraggio e l'ottimizzazione e alla gestione al fine di ottenere un funzionamento efficiente sotto l'ottica energetica, economica e della sicurezza delle apparecchiature. Con Building Automation and Control System si comprendono, invece, tutti i prodotti e i servizi ingegneristici relativi a BAC; un sinonimo da utilizzare in alternativa è BMS, Building Management System. Con Technical Building Management, infine, si coinvolge la totalità di processi e prestazioni relativi all'esercizio e alla gestione degli edifici e degli impianti tecnici; questo implica, dunque, la presenza di BACS e la loro effettiva integrazione in esercizio, rappresentando una parte del Facility Management. (Aste et al., 2017)

Diversi appellativi sono comunque attribuiti a questa disciplina (Tang et al., 2020):

- Energy Management Control Systems (EMCS)
- Home Control Network (HCN)
- Home Control Systems (HCS)
- Home and Building Management Systems (HBMS)
- Building Energy Management Systems (BEMS)
- Facility Management Systems (FMS)

Altro dettaglio a cui è necessario prestare attenzione è la differenza tra automazione degli edifici e controllo degli edifici, termini molto simili e spesso confusi. L'*Association of German Engineers*¹, VDI, definisce l'automazione come la misurazione, il controllo e la gestione computerizzati dei servizi di costruzione. Da questa, è possibile dedurre che il controllo degli edifici fa parte dell'automazione stessa, è dunque una sua suddivisione specifica che si

¹ Organizzazione di oltre 150.000 ingegneri e scienziati naturali fondata nel 1856. Promuove il progresso della tecnologia e rappresenta gli interessi degli ingegneri e delle imprese di ingegneria in Germania.

concentra soprattutto sugli impianti elettrici. (Merz et al., 2009)

È pertanto chiaro come un BACS aspiri a rendere automatiche le attività in ambienti tecnologicamente autorizzati, coordinando una serie di apparati elettrici e meccanici interconnessi in modo distribuito, tramite le reti di controllo sottostanti. Questi sistemi possono essere implementati in infrastrutture industriali, edifici aziendali e centri commerciali, o anche in ambito domestico.

Negli ultimi due decenni l'ambito dell'automazione degli edifici ha ricevuto una maggiore attenzione grazie alla possibilità di ridurre il consumo di energia e di semplificare la funzione, il monitoraggio e la manutenzione dell'edificio, garantendo al contempo la compiacenza dei suoi residenti. Ciò è reso possibile usufruendo di un'ampia gamma di sensori atti, ad esempio, al rilevamento della concentrazione di CO₂, della temperatura, del flusso d'aria e dei livelli di occupazione e illuminazione, i quali forniscono delle informazioni fondamentali che permettono di prendere alcune decisioni. Queste ultime sono volte al controllo delle apparecchiature dell'edificio consentendo agli occupanti di ridurre le spese e gli sprechi attraverso atteggiamenti consapevoli, ponendo così particolare attenzione alla salvaguardia dell'ambiente e soddisfacendo i requisiti di comfort. (Domingues et al., 2016)

La possibilità di sviluppo del concetto di automazione pone le sue basi nella digitalizzazione dell'ambiente costruito e, dunque per prima cosa, nei progressi fatti con il Building Information Modeling, il BIM. Quest'ultimo si sta espandendo rapidamente come piattaforma di distribuzione e gestione delle informazioni, tanto da poterlo definire uno standard universale nel settore edile: dal 2025, difatti, rappresenterà un processo di modellazione obbligatorio per tutte le opere.

Con il BIM si denota un sistema informativo digitale della costruzione delineato da un modello tridimensionale integrato da dati fisici, funzionali e prestazionali dell'edificio. Si può così notare il potenziale del BIM che è incernierato nell'opportunità di generare un modello informativo dinamico, interdisciplinare e condivisibile che racchiude in sé la totalità delle informazioni sull'intero ciclo di vita di un manufatto.

Ad oggi, nonostante possa prendere parte delle fasi di architettura, ingegneria, costruzione e funzionamento, è ancora strettamente legato alle sole fasi di progettazione e costruzione, ma se posto a supporto del processo di monitoraggio e controllo, durante la fase post-costruzione, un modello BIM si può collegare ai sistemi di gestione intelligente degli edifici producendo un flusso di lavoro digitalizzato. Con la crescente popolarità delle piattaforme BIM, infatti, si crea la possibilità di sfruttare tale tecnologia in modo da plasmare delle piattaforme aperte capaci di sincronizzarsi con diverse fonti di informazioni, come sensori wireless e, di conseguenza, anche gli stessi sistemi di automazione degli edifici.

Dati questi precedenti, gli standard BIM aperti non proprietari, come l'IFC ad esempio, si posizionano come candidati naturali per la modellazione e lo scambio di informazioni riguardanti l'automazione, permettendo di conservare e scambiare dati coerenti tra i membri di un team di progetto, sia tra piattaforme con software simili che differenti, uniformando i

flussi di lavoro.

Tuttavia, l'ambito edilizio è permeato da silos di dati e dalla mancanza di una standardizzazione in grado ordinare il flusso delle informazioni: BIM e sistemi di edifici intelligenti sono colmi di formati di file ed ecosistemi serrati, i quali non ammettono la libera condivisione di nozioni tra le parti interessate. In particolare, il BIM si presenta ancora come uno strumento complesso, riservato agli esperti del settore, rendendo problematica la mera condivisione di informazioni e dall'altra parte, guardando al sottodominio del controllo e automazione degli immobili, questo è fin troppo caratterizzato da protocolli e standard aperti come KNX, LonWorks, ModBus, Zigbee, oBIX, BACnet e tanti altri che non dialogano affatto con i sistemi informativi digitali.

Emerge così una lacuna nella ricerca dell'integrazione dei dati dell'ambiente costruito e nello sviluppo di interfacce utente che migliorino il tenore di vita degli abitanti, i quali rappresentano una delle maggiori sfide per l'implementazione di queste tecnologie nel settore.

LA BUILDING AUTOMATION E LA DIGITALIZZAZIONE

1 L'EVOLUZIONE DIGITALE

Il XXI secolo sta assistendo ad una rapida evoluzione digitale e il settore edile ne è significativamente coinvolto. Negli ultimi anni, infatti, la ricerca si è focalizzata su edifici, città e infrastrutture intelligenti e i progressi fatti con tecnologie come il Building Information Modeling (BIM), l'Internet of Things (IoT), i sensori wireless e l'elaborazione e analisi dei dati gettano le basi per la trasformazione del modo in cui interagiamo con l'ambiente costruito. Uno dei motivi al principio di questa ricerca è lo sviluppo di un approccio più attento e consapevole dal punto di vista energetico senza nuocere al livello di benessere, bensì migliorando l'esperienza per gli utenti finali e i fornitori di servizi.

Il concetto di edificio intelligente è nato dal controllo automatico, e appunto intelligente, dei processi tipici dei servizi edilizi e degli apparati di comunicazione. Insieme all'evoluzione e all'integrazione della tecnologia elettronica, informatica e dell'informazione i sistemi costruttivi intelligenti stanno diventando sempre più progrediti tendendo alla convergenza dei sistemi informativi.

Più nel particolare, inizialmente l'automazione degli edifici venne integrata nelle strutture commerciali al fine di abilitare funzioni eseguibili in maniera automatica, senza l'intervento diretto di una persona fisica per la regolazione degli impianti. Furono implementati i sistemi tradizionali mediante l'introduzione dei primi controller di dati digitali diretti, cosiddetti DDC (Direct Digital Controllers) ai sistemi di riscaldamento, ventilazione e gestione dell'aria condizionata (Heating Ventilation Air Conditioning, HVAC). Così facendo, aggiungendo un controller centrale a questi sistemi, fu possibile mettere in esecuzione e monitorare i sistemi più efficacemente e creare così una rete di sistema. (Merz et al., 2009)

L'evoluzione di questi sistemi è illustrata nella figura che segue, aggiornata dalle *Intelligent Building Pyramids* sviluppate dallo *European Intelligent Building Group*. La piramide illustra il contenuto e l'evoluzione dell'automazione degli edifici dal 1980 ad oggi. Il disegno è volutamente aperto nella sommità, rimarcando come i sistemi di costruzione intelligenti non siano più racchiusi all'interno degli edifici, ma fusi con quelli di altri edifici e con altri sistemi informativi tramite l'IoT.

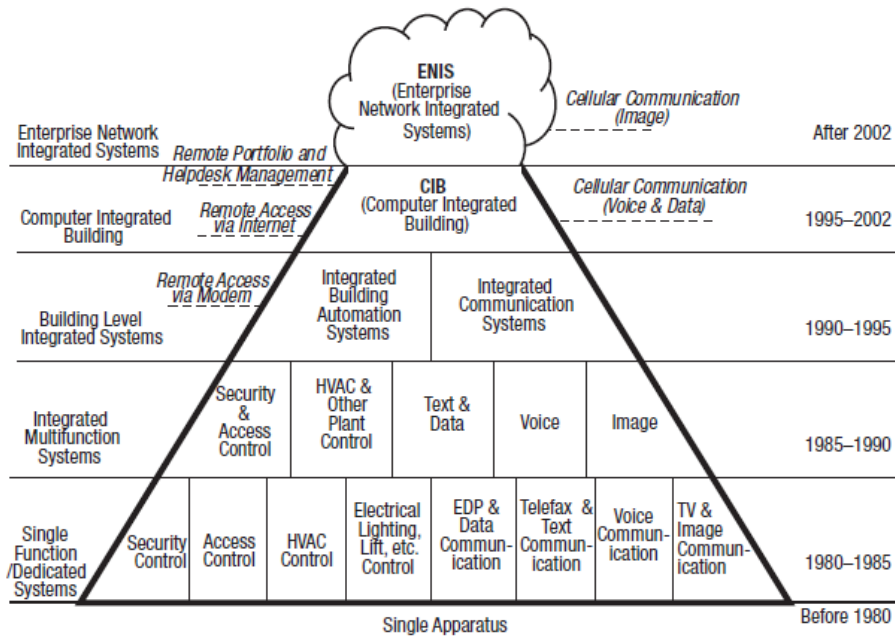


Figura 1.1: Percorso evolutivo dei Sistemi di Automazione e Controllo degli Edifici. (Aste et al., 2017)

Prima del 1980, l'automazione era intesa a livello del singolo apparato o dispositivo. Successivamente i sistemi di costruzione intelligenti sono entrati nelle fasi di definizione: sono stati accurati grandi progressi nell'integrazione del sistema di intelligenza degli edifici in termini sia di tecnologia che di scala. I sistemi successivi al 1980 possono essere suddivisi in cinque fasi come segue:

- Sistema integrato monofunzionale o dedicato (1980-1985)
- Sistemi multifunzione integrati (1985-1990)
- Sistemi integrati a livello di edificio (1990-1995)
- Edificio integrato con computer (1995-2002)
- Sistemi integrati di rete aziendale (2002-)

Nella fase dei sistemi integrati monofunzionali o dedicati tutti i sottosistemi di automazione degli edifici e di automazione della comunicazione dei dati sono stati integrati a livello di funzione singola o individuale; in più, non era possibile attuare la comunicazione tra sistemi di automazione di diversi sottosistemi.

Nella fase dei sistemi multifunzione integrati sono integrate la sicurezza e il controllo delle entrate. Sono stati integrati i sistemi di automazione degli impianti edilizi, o dei sistemi di servizi, tramite reti unificate in relazione alla comunicazione di testo e dati, la comunicazione vocale e la comunicazione di immagini. È evidente come in questa fase venga realizzata l'integrazione di sistemi operanti nello stesso ambiente.

Nella fase dei sistemi integrati a livello di edificio sia la Building Automation sia i sistemi di comunicazione sono integrati a livello di edificio come sistema di automazione degli edifici

(BAS) e sistema di comunicazione integrato (ICS). In questa fase è attuabile l'accensione da remoto di un sistema di automazione tramite rete telefonica o modem, fintanto che viene introdotto sul mercato il cellulare per la comunicazione voce e dati.

Durante e dopo la fase di costruzione integrata di computer le reti di convergenza diventano fruibili e sono impiegate nella pratica gradualmente, grazie all'uso sempre più comune delle tecnologie di rete del protocollo Internet (IP) e all'aumento della capacità della rete. In questa fase, l'integrazione è a livello di edificio e monitoraggio e controllo remoti possono essere raggiunti tramite Internet.

Nella fase di sistema integrato di rete aziendale i sistemi di automazione possono essere adattati e gestiti non solo a livello aziendale, ma anche cittadino. I sistemi di costruzione intelligenti ora non sono più racchiusi all'interno di singoli edifici, ma vengono fusi con quelli di altri edifici e con altri sistemi informativi tramite l'infrastruttura Internet globale. L'integrazione e la gestione a questo livello diventano possibili grazie alle applicazioni di tecnologie IT avanzate come Web Services, XML, remote portfolio management e helpdesk management. (Wang, 2009)

Come già accennato, l'uso delle nuove tecnologie è un presupposto fondamentale per lo sviluppo di edifici intelligenti che includono, ma non si limitano a, l'implementazione di sensori, l'ingegneria e l'analisi dei big data, il cloud e il fog computing, lo sviluppo dell'ingegneria del software e degli algoritmi di interazione uomo-computer. Date queste tecnologie, una che emerge fortemente in questi tempi è lo sviluppo dell'Internet of Things (IoT). Con l'uso dell'IoT c'è un enorme potenziale per compiere notevoli progressi. Il perfezionamento del Building Information Modeling (BIM) con i flussi di dati in tempo reale dell'IoT rappresenta un potente modello per le applicazioni al fine di affinare la costruzione e l'efficienza operativa.

1.1 Industry Foundation Classes

Ad oggi, il Building Information Modeling (BIM) è diventato un modello consolidato per lo sviluppo di pratiche avanzate di consegna dei progetti. Se opportunamente sviluppato e gestito, rende disponibili pacchetti di dati dei componenti dell'edificio ad alta fedeltà, meticolosi dal punto di vista geometrico e della geometria spaziale, congiunti a una vasta serie di metadati descrittivi e sfruttabili.

Il BIM si compone di fatto di un database che assimila informazioni grafiche e alfanumeriche, ordinate all'interno di oggetti 3D, il tutto sfruttando standard aperti e l'IT, per mezzo del quale più parti interessate possono progettare, costruire e gestire in modo collaborativo una struttura durante le varie fasi di un progetto. Si evince, dunque, quanto sia importante la scelta del formato utilizzato per definire il modello: è necessario un linguaggio comune che funga da mezzo di interscambio di dati tra domini, fasi e parti per le applicazioni di BIM, ma anche per la Building Automation.

Lo standard più adatto al caso è l'IFC, acronimo per *Industry Foundation Classes*. È un

formato dati *platform neutral*, il che significa che il suo sviluppo non appartiene a nessuna società, azienda o ente specifico e non è controllato da alcuna casa produttrice di software. È proposto per la prima volta nel 1994 da parte di BuildingSMART International, consorzio industriale che ancor oggi provvede al mantenimento e all'aggiornamento delle sue versioni. L'associazione è promotrice di tre standard tra loro congiunti e messi a disposizione nella piattaforma online:

- lo standard IFD (International Frameworks for Dictionaries) relativo ai Termini,
- lo standard IDM (Information Delivery Manual) relativo ai Processi,
- lo standard IFC, relativo ai Dati.

All'epoca, viene evidenziata la necessità di trovare un unico sistema di comunicazione per affrontare la digitalizzazione già in corso. Da allora IFC si è evoluto progressivamente, aggiungendo classi e relazioni fra le medesime come osservabile in figura, fino ad arrivare all'ultima release ufficiale dello standard (IFC4 Add2).

Ad oggi è il più diffuso formato di interscambio dati tra applicativi BIM, nonché l'unico recepito a partire dal 2013 dall'International Organization for Standardization (ISO), l'ente normativo internazionale con sede a Ginevra, come standard di scambio dati in ambito costruttivo. Con la pubblicazione della *ISO 16739 – Industry Foundation Classes (IFC) for data sharing in the construction and facility management industries*, si dimostra il rispetto di determinati standard di qualità per quanto riguarda stabilità e maturità. Ciò dovrebbe aiutare i fornitori a fare affidamento sulle implementazioni del modello per la compatibilità futura.

Lo schema IFC si basa su un meta-modello con la finalità di assicurare la comunicabilità dei dati tramite individui che potenzialmente potrebbero usare piattaforme di authoring differenti, appartenenti a case software differenti e con operatori che potrebbero interrogare i database per i più svariati fini: in sostanza mira all'interoperabilità tra diversi domini di professione.

Version	Name (HTML Documentation)	ISO publication	Published (yyyy-mm)	Current Status	HTML	EXPRESS	XSD	pSet XSD	OWL HTML	RDF	TTL
4.4-dev	IFC 4.4.0 development	not started		Extension of 4.3.0. Adding additional functionality (mainly for Tunnels). Currently under development in the Standards & Solutions program							
4.3.1.0	IFC4.3.1.x dev	Updates from 4.3.1.x might be used as input into the ISO DIS process.	Updates after 4.3.0.1 coming from the Implementer Forum.	Focussed on documentation improvement, clarifications and further detailing of implementation.	Latest HTML	GitHub output		PSD output			
4.3.0.1	IFC4.3 TC1 (zip)	Under ISO DIS Voting		under ISO DIS Voting	HTML						
4.2.0.0	IFC4.2	-	2019-04	Withdrawn	ZIP	EXP	IFC4x2.xsd	-			
4.1.0.0	IFC4.1	-	2018-06	Withdrawn	ZIP	EXP	IFC4x1.xsd	-	ifcOWL IFC4.1	RDF	TTL
4.0.2.1	IFC4 ADD2 TC1	ISO 16739-1:2018	2017-10	Official	ZIP	EXP	IFC4.xsd	-	ifcOWL IFC4 ADD2 TC1	RDF	TTL
4.0.2.0	IFC4 ADD2	-	2016-07	Retired	ZIP	EXP	IFC4_ADD2.xsd	-	ifcOWL IFC4 ADD2	RDF	TTL
4.0.1.0	IFC4 ADD1	-	2015-06	Retired	ZIP	EXP	IFC4_ADD1.xsd	-	ifcOWL IFC4 ADD1	RDF	TTL
4.0.0.0	IFC4	ISO 16739:2013	2013-02	Retired	ZIP	EXP	ifcXML4.xsd	PSD_IFC4.xsd	ifcOWL IFC4	RDF	TTL
2.3.0.1	IFC2x3 TC1	ISO/PAS 16739:2005	2007-07	Official	ZIP	EXP	IFC2X3.xsd	PSD_R2x3.xsl	ifcOWL IFC2x3 TC1	RDF	TTL
2.3.0.0	IFC2x3	-	2005-12	Retired	ZIP	EXP	-	-	ifcOWL IFC2x3	RDF	TTL
2.2.1.0	IFC2x2 ADD1	-	2004-07	Retired	ZIP	EXP	-	-	-	-	-
2.2.0.0	IFC2x2	-	2003-05	Retired	ZIP	EXP	-	-	-	-	-
2.1.1.0	IFC2x ADD1	-	2001-10	Retired	ZIP	EXP	-	-	-	-	-
2.1.0.0	IFC2x	-	2000-10	Retired	ZIP	EXP	-	-	-	-	-
2.0.0.0	IFC2.0	-	1999-10	Retired	-	-	-	-	-	-	-

Figura 1.2: Database aggiornato delle release IFC. (van Berlo, 2022)

Lo stesso logo di IFC, mostrato in fig. 1.3, è lo specchio dell'ambizione a diventare un linguaggio comune per lo scambio delle informazioni sull'edificio alle varie fasi del processo: procure, design, assemble, operate.



Figura 1.3: Il logo di IFC (Fonte: BIS-LAB® building innovation & skills-lab, 2017)

Il supporto per IFC è ormai deliberato nella maggior parte dei software di composizione odierni e ampiamente riconosciuto come un mezzo prezioso per garantire una migliore integrazione del processo del ciclo di vita degli edifici. Affinché le applicazioni divengano conformi alla specifica IFC, è necessario un processo di certificazione pubblica in due fasi: prima è doveroso testare i file di prova durante un workshop, poi è previsto un periodo di test per l'utente finale. L'intero processo è scandito e custodito da BuildingSMART.

1.1.1 Regole per la comprensione e interfaccia grafica dello schema IFC

Lo studio condotto si riferisce alla versione 4.0.2.1 dello schema IFC, consultabile e visualizzabile online in formato HTML (BuildingSMART International, 2020). Nel seguente paragrafo saranno dunque illustrati e chiariti le definizioni, i concetti e le regole di base utili alla comprensione dell'elaborato, nonché una spiegazione dell'interfaccia grafica della pagina web.

Consultando la pagina HTML dello schema, in alto si trovano fissi i capitoli di riferimento dello standard, divisi per argomento e raggruppati in cinque diverse sezioni (fig. 1.4).

IFC4_ADD2_TC1 - 4.0.2.1 [Official]		© 1996-2020 buildingSMART International Ltd.		
Cover	1. Scope	5. Core data schemas	A. Computer interpretable listings	E. Examples
Contents	2. Normative references	6. Shared element data schemas	B. Alphabetical listings	F. Change logs
Foreword	3. Terms, definitions, and abbreviated terms	7. Domain specific data schemas	C. Inheritance listings	Bibliography
Introduction	4. Fundamental concepts and assumptions	8. Resource definition data schemas	D. Diagrams	Index

Figura 1.4: Sezioni e relativi capitoli di riferimento dello schema IFC. (BuildingSMART International, 2020)

La prima sezione introduce lo schema, i suoi contenuti e gli obiettivi.

La seconda fornisce l'ambito, i riferimenti normativi, i termini, le definizioni e le abbreviazioni e i concetti fondamentali su cui si basa lo schema.

La terza sezione riporta gli schemi individuali riferiti ai quattro layer dell'architettura di IFC: Core data schemas, Shared element data schemas, Domain specific data schemas, Resource definition data schemas, i quali saranno descritti al paragrafo 1.1.2 del medesimo capitolo.

La quarta sezione, assieme alla quinta, riporta gli allegati comprendenti l'elenco delle entità in ordine alfabetico e gerarchico, i diagrammi, gli esempi, i sommari delle modifiche apportate con le nuove release di schema, e infine la biografia e l'indice.

Come accennato, nella seconda sezione sono presenti, al capitolo 3, tutte le definizioni necessarie per la comprensione dello standard. Vengono di seguito riportate le principali al fine di garantire una maggiore scorrevolezza alla lettura del lavoro svolto (BuildingSMART International, 2020):

- Attribute (attributo): unità di informazione interna ad una entità, definita da un particolare tipo o riferita ad una particolare entità. Vi sono tre tipo di attributi: diretti (direct), inversi (inverse) e derivati (derived).
- Entity (entità): classe di informazione definita da attributi comuni e vincoli come definito dallo standard ISO 10303-11.
- Instance (istanza): è l'occorrenza di un'entità.
- Type: costruito di informazione di base derivato da una entità primitiva, una enumerazione o una selezione di entità.
- Enumeration (enumerazione): costruito che consente ad un valore di un attributo di essere uno tra più valori predefiniti identificati dal nome.
- Property (proprietà): unità di informazioni definita dinamicamente come una particolare istanza di entità.

Altro concetto che si rivelerà utile in seguito è quello di aggregazione. In generale, sono supportate quattro diverse relazioni di aggregazione:

- ARRAY = una collezione di entità stabilite la cui rappresentazione è del tipo: A [1:?];
- BAG = una collezione di entità priva di ordine e che permette la duplicazione degli oggetti contenuti, la cui rappresentazione è del tipo: B [1:?] (Non utilizzata in IFC);
- LIST = una collezione di oggetti in sequenza ordinata, la cui rappresentazione è del tipo: L [1:?];
- SET = una collezione di oggetti priva di ordine e senza duplicazione degli stessi, la cui rappresentazione è del tipo: S [1:?].

Addentrando nello schema IFC, si vede come i due tipi di aggregazione più diffusi siano gli ultimi due, dunque LIST e SET.

I numeri tra parentesi quadre specificano la cardinalità delle relazioni e danno un'indicazione nel numero di oggetti che possono o devono essere relazionati mediante l'aggregazione:

- [0:1] = la relazione di aggregazione è opzionale, e può essere ricondotta un unico oggetto dell'aggregazione;
- [0:?] = la relazione di aggregazione è opzionale, e può essere ricondotta ad un numero massimo di oggetti non specificato dell'aggregazione;
- [1:1] = la relazione di aggregazione è obbligatoria, e deve essere ricondotta ad

- un unico oggetto dell'aggregazione;
- [1:?] = la relazione di aggregazione è obbligatoria, e deve essere ricondotta ad almeno un oggetto dell'aggregazione, fino ad un numero massimo di oggetti non specificato;
- Nel caso in cui il numero massimo di elementi di aggregazione è noto e limitato, il numero va identificato e collocato al posto del carattere di non specificazione (?).

In generale, i termini e i concetti riportati dallo schema sono espressi in lingua inglese e seguono una convenzione per la loro nomenclatura:

- I nomi dei dati per tipi, entità, regole e funzioni cominciano con il prefisso "Ifc" e continuano con le parole inglesi seguendo la convenzione CamelCase (nessuna spaziatura, con la prima lettera di ogni parola in maiuscolo);
- I nomi di attributo seguono la convenzione CamelCase ma senza prefisso;
- I nomi di definizioni di set di proprietà presentano il prefisso "Pset_" e continuano seguendo la convenzione CamelCase;
- I nomi di definizioni di set di quantità presentano il prefisso "Qset_" e continuano seguendo la convenzione CamelCase.

1.1.2 Struttura dello schema dati IFC

Lo schema segue una struttura ad oggetti complessa, chiamati entità, i quali si connettono tra loro attraverso relazioni. Ad ogni oggetto sono legati dei valori che riempiono campi standard al fine di descrivere sé stessi. Le specifiche di ogni costrutto, abbinate all'insieme dei propri attributi, definiscono una classe.

Esempi di entità sono elementi costruttivi come per esempio *IfcWall*, elementi geometrici, come *IfcExtrudedAreaSolid* e costrutti di base come *IfcCartesianPoint*. Al livello più ideale, IFC divide tutte le entità in radicate e non radicate. Le prime si creano a partire da *IfcRoot* e hanno un concetto di identità, insieme agli attributi di nome, descrizione, e controllo. Le seconde, invece, non hanno alcuna identità e le istanze esistono solo se si fa riferimento ad un'istanza radicata.

IfcRoot, nello specifico, è la classe più astratta, determinabile come il "supertipo" comune a tutte le entità IFC; è suddiviso in tre concetti astratti: definizioni di oggetti, relazioni e insiemi di proprietà.

L'architettura dei *data schemas* di IFC va a definire quattro livelli concettuali, che catalogano e raccolgono tutti i modelli usati per definire le entità usate in IFC. Il tutto è organizzato in maniera gerarchica, cosicché ogni data schema sia assegnato puntualmente a un livello concettuale e ogni specifica possa essere messa in relazione solo ad altre di livello pari o inferiore al proprio.

Questi livelli sono:

- Resource schema: include tutte quelle entità di supporto, non radicate, che non possono esistere in maniera indipendente, ma che possono esistere solo in riferimento ad una o più classi degli altri schemi per la definizione delle risorse (ad esempio *IfcDate*, *IfcPoint*, *IfcAddress*, *IfcMaterial*). Questo livello racchiude dei *booster* del modello e dunque dei concetti essenziali, delle risorse per l'appunto, che completano i domini disciplinari. I *data schemas* di questo livello, infatti, definiscono le proprietà base, come geometria, materiali e costi, racchiudendo informazioni a cui si può assegnare un valore o una descrizione.
- Core schema: in questo livello si trovano entità astratte che costituiscono la struttura di base, le relazioni fondamentali e i concetti comuni, necessarie per le successive definizioni specifiche delle diverse discipline (ad esempio *IfcActor*, *IfcTask*, *IfcElement*).

Tutte le *entity* qui presenti hanno un unico ID e derivano da *IfcRoot*.

La struttura proposta è ben definita, suddivisa in:

- *Control Extension*: raggruppa entità per il controllo e l'assegnazione degli oggetti;
- *Product Extension*: raggruppa le entità che descrivono i prodotti;
- *Process Extension*: vi fanno parte le entità che racchiudono le informazioni riguardanti processi, sequenze, pianificazione, e controllo dei lavori;
- *Kernel*: parte più astratta del *Core schema*. Racchiude entità che descrivono aspetti generali degli oggetti, proprietà e relazioni fra di essi, sequenza dei processi e raggruppamenti.
- Shared schema: classi che descrivono relazioni, processi e oggetti di risorse più specifici, condivisi da diverse discipline (ad esempio *IfcBeam*, *IfcColumn*, *IfcDoor*, *IfcFastener*, *IfcFurniture*). Le entità definite in questo livello possono essere referenziate e specializzate da tutte le entità sopra nella gerarchia.
- Domain schema: è il layer più alto, include le classi che afferiscono a specifiche discipline e che non possono essere ulteriormente specificate da altre classi (ad esempio *IfcAlarm*, *IfcCoil*, *IfcStructuralAction*). Queste definizioni sono tipicamente utilizzate per scambi e condivisioni di informazioni all'interno di uno stesso dominio disciplinare, organizzate secondo le varie discipline.

Nello schema successivo è possibile visualizzare i livelli appena descritti in ordine gerarchico:

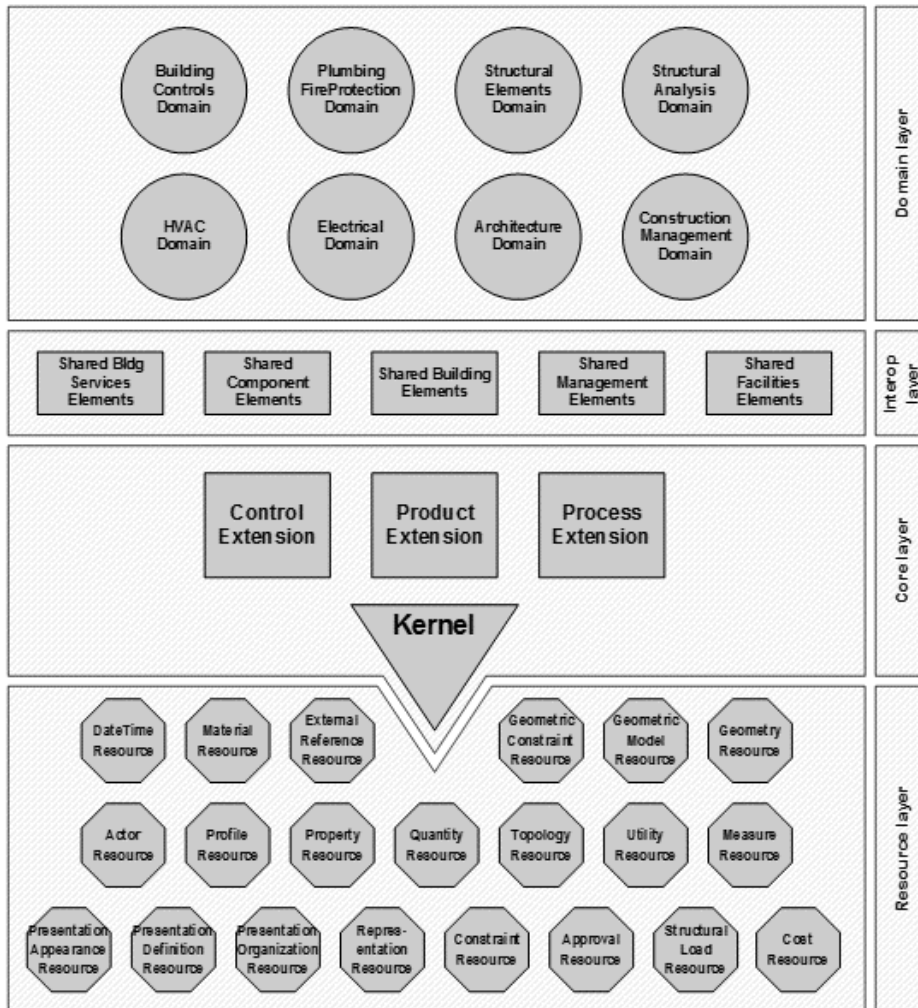


Figura 1.5: Organizzazione delle classi di IFC. (BuildingSMART International, 2020)

1.1.3 Formati file dello schema IFC

Il formato file di IFC si basa su STEP (Standard for the Exchange of Product model data), normato dalla ISO 10303. Questo standard per lo scambio di dati del modello è un modo per descrivere i dati per lo scambio di informazioni sul prodotto. Dichiara essenzialmente le definizioni delle risorse adottate dall'IFC, il motivo principale per l'utilizzo del formato EXPRESS e non, ad esempio, dell'Unified Modeling Language (UML) che è orientato agli oggetti: l'approccio IFC tiene conto della necessità di un modello più specifico del dominio nelle discipline edilizie.

A seconda della produzione e dell'utilizzo del modello, IFC definisce più formati di file che possono essere utilizzati, definendo varie codifiche dei medesimi dati presenti. Il tutto è consultabile in figura 1.6, la quale comprende anche i formati provvisionali e sperimentali.

IFC-SPF (Standard for the Exchange of Product model data) è un formato di testo definito nella norma ISO 10303-21 ("STEP-File"). Ogni riga di codice si compone di un singolo oggetto registrato e ha estensione ".ifc". È il formato IFC più utilizzato, con il vantaggio di avere

dimensioni compatte e una rappresentazione in formato decifrabile del data model in EXPRESS.

IFC-XML è un formato Extensible Markup Language (XML) definito nella norma ISO 10303-28 (“STEP-XML”), con estensione “.ifcXML”. Questo formato è sicuramente adatto per l’interoperabilità di strumenti XML e per lo scambio di modelli di edifici parziali, ma causa delle grandi dimensioni dei modelli tipici di un edificio, è meno pratico poiché risulta 3/4 volte più pesante di un file “.ifc”.

IFC-ZIP è un formato compresso caratterizzato da un file IFC-SPF incorporato, con estensione “.ifcZIP”. Questo rende possibile una riduzione del 70% del peso del formato STEP.

IFC-OWL fornisce una rappresentazione di linguaggio ontologico web (OWL) dello schema IFC. L’ontologia ifcOWL ha lo stesso stato degli schemi EXPRESS e XSD di IFC. Le ontologie ifcOWL sono generate direttamente dallo schema EXPRESS di IFC, mediante una procedura specifica di conversione, e si basano su una rete semantica e tecnologie di dati collegati (Linked Data). Il risultato è una rete di dati collegati che amplifica le opportunità per la gestione e interscambio dei dati.

Format	Extension	MIME Type	Text	Indexed	Size	Summary
Official						
STEP Physical File (SPF)	.ifc	application/x-step	Yes	No	100%	STEP Physical Format (SPF or IFC-SPF) is the most widely used format for IFC in practice, which is the most compact of the formats listed that can be read as text. IFC-SPF is based on the ISO standard for clear text representation of EXPRESS data models ISO 10303-21
Extensible Markup Language (XML)	.ifcXML	application/xml	Yes	No	113%	Extensible Markup Language (XML) provides enhanced readability and benefits from a broad range of software tools. ifcXML is based on the ISO standard for representation of STEP data in XML format ISO 10303-28
ZIP	.ifcZIP	application/zip	No	No	17%	IFC data may embedded within a ZIP file. The embedded data may be encoded as either SPF or XML, where the resulting size is typically comparable.
Terse RDF Triple Language (Turtle)	.ttl based on ifcOWL	text/turtle	Yes	No	1372%	More info on: ifcOWL
Resource Description Framework (RDF/XML)	.rdf based on ifcOWL	application/rdf+xml	Yes	No	816%	More info on: ifcOWL
Provisional/Candidate						
JavaScript Object Notation (JSON)	.json	application/json	Yes	No	148%	JSON provides enhanced readability and benefits from a broad range of software tools.
Hierarchical Data Format (HDF)	.hdf	application/x-hdf	No	Yes	n/a	HDF5 may store IFC data within hierarchical database, which provides high performance access to engineering data. HDF is based on the ISO standard for STEP data representation ISO 10303-26
Experimental/Unsupported						
SQLite	.sqlite	application/x-sqlite3	No	Yes	n/a	SQLite may store IFC data within a relational database, which provides indexed access to data within large models and benefits from a broad range of software tools.

Figura 1.6: Elenco dei formati ufficiali di IFC e loro caratteristiche. (BuildingSMART International, 2020)

1.2 L'Internet of Things

1.2.1 La nascita dell'IoT

Il termine "Internet of Things", abbreviato per semplicità con IoT, è relativamente nuovo ed è protagonista di un certo ronzio globale negli ultimi due decenni, ma l'idea di associare computer e reti per il monitoraggio e il controllo di dispositivi esiste da molti anni prima.

Alla fine degli anni Settanta, ad esempio, i sistemi per monitorare da remoto i contatori sulla rete elettrica tramite linee telefoniche erano già in uso commerciale. Negli anni Novanta, i perfezionamenti nella tecnologia wireless hanno consentito soluzioni aziendali e industriali *machine-to-machine* (M2M) per propagare il monitoraggio e il funzionamento delle apparecchiature. Numerose di queste prime soluzioni, tuttavia, erano fondate sull'utilizzo di reti chiuse, appositamente costruite, e di standard proprietari o specifici del settore, piuttosto che su reti basate su protocollo Internet (IP) e standard Internet, il che rappresenta un ostacolo per lo sviluppo delle stesse.

Nemmeno l'uso dell'IP per connettere oggetti comuni, diversi dai computer dunque, a Internet non è un'idea sviluppata negli ultimi anni e, anzi, le basi dell'IoT vantano esempi piuttosto stravaganti. Il primo *Internet Device* è un tostapane a che poteva essere acceso e spento su Internet tramite IP, presentato in una Conferenza su Internet nel 1990. Negli anni successivi si ricordano un distributore di bibite alla Carnegie Mellon University negli USA e una caffettiera nella Trojan Room dell'Università di Cambridge nel Regno Unito, la quale rimase collegata ad Internet fino al 2001.

L'espressione "Internet of Things" è coniata per la prima volta nel 1999 quando Kevin Ashton, pioniere tecnologico britannico, la usa per rappresentare un sistema in cui oggetti propri del mondo fisico potrebbero essere collegati a Internet tramite dei sensori.

Ashton ha co-fondato l'Auto-ID Center al Massachusetts Institute of Technology (MIT), in cui è stato creato il sistema standard per l'RFID, ossia la Radio Frequency Identification. Questa tecnologia permette l'identificazione automatica, fondata sulla propagazione nell'aria di onde elettro-magnetiche, permettendo la rilevazione univoca, hand free, massiva e a distanza di oggetti, persone e animali, siano essi statici o in movimento. L'ingegnere inglese ha inventato il termine IoT proprio per intendere il potere di collegare i tag RFID, utilizzati nelle catene di approvvigionamento aziendali, a Internet allo scopo di contare e tracciare le merci senza la necessità dell'intervento umano. Ashton, a riguardo, sostiene che:

Nel ventesimo secolo i computer erano essenzialmente cervelli senza i cinque sensi, limitandosi a eseguire ordini che noi gli davamo. Questa era una grande limitazione: nel ventunesimo secolo, grazie all'IoT, i computer possono essere più indipendenti, perché li abbiamo programmati per pensare e trovare soluzioni autonomamente a seguito di particolari

*impulsi.*²

Kevin Ashton

Il National Intelligence Council degli Stati Uniti, nel 2008, ha concepito l'IoT come una delle aree con una potenziale influenza sul potere nazionale globale degli Stati Uniti e l'ha inclusa nell'elenco delle "sei tecnologie civili dirompenti" che potrebbero emergere entro il 2025. (Jia et al., 2019) Queste vengono assimilate attraverso un processo di studio pilotato da analisti tecnologici della sede di SRIC-BI a Menlo Park, in California, e del suo ufficio europeo a Croydon, in Inghilterra.

Una tecnologia dirompente, secondo il NIC, non è altro che una tecnologia capace di causare un perfezionamento o un decadimento notevole, anche se precario, in uno degli elementi del potere nazionale statunitense: coesione geopolitica, militare, economica o sociale. Oltre l'IoT sono qui esaminate: la biogeronteologia, i materiali di accumulo di energia, i biocarburanti e prodotti chimici di base biologica, le tecnologie del carbone pulito e, infine, la robotica di servizio. (Intelligence Council, 2008)

Il concetto di IoT, dunque, non è del tutto nuovo ma è un argomento di nuova popolarità oggi. Questo è dovuto alla confluenza di diverse tecnologie e tendenze di mercato, le quali rendono pratica l'interconnessione di dispositivi in maniera via via più semplice e accessibile, economicamente parlando. I fattori sono molteplici:

- Connettività onnipresente: connettività di rete a basso costo, ad alta velocità e pervasiva, soprattutto attraverso tecnologia e servizi wireless con e senza licenza, rende quasi tutto interconnesso.
- Adozione propagata di reti basate su IP: l'IP è lo standard globale dominante per networking, fornendo una piattaforma di software e strumenti ben definita e ampiamente implementata che può essere incorporata in un'ampia gamma di dispositivi.
- Economia dell'informatica: trainata dagli investimenti dell'industria in ricerca, sviluppo e manifatturiero, la legge di Moore continua a fornire una maggiore potenza di calcolo a prezzi inferiori e un minor consumo energetico.
- Riduzione: i progressi della produzione consentono l'elaborazione e le comunicazioni all'avanguardia tecnologia da incorporare in oggetti molto piccoli. Insieme a una maggiore economia informatica, questo ha alimentato il progresso di dispositivi sensori piccoli ed economici, che guidano molte applicazioni IoT.
- Progressi nell'analisi dei dati: nuovi algoritmi e rapidi aumenti della potenza di calcolo, dell'archiviazione dei dati e dei servizi cloud consentono il raggruppamento, la correlazione e lo studio di grandi quantità di dati; questi set di dati concedono nuove opportunità per estrarre informazioni e conoscenze.

² <https://www.tiot.it/campi-di-applicazione-iot/iot-il-punto-di-vista-di-kevin-ashton/>

- Rise of Cloud Computing: il cloud computing, che sfrutta le risorse informatiche remote e in rete per elaborare, gestire e archiviare i dati, consente a dispositivi piccoli e distribuiti di interagire con potenti funzionalità di controllo e analisi back-end.

Dati questi concetti e questa rapida crescita, l' IoT disegna una convergenza di varie tendenze informatiche e di connettività, evolute per decenni. È diventato un fulcro tecnologico per il mondo accademico, per i settori industriali (automobilistico, elettronico casalingo e di consumo, sanitario, manifatturiero e tanti altri) e persino per le organizzazioni governative.

1.2.2 Definizioni e struttura dell' IoT

Nonostante questa diramazione globale del concetto, non esiste una definizione inequivocabile e universalmente accettata per il termine. Definizioni diverse vengono utilizzate da vari gruppi per descrivere o promuovere una visione particolare del significato dell' IoT e dei suoi attributi più importanti. Alcune definizioni specificano il concetto di Internet o di Internet Protocol (IP), mentre altre non se ne occupano nello specifico.

Il gruppo RFID definisce l'Internet delle cose come la rete mondiale di dispositivi, o meglio oggetti, interconnessi e orientabili in modo univoco sulla base di protocolli di comunicazione standard.

Le Nazioni Unite, in occasione di una conferenza a Tunisi nel 2005, prevedono lo sviluppo dell' IoT e affermano come i cambiamenti provocati da Internet saranno sottovalutati rispetto quelli provocati dalla rete di oggetti quotidiani.

Secondo i cluster di progetti di ricerca europei le *cose* non sono altro che partecipanti attivi nei processi aziendali, informativi e sociali. Qui sono abilitati a interagire e comunicare tra loro e con l'ambiente, scambiando dati e informazioni raccolti, reagendo senza vincoli agli eventi del mondo fisico e influenzandolo, eseguendo processi che attivano azioni e creano servizi con o senza intervento umano diretto.

Altri ancora definiscono l'ambiente intelligente come un ambiente capace di utilizzare le tecnologie dell'informazione e della comunicazione per rendere più consci, interagenti e proficui i componenti e i servizi infrastrutturali critici dell'amministrazione, dell'istruzione, della sanità, della sicurezza pubblica, del settore immobiliare, dei trasporti e dei servizi pubblici di una città. (Gubbi et al., 2013)

Se si guarda alla struttura dell'Internet of Things si vede come questo sia fondato su tre paradigmi: orientato a Internet (middleware), orientato alle cose (sensori) e orientato alla semantica (conoscenza). Questo tipo di descrizione si rivela necessario a causa della natura interdisciplinare dell'argomento e l'utilità dell'IoT può essere resa libera solo in un dominio applicativo in cui questi paradigmi si intersecano, come mostrato nell'immagine seguente. (Atzori et al., 2010)

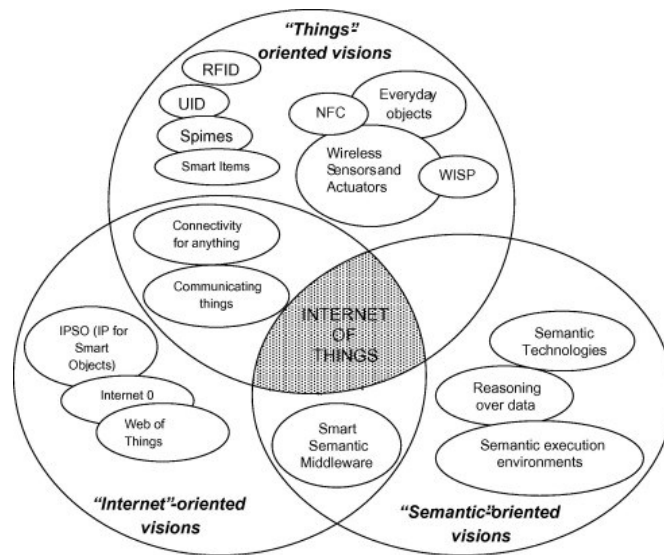


Figura 1.7: Paradigma "Internet of Things" come risultato della convergenza di visioni diverse (Atzori et al., 2010)

È lecita allora la confusione apparente odierna a riguardo, la quale è una diretta conseguenza del nome stesso "Internet of Things". Guardando alla sintattica questo è composto da due termini: il primo si dirige verso un'ottica *network oriented*, mentre il secondo si concentra a focalizzare gli oggetti, a livello generico, da integrare in un framework comune.

Per molte istituzioni la questione dell'IoT sarà fortemente influenzata dall'intelligenza dei dispositivi, la quale va radicalmente implementata. Da qui è di conseguenza emerso un neologismo fondamentale: *spime*. Questo rappresenta un oggetto che può essere segnalato attraverso lo spazio e il tempo per tutta la sua esistenza e che sarà sostenibile, migliorabile e distinguibile in modo non ambiguo. Nonostante sia un concetto ancora in via di definizione, si trovano alcuni esempi negli *Smart Items*: una sorta di sensori dotati di potenzialità superiori rispetto a quelle basiche (comunicazione wireless, memoria ed elaborazione dati). Essi sono contrassegnati, ad esempio, da un comportamento autonomo e dinamico, consapevolezza del contesto, capacità comunicativa ed elaborazione di cooperazione.

L'associazione delle parole "internet" e "cose" introduce un livello di innovazione travolgente nel mondo delle Information and Communications Technology, ICT, di oggi. Le differenze tangibili nelle visioni di IoT stanno di fatto negli interessi e nelle finalità degli stakeholder i quali, a seconda dei vari sfondi, iniziano a considerare il concetto da una prospettiva orientata alle cose oppure a Internet. (Atzori et al., 2010)

Prestando ora attenzione alla semantica si vede come l'Internet delle cose rappresenti esattamente quanto detto dal gruppo RFID: una rete globale di oggetti interconnessi indirizzabili in modo univoco, fondata su protocolli di comunicazione standard, il che porta alla formazione di un'infrastruttura di rete globale dinamica ed eterogenea con capacità di autoconfigurazione. L'indirizzamento esclusivo dell'oggetto, la riproduzione e apprendimento mnemonico delle informazioni scambiate diventano le questioni più concorrenti, portando alla prospettiva "Semantic oriented" dell'IoT.

In questo senso, l'IoT potrebbe supportare in modo efficace la futura evoluzione di BACS e TBM. Per di più, in un orizzonte di risorse limitate, i sistemi infrastrutturali (Internet, Smart Grid, ecc.) dovranno necessariamente essere in grado di interagire positivamente con gli utenti finali e l'ambiente costruito, al fine di massimizzare l'efficienza complessiva e minimizzare costi e rischi di guasti e interruzioni.

1.2.3 Domini applicativi e implicazioni dell'IoT

L'Internet of Things è una tecnologia dalle potenzialità apprezzabili, le quali auspicano allo sviluppo di innumerevoli applicazioni, ma solo una parte esigua è ora disponibile nella società odierna. Queste ultime possono dipendere dal tipo di disponibilità della rete, dalla copertura, scala, dall'eterogeneità, dalla ripetibilità e dal coinvolgimento degli utenti, senza dimenticare l'impatto ambientale.

I domini di applicazione e gli ambienti in cui i nuovi tecnicismi potrebbero ottimizzare la qualità di vita sono molteplici: a casa, in viaggio, in caso di malattia, al lavoro e in palestra, per riportarne alcuni. I contesti citati sono abilitati ad un tipo di intelligenza primordiale e molto spesso sono incapaci di comunicare con gli utenti. Renderli adeguati alla comunicazione e all'elaborazione dei dati e delle informazioni percepite dall'ambiente comporta avere spazi diversi in cui è possibile implementare una serie di applicazioni molto più vasta.

Nella figura 1.8 proposta di seguito si possono distinguere i principali domini di applicazione:

- Trasporti e logistica
- Sanitario
- Ambiente intelligente
- Ambito personale e sociale
- Applicazioni futuristiche

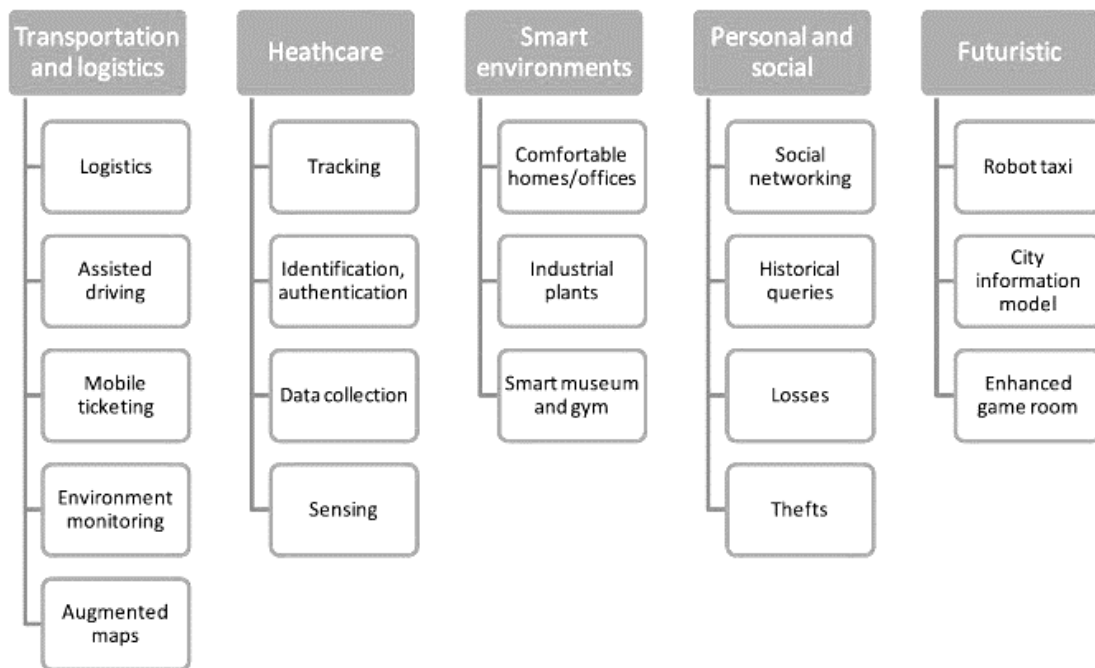


Figura 1.8: Domini applicativi e scenari principali rilevanti dell'IoT. (Atzori et al., 2010)

Un ambiente intelligente è quello in grado di rendere la vita dell'utente facile e confortevole grazie all'intelligenza degli oggetti contenuti, sia esso un ufficio, una casa, uno stabilimento industriale, o un ambiente di svago. Per il caso studio sono considerati, nello specifico, l'ambiente intelligente e le sue sfumature nei contesti.

Partendo dalle installazioni di sensori e attuatori nelle case, come peraltro negli uffici, queste sono in grado di restituire una vita più agevole agli utenti sotto molteplici aspetti: il grado di illuminazione della stanza può mutare di ora in ora, il riscaldamento o raffreddamento può essere arrangiato in base al clima esterno e alle proprie predilezioni, incidenti o infortuni casalinghi possono essere scongiurati grazie ad opportuni sistemi di monitoraggio e allarme, basti pensare a quelli per la rilevazione del fumo. È, inoltre, attuabile un sistema di risparmio energetico grazie all'automatico spegnimento degli elettrodomestici quando non necessari per l'utilizzo. Una logica di automazione intelligente è anche capace di ottimizzare i costi di consumo energetico durante la giornata, indagando sui momenti in cui i prezzi, forniti dai servizi esterni, sono più convenienti e avendo cura delle esigenze specifiche delle apparecchiature.

Anche gli stabilimenti industriali possono essere migliorati dalla presenza di automazione negli impianti, in particolare con un largo investimento di tag RFID affiliati ai settori di produzione. Si pensi ad aggiungere un tag alla fine di un nastro di produzione generico, capace di comunicare il punto di lavorazione associata: questo viene decifrato da un lettore RFID, il quale genera un evento da pubblicare in rete. La macchina produttiva associata riceve una notifica e legge l'evento in questione, di conseguenza preleva la parte di produzione in autonomia. Un altro esempio può risiedere nella capacità di controllo qualità: installando sensori wireless in macchine specifiche, questi possono monitorare la vibrazione della stessa e

produrre un allarme per fermare nell'immediato il processo di produzione se l'oscillazione supera una soglia prestabilita. A tutto ciò sono accomunate reazioni posteriori dell'intero impianto di produzione industriale, comprese notifiche di emergenza al personale responsabile. Quest'ultimo giova per molti aspetti della presenza di tag e sensori poiché può controllare nell'immediato lo stato degli ordini, lo sviluppo della produzione, lo stato delle apparecchiature e, in generale, ha una visione completa di tutti gli elementi che compongono l'impianto e dei possibili effetti collaterali che interessano la linea di produzione.

Nell'ambito del tempo libero e delle relazioni sociali, esempi rappresentativi delle tecnologie dell'IoT si trovano in ambienti come la palestra: determinate macchine dotate di tag RFID sono capaci di riconoscere gli utenti che le utilizzano, modificando il loro funzionamento in base al profilo personale ad esso collegato. È possibile, in questa occasione, monitorare i parametri vitali durante la sessione di allenamento e controllare se il fruitore è in sovrallenamento o troppo rilassato per evitare l'esecuzione errata degli esercizi e per scongiurare problemi di salute, garantendo un allenamento adeguato.

Passando ad esempi che interessano la singola persona, può essere utile un motore di ricerca per gli oggetti smarriti basato sull'applicazione RFID sul web. Contrassegnare oggetti specifici, come chiavi di casa, laptop o altri effetti personali, permette di risalire all'ultima loro posizione registrata oppure di cercare la posizione particolare nel momento, utile in caso di notifica di spostamento dell'oggetto da un'area riservata senza il consenso del proprietario e, dunque, di probabile furto. In questo caso il sistema avvisa nell'immediato il detentore o gli addetti alla sicurezza che l'oggetto rubato lascia la sua posizione senza una precedente autorizzazione. Il tutto si può implementare pensando di notificare il destinatario quando la posizione degli oggetti interessati soddisfa determinati requisiti.

In sintesi, le conseguenze dell'IoT sono duplici:

- Integrazione della capacità di rilevamento, archiviazione, rete, programmazione ed elaborazione dati in oggetti di uso quotidiano e casalingo come elettrodomestici, finestre, luci, rilevatori di fumo, e loro pubblicazione online, nonostante originariamente questi non siano prodotti con tale facoltà. Ciò è contrario alla maggior parte dei dispositivi all'odierno su Internet e progettati da principio per farne parte come smartphone e laptop, ad esempio.
- Integrazione di reti che comprendono gli oggetti sopra menzionati, i quali sono così accessibili tramite la rete.

1.2.4 Architettura dell'IoT

Per quanto concerne la composizione di un sistema IoT, questo è formato da cinque principali costituenti che si distinguono a seconda del contributo e della funzione, come mostrato nella figura seguente. Si riconoscono allora:

- Dispositivi o sensori (terminale)

- Reti (infrastruttura di comunicazione)
- Cloud (deposito e infrastruttura di elaborazione dei dati)
- Analytics (algoritmo computazionale e di data mining)
- Attuatori e interfacce utente (servizi a disposizione)

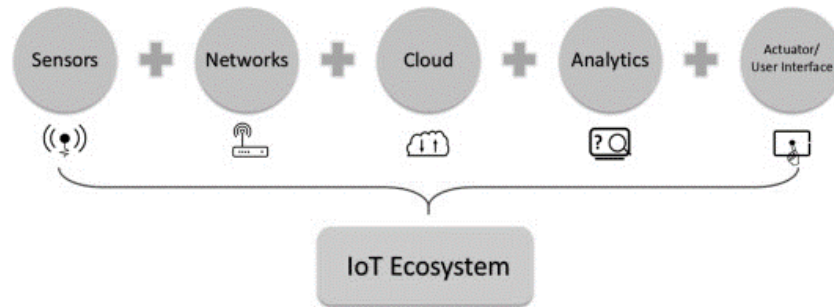


Figura 1.9: Componenti di un sistema IoT. (Jia et al., 2019)

L'architettura di un sistema IoT è normalmente frazionata su una base a strati e i ricercatori, a seconda delle esigenze dei propri utenti, propongono modelli differenti. Alcuni schemi comuni si compongono di tre strati, altri a cinque strati, possono essere basati su Service-Oriented Architecture (SOA) o su middleware. Per l'ambito edile l'architettura a tre strati è la più applicabile, più adatta ed è, soprattutto, sufficiente per l'adozione efficace dell'IoT per lo sviluppo delle funzionalità generali.

L'architettura a tre livelli dell'IoT, rappresentata in figura 1.10, è composta da:

- livello di percezione
- livello di rete
- livello di applicazione

Lo strato di percezione include i nodi e le reti di percezione, è inserito nel target ed è responsabile della ricerca e della raccolta dei dati e delle informazioni nel mondo reale. È ordinariamente descritto da tecnologie di rilevamento, come i sensori, e di attuazione.

Il livello di rete è il livello più importante, considerabile come il nucleo tecnico del sistema. È adibito al trasporto, all'elaborazione e alla trasmissione dei dati grezzi ottenuti dal livello inferiore di percezione. È sede della convergenza di vari dispositivi, come ad esempio gateway, e l'infrastruttura di comunicazione. Per trasmettere i dati si serve di due principali modalità di connessione: comunicazioni cablate o wireless.

Infine, il livello di applicazione è il livello che funge da interfaccia front-end per restituire analisi e risultati decisionali ai fruitori del servizio in domini correlati. Prende i dati trasmessi e li presenta agli utenti per implementare ulteriori servizi. In questo livello le tecnologie abilitanti variano in base al dominio applicativo del sistema IoT.

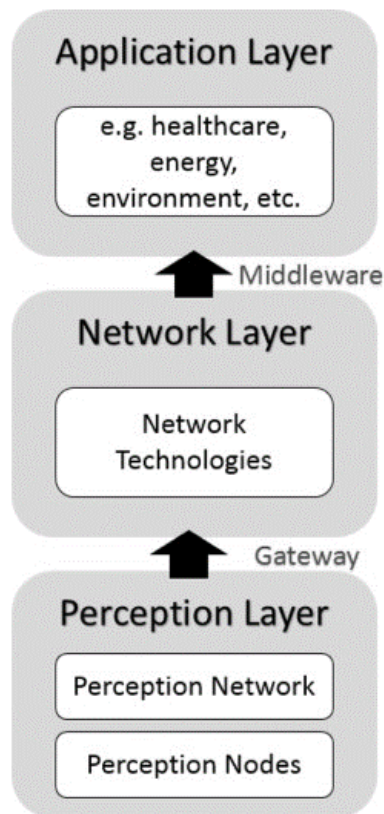


Figura 1.10: Architettura IoT a tre strati (Jia et al., 2019)

Capitolo Secondo

LA BUILDING AUTOMATION E LE TECNOLOGIE DI COMUNICAZIONE

1 CONCETTI GENERALI DEI SISTEMI DI AUTOMAZIONE E CONTROLLO DEGLI EDIFICI

I sistemi di automazione e controllo degli edifici si possono immaginare come l'insieme di varie apparecchiature, siano essi sistemi HVAC, di illuminazione o motorizzazione. Le apparecchiature citate, a loro volta, sono scomponibili, nel senso che elementi più complessi sono composti da elementi più semplici fino ad arrivare a unità elementari non ulteriormente scomponibili che saranno trattate in seguito.

È fondamentale porre attenzione alla diversificazione dei significati di apparecchiatura e dispositivo. Quest'ultimo è un'apparecchiatura atta ad accertare, attuare o coordinare l'azione esplicitamente sull'ecosistema in cui si trova oppure su altre apparecchiature. Si definisce, inoltre, un dispositivo di controllo, il quale è collegabile ad un bus di controllo o ad una rete per trasmettere o captare istruzioni per il controllo. Sono capaci di comunicare tra loro per mezzo di porte logiche indirizzabili predeterminate, comunemente chiamate *datapoint*. Un esempio si trova nei sensori, negli attuatori, nei controllori logici, nei gateway e nei contatori.

Concetti come datapoint, endpoint, tag o punti hanno nel complesso definizioni diverse, ciò nonostante, a loro modo tutti delineano le entità come un punto, il quale è indirizzabile di influenza reciproca tra il sistema di controllo e i suoi oggetti di dominio.

La totalità dei datapoint possiede dei metadati associati, i quali completano un insieme di regole riguardo l'interazione dello stesso datapoint. Qui normalmente si trovano le informazioni riguardanti il tipo di accesso, la posizione di installazione, il tipo di dati, la zona di influenza e la velocità di aggiornamento del dato per le operazioni di lettura e scrittura.

Esistono due possibilità di descrizione dei datapoint: fisica o virtuale. Nel primo caso si parla di un punto dati fisico direttamente relazionato ad un dispositivo collegato al sistema, come possono essere le porte input/output di un dispositivo, le quali possono tutte essere catalogate come un punto dati. Dall'altra parte, un datapoint virtuale si comporta come un metodo di indirizzamento di oggetti virtuali come servizi procurati dai dispositivi: un punto dati che guida un parametro di configurazione di un certo dispositivo, in modo che la scrittura del determinato punto influenzi il parametro di configurazione ad esso associato. Questo tipo di datapoint è utilizzato anche per indirizzare i dispositivi. I punti dati virtuali possono essere utilizzati anche per indirizzare i dispositivi *stub*³ a scopo di test.

³ Uno stub è una porzione di codice usata al fine di simulare il comportamento di funzionalità software.

I punti di dati possono essere capaci di lettura, scrittura o entrambi: i datapoint leggibili sono quelli di sola lettura e tipicamente sono riferiti ai dispositivi come i sensori; quelli scrivibili sono di sola scrittura e si riferiscono agli attuatori: nella pratica, la scrittura sui datapoint corrisponde all'aggiornamento dello stato del sistema; infine, i datapoint che sono capaci garantire ambedue i servizi si possono impiegare per la scrittura di aggiornamenti nei dispositivi e per leggere lo stato interno del dispositivo, ad esempio, per sondare lo stato di attivazione o disattivazione dello stesso.

Di norma a ciascun punto di dati è accostato un determinato tipo di dati, il quale suggerisce alle applicazioni client come sono predisposte le informazioni quando esse decifrano un datapoint e come queste devono essere impostate durante la scrittura su quel datapoint. In alcuni sistemi, i datapoint con il medesimo tipo di dati vengono collegati tra loro in un processo noto come *binding*⁴. Ciò comporta che, quando un punto dati viene aggiornato, i punti dati ad esso collegati vengono avvisati e possono eseguire un'azione, come ad esempio acquisire il valore equivalente.

I datapoint fisici sono sempre associati a dei dispositivi collocati in una posizione fisica di un edificio. La conoscenza della locazione di un punto dati è fondamentale, soprattutto nel caso in cui esso sia riferito ad un sensore. Inoltre, i datapoint possiedono anche una zona di influenza, la quale rappresenta lo spazio coinvolto dalle loro attivazioni, che in alcuni casi non combacia con il luogo di installazione: un sistema HVAC di norma risiede in una stanza specifica dell'edificio e concerne anche molte altre stanze, mentre un dispositivo come una lampada riguarda la sola stanza in cui è installato.

1.1 Architettura di sistema

Come già accennato precedentemente, si può definire un BACS come un sistema distribuito e orientato al controllo e alla gestione computerizzata delle prestazioni di un edificio. Tale sistema distribuito si organizza, generalmente, in tre livelli:

- Field Layer: noto anche come livello di campo, è lo strato più basso e qui si realizza l'interazione delle varie apparecchiature come sensori e attuatori, chiamati dispositivi di campo e relazionati tra loro tramite cablaggio, con l'ambiente costruito. Vengono ricavati i dati ambientali (misurazioni, conteggi) e plasmati in una corrispondenza idonea alla trasmissione ed elaborazione. In egual modo, sono controllati dal personale i parametri dell'ambiente in risposta ai comandi riscossi dal sistema.
- Automation Layer: è lo strato intermedio dove vengono coordinati i dispositivi di controllo a livello di campo, ovvero sono sviluppate le misurazioni e raccolti i dati prodotti dai dispositivi del livello inferiore, e si stabiliscono le logiche e i

⁴ Processo attraverso il quale viene ultimato il collegamento fra una entità di un software ed il suo corrispettivo valore.

loop di controllo prefissati. Se necessario le apparecchiature possono comunicare tra loro valori di interesse globale, in questo caso si parla di un tipo di scambio dati per comunicazione orizzontale.

- **Management Layer:** è il livello di gestione, il quale riceve i rapporti dai controller e ne governa la configurazione, la registrazione e l'archiviazione. È dotato di interfacce utente per le mansioni di amministrazione manuale come il monitoraggio dello stato di sistema e la configurazione delle variabili di sistema, l'accesso ai dati passati e alle notifiche di allarme. Altro compito destinato a questo livello è quello di garantire l'interoperabilità tra gli svariati sistemi e le tecnologie adoperate dai livelli inferiori. Per far ciò, è necessario scandire la copertura delle funzionalità dei sistemi.

I livelli citati rispecchiano i molteplici aspetti funzionali di BACS e il sistema di funzionamento, come si può osservare in fig. 2.1, è piramidale, ovvero i livelli superiori gestiscono gli inferiori e la quantità di dati presenti all'interno di un certo dispositivo cresce salendo di livello.

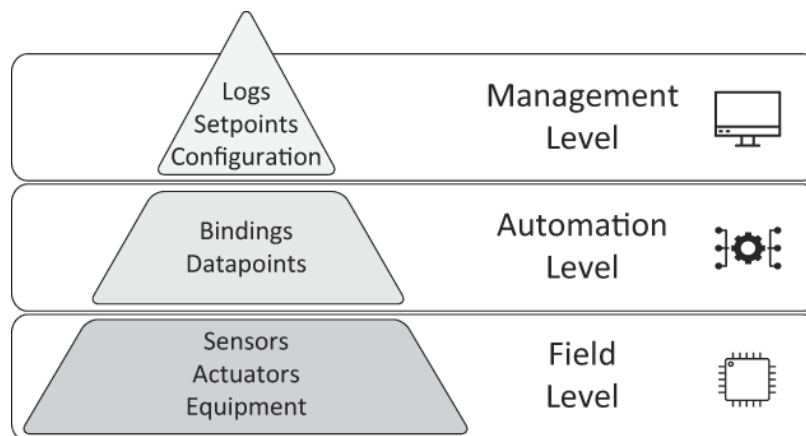


Figura 2.1: Stacking dei livelli gerarchici di BACS (Vieira et al., 2020)

Il livello di campo è un livello distribuito per natura, così come può esserlo quello di automazione. Ciò garantisce vari vantaggi al sistema come la riduzione di eventuali latenze nei processi di controllo e il rischio di colli di bottiglia nelle prestazioni e la possibilità di evitare singoli punti di guasto. In più consente operazioni di manutenzione programmata senza influire su tutto l'organismo, mettendo fuori servizio sottosistemi localizzati. La produzione di questo tipo di sistemi distribuiti è però più difficoltosa rispetto a quelli centralizzati.

I sistemi di automazione più moderni e sviluppati sono propensi a separare la logica di automazione dall'interfaccia utente mediante astrazioni indirizzate ai servizi, procurando un accesso adattabile al BACS da diverse piattaforme e ubicazioni.

1.2 Sensori, attuatori e controller

Un BACS è composto, come già detto, da apparecchiature elementari quali attuatori, sensori, moduli hardware e controller.

Gli attuatori sono meccanismi capaci di reagire ai segnali esterni al fine di chiudere circuiti o variare l'intensità dei carichi elettrici, un esempio fisico sono le plafoniere e le tapparelle. Spesso questi vengono scambiati con i carichi elettrici o con lo stesso strumento fisico che mettono in moto, ad esempio, un attuatore per tapparelle è, nello specifico, un attuatore a motore associato poi ad una tapparella.

I sensori, invece, sono capaci di convertire le realtà fisiche in segnali misurabili dalle attrezzature. Alcuni dispositivi sono adibiti sia alla rilevazione sia all'attivazione, ma in generale e per semplicità di catalogazione si dividono in due gruppi distinti: quelli in grado di rilevare e quelli in grado di attivare. Infine, entrambi i dispositivi sono allacciati alle porte di input e output (I/O) dei moduli hardware che generano dei segnali elettrici a seconda dei comandi di uscita digitali e creano letture dai segnali di ingresso.

L'influenza reciproca tra i sensori e attuatori deve essere strumentata per mezzo di un certo tipo di logica di controllo. Quest'ultima si colloca all'interno di componenti noti come controller. Nei sistemi di controllo e automazione degli edifici, un controller è normalmente composto da un hardware caratteristico per l'applicazione con un software integrato, il quale ispeziona assiduamente gli attuatori fisici, come luci e tapparelle, a seconda della retroazione fornita dagli ingressi monitorizzati, come sensori di illuminazione o di occupazione, o riscuotendo comandi dal sistema.

In termini applicativi, i controller esibiscono differenti campioni di oggetti logici che possono essere letti o anche scritti. Al variare della ricercatezza del controller, questo può essere capace di adempiere più di un programma di controllo simultaneamente, leggere e scrivere porte I/O o conversare con altri controller sulla rete dello stesso bus di campo.

Il sistema logico della funzione di controllo può presentare disparati livelli di sofisticatezza, che vanno da semplici condizioni binarie o blocchi funzione, a espressioni matematiche o algoritmi più artificiosi. A seconda della complessità delle funzioni di controllo i controllori possono essere identificati come Controllori a Logica Programmabile (PLC) o Controllori Digitali Diretti (DDC). I primi, in linea di massima, predispongono di funzioni più semplici e inflessibili che richiedono poca o a volte nessuna configurazione, mentre i secondi sono più elastici e di norma attivano funzioni che richiedono una configurazione più articolata, come la programmazione o la coordinazione degli scenari.

Le funzioni di controllo possono essere ottemperate anche in software non incorporati in esecuzione in un server, accentrando così le funzioni di ispezione di una building automation nel livello di gestione, dove la logica di controllo può fare uso di dati aggregati nativi da differenti segmenti del bus di campo. Il vantaggio di avere questo tipo di controller sta nella capacità di prendere decisioni più articolate e di perlustrare dati aggiuntivi conseguiti dal sistema.

Nella figura che segue è disponibile un quadro generale di quanto detto finora per sistemi di automazione con gerarchia funzionale a tre livelli.

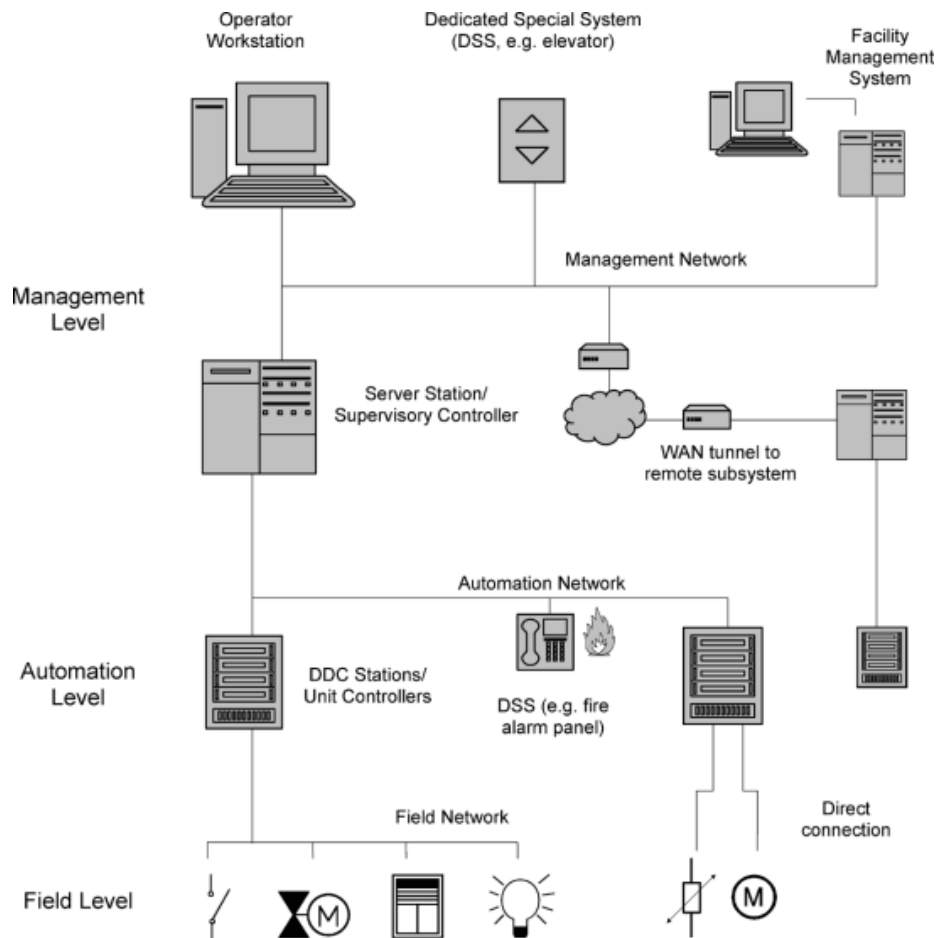


Figura 2.2: Gerarchia funzionale a tre livelli di un sistema di automazione. (Kastner et al., 2005)

Partendo dal basso, ovvero dal livello di campo, si vedono sensori e attuatori collegati in modo diretto ai controller del livello successivo attraverso delle interfacce standard o una rete di campo. Il controllo del processo è compiuto dalle stazioni DDC, le unità di controllo. Salendo ancora, una stazione server si occupa del controllo di supervisione, della registrazione e dello studio dell'andamento di un gruppo di unità di controllo. I controller di supervisione e le unità di controllo sono allacciati tramite una propria rete di automazione, dove possono essere connessi anche i sistemi speciali dedicati (DSS). Le workstations utilizzano i dati preparati dalle stazioni server per presentare l'interfaccia utente e anche i sistemi speciali dedicati che non devono essere integrati in uno stretto schema di controllo automatico possono essere vincolati a questo livello. Questa unione comporta la possibilità di conseguire la visualizzazione di un'unica stazione di lavoro per tutti i sistemi presenti.

Ad ogni modo, i dati di misurazione e, in generale, la totalità degli altri dati sull'utilizzo possono essere trasportati in database a livello aziendale, come sono i sistemi CAFM (*Computer Aided Facility Management*), per il mantenimento predittivo e la ripartizione dei costi. Per

quanto riguarda le stazioni remote, queste sono inquadrare nella rete di gestione su richiesta per mezzo di una connessione dial-up, oppure di un tunnel WAN, quando è ritenuto necessario lo scambio di dati. I messaggi di avviso possono essere trasmessi all'operatore attraverso messaggi al cellulare o posta elettronica.

1.3 Modello del dispositivo

Rispetto ad un sistema di automazione, i dispositivi ad esso collegati presentano due interfacce: un'interfaccia elettrica, che dimostra come connettere il dispositivo al resto del sistema, e un'interfaccia applicativa che autorizza gli altri dispositivi e le applicazioni software di interagire con il dispositivo stesso per mezzo di punti dati esposti.

L'hardware di automazione è profondamente eterogeneo e tende ad essere astratto in modo da permettere alle applicazioni software di essere il più indipendenti possibile dalle specificità dell'hardware in questione.

Un driver di dispositivo è una parte costituente dedicata al controllo dei dispositivi collegati a un sistema, allo stesso tempo fornisce un livello di astrazione che ne agevola il funzionamento. Ogni driver di dispositivo si collega a un particolare tipo di dispositivo, ciò implica che se un sistema deve supportare un centinaio di dispositivi diversi, questo deve installare centinaia di driver di dispositivo. Data l'innumerabile diversità di dispositivi ad oggi disponibili, la creazione di un driver di dispositivo generico, che si adatta dunque a tutti i dispositivi con interfacce compatibili, diventa una vera e propria sfida. La compatibilità tra dispositivi in questione è data se alcune proprietà di un'interfaccia sussistono nell'altra e se condividono lo stesso tipo di dati.

È possibile un'ulteriore diversificazione tra driver di dispositivo hardware e driver di dispositivo software.

I driver dei dispositivi hardware sono composti da moduli hardware con porte I/O e un microcontrollore: le porte I/O sono congiunte a dei dispositivi fisici che in generale non godono di intelligenza e che sono innestati direttamente tramite dei segnali elettrici, come un termistore, una lampada, o la ventola di un deumidificatore; il microcontrollore è responsabile della gestione pilotata di questi dispositivi e della presentazione di un'interfaccia che può essere impiegata per instradare ciascuna porta I/O per fini di lettura e scrittura. Questo modulo hardware può essere collegato a un computer, a una rete o anche a un altro modulo hardware.

I driver dei dispositivi software, invece, sono formati dalle interfacce di programmazione di quelle applicazioni adoperate per permettere a ulteriori applicazioni di scambiare dati con vari dispositivi. Tali driver di dispositivo sono usati per due diversi motivi: per convertire i segnali elettrici, recepiti in valori registrati, in variabili o in oggetti capaci di essere decifrati da applicazioni software e, secondo, tramutano i valori registrati in segnali elettrici che manovrano gli attuatori.

In conclusione, è possibile affermare che i dispositivi fisici sono codificati in driver di

dispositivo hardware procurando una quota di astrazione maggiore, altrimenti su driver di dispositivo software che rappresenta un livello di astrazione hardware. Quest'ultimo consente di prestare servizio ad altre applicazioni software con la facoltà di far operare questi dispositivi, senza porre l'attenzione in dettagli hardware di basso livello, come possono essere i protocolli di comunicazione dei vari dispositivi.

2 STANDARD PER LA COMUNICAZIONE DEI SISTEMI DI AUTOMAZIONE

La building automation è stata, fin dal principio, popolata da una pletera di soluzioni proprietarie per quanto concerne le tecnologie e gli standard⁵ per la comunicazione. Alla base di questo si trovano i requisiti propri delle prestazioni che incentivano ancora un tipo di approccio pensato ad hoc; tuttavia, data la domanda notevole nel mercato e la necessità di adottare sistemi aperti, ad oggi i maggiori produttori stanno progressivamente desistendo dalla produzione di progetti proprietari.

I diversi protocolli informativi raffigurano, contraddistinguono e relazionano concetti di uno specifico dominio appoggiandosi a un modello di dominio diffuso in cui le varie applicazioni sono in grado di condividere le informazioni. Da quest'ultime sono ricavati i dettagli di basso livello inerenti alle particolarità dei materiali e di rete, come le architetture dei dispositivi e i dati del protocollo.

Al giorno d'oggi è ordinariamente utilizzata la modellazione di strutture dati come oggetti, incluse le strutture del dominio di controllo. A tale scopo vengono utilizzati diversi standard per i sistemi distribuiti orientati agli oggetti attraverso l'internet, rendendoli così indicati per l'uso nei gateway del livello applicativo. Tali standard per lo scambio di informazioni su reti IP sono forniti dalla Common Object Request Broker Architecture (CORBA), dall'interfaccia Java Remote Method Invocation (RMI), dal Microsoft Distributed Component Object Model (DCOM) o dal Simple Object Access Protocol (SOAP) utilizzando la notazione XML, tuttavia, tutte le tecnologie citate sono soluzioni proprietarie.

Tra i primi standard aperti orientati agli oggetti per l'accessibilità ai dati di processo e tra i più utilizzati dai fornitori spicca l'OPC, acronimo di *Open Process Control* o anche *Open Platform Communication*. Questo è uno standard di interoperabilità, basato su DCOM⁶ per lo scambio sicuro e affidabile di dati di automazione industriale, ma ampiamente usato al momento anche nel campo edile. Tale standard nasce con la volontà di estrapolare i protocolli di comunicazione dei controllori logici programmabili (PLC), come linguaggio indipendente dalle piattaforme e, inoltre, garantisce il flusso continuo di informazioni tra i dispositivi di più

⁵ Gli standard che riguardano la tecnologia dei BACS sono principalmente creati negli USA e in molteplici organismi di standardizzazione europei e internazionali. (Kastner et al., 2005)

⁶ Estensione del Component Object Model (COM), creata da Microsoft, che consente ai componenti software di comunicare tra loro su diversi computer su una rete locale (LAN), su una WAN (Wide Area Network) o su Internet.

fornitori.

OPC vede l'unione di varie specifiche sviluppate dai fornitori del settore, dagli utenti finali e dagli sviluppatori di software. Tali specifiche determinano l'interfaccia tra client e server, ma anche tra server e server; tra i servizi implementati si trovano l'accesso ai dati in tempo reale, il monitoraggio di allarmi ed eventi, l'accesso ai dati storici e altre applicazioni.

I modelli di automazione aperti che vantano la capacità innata di coprire le applicazioni dell'automazione degli edifici nella loro totalità sono BACnet, LonWorks, ZigBee e EIB/KNX. I primi hanno guadagnato un'importanza degna di nota a livello mondiale, mentre gli ultimi due spiccano piuttosto nel mercato europeo; tutti sono scelti per un tipo di soluzione di sistema completa.

2.1 BACnet

Il Building Automation and Control Networking protocol (BACnet) è stato sviluppato a partire dal 1987 da un comitato di progetto istituito dall'American Society of Heating, Refrigeration and Air-Conditioning Engineers, detta ASHRAE, quando questi non riuscirono a trovare un protocollo esistente che soddisfacesse in modo appagante tutti i criteri che i membri del comitato ritenevano necessari per una comunicazione standard adeguata. BACnet, allora, si sviluppa a livello industriale per soddisfare le esigenze di comunicazione dei sistemi di automazione per diverse applicazioni, come ad esempio: riscaldamento, controllo dell'illuminazione e sistemi di rilevazione incendi.

Lo sforzo di sviluppo è stato portato ad una prima soddisfazione nel 1995, quando BACnet viene pubblicato per la prima volta come standard americano ANSI/ASHRAE. Successivamente, nel 2003 BACnet è adottato sia come standard CEN che ISO all'interno della serie ISO 16484 e diventa uno standard nazionale dei paesi membri dell'Unione Europea ai sensi dei regolamenti CEN. Infine, è stato anche scelto come standard nazionale dalla Corea, Russia, Cina e Giappone, il che evidenzia quanto tale protocollo riesca ad attrarre l'interesse dei produttori a livello globale.

La specifica BACnet è in continua manutenzione e ulteriore sviluppo. Tutto ciò è gestito dall'ASHRAE Standing Standard Project Committee 135 (SSPC). I membri di SSPC rappresentano i settori dell'industria nella loro completezza, ma in generale la partecipazione al processo di sviluppo è completamente aperta. Tutte le parti interessate sono incoraggiate attivamente a fornire commenti e suggerire modifiche.

In quanto protocollo di comunicazione aperto ed esente da ogni tipo di licenza e canone, BACnet mira a risolvere i problemi di interoperabilità tra i diversi fornitori di dispositivi creando un tipo di linguaggio di base comune e plasmando le informazioni scambiate. Il tutto si basa su un tipo di organismo dati orientato, con le proprie informazioni di scambio, attorno agli oggetti BACnet stessi, creando una struttura object-oriented. Ciò significa che il complesso di elementi dati si articola a partire dagli oggetti, e affinché le apparecchiature funzionino

insieme, i componenti devono essere in grado di scambiare e comprendere i messaggi BACnet.

Le specifiche del protocollo, come ad esempio la sua architettura orientata agli oggetti e le sue proprietà sono illustrate dettagliatamente nel paragrafo 3 del Capitolo Secondo.

2.2 KNX

Lo standard KNX nasce dalla convergenza di diverse associazioni europee operanti nel settore dell'automazione e della gestione di sistemi intelligenti.

Alcuni produttori di origine tedesca, nel maggio del 1990 a Bruxelles, si riunirono fondando l'EIBA, l'*European Installation Bus Association*. In tale occasione proposero un nuovo standard aperto basato su norme ISO, in grado di collegare quei dispositivi dotati di una propria intelligenza senza la necessità di un controller centrale, chiamandolo EIB. L'obiettivo principale era quello di creare un concetto unificato di impianto elettrico e di gestione degli edifici di tutte le dimensioni, separando la trasmissione delle informazioni di controllo dal cablaggio di rete tradizionale.

Seguendo lo stesso principio, la società francese formata da Merlin, Gerin, Airlec, EDF, Landis&Gir, nel 1989 creò un altro protocollo chiamato Batibus. Pur essendo un protocollo precursore, non ebbe la stessa diffusione nel mercato di EIB, probabilmente a causa della rigidità del sistema nel lato fisico: questo si appoggiava sostanzialmente solo su un doppino telefonico che veniva schermato per ridurre ed eliminare eventuali interferenze.

Infine, un'associazione belga chiamata EHSA, si occupò di creare EHS, un protocollo di comunicazione per bus di campo dedito al controllo di elettrodomestici e delle telecomunicazioni, tramite Power Line Communication (PLC) con funzionalità di plug and play.

Le tre associazioni, pur sviluppando standard diversi, avevano come scopo comune la standardizzazione, la compatibilità e la possibilità di scambio di diverse esperienze maturate sul campo. Proprio per questo motivo nel 1998, fondendo gli aspetti più prestanti delle loro tecnologie, decisero di convergere verso un unico protocollo di comunicazione e nell'anno successivo nacque la Konnex Association, da cui emerge KNX⁷.

Si può definire KNX come il primo standard aperto e brevettato per l'automazione nelle case e negli edifici, che segue un approccio di sistema distribuito. È riconosciuto come standard europeo a partire dal 2004 come norma EN 50090, e successivamente nel 2006, a livello mondiale con la ISO/IEC 14543-3. (Domingues et al., 2016)

Per la messa in servizio, la diagnosi e la manutenzione delle installazioni KNX, la Konnex coordina un unico strumento software chiamato Engineering Tool Software⁸ (ETS), il quale è

⁷ Spesso si vede scritto come EIB/KNX poiché la tecnologia del sistema EIB continua a esistere inalterata come un insieme di profili all'interno di KNX.

⁸ È uno strumento software di configurazione indipendente dal produttore, utile a progettare e configurare le installazioni intelligenti di controllo di edifici e case con il protocollo KNX.

capace di fornire al fruitore un livello di astrazione maggiore rispetto alla complicatezza della configurazione del sistema.

Il principio funzionale di KNX si basa sul controllo dei dispositivi tramite blocchi funzionali, standardizzati dalla stessa tecnologia. Quest'ultimo è costituito da un gruppo di datapoint, definiti per tipo, e da una specifica che astrae gli aspetti comportamentali del dispositivo di controllo; un esempio può essere un interruttore on/off per l'illuminazione riprodotto dal blocco funzionale *Light Switching Actuator Basic*. Ogni blocco funzionale è relazionabile ad un solo dispositivo, nonostante un dispositivo debba necessariamente implementare al minimo un blocco funzionale, senza alcun limite superiore.

Un punto dati rappresenta gli input e gli output, dunque ingressi e uscite, di un blocco funzionale e i parametri di configurazione associati ad un'applicazione, come ad esempio una variabile interna di setpoint che memorizza un valore massimo o minimo che un controller può fornire, o un trigger di allarme. Si può allora affermare che i datapoint sono l'interfaccia di un blocco funzionale per l'invio e la ricezione dei dati.

KNX specifica tre modi distinti di associazione dei punti dati: associazione libera, associazione strutturata e associazione con tag con livelli crescenti di semantica. In caso di free binding i datapoint dello stesso tipo possono essere collegati senza impedimenti. L'associazione strutturata, invece, accoglie uno specifico schema definito dal modello informativo per il collegamento di datapoint a seconda dei blocchi funzionali dei dispositivi. Infine, l'associazione con tag prevede che una parte dell'indirizzo del datapoint includa certe informazioni al fine di mettere in luce alcuni aspetti, come può essere il gruppo di appartenenza del datapoint.

Una rete KNX è costituita alla base da una linea, la quale contiene al suo interno un massimo di 254 dispositivi in topologia libera. Le linee sono caratterizzate da una struttura ad albero e sono disposte in tre diversi livelli, le linee secondarie sono allora collegate da quelle principali attraverso i router, denominati accoppiatori di linea, al fine di formare una zona. Queste ultime, le zone, a loro volta sono appaiate da una linea dorsale come è verificabile in figura 2.3.

Le partizioni di rete su supporti aperti sono generalmente unite alla topologia come una linea o una zona separata. Il tunneling⁹ IP viene solitamente adoperato per le linee principali e per la linea dorsale, attraverso router EIBnet/IP che fungono da accoppiatori di linea. Nell'insieme, una rete KNX è in grado di ospitare un massimo di circa 60mila dispositivi.

⁹ Il tunneling è un protocollo di comunicazione che consente ad un certo utente di fornire oppure di accedere ad un dato servizio non supportato o non fornito in linea diretta dalla rete che sta utilizzando.

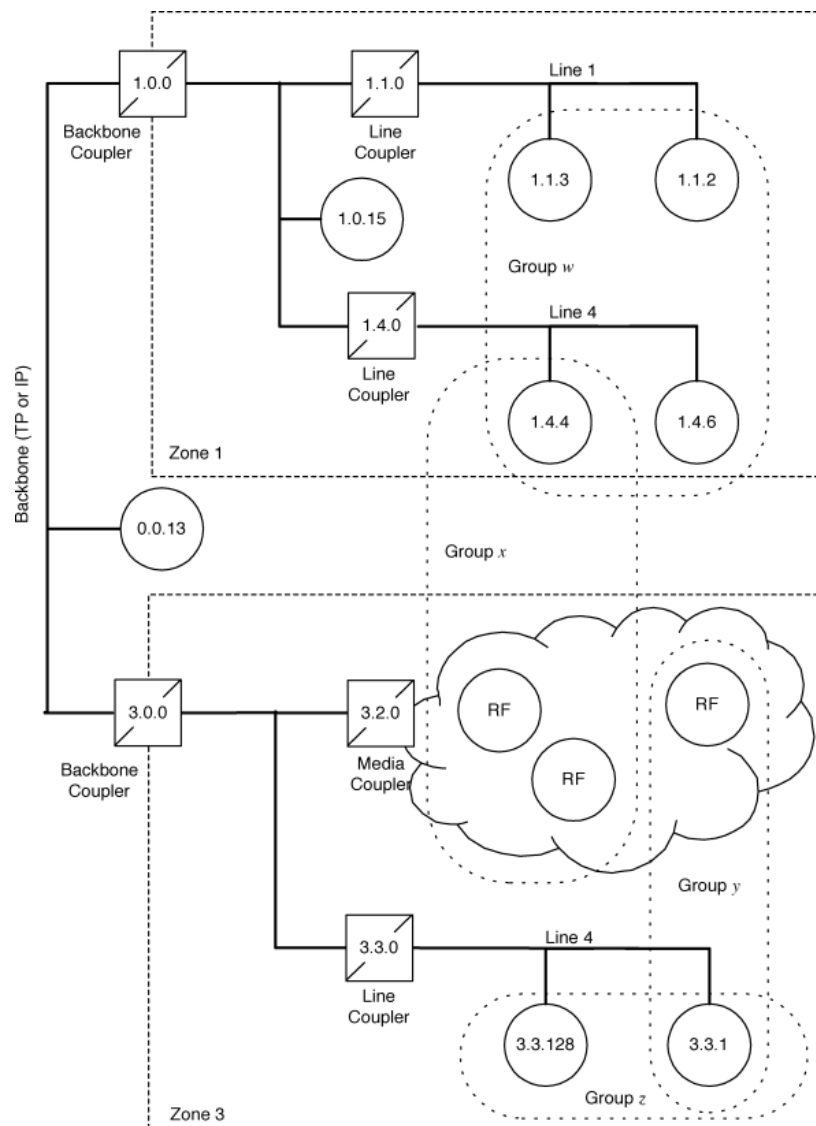


Figura 2.3: Topologia di rete EIB/KNX. (Kastner et al., 2005)

Ogni nodo presente nella rete è caratterizzato da un indirizzo individuale assegnato che collima con la sua posizione nella struttura topologica della rete stessa e pertanto restituisce informazioni riguardo la zona, la linea e il dispositivo. Il corrente indirizzo viene usato unicamente per le comunicazioni di tipo unicast, che rappresentano una connessione tra due punti e di conseguenza molto affidabile.

Il tipo di indirizzamento multicast è, invece, implementato in caso di collegamento dati. Per questo fine, sono assegnati ai nodi degli indirizzi MAC¹⁰ non univoci supplementari, detti indirizzi di gruppo, che consentono una propagazione dati enormemente efficiente e sono instradati attraverso l'intera rete. Ciò nonostante, i riconoscimenti sono assegnati sono

¹⁰ Sono componenti di basso livello di una rete Ethernet o di una rete Wi-Fi. Permettono al dispositivo di scambiare informazioni con una macchina sulla rete fisica locale (LAN) senza il supporto della rete Internet.

all'interno di un segmento elettrico, che corrisponde al secondo strato. Il gruppo, nella sua interezza, risponde subito ma con identificazioni negative, le quali prevalgono sulle positive.

Una peculiarità fondamentale di KNX è la sua metodologia di scambio di informazioni fondata su un modello di osservazione: svariati dispositivi di osservazione vengono avvisati simultaneamente tramite messaggio delle modifiche ai dati compiute su un'unica fonte. Il normale scambio di informazioni, pertanto, non viene effettuato tramite messaggi da punto a punto, ma tramite la comunicazione di gruppo multicast. A tale scopo sono previsti degli indirizzi di gruppo particolari, che autorizzano un dispositivo ricevente di decretare se è un membro di tale gruppo, e di ignorare o elaborare i messaggi ricevuti.

Per quanto riguarda i supporti fisici, il mezzo principale di comunicazione, nonché il primo utilizzato dalla Konnex, è il cablaggio a doppino intrecciato, definito come KNX TP1, che permette il cablaggio svincolato della topologia con una lunghezza del cavo fino a 1000 m per segmento fisico. È possibile concatenare un massimo di quattro segmenti attraverso l'utilizzo di ripetitori di linea, semplificabili come dei ponti di comunicazione.

Le specifiche di KNX presentano alcune limitazioni, innanzitutto queste non si possono attribuire in maniera nativa alle funzionalità di accesso ai dati storici, alle notifiche di eventi e allarmi, all'organizzazione delle attività e alle funzionalità di gestione degli scenari. Ciò permette a tutti i fruitori di implementare liberamente le funzionalità a livelli superiori senza rincorrere ad una linea guida di base, causando problemi di comunicazione e, di conseguenza, interoperabilità.

2.3 LonWorks

LonWorks, o anche Local Operating NetWork, è una piattaforma per lo sviluppo della comunicazione digitale basata su fieldbus e implementata dalla Echelon Corporation dal 1988. Da anni si afferma come una rete di controllo aperta e generica con l'obiettivo di supportare una vasta gamma di applicazioni appartenenti a vari domini come sistemi di building automation, ma anche in linee di produzione in ambito industriale e nel settore dei trasporti.

Tale tecnologia si basa sulla sussistenza di dispositivi indipendenti dal punto di vista della logica, che sono in grado di comunicare ed eseguire comandi attraverso una rete di controllo che usa il protocollo LonTalk sviluppato dalla stessa Echelon Corporation. Dal 1999 quest'ultimo è pubblicato come standard formale ANSI/EIA-709 e ANSI/CEA-709, tanto che è pratica comune riferirsi al protocollo di comunicazione standardizzato con il termine EIA-709. È inoltre disponibile dal 2005 come norma europea EN 14908 e a livello mondiale, dal 2008, come norma ISO/EIC-14908. (Malinowsky & Kastner, 2010)

In LonWorks un singolo dispositivo di rete viene denominato nodo e una rete di comunicazione è composta da nodi differenti, identificati da un proprio indirizzo e da un ID univoco mondiale a 48 bit, chiamato selettore. Ogni nodo è capace di implementare diversi profili funzionali.

Tali nodi si basano su un chip proprio di Echelon, della serie *Neuron* oppure su di altri controller incorporati, come ad esempio il controller LC3020 di *Loytec*¹¹, il quale esegue i sette livelli di protocollo OSI¹² e il programma applicativo associato, il quale si collega con sensori e attuatori congiunti attraverso l'interfaccia input/output. Una tipica architettura del nodo di rete è mostrata nella figura 2.4.

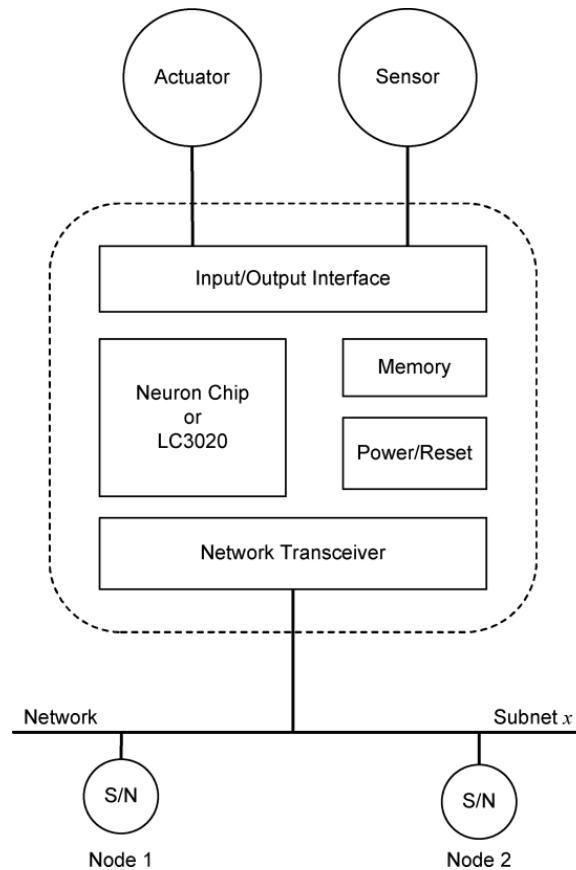


Figura 2.4: Tipica architettura di un nodo di rete in LonWorks.

I profili funzionali esprimono nel dettaglio gli aspetti dell'interfaccia del livello applicativo di un dispositivo, ovvero le sue funzionalità e le proprietà previste e include le configurazioni relazionali con altri nodi e le variabili di rete. I nodi citati si esprimono tra loro modificando le reciproche variabili di rete, le quali sono dei datapoint esposti da un singolo dispositivo agli altri, nella stessa rete, per lo scambio di informazioni. Queste hanno un tipo di dati associato che definisce la struttura dei dati come le unità o il ridimensionamento.

¹¹ LOYTEC Electronics GmbH è oggi uno dei principali fornitori europei di prodotti di controllo in rete intelligenti per l'automazione degli edifici. (Fonte: <https://www.loytec.com/about/who-we-are>)

¹² Il Modello ISO OSI (Open Systems Interconnection) è un modello concettuale che determina il metodo in cui le reti scambiano i dati dal mittente al destinatario. Si utilizza al fine di descrivere le componenti nell'ambito della comunicazione dei dati in modo tale da consentire la definizione di standard. Il modello OSI contiene sette livelli disposti concettualmente dal basso verso l'alto: fisico, collegamento dati, rete, trasporto, sessione, presentazione, e applicazione.

Un esempio pratico può risiedere in un dispositivo capace di implementare il profilo funzionale di un sensore di luce, il quale esibisce una variabile di rete che raffigura la quantità di lux ricavata in un determinato momento. È possibile che tale variabile si leghi ad un'altra attraverso un nodo che rappresenta un controller e che gestisce il profilo funzionale "Constant Light Controller" il quale riceve come input il livello di lux prima ricavato. (Vieira et al., 2020)

Le variabili di rete aventi il medesimo tipo di dati sono dunque associabili tra loro e seguono delle regole ben precise per garantire l'interoperabilità tra i vari nodi e i dispositivi, descritte nella specifica LonWorks Standard Network Variables Types (SNVT). Alla base di questo concetto è presente l'intelligenza necessaria di ogni dispositivo per decretare cosa sia opportuno fare in occasione di un nuovo input proveniente da una certa fonte. Lo sviluppo di nuovi nodi è così semplificato poiché sono isolati dalla conoscenza dei dettagli degli altri nodi.

LonWorks sostiene la modellazione di reti virtuali e logiche nella struttura di rete fisica, cioè nei domini e nelle sottoreti, utilizzabili per l'implementazione della suddivisione in zone. Tale principio è utile nel momento in cui si rivela necessaria la separazione di gruppi indipendenti estesi di dispositivi presenti in una rete, come in caso di disgiunzione di sistemi di ventilazione con i sistemi di illuminazione in un edificio.

Si possono definire le sottoreti come dei gruppi fisici oppure logici di apparecchi che si trovano all'interno di un dominio, come ad esempio un gruppo formato da tutti i dispositivi posti in una stanza. Sono proprio le sottoreti e i domini a procurare a LonWorks l'efficienza di zonizzazione.

I nodi che si trovano nelle sottoreti possono essere guidati singolarmente oppure attraverso broadcast, altrimenti per procedere con l'associazione di dispositivi in modo indipendente dall'indirizzamento nodo-settore-dominio è possibile creare dei gruppi. Questi ultimi non sono altro che un insieme di apparecchi, nei quali ogni membro è in grado di esprimersi e dunque conversare con gli altri, adoperando l'identificazione propria del gruppo. Se ne deduce che i gruppi sono utilizzabili anche per la propagazione di informazioni. Le sottoreti sono caratterizzate da un tipo di gerarchia ad albero come mostrato nella figura 2.5.

Il dominio, invece, è l'intero spazio di indirizzi gestibile in una rete EIA-709 e viene identificato da un ID avente massimo 48 bit, il quale corrisponde a determinati requisiti; in particolare, in supporti aperti sarà il più lungo possibile al fine di evitare interferenze logiche. Un singolo dominio è in grado di ospitare fino a 255 sottoreti aventi un massimo di 127 nodi ognuno.

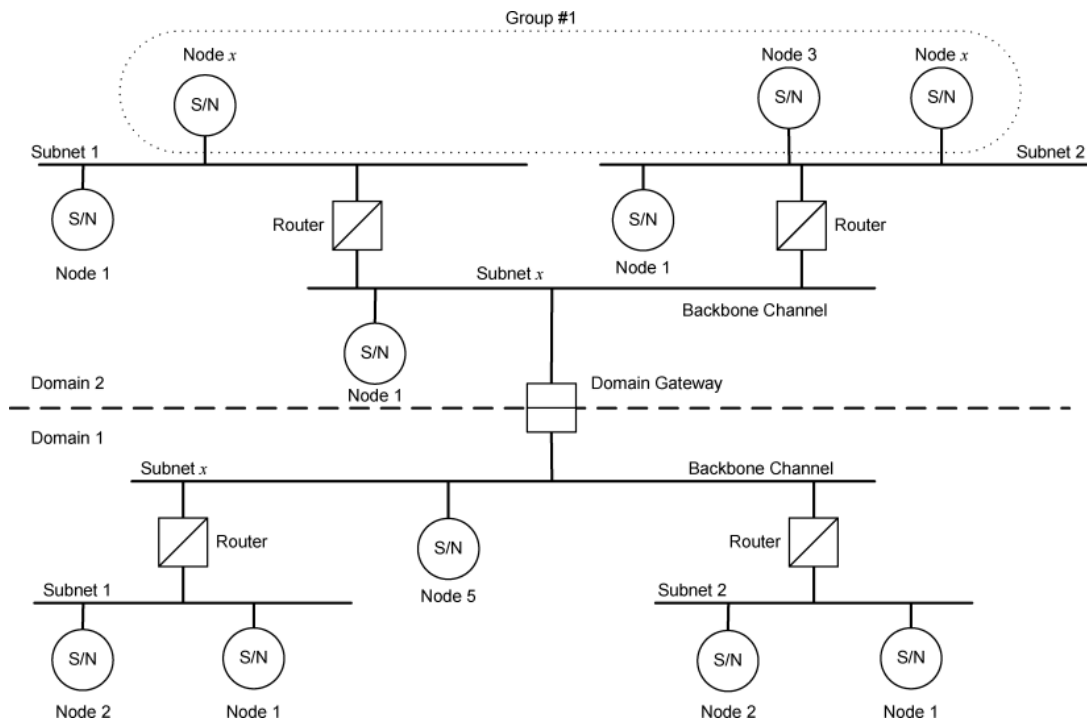


Figura 2.5: Tipica gerarchia logica delle sottoreti in LonWorks. (Kastner et al., 2005)

I profili funzionali possono comunque rappresentare un inconveniente, a causa dell'incapacità delle specifiche di prevedere ogni singola esigenza. Per questo motivo, spesso i produttori ricorrono all'implementazione di estensioni proprietarie, creando degli ostacoli per gli obiettivi di interoperabilità. Si può trovare un esempio di tale ostacolo nella mancanza di variabili di rete create ad hoc per l'accessibilità ai dati storici. Per sopperire a tale carenza il protocollo può ricorrere a dispositivi programmabili capaci di supportare le proprietà non coperte dalle specifiche, a discapito della privazione di interoperabilità come prima accennato.

2.4 ZigBee

ZigBee è una tecnologia radio di prossimità creata nel 2004 per realizzare reti di tipo Wireless Personal Area Network. Il protocollo viene standardizzato dalla ZigBee Alliance, un'associazione non-profit di circa 150 aziende, tra cui spiccano Mitsubishi, Philips, Samsung, Siemens, che si propone di definire e sviluppare il protocollo ZigBee a partire dal livello fisico fino a raggiungere quello applicativo, sulla base dello standard IEEE 802.15.4.

Lo standard IEEE 802.15.4 determina i livelli di accesso fisici e medi per le reti personali wireless a bassa velocità. Racchiude in sé i metodi per fornire, in modo cumulativo, estesi periodi di sonno profondo, indispensabili per il risparmio energetico ottenuto dallo sfruttamento delle transizioni rapide tra la modalità di sospensione e lo stato attivo. Un dispositivo adibito a coordinatore può ciclicamente trasmettere frame beacon, che vengono utilizzati per sincronizzare i dispositivi collegati. I beacon sono dei piccoli trasmettitori che funzionano con tecnologia BLE (Bluetooth Low Energy), composti da una quantità di dati

marginale e che di frequente trasmettono dei segnali radio a smartphone o tablet posti nelle vicinanze. Sono dispositivi in grado solo di divulgare le informazioni e non riceverle e dunque, operano in una singola direzione. I dispositivi che richiedono dati regolarmente, ad una velocità precedentemente definita, sono in grado di attivarsi solo per ricevere il frame beacon, il quale denota se i dati sono effettivamente disponibili per loro. Viceversa, quei dispositivi che adibiti alla trasmissione dei dati con una frequenza definita dall'applicazione o da un sensore esterno possono attivarsi, sincronizzarsi con il beacon, trasmettere e tornare a dormire.

ZigBee, dunque, specifica la rete e il livello dell'applicazione e permette ai dispositivi che lo usano una comunicazione di tipo wireless, il che consente a livello di campo una notevole riduzione dei costi di cablaggio e di installazione dei sensori e un miglior impatto estetico nell'installazione del sistema, procurando un protocollo semplice, a bassa velocità, a bassa potenza ed economico per le applicazioni che usano la radiofrequenza. Tuttavia, per godere di questo vantaggio, i nodi del protocollo devono lavorare a batterie per mesi, o in casi migliori anche per anni. Le applicazioni di controllo nella building automation non richiedono un'elevata larghezza di banda, ma richiedono comunque una latenza ragionevolmente bassa. gruppo

La ZigBee Alliance aggiunge livelli aggiuntivi allo standard IEEE 802.15.4. Forniscono infatti funzionalità a livello di rete con sicurezza aggiuntiva, tra cui l'Advanced Encryption Standard¹³ e delle funzionalità di routing per estendere la singolare portata di 50 m di un singolo segmento di rete, supportando topologie di rete mesh¹⁴ per la creazione dinamica, il consolidamento e le divisioni. ZigBee aggiunge anche un livello di supporto dell'applicazione con discovery e binding più dei blocchi funzionali, che attualmente coprono applicazioni di automazione degli edifici, controllo domestico e degli impianti.

In genere, il modello informativo di ZigBee è fondamentalmente definito su tre livelli, i quali coordinano gli oggetti dispositivo, gli endpoint e i collegamenti tra di essi. Tali livelli sono noti come Network Layer (NWK), Application Support Sublayer (APS), e Application Framework Layer (AF). È poi presente un'entità a più livelli denominata ZigBee Device Object (ZDO).

Dei livelli visibili nell'immagine 2.6, il livello fisico (PHY) e il livello di controllo di accesso medio (MAC) non fanno propriamente parte della specifica ZigBee, ma derivano dallo standard IEEE 802.15.4. (Vasseur & Dunkels, 2010)

¹³ È una specifica per la crittografia dei dati elettronici deliberata dal National Institute of Standards and Technology (NIST) degli USA nel 2001.

¹⁴ È una rete composta da un insieme di dispositivi di connettività, come ad esempio router Wi-Fi, che funzionano come un'unica rete.

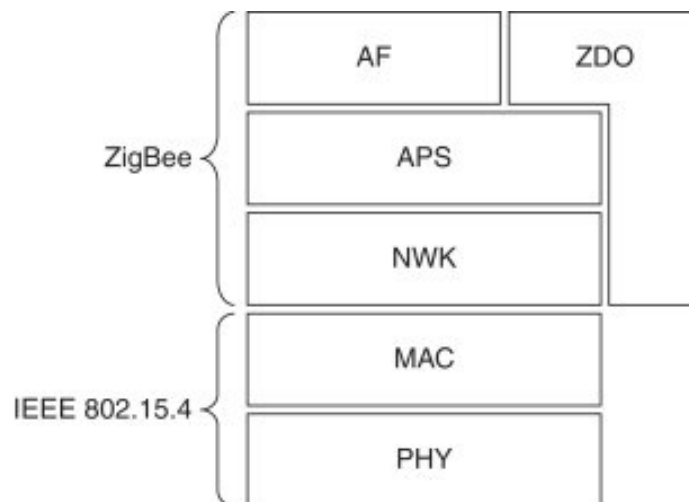


Figura 2.6: Stack del protocollo ZigBee. (Vasseur & Dunkels, 2010)

Il livello di rete o di Network esegue l'indirizzamento e il routing ed è l'equivalente del livello IP nell'architettura IP. Esso presenta due forme di consegna dei dati: broadcast e unicast. Tuttavia, è supportato anche il multicast, ma in questo caso i dati vengono restituiti utilizzando la trasmissione con il filtraggio software dei pacchetti in arrivo al ricevitore.

L'Application Support Sublayer è sostanzialmente un livello di trasporto poiché funge da intermediario tra il livello di rete e quello di applicazione. È infatti uno strato responsabile dell'associazione degli endpoint, del recapito dei messaggi tra i vari dispositivi associati e dell'amministrazione dei gruppi.

Seguendo poi, l'Application Framework accoglie le applicazioni dei dispositivi. Un oggetto applicazione indica quali servizi è capace di mettere a disposizione il dispositivo, ad esempio, un oggetto applicazione può essere un dispositivo come un interruttore, una lampadina o anche una linea. Ogni dispositivo che appartiene a ZigBee può avere nella totalità fino a 240 applicazioni, ciò significa che in ogni dispositivo 240 endpoint sono destinati a questo fine.

Il livello ZigBee Device Object governa la gestione d'insieme del dispositivo ed è puntualmente responsabile: della descrizione della modalità esecutiva del dispositivo, indicando se un dispositivo regola le comunicazioni di rete o rappresenta da dispositivo finale, del rilevamento e dell'indicazione di quali servizi applicativi sono previsti dall'apparecchio, dove ogni servizio applicativo collima con un endpoint di un dispositivo e, infine, della gestione delle richieste di associazione da altri dispositivi o coordinatori.

Per riuscire nella comunicazione con altri dispositivi, le applicazioni si avvalgono di unità dati di protocollo particolari, chiamate cluster, che constano in strutture di messaggi predeterminate e formate da comandi e attributi, somiglianti ad un oggetto in un contesto di programmazione orientato agli oggetti, di cui entrambi i dispositivi comunicanti sono a conoscenza, dotando così la rete di interoperabilità. ZigBee ha a sua disposizione una libreria

di cluster che procura protocolli per la gestione di gruppi di dispositivi, scenari e notifiche di allarme.

2.5 Altri standard

Altre tecnologie utilizzate regolarmente nella building automation sono: EnOcean, Insteon, ModBus e Z-Wave.

EnOcean e Z-Wave due determinano i protocolli di comunicazione wireless a bassa potenza e i loro aspetti elettrici appartenenti e sono abitualmente sfruttati nell'automazione domestica e in quella industriale. Sullo stesso principio, Insteon fornisce sostegno, ma non si limita soltanto, a comunicazioni wireless ed è normalmente utilizzato per l'automazione in ambito domestico. Infine, ModBus distingue un protocollo di messaggistica a livello di applicazione per la comunicazione dei dispositivi di automazione, fondato soprattutto sullo scambio di messaggi da controller a controller, che può essere implementato utilizzando diversi tipi di bus e reti.

Ad ogni modo, gli standard appena citati specificano in via generale protocolli di trasporto per messaggi e non determinano dei modelli informativi interessanti nel contesto dell'interoperabilità. Infatti, le decisioni relative all'esecuzione di comandi specifici del dispositivo e i concetti di primo livello, come possono essere scenari o allarmi, sono una libera scelta dei produttori di hardware.

2.5.1 ModBus

ModBus si può considerare come un protocollo aperto, nonostante non sia mai stato ufficialmente standardizzato. Questo protocollo è stato progettato alla fine degli anni Settanta ed è al momento supportato dalla maggior parte dei controllori logici programmabili (PLC).

La sua implementazione è possibile senza alcuna licenza, il che lo rende particolarmente appetibile per l'integrazione e l'interfacciamento tra i BACS e gli altri sistemi. È attualmente ancora supportato con una certa importanza da numerosi controller, soprattutto per quanto riguarda la comunicazione HVAC da controller a controller, come ad esempio accade con i refrigeratori. ModBus, comunque sia, è presente anche in dispositivi appartenenti ad altre discipline nell'ambito edilizio, come in contatori o sistemi di allarme.

Il livello di applicazione del protocollo è sostanzialmente circoscritto alla lettura e alla scrittura dei valori dati dai registri, per mezzo di un semplice protocollo di richiesta/risposta. Ne consegue un livello applicativo molto flessibile e poliedrico, che però implica un elevato sforzo ingegneristico informatico, considerando che si rivela necessaria la coordinazione anche del formato dei tipi di dato primitivi.

ModBus supporta la comunicazione di serie per mezzo di un semplice protocollo master-slave basato su EIA-485; questo consente di indirizzare un totale di 247 diversi slave. È inoltre definita una modalità di trasmissione su TCP/IP, in cui ogni nodo può essere sia client che server.

2.5.2 *EnOcean*

EnOcean è una tecnologia di comunicazione wireless a bassissima potenza, che consente tecniche di raccolta di energia e viene massimamente implementato nell'automazione domestica e dell'ufficio. Rispetto ad altre tecnologie come ZigBee o anche semplicemente Bluetooth, si rivela essere molto più proficua sotto il punto di vista energetico. Il protocollo armonizza convertitori di microenergia con elettronica a bassissima potenza, creando delle comunicazioni wireless affidabili per mezzo di una semplice fusione di più protocolli.

L'hardware è di norma ideato da un system-on-chip, alimentato unicamente da un micro-convertitore di energia, il che può presentarsi come un pannello solare, un termogeneratore o un generatore elettromeccanico, creando una tecnica di sostentamento autosufficiente. Ciò significa che non è indispensabile l'uso di batterie, riducendo notevolmente o quasi annullando i costi di manutenzione. I dispositivi supportati sono attuatori, controller, sensori, gateway e anche dispositivi di controllo e monitoraggio centrali.

Ci sono quattro livelli nella specifica standard di EnOcean: il livello fisico, il livello di collegamento dati, il livello di rete e il livello di applicazione. Lo standard internazionale ISO/IEC 14543-3-10 specifica i livelli OSI da 1 a 3 del protocollo Wireless Short-Packet (WSP) e l'EnOcean Alliance, organizzazione aperta e indipendente, uniforma il livello di applicazione.

2.5.3 *Insteon*

Insteon è una tecnologia sviluppata per la domotica da SmartLabs¹⁵ e promossa dall'Insteon Alliance. Si distingue dalle altre soluzioni poiché consente la creazione di semplici reti a basso costo utilizzando una topologia mesh con collegamenti sia in radiofrequenza che su linee elettriche.

I dispositivi sono collegati tra loro tramite un collegamento punto a punto, il che significa che ogni dispositivo può trasmettere, ricevere o ripetere messaggi proiettandoli contemporaneamente in piccoli intervalli di tempo sincronizzati con lo zero crossing della linea elettrica. I dispositivi che sono posizionati nella rete in modo irregolare, o che non si trovano all'interno della stessa portata, comunicano tra loro per mezzo di un approccio multi-hop, che viene eseguito utilizzando uno schema di sincronizzazione di slot di tempo, attraverso il quale le trasmissioni di dati sono concesse in specificati intervalli di tempo e i dispositivi all'interno dello stesso intervallo non trasmettono messaggi diversi simultaneamente. Il numero massimo di salti per ogni messaggio è limitato a quattro, come accade per Z-Wave.

I dispositivi Insteon utilizzano un tipo di approccio chiamato simulcast, invece del routing, il quale evita la necessità di memorizzare lo stato per rendere possibile la

¹⁵ SmartLabs è una startup innovativa specializzata nello sviluppo di soluzioni tecnologiche avanzate in ambito Big Data, Business Intelligence, IoT, Blockchain. (Fonte: <https://www.smartlabs.eu>)

comunicazione multi hop, poiché si basa sulla probabilità molto bassa che più segnali simultanei vengano cancellati sul ricevitore. Il vantaggio del simulcast è che i dati possono essere raggiunti attraverso un percorso alternativo quando un dispositivo intermedio non è disponibile.

2.5.4 Z-Wave

Z-Wave è un'architettura di protocollo wireless proveniente da un'implementazione di un substrato IoT completo, contenente protocolli di comunicazione, rete e livello applicativo ben definiti. È sviluppata nel 2001 da una startup danese Zen-Sys, acquisita dal 2008 dall'azienda americana Sigma Designs, e promossa dalla Z-Wave Alliance per l'automazione in ambienti residenziali e commerciali di piccola entità. Il suo scopo principale è quello di consentire la trasmissione affidabile di brevi messaggi da un'unità di controllo a uno o più nodi della rete.

Z-Wave è organizzato secondo un'architettura composta da cinque livelli principali: i livelli PHY, MAC, trasferimento, routing e applicazione.

Il livello MAC di Z-Wave definisce un meccanismo di prevenzione delle collisioni, il quale rende possibile la trasmissione di un frame quando il canale è disponibile. In caso contrario, il tentativo di trasmissione viene rinviato per un periodo di tempo casuale. Il livello di trasferimento gestisce la comunicazione tra due nodi consecutivi. Questo livello fornisce un meccanismo di ritrasmissione opzionale basato su ACK.

Il livello di routing Z-Wave esegue il routing in base a un tipo di approccio di origine. Quando un controller trasmette un pacchetto, in esso include il percorso da seguire. Questo può essere trasmesso su un massimo di quattro hop, il che è sufficiente in uno scenario residenziale. Un controller mantiene una tabella che rappresenta la topologia completa della rete, mentre un controller portatile, come ad esempio un telecomando, tenta prima di raggiungere la destinazione tramite trasmissione diretta. Se tale opzione non riesce, il controller stima la sua posizione e calcola il percorso migliore per la destinazione.

Sebbene le specifiche di PHY, MAC e una parte del livello applicativo dello stack del protocollo Z-Wave siano pubblicamente accessibili, esistono pochi dettagli relativi al livello di rete all'interno del pubblico dominio.

Z-Wave definisce due tipi di dispositivi: controller e slave. I controller eseguono il polling o inviano comandi agli slave, che rispondono ai controller o eseguono i comandi. I secondi sono adatti generalmente per il monitoraggio di sensori, nei quali il ritardo apportato dal polling è accettabile, nonché per attuatori che eseguono azioni in risposta a comandi di attivazione. Gli slave possono fungere da router, questi memorizzano percorsi statici in genere verso i controller e sono in grado di inviare messaggi ad altri nodi senza che venga richiesto di farlo, vengono utilizzati per applicazioni di trasmissione time-critical e non richieste, come l'attivazione di allarmi.

3 BACNET

La specifica BACnet è in continua manutenzione e ulteriore sviluppo. Tutto ciò è gestito dall'ASHRAE Standing Standard Project Committee 135 (SSPC). I membri di SSPC rappresentano i settori dell'industria nella loro completezza, ma in generale la partecipazione al processo di sviluppo è completamente aperta. Tutte le parti interessate sono incoraggiate attivamente a fornire commenti e suggerire modifiche.

Come già accennato la struttura di BACnet si contraddistingue dalle altre tecnologie di comunicazione per un tipo di organizzazione dei dati orientata agli oggetti, i quali hanno tutti delle determinate proprietà e garantiscono al protocollo lo sviluppo di servizi.

3.1 Dispositivi BACnet

Per facilitare il processo di determinazione della conformità BACnet, lo standard classifica i dispositivi in otto categorie.

Affinché un meccanismo possa essere classificato come dispositivo standard BACnet, deve essere conforme a una serie di capacità funzionali definite, chiamate BIBBs¹⁶. L'elenco dei BIBBs richiesti per ciascuna classificazione del dispositivo è denominato *Device Profile*. Il dispositivo più semplice è lo Smart Sensor ed è programmato solo per eseguire la richiesta di dati, poiché è appunto un sensore.

Man mano che si sale di complessità nella definizione dei dispositivi aumenta la quantità minima di BIBBs richiesta; pertanto, ogni profilo contiene tutti i profili minimi dei precedenti. Ad esempio, i profili BACnet per uso generico aumentano di complessità man mano che si passa da Smart Sensor a Building Controller. Tutti i BIBB inclusi in un profilo Smart Sensor sono sempre inclusi in un profilo Smart Actuator e tutti i BIBB inclusi in questi due profili sono sempre inclusi in un Application Specific Controller e così via.

3.1.1 BACnet Operator Workstation (B-OWS)

È progettato per fornire ad un operatore tutte le informazioni e le capacità di modifica di cui ha bisogno per gestire un sistema su base giornaliera. Oltre a visualizzare e modificare gli oggetti BACnet selezionati, è possibile visualizzare tendenze, programmi e altri oggetti specializzati. Può anche visualizzare report e grafici.

Gli allarmi forniscono un buon esempio di come un Operator Workstation fornisce più funzionalità rispetto ad un Operator Display BACnet: entrambe le interfacce avviseranno l'operatore che si è verificato un allarme, ma una Operator Workstation consentirà anche all'operatore di riconoscere il problema. Esso fornirà anche un riepilogo degli allarmi e permetterà all'operatore di regolare le soglie di allarme degli oggetti analogici. Un esempio è visibile nella figura seguente.

¹⁶ Si veda il paragrafo 3.5 per maggiori informazioni.



Figura 2.7: Esempio di (B-OWS), lo StruxureWare Building Operation - Automation Server di Schneider Electric.

3.1.2 BACnet Advanced Operator Workstation (B-AWS)

È possibile utilizzare una Advanced Operator Workstation per configurare e mettere in servizio un sistema BACnet e inoltre, può fornire servizi avanzati di risoluzione dei problemi di rete. Ciò non significa che se un fornitore non fornisce un B-AWS non avrà modo di commissionare il sistema, in generale nessun fornitore potrebbe rimanere in attività se non fornisse gli strumenti necessari per configurare, commissionare e risolvere i problemi del proprio sistema. La differenza è che una BACnet Advanced Workstation fornisce gli strumenti BACnet per completare queste attività. Ciò significa che è possibile utilizzarlo per commissionare e riconfigurare un sistema con componenti forniti da più fornitori BACnet.

3.1.3 BACnet Operator Display (B-OD)

Un BACnet Operator Display (fig. 2.8) fornisce all'operatore la possibilità di visualizzare un insieme limitato di oggetti BACnet e apportare modifiche setpoint e altri parametri. Questi requisiti di interfaccia sono abbastanza semplici e possono essere soddisfatti da un display montato direttamente sull'apparecchiatura, oppure potrebbero essere inclusi in un dispositivo portatile a cui si collega l'operatore con un controller BACnet.



Figura 2.8: Esempio di B-OD della serie SmartX di Schneider Electric.

3.1.4 BACnet Building Controller (B-BC)

Un controller è un dispositivo specializzato utilizzato per automatizzare parzialmente e fornire il controllo elettronico dei processi elettromeccanici. Sono progettati pensando a molteplici disposizioni di ingresso e uscita e sono tipicamente costruiti per gestire temperature estreme, rumore elettrico e vibrazioni che si incontrano spesso nei processi industriali.

I BACnet Building Controller (B-BC) sono la categoria di controller di livello superiore e vengono utilizzati per l'instradamento delle informazioni tra controller e utenti come, ad esempio, tramite comunicazioni IP a un'interfaccia operatore (fig. 2.9). Come parte di un sistema completo di automazione degli edifici interoperabile, fornisce un monitoraggio ed un controllo precisi dei punti collegati attraverso una serie di sensori discreti e analogici. Nel controllore sono integrati un router BACnet, un server Web e I/O espandibili in un dispositivo BACnet nativo.



Figura 2.9: Esempio di B-BC, l'Andover Continuum™ bCX1 di Schneider Electric.

In generale, le loro capacità possono variare notevolmente: alcune non hanno capacità di punto e forniscono il controllo di supervisione degli ASC/AAC, ad esempio come programmazione, tendenza, allarme, o possono semplicemente agire come router di informazioni.

3.1.5 BACnet Advanced Application Controller (B-AAC)

Tali controller hanno, rispetto ad altri, una gamma più ampia di conteggi/tipi di punti insieme a capacità di espansione degli stessi. Supportano la pianificazione e possono anche avere capacità di tendenza e di generare allarmi, specialmente se necessario per gli ASC associati. Le comunicazioni MS/TP e IP sono spesso entrambe disponibili. Alcuni AAC potrebbero essere meglio classificati come ASC a causa dell'uso di scelte di sequenza programmate in fabbrica, ad esempio per RTU¹⁷, mentre altri AAC offrono maggiore flessibilità,

¹⁷ Acronimo di Remote Terminal Unit, dispositivo elettronico di controllo a microprocessore che interfaccia oggetti del mondo fisico a un sistema di controllo distribuito o uno SCADA (supervisory control and data acquisition system) attraverso la trasmissione di dati acquisiti dalla strumentazione collegata al sistema di supervisione.

ad esempio per AHU¹⁸ personalizzate. Esempio in figura 2.10.



Figura 2.10: Esempio di B-AAC, gli SmartX IP Controllers MP-C and RP-C di Schneider Electric.

3.1.6 BACnet Application Specific Controller (B-ASC)

Questo tipo di controller (Fig. 2.11) è pensato per un'applicazione specifica come, ad esempio scatole VAV, e quindi hanno un conteggio e un tipo di punti pensato per quello scopo. Possono venire con una serie fissa di scelte di sequenza programmate in fabbrica, sebbene alcuni produttori offrano maggiore flessibilità rispetto a questa. Variano anche nel grado in cui possono operare autonomamente; alcuni hanno orologi per la programmazione di avvio/arresto, capacità di trending e capacità di generare allarmi. Quasi tutti gli ASC comunicano tramite MS/TP sebbene stiano comparando versioni che utilizzano IP.



Figura 2.11: Esempio di B-ASC, lo SmartX IP Controllers IP-IO Modules di Schneider Electric.

3.1.7 BACnet Smart Actuator (B-SA)

Gli attuatori sono dispositivi meccanici che consentono ad una persona di applicare un

¹⁸ Acronimo per Air Handling Unit, unità di trattamento dell'aria, è la composizione di elementi montati in grandi unità scatolari accessibili chiamate moduli, che ospitano gli opportuni requisiti di ventilazione per purificare, climatizzare o rinnovare l'aria interna di un edificio o di un locale.

movimento o una forza specifici indirettamente a un prodotto o processo, piuttosto che manualmente. Un attuatore intelligente (fig. 2.12) è definito come un attuatore integrato di tutti i componenti come motore, controller, sensori e unità di comunicazione. Tale dispositivo consente il controllo autonomo o semi-autonomo nel settore del riscaldamento, della ventilazione e della climatizzazione. Ulteriori vantaggi sono la semplicità di installazione e messa in servizio, la connessione al cloud con la libreria di applicazioni e l'accesso remoto ai dati tramite la rete BACnet.



Figura 2.12: Esempio di B-SA della serie EM4800 multi-circuit energy meter di Schneider Electric.

3.1.8 BACnet Smart Sensor (B-SS)

La forma più semplice di profilo BACnet. In genere sono dei dispositivi che convertono le proprietà fisiche in segnali elettrici, per esempio un sensore di temperatura, sensore di umidità, sensore rilevatore di presenza e così via (fig. 2.13). Sono definiti come sensori con piccola memoria e connessione fisica standardizzata per consentire la comunicazione con il processore e la rete dati BACnet.



Figura 2.13: Esempi di B-SS, gli SXWS sensors di Schneider Electric.

3.2 Oggetti BACnet

Gli oggetti rappresentano servizi applicativi lungo i relativi input e output, come possono essere dispositivi, calendari e pianificazioni, comandi e circuiti di controllo. In BACnet un dispositivo è rappresentato da un oggetto dispositivo, il quale caratterizza le proprietà dello stesso come il nome del modello, il fornitore, lo stato e l'elenco di altri oggetti BACnet associati all'apparato. Un dispositivo con due ingressi analogici, ad esempio, presenterà due istanze di oggetti di ingresso analogico associati, oppure ancora un oggetto di ingresso analogico che riporta la temperatura dell'ambiente in cui si trova avrà prima di tutto una proprietà "valore attuale" associata alla temperatura ambiente effettiva letta dall'ingresso fisico. Altre proprietà descrivono il sensore, i valori minimo e massimo dell'ingresso, la risoluzione e le unità ingegneristiche del valore e indicano lo stato di affidabilità del sensore.

Oltre a ciò, il protocollo prevede che i produttori abbiano la possibilità di creare anche tipi di oggetti proprietari per consentire loro di registrare funzionalità specifiche e necessarie, non coperte da oggetti standard, a discapito però dell'interoperabilità tra i dispositivi. Ciò può essere fatto da qualsiasi implementatore senza ottenere alcuna approvazione ufficiale e tali nuove funzionalità non interferiranno con estensioni simili realizzate da altri, a condizione che l'implementatore utilizzi il suo *Vendor ID Code*, disponibile gratuitamente da ASHRAE. (Kastner et al., 2005)

Attualmente, ci sono circa 55 tipi di oggetti (*BACnet Object Types*) definiti nella norma americana ANSI/ASHRAE 135:2016 ed elencati per categorie in Tabella 1. All'interno del protocollo, le istanze di *Object Types* sono chiamate *Objects*. Un dispositivo può essere rappresentato come un'aggregazione di oggetti BACnet, ciascuno contenente informazioni che fanno riferimento a un componente di un sistema BACS, ad esempio un sensore o un attuatore. Queste informazioni sono raggruppate all'interno di set di proprietà, le quali consentono la differenziazione degli oggetti e possono essere viste come datapoint, ovvero come una rappresentazione logica dei dati di processo.

Tabella 1: Elenco dei tipi di oggetti standard BACnet divisi per categorie. (Newman, 2013)

<p>Basic Device Object Types Device Analog Input Analog Output Analog Value Binary Input Binary Output Binary Value Multi-state Input Multi-state Output Multi-state Value File</p> <p>Process-related Object Types Averaging Loop Program</p> <p>Control-related Object Types Command Load Control</p> <p>Meter-related Object Types Accumulator Pulse Converter</p> <p>Presentation-related Object Types Group Global Group Structured View</p> <p>Schedule-related Object Types Calendar Schedule</p> <p>Notification-related Object Types Event Enrollment Notification Class Notification Forwarder Alert Enrollment</p>	<p>Logging Object Types Event Log Trend Log Trend Log Multiple</p> <p>Life Safety and Security Object Types Life Safety Point Life Safety Zone Network Security</p> <p>Physical Access Control Object Types Access Zone Access Point Access Door Access User Access Rights Access Credential Credential Data Input</p> <p>Simple Value Object Types CharacterString Value DateTime Value Large Analog Value BitString Value OctetString Value Time Value Integer Value Positive Integer Value Date Value DateTime Pattern Value Time Pattern Value Date Pattern Value</p> <p>Lighting Control Object Types Channel Lighting Output</p>
---	--

Uno degli oggetti BACnet standard è l'oggetto ingresso analogico, che rappresenta un ingresso sensore analogico come un termistore. La figura 2.14 mostra un diagramma di un oggetto di input analogico così come potrebbe essere visto sulla rete attraverso cinque delle sue proprietà. Proprietà come *Description*, *Device_Type* e *Units*, vengono impostate dai fruitori durante l'installazione; altri invece, inclusi *Present_Value* e *Out_Of_Service*, forniscono lo stato dell'ingresso del sensore rappresentato dall'oggetto ingresso analogico. Altri ancora, considerando che un oggetto di ingresso analogico può avere fino a 25 proprietà, possono essere impostati dal produttore dell'apparecchiatura.

I tipi di oggetto BACnet disponibili, così come le proprietà in essi iscritte, sono per lo più sommarie: ad esempio, BACnet definisce tipi di oggetti generici come *BACnet Binary Output* oppure *BACnet Analog Input*, i quali sono poi assimilabili a esempi pratici d'uso. Oltre a questi, si sta sviluppando una serie di tipi più complessi, se pur ancora generici, relativi a funzionalità di pianificazione, tendenze, allarmi e sicurezza della vita.

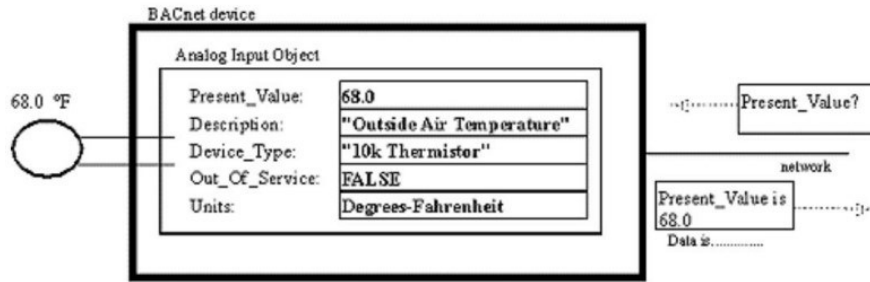


Figura 2.14: Un oggetto standard BACnet, in particolare un ingresso analogico. (Swan, 1996)

Ogni dispositivo di automazione in BACnet può avere zero, uno o più oggetti di ogni tipo di oggetto ad eccezione dell'oggetto speciale *Device Object*, che deve essere presente in ogni dispositivo. Questo oggetto assegna le informazioni di base sul dispositivo BACnet come varie indicazioni sul fornitore, sulla versione del firmware e del protocollo e ora e data locali, inoltre, contiene un'enumerazione di tutti gli altri oggetti esistenti nel dispositivo. È importante specificare che il suo *Object Identifier* e il suo *Object Name* devono essere univoci all'interno dell'intera rete BACnet e pertanto possono essere utilizzati per identificare il dispositivo nella rete.

La scelta di quali oggetti specifici debbano essere presenti in un dispositivo BACnet è determinata dalla funzione e dalle capacità del dispositivo. Lo standard BACnet, infatti, non reclama la presenza di tutti gli oggetti in tutti i dispositivi BACnet.

Il protocollo supporta anche il raggruppamento di oggetti mediante un oggetto gruppo specifico che consta in un elenco di oggetti che fanno parte di un gruppo desiderato con associato un secondo elenco delle proprietà selezionate di ciascuno di tali oggetti. I gruppi possono essere utilizzati per creare gruppi logici come le zone.

Saranno esaminati nei paragrafi seguenti alcuni oggetti BACnet. (Wichenko, 2013)

3.2.1 Accumulator

Il tipo di oggetto Accumulator definisce un oggetto standardizzato le cui proprietà rappresentano le caratteristiche visibili esternamente di un dispositivo che indica le misure effettuate contando gli impulsi¹⁹. Questo oggetto mantiene la misurazione precisa dei valori di conteggio in ingresso, accumulati nel tempo e si occupa anche della rappresentazione accurata dei valori presentati sulle letture dei contatori. Ciò include la possibilità di impostare inizialmente la proprietà *Present_Value* sul valore attualmente visualizzato dal misuratore. Le applicazioni tipiche di tali dispositivi sono nella gestione dei picchi di carico e nei sistemi di gestione della contabilità e della fatturazione.

¹⁹ L'accumulo di impulsi rappresenta la quantità misurata in unità intere senza segno.

3.2.2 Alert Enrollment

Questo oggetto fornisce un modo semplice per un dispositivo di notificare a un altro dispositivo che si è verificato un evento. Gli avvisi offrono agli utenti flessibilità in quanto non utilizzano gli algoritmi di eventi standard BACnet come CHANGE_OF_STATE e OUT_OF_RANGE. Gli avvisi sono adatti per l'invio di notifiche di manutenzione a un altro dispositivo. È da specificare che gli avvisi informativi non sostituiscono gli allarmi BACnet.

3.2.3 Analog Input

Rappresenta un oggetto standardizzato le cui proprietà rappresentano le caratteristiche visibili dall'esterno di un ingresso analogico come, ad esempio, la temperatura letta da un generico sensore.

3.2.4 Analog Output

Definisce un oggetto standardizzato le cui proprietà rappresentano le caratteristiche visibili dall'esterno di un'uscita analogica come, ad esempio, un comando di controllo.

3.2.5 Analog Value

Definisce un oggetto standardizzato le cui proprietà rappresentano le proprietà visibili esternamente di un valore analogico. Un valore analogico è un parametro del sistema di controllo che risiede nella memoria del dispositivo BACnet.

3.2.6 Averaging

Oggetto che calcola gli indicatori statistici riguardanti un dato specifico (valore minimo o valore massimo, media, varianza).

3.2.7 Binary Input

Definisce un oggetto standardizzato le cui proprietà rappresentano le proprietà visibili dall'esterno di un ingresso binario. Un input binario è un dispositivo fisico o un input hardware che può trovarsi solamente in uno di due stati distinti (ad esempio, ATTIVO/INATTIVO, I/O). Un uso tipico di un ingresso binario è quello di indicare se un particolare pezzo di apparecchiatura meccanica, come un ventilatore o una pompa, è in funzione o inattivo.

3.2.8 Binary Output

Definisce un oggetto standardizzato le cui proprietà rappresentano le caratteristiche visibili dall'esterno di un'uscita binaria. Un output binario è un dispositivo fisico o un output hardware che può trovarsi solo in uno di due stati distinti, come il caso precedente. Un uso tipico di un'uscita binaria è l'accensione o lo spegnimento di una particolare apparecchiatura meccanica, come un ventilatore o una pompa.

3.2.9 Binary Value

Gli oggetti Binary Value hanno caratteristiche e capacità software simili a un oggetto di input binario; tuttavia, non sono associati ad alcun hardware fisico e sono il risultato di un processo di controllo o di un inserimento da parte dell'operatore. Questo oggetto consente di contenere un valore binario a cui altri oggetti possono fare riferimento. Ad esempio, un interblocco²⁰ può fare riferimento e basare il proprio funzionamento su un valore binario come duty/standby.

3.2.10 BitString Value

Tale oggetto viene utilizzato per rappresentare una sequenza di valori true/false a bit singolo che sono sempre presi insieme come un gruppo. Un BitString Value ha un Present_Value il cui tipo di dati è, appunto, a bitstring²¹.

Emulando il sistema per cui i valori multistrato hanno sia valori di stato numerici che a stringa di caratteri per ogni stato, i valori BitString possono avere un nome per ogni bit. La proprietà Bit_Text degli oggetti BitString Value è un array di stringhe di caratteri ciascuna rappresentante il nome del bit corrispondente.

In BACnet, i bit della stringa sono "arretrati" rispetto al modo in cui i programmatori di solito li considerano.

3.2.11 Calendar

Definisce un oggetto standard utilizzato per descrivere un elenco di date del calendario, che potrebbe essere considerato come festività, eventi speciali o semplicemente come un elenco di date, utile per lo scheduling.

3.2.12 Channel

Tale oggetto è stato aggiunto allo standard successivamente, poiché necessario per il settore dell'illuminazione allo scopo di consentire la scrittura rapida di un valore di luminosità per decine di singole luci in molti controller di illuminazione. Potrebbero essere utilizzate le richieste di servizio WritePropertyMultiple già esistenti, ma questo approccio non fornirebbe una risposta rapida necessaria per il settore.

3.2.13 Character String Value

Questo oggetto viene utilizzato per rappresentare una stringa di testo composta da lettere, cifre, simboli ecc. Il testo è memorizzato in una sequenza di ottetti dove generalmente ogni ottetto rappresenta un codice per un particolare simbolo di carattere. Lo standard BACnet originale del 1995 ha proposto che le stringhe di caratteri potessero contenere simboli da uno dei diversi caratteri possibili definiti da standard nazionali e internazionali. Il più comune di

²⁰ L'oggetto Interlock fornisce un mezzo per stabilire il controllo condizionale su uno o più altri oggetti

²¹ Sequenza di bit.

questi era il set di caratteri "ANSI" da ANSI x3.64 che è un super set del venerabile carattere di codifica ASCII. Un'altra opzione era il cosiddetto "Unicode" o ISO10646 e vari altri.

Viene spesso utilizzato come sostituzione dell'oggetto MultistateValue, il quale deve essere convertito in testo comprensibile dall'utente per l'utente.

3.2.14 Command

Utilizzato per scrivere valori multipli su oggetti multipli e in molteplici dispositivi, i quali dovranno adempiere ad uno specifico scopo, come ad esempio per passare da una modalità giorno a una modalità notte, o modalità emergenza dove in generale possono essere coinvolti più entità.

L'oggetto Command può essere un aspetto potente con applicazioni vantaggiose, ma allo stesso tempo questo oggetto può causare confusione e problemi allo sviluppatore. Si deve dunque prestare particolare attenzione a:

- Configurazione corretta dell'oggetto Command.
- Corretta configurazione del dispositivo relativamente all'oggetto Command.
- Monitorare attentamente la proprietà In_Process dell'oggetto Command, a causa della possibilità che l'oggetto comandi sé stesso.

3.2.15 Date Pattern Value

Tale oggetto consente ad un carattere jolly del modello di data, come può essere un asterisco (*), di rappresentare un certo valore. Ad esempio, in 07-*-2021, l'asterisco rappresenta qualsiasi giorno di luglio dell'anno specificato.

3.2.16 Date Value

Questo oggetto consente di rappresentare una singola data specifica, come ad esempio 20 settembre 2021.

3.2.17 DateTime Pattern Value

Vede la combinazione dei concetti degli oggetti Date e Time Value, il valore attuale di questo oggetto potrebbe apparire come 07-*-2013, 12.*.*. In questo caso, oltre a rappresentare qualsiasi valore, l'asterisco può rappresentare giorni o mesi pari o dispari.

3.2.18 DateTime Value

L'oggetto in questione permette di rappresentare data e ora specifiche come 20 settembre 2021, 12:34:56.

3.2.19 Device

Le Properties di questo oggetto mantengono informazioni come, ad esempio, quali oggetti e servizi il dispositivo supporta, ma anche informazioni specifiche del dispositivo come

il produttore o la revisione del firmware ecc. Questo oggetto è obbligatorio e dunque sempre presente in un'unica istanza all'interno di un dispositivo BACnet.

3.2.20 Event Enrollment

L'oggetto Event Enrollment è definito dallo standard come informazione relativa alla gestione degli eventi per i sistemi BACnet. Gli eventi sono essenzialmente cambiamenti nel valore di qualsiasi proprietà riguardante qualsiasi oggetto che soddisfi criteri specifici, come, ad esempio, può essere una condizione di errore: l'Input è out of range, o un allarme che altri dispositivi sanno riconoscere. L'oggetto può dire direttamente a un dispositivo di usare un oggetto della Notification Class o dirlo a molteplici altri dispositivi.

Lo scopo principale dell'oggetto è definire un evento e offrire allo sviluppatore una connessione tra il verificarsi dell'evento e la trasmissione dei messaggi di notifica.

3.2.21 Event Log

Questo oggetto registra le notifiche²² degli eventi con timestamp e altri dati pertinenti in un buffer interno per il successivo recupero. Ciascuna voce del buffer con timestamp è denominata "record" nel registro eventi. In generale, è molto simile all'oggetto Trend Log.

Ogni oggetto Event Log mantiene un buffer interno, facoltativamente di dimensioni fisse, il quale si riempie o aumenta man mano che vengono aggiunti i record del registro eventi. Se il buffer non ha più capacità di archiviazione a causa della saturazione della memoria data dai troppi file, i record meno recenti vengono sovrascritti quando se ne aggiungono di nuovi, oppure la raccolta può essere impostata per interrompersi autonomamente.

I record del registro eventi vengono trasferiti come BACnet Event Log Records utilizzando il servizio ReadRange sviluppato per l'oggetto Trend Log.

Il buffer può essere cancellato in ogni momento scrivendo uno zero nella proprietà Record_Count.

3.2.22 File

Permette l'accesso in lettura e in scrittura ai dati del file supportati dal dispositivo. Lo scopo principale dell'oggetto File è identificare un file specifico attraverso le seguenti descrizioni:

- Nome del file
- Dimensione del file
- Data di creazione
- Tipo di file

²² L'utente può decidere quali notifiche inserire nel registro.

3.2.23 Global Group

Questo tipo di oggetto definisce un mezzo standardizzato per rappresentare una raccolta di proprietà da uno o più oggetti. Viene utilizzato per semplificare lo scambio di informazioni tra dispositivi BACnet fornendo un modo rapido per specificare tutti i membri del gruppo contemporaneamente come, ad esempio, tutte le temperature ambiente in un edificio.

Il Global Group supporta anche la segnalazione di eventi intrinseci ed espone un metodo per l'invio di notifiche periodiche di cambio di valore (COV). Questo tipo di oggetto può avviare un messaggio di notifica dell'evento che trasmette i valori di tutti i membri del gruppo contemporaneamente. Un oggetto Global Group differisce dall'oggetto Group in quanto i suoi membri non sono limitati allo stesso dispositivo in cui si trova l'oggetto Group, ma possono provenire da qualsiasi dispositivo sulla rete BACnet. Questo oggetto differisce anche dall'oggetto Structured View, poiché il primo fornisce i valori attuali delle proprietà dell'oggetto di riferimento, mentre il secondo definisce la posizione dell'oggetto, consentendo al dispositivo client di trovare l'oggetto e leggerne il valore attuale.

3.2.24 Group

L'oggetto è descritto dallo standard BACnet come una raccolta di altri oggetti e una o più delle loro proprietà. Esso è essenzialmente utilizzato per semplificare lo scambio di informazioni tra dispositivi BACnet attraverso un metodo abbreviato per specificare tutti i membri del gruppo contemporaneamente. Ad esempio, è possibile usare tale oggetto per monitorare la temperatura in diverse stanze in tutto un piano di un edificio.

Un gruppo può essere formato tramite qualsiasi combinazione di tipi di oggetto.

3.2.25 Integer Value

Questo oggetto consente di rappresentare un numero intero negativo come -138 o un numero intero positivo come +831. Questo tipo di oggetto può essere utilizzato per rappresentare valori positivi per rpm in senso orario e valori negativi per rpm in senso antiorario.

3.2.26 Large Analog Value

Un dispositivo BACnet può utilizzare un oggetto Large Analog Value per rendere qualsiasi tipo di valore di dati a doppia precisione accessibile ad altri dispositivi BACnet. Un uso eccellente di questo oggetto è rappresentare i kWh di un contatore elettrico.

3.2.27 Life Safety Point

Il tipo di oggetto Life Safety Point è definito dallo standard come quegli oggetti relativi alle applicazioni antincendio, di sicurezza e di protezione. La condizione di questo tipo di oggetto è determinata dai seguenti due fattori:

- Modalità: solitamente sotto il controllo dell'operatore del dispositivo
- Stato: indica la condizione del controllore in funzione della logica interna del dispositivo.

Generalmente, le applicazioni dell'oggetto Life Safety Point possono includere i seguenti dispositivi relativi all'automazione: rivelatori di incendio automatici, sirene, stazioni di trazione.

3.2.28 Life Safety Zone

Il tipo di oggetto Life Safety Zone ha qualità condizionali e di proprietà simili a quelle del tipo di oggetto Life Safety Point, ma l'attenzione è concentrata su un gruppo di oggetti, il quale dunque consente di definire zone di rivelazione incendio costituite da un elenco di rilevatori.

3.2.29 Lighting Output

Definisce un oggetto standardizzato le cui proprietà rappresentano le caratteristiche visibili esternamente di un output di illuminazione e include funzionalità dedicate specifiche per il controllo dell'illuminazione che altrimenti richiederebbero una programmazione esplicita.

L'emissione luminosa è di natura analogica e il livello di uscita fisica, o intervallo non normalizzato, è specificato come percentuale linearizzata (da 0% a 100%), dove il valore effettivo può variare dal 5% all'85%. Questi valori possono essere definiti come un sottoinsieme di livelli di output fisici definiti da `Min_Actual_Value` e `Max_Actual_Value`. Il livello delle luci può essere modificato direttamente in un livello assoluto scrivendo nel `Present_Value`. Questa proprietà è comandabile e utilizza il meccanismo dell'array di priorità a 16 livelli di BACnet per arbitrare tra più scrittori sull'uscita di illuminazione.

Questa nozione di avere più di un dispositivo che scrive su un oggetto non è molto usata nel mondo HVAC, ma è comune nel mondo dell'illuminazione. La luce potrebbe avviarsi con un livello di luminosità definito durante l'occupazione, il quale potrebbe essere ridotto da un sensore di illuminazione diurna se il sole splende nello spazio ufficio. Potrebbe esserci una strategia di eliminazione del carico di illuminazione controllata da un oggetto Load Control che potrebbe quindi ridurre l'emissione di illuminazione. L'utente locale normalmente ha l'ultima parola e può impostare la luminosità delle luci della stanza tramite un controller locale sul muro.

`Lighting_Command` è una proprietà scrivibile che fornisce funzionalità di illuminazione aggiuntive con speciali funzioni specifiche di illuminazione come rampa, gradino e dissolvenza. Il `Lighting_Command` fornisce anche un meccanismo di avviso lampeggiante per avvisare gli occupanti della stanza che le luci stanno per spegnersi. Il meccanismo di avviso di lampeggiamento è interno al tipo di oggetto Lighting Output e può causare l'accensione e il lampeggiamento delle luci fisiche o emettere una notifica in altro modo.

L'esempio in figura 2.15 illustra come un Lighting Output object type può essere utilizzato in un tipico scenario di ufficio: prima delle 7:00 le luci sono spente poiché il tipo di oggetto Lighting Output viene controllato al valore predefinito di abbandono di 0,0% o

disattivato. Alle 7:00 uno scheduler (ad esempio un oggetto BACnet Schedule o un altro processo automatizzato) accende le luci fisiche scrivendo 100% nella proprietà Present_Value con priorità 9. Alle 18:00 viene eseguito un comando di illuminazione WARN_RELINQUISH con

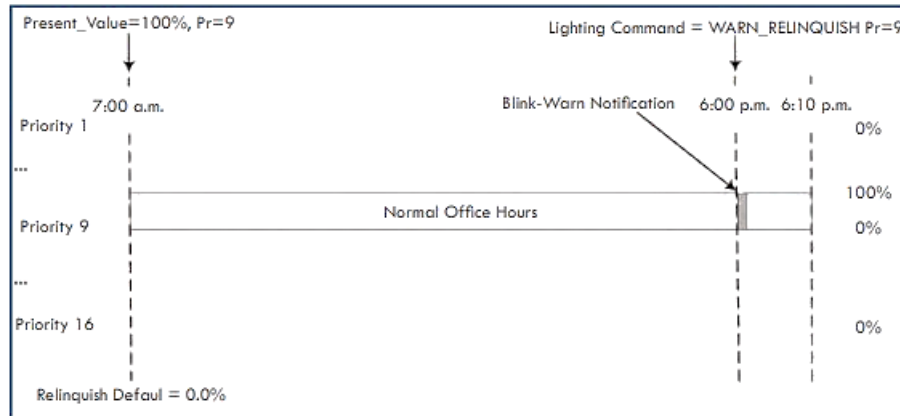


Figura 2.15: Programma giornaliero di gestione illuminazione utilizzando Lighting Output. (Wichenko, 2013)

priorità 9, che provoca un'immediata notifica di avviso di lampeggiamento ma lascia le luci accese finché il timer di uscita non è scaduto. Supponendo che sia specificato un tempo di uscita di 10 minuti (600 secondi), il valore con priorità 9 viene abbandonato alle 18:10 e Present_Value ritorna a 0%.

3.2.30 Load Control

Il tipo di oggetto Load Control definisce un oggetto standardizzato le cui proprietà rappresentano le caratteristiche visibili esternamente di un meccanismo per il controllo dei requisiti di carico. Un dispositivo BACnet può utilizzare un oggetto Load Control per consentire il controllo esterno sull'eliminazione di un carico che controlla.

3.2.31 Loop

Tale oggetto rappresenta gli attributi fisici esterni di qualsiasi categoria del loop di controllo. La flessibilità ottimale di questo oggetto si ottiene includendo tre costanti di guadagno specifiche senza valori assunti come unità:

- Costante proporzionale
- Costante integrale
- Costante derivata

3.2.32 Multi-state Input

Definisce un oggetto standardizzato il cui Present_Value rappresenta il risultato di un processo algoritmico all'interno del dispositivo BACnet in cui risiede l'oggetto. Il processo algoritmico stesso è una questione locale e non è definito dal protocollo. Il Present_Value o lo stato dell'oggetto Multi-state Input può essere il risultato di una combinazione logica di più

ingressi binari o la soglia di uno o più ingressi analogici o il risultato di un calcolo matematico. Ad esempio, può modellare lo stato di un refrigeratore che può essere in stato On, Off o nel ciclo di sbrinamento.

3.2.33 Multi-state Output

Riproduce un oggetto standardizzato le cui proprietà rappresentano lo stato desiderato di una o più uscite fisiche o processi all'interno del dispositivo BACnet in cui risiede l'oggetto. Le effettive funzioni associate a uno stato specifico sono una questione locale e non specificate dal protocollo. Ad esempio, uno stato particolare può rappresentare la condizione attiva/inattiva di diverse uscite fisiche o forse il valore di un'uscita analogica.

3.2.34 Multi-state Value

Il Multi-state Value Object è essenzialmente un parametro del sistema di controllo nella memoria di un dispositivo BACnet. Analogamente agli altri oggetti multi-stato, le funzioni associate sono una questione locale e non specificate dallo standard BACnet. Gli esempi di stato possono includere la condizione attiva/inattiva di ingressi e uscite o il valore di un ingresso o uscita analogica.

3.2.35 Notification Class

Il tipo di oggetto Notification Class definisce un oggetto standardizzato che rappresenta e contiene le informazioni necessarie per la distribuzione delle notifiche degli eventi all'interno dei sistemi BACnet. Le classi di notifica sono utili per oggetti che avviano eventi che hanno esigenze identiche in termini di gestione di notifica, quali dovrebbero essere le destinazioni per le loro notifiche e come dovrebbero essere riconosciute.

Una classe di notifica definisce come verrà assegnata la priorità alle notifiche degli eventi nella loro gestione in base agli eventi TO-OFFNORMAL, TO-FAULT e TO-NORMAL, definisce se queste categorie di eventi richiedono un riconoscimento, il che avviene sempre da parte di un operatore umano e, infine, quali dispositivi o processi di destinazione dovrebbero ricevere le notifiche.

Lo scopo della definizione delle priorità è fornire un mezzo per garantire che gli allarmi o le notifiche di eventi con considerazioni temporali critiche non vengano ritardati inutilmente. Il possibile intervallo di priorità è compreso tra 0 e 255, dove un numero inferiore indica una priorità più alta.

3.2.36 Notification Forwarder

Il tipo di oggetto Notification Forwarder consente ai dispositivi in grado di distribuire notifiche a un numero limitato di destinazioni di ricevere le proprie notifiche da molte destinazioni. Questo tipo di oggetto differisce dal Notification Class in quanto non viene utilizzato per originare notifiche di eventi, ma piuttosto viene utilizzato per inoltrare notifiche di eventi a un numero diverso e potenzialmente maggiore di destinatari.

Può essere pensato come un elenco di destinatari che devono essere avvisati quando scatta l'antifurto. Il sistema di sicurezza dell'edificio invia l'allarme alla società di allarme. L'azienda (che rappresenta il tipo di oggetto Notification Forwarder) notifica quindi al personale e alle forze dell'ordine che si è verificato un allarme. Questo oggetto è ideale per dispositivi di piccole dimensioni, come le scatole VAV. La casella invia la notifica a un controllore di supervisione, che a sua volta invia l'evento a un Recipient_List in un oggetto Notification Class.

3.2.37 Positive Integer Value

Questo oggetto consente di rappresentare solo un numero intero positivo come +125, potrebbe rappresentare il conteggio del numero di spazi per uffici in un edificio che richiedono il riscaldamento.

3.2.38 Program

Dato un programma applicativo eseguibile nel dispositivo, con questo oggetto è possibile conoscere lo stato di avanzamento. In questo contesto, un programma applicativo è una rappresentazione astratta di un processo all'interno di un dispositivo BACnet, che sta eseguendo un particolare corpo di istruzioni che agiscono su una particolare raccolta di strutture di dati.

L'oggetto permette dall'esterno la modifica dello stato di avanzamento, ad esempio, si può decidere di mandare in esecuzione, stoppare, caricare o scaricare il programma.

3.2.39 Pulse Converter

Il tipo di oggetto Pulse Converter rappresenta un processo in base al quale le misurazioni in corso effettuate su una certa quantità, come l'utilizzo di energia elettrica o acqua o gas naturale, e rappresentate da impulsi o conteggi, potrebbero essere monitorate su alcuni intervalli di tempo per applicazioni come la gestione dei picchi di carico, dove è necessario effettuare misurazioni periodiche, ma dove non è richiesta una contabilizzazione precisa di ogni impulso o conteggio in ingresso.

L'oggetto in questione potrebbe rappresentare un input fisico. In alternativa, potrebbe acquisire i dati dal Present_Value di un oggetto Accumulator, che rappresenta un input nello stesso dispositivo dell'oggetto Pulse Converter. Questo collegamento è illustrato dalla linea tratteggiata nella figura seguente (fig. 2.16). Ogni volta che viene incrementata la proprietà Present_Value dell'oggetto Accumulator, viene incrementata anche la proprietà Count dell'oggetto Pulse Converter. La proprietà Present_Value dell'oggetto Pulse Converter può essere regolata in qualsiasi momento scrivendo nella proprietà Adjust_Value, che provoca la regolazione della proprietà Count e il ricalcolo di Present_Value da Count.

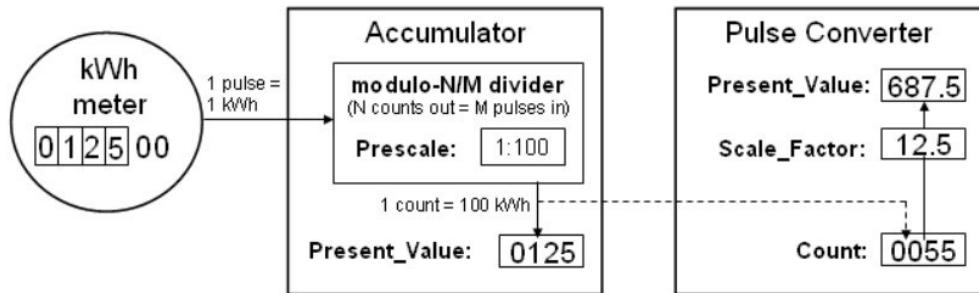


Figura 2.16: Esempio di utilizzo di un Pulse Converter associato ad un Accumulator.

3.2.40 Schedule

Definisce un oggetto standardizzato utilizzato per descrivere una pianificazione periodica che può ricorrere durante un intervallo di date, con eccezioni facoltative in orari arbitrari in date arbitrarie. L'oggetto Schedule funge anche da collegamento tra questi orari pianificati e la scrittura di "valori" specificati in proprietà specifiche di oggetti specifici in tali orari.

Gli orari sono suddivisi in giorni, di cui esistono due tipi: giorni normali all'interno di una settimana e giorni di eccezione. Entrambi i tipi di giorni possono specificare eventi di pianificazione per l'intera giornata o porzioni di giornata e un meccanismo di priorità definisce quale evento pianificato ha il controllo in un dato momento.

Lo stato corrente dell'oggetto Schedule è rappresentato dal valore della relativa proprietà Present_Value, che in genere viene calcolata utilizzando le coppie ora/valore dalle proprietà Weekly_Schedule ed Exception_Schedule, con un valore predefinito da utilizzare quando non sono attive pianificazioni.

3.2.41 Time Pattern Value

Gli oggetti Time Pattern Value consentono ai caratteri jolly come l'asterisco (*) di rappresentare qualsiasi valore nel campo. Il valore jolly dell'oggetto in questione potrebbe avere un valore attuale di 12.*.*.*, dove gli asterischi tra i punti rappresentano qualsiasi minuto, secondo o centesimo di secondo.

3.2.42 Time Value

Questo oggetto consente di rappresentare un singolo valore temporale specifico come 12:34:56.

3.2.43 Trend Log

L'oggetto Trend Log è definito dallo standard come un monitor di proprietà per un oggetto specifico referenziato. Quando vengono raggiunte determinate condizioni predefinite, salva (da cui deriva *log*) il valore della proprietà e viene prodotto un registro con il valore misurato e un indicatore di data/ora che viene inserito in un buffer per il recupero futuro.

Questo buffer può essere facoltativamente corretto in termini di dimensioni dallo sviluppatore.

L'oggetto offre allo sviluppatore la possibilità di registrare i valori periodicamente oppure su variazione di valore.

Ogni voce del buffer con timestamp è chiamata "record" del registro delle tendenze. L'oggetto di riferimento può risiedere nello stesso dispositivo dell'oggetto Trend Log o in un dispositivo esterno.

Ciascun oggetto Trend Log mantiene un buffer interno di dimensioni fisse. Questo buffer si riempie o aumenta man mano che vengono aggiunti record di registro. Se il buffer si riempie, il record meno recente viene sovrascritto quando viene aggiunto un nuovo record oppure la raccolta può essere interrotta. In generale, la registrazione delle voci può essere attivata o disattivata in qualsiasi momento.

Una protezione contro la perdita di dati dei valori registrati fornisce la voce di configurazione opzionale UserBitTrendLog, ad esempio se un oggetto trend log è stato cancellato e quindi anche i valori memorizzati nella proprietà Log_Buffer non sono più disponibili.

3.2.44 Trend Log Multiple

Il tipo di oggetto Trend Log Multiple funziona allo stesso modo del tipo di oggetto Trend Log originale, ad eccezione del fatto che il presente oggetto monitora una o più proprietà di uno o più oggetti di riferimento, nello stesso dispositivo dell'oggetto Trend Log Multiple o in un dispositivo esterno. Ad esempio, il tipo di oggetto Trend Log Multiple consente a un utente di tracciare l'andamento della temperatura dell'aria esterna nel controller di un fornitore e del setpoint dell'acqua di alimentazione della caldaia nel controller di un altro fornitore.

3.3 Proprietà degli oggetti BACnet

Lo standard BACnet identifica 123 diverse proprietà degli oggetti e per ogni tipo di oggetto ne viene specificato un sottoinsieme diverso, in figura 2.17 ne sono elencate alcune. Lo scopo di una proprietà è di permettere agli altri device BACnet di leggere le informazioni sull'oggetto in esame e, potenzialmente, settare un valore differente nella proprietà.

acked-transitions	(0),	controlled-variable-value	(21),	file-access-method	(41),
ack-required	(1),	cov-increment	(22),	file-size	(42),
action	(2),	datelist	(23),	file-type	(43),
action-text	(3),	daylight-savings-status	(24),	firmware-revision	(44),
active-text	(4),	deadband	(25),	high-limit	(45),
active-vt-sessions	(5),	derivative-constant	(26),	inactive-text	(46),
alarm-value	(6),	derivative-constant-units	(27),	in-process	(47),
alarm-values	(7),	description	(28),	instance-of	(48),
all	(8),	description-of-halt	(29),	integral-constant	(49),
all-writes-successful	(9),	device-address-binding	(30),	integral-constant-units	(50),
apdu-segment-timeout	(10),	device-type	(31),	issue-confirmed-notifications	(51),
apdu-timeout	(11),	effective-period	(32),	limit-enable	(52),
application-software-version	(12),	elapsed-active-time	(33),	list-of-group-members	(53),
archive	(13),	error-limit	(34),	list-of-object-property-references	(54),
bias	(14),	event-enable	(35),	list-of-session-keys	(55),
change-of-state-count	(15),	event-state	(36),	local-date	(56),
change-of-state-time	(16),	event-type	(37),	local-time	(57),
--	(17),	event-parameters	(83),	location	(58),
--	(18),	exception-schedule	(38),	low-limit	(59),
controlled-variable-reference	(19),	fault-values	(39),	manipulated-variable-reference	(60),
controlled-variable-units	(20),	feedback-value	(40),	maximum-output	(61),

Figura 2.17: Elenco delle proprietà degli oggetti BACnet. (Swan, 1996)

La specifica richiede che alcune proprietà debbano essere presenti per ogni oggetto, mentre altre, specificate dalla casa produttrice, sono facoltative (si veda la Tabella 2). In entrambi i casi, le proprietà implementate hanno comportamenti specifici definiti, in particolare quelli coinvolti nelle notifiche di allarmi o eventi e quelli che hanno un effetto su valori o stati di controllo.

Le proprietà *Limit_Enable*, *Event_Enable*, *Acked_Transitions* e *Notify_Type* si occupano tutte del rilevamento di condizioni insolite e possibilmente pericolose al sensore e generando le notifiche appropriate o allarmi in risposta.

Alcune delle proprietà standard sono richieste dalle specifiche BACnet per essere scrivibili mentre altre possono essere scrivibili a discrezione del produttore. In generale tutto può essere letto da tutti.

Tabella 2: Esempio delle proprietà BACnet obbligatorie e facoltative in un BACnet Analog Input Object. (Swan, 1996)

PROPERTY	BACnet	EXAMPLE
Object_Identifier	Required	Analog Input #1
Object_Name	Required	"AI 01"
Object_Type	Required	Analog Input
Present_Value	Required	68.0
Description	Optional	"Outside Air Temperature"
Device_Type	Optional	"10k Thermistor"
Status_Flags	Required	In_Alarm, Fault, Overridden, Out_Of_Service flags
Event_State	Required	Normal (plus various problem-reporting states)
Reliability	Optional	No_Fault_Detected (plus various fault conditions)
Out_Of_Service	Required	False
Update_Interval	Optional	1.00 (seconds)
Units	Required	Degrees-Fahrenheit
Min_Pres_Value	Optional	-100.0, minimum reliably read value
Max_Pres_Value	Optional	+300.0, maximum reliably read value
Resolution	Optional	0.1
COV_Increment	Optional	Notify if Present_Value changes by increment: 0.5
Time_Delay	Optional	Seconds to wait before detecting out-of-range: 5
Notification_Class	Optional	Send COV notification to Notification Class Object: 2
High_Limit	Optional	+215.0, Upper normal range
Low_Limit	Optional	-45.0, Lower normal range
Deadband	Optional	0.1
Limit_Enable	Optional	Enable High-limit-reporting, Low-limit-reporting.
Event_Enable	Optional	Enable To_Offnormal, To_Fault, To_Normal change reporting.
Acked_Transitions	Optional	Flags indicating received acknowledgments for above changes.
Notify_Type	Optional	Events or Alarms

In generale, ci sono tre proprietà obbligatorie che devono essere definite per ogni singolo oggetto BACnet:

- Object_Identifier: è un codice a 32 bit, dove i primi 10 bits sono dedicati all'Object Type e gli altri 22 sono per l'Instance Number;
- Object_Name: è un testo univoco che identifica l'oggetto nel device;
- Object_Type: sono riconosciuti da 0 a 127 tipi di oggetti standard BACnet, dal numero 128 a 1023 sono parametri personalizzabili.

Le prime due proprietà devono assolutamente essere univoche all'interno di un dispositivo BACnet, poiché sono responsabili del riconoscimento dell'oggetto all'interno dello stesso.

L'Instance Number è un numero, da 0 a 4194302, che identifica un dispositivo in modo univoco sull'intera rete BACnet interconnessa. Nessun altro dispositivo in qualsiasi punto della rete deve avere lo stesso numero di istanza BACnet, anche se si trovano su bus diversi o addirittura in edifici diversi. Potenzialmente, dunque, un dispositivo BACnet potrebbe avere oltre quattro milioni di oggetti di un particolare tipo.

Ogni proprietà ha un identificatore univoco (*Property Identifier*), un tipo di dati designato (*Property Datatype*) e un codice di conformità (*Conformance Code*). Quest'ultimo definisce i permessi di accesso di una proprietà e specifica se una proprietà è obbligatoria (R) o facoltativa (O), di sola lettura o deve essere scrivibile, come mostrato in fig. 2.18. Per quanto riguarda invece il Property Identifier, questo non è altro che un'enumerazione che rappresenta una determinata proprietà e, da standard BACnet, se ne definiscono 192 totali.

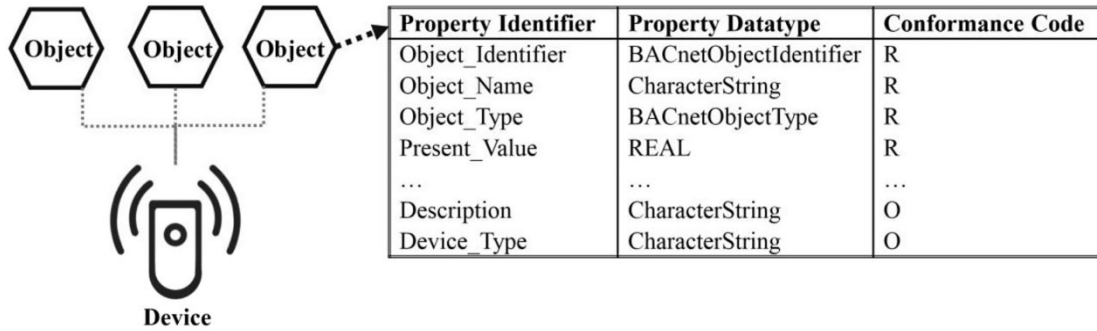


Figura 2.18: Schema esemplificativo di un dispositivo BACnet, con la sua aggregazione interna di BACnet Objects e le relative proprietà. (Tang et al., 2020)

BACnet consente ai fornitori di aggiungere proprietà proprietarie ma, come accade per gli oggetti proprietari, queste potrebbero non essere comprese o accessibili da apparecchiature di altri produttori.

3.4 Servizi applicativi BACnet

I servizi sono i mezzi con cui un dispositivo BACnet acquisisce informazioni da un altro dispositivo, comanda un altro dispositivo per eseguire alcune azioni o annuncia a uno o più dispositivi che si è verificato un determinato evento. Ogni richiesta di servizio emessa, con il seguente riconoscimento del servizio restituito e cioè la risposta data, diventano un pacchetto di messaggi trasferiti in rete dal dispositivo di invio a quello di ricezione.

In generale i servizi seguono le basi del modello client/server; dunque, un client BACnet è un device che richiede un servizio e, viceversa, un server BACnet è un device che fornisce un servizio

Un *Application Program* in esecuzione sul dispositivo BACnet emette richieste di servizio e le elabora al ricevimento. Il programma applicativo è un software a tutti gli effetti, il quale esegue le operazioni richieste dal dispositivo in cui è installato. Nel caso di una workstation dell'operatore, tale software potrebbe mantenere una visualizzazione di diversi ingressi del sensore, poiché l'operatore non emetterà periodicamente richieste di servizio agli oggetti appropriati nei dispositivi di destinazione per ottenere il valore più recente dagli ingressi. Nel dispositivo monitorato, la richiesta di servizio viene elaborata nel suo programma di applicazione e la risposta contenente i dati richiesti viene restituita.

BACnet definisce attualmente 40 servizi applicativi raggruppati in cinque diverse categorie di servizio, catalogabili come: Alarm and Event, File Access, Object Access, Remote

Device Management and Virtual Terminal Services, sebbene questi ultimi servizi siano stati largamente rimpiazzati da strumenti basati sul web.

Per ciascuno dei servizi "confermati" si prevede una risposta, in genere con dati, etichettata "C", e per ciascuno dei servizi "non confermati" non è prevista alcuna risposta, etichettata "U". In entrambi i casi il dispositivo BACnet può avere la possibilità di avviare la richiesta di servizio o anche avere la capacità di elaborare e rispondere a una richiesta ricevuta di quel tipo, o entrambi.

BACnet fornisce tre metodi distinti, seppur complementari, per la gestione degli eventi, compresi quelli considerati tanto importanti da essere designati come allarmi.

Il primo si chiama *Intrinsic Reporting* e fa uso di parametri incorporati in singoli oggetti e di algoritmi di tipo di evento standardizzati. Ad oggi sono definiti nove algoritmi di questo tipo, come *Out_Of_Range*, *Change_Of_State*, ma il protocollo li applica inflessibilmente alle proprietà specificate degli oggetti standard.

Il secondo, *Algorithmic Change Reporting* usufruisce degli stessi algoritmi del primo, consentendo loro però di essere applicati in modo più ampio a qualsiasi proprietà di qualsiasi oggetto. I parametri associati all'algoritmo selezionato, come ad esempio limite superiore, limite inferiore, ritardo temporale, sono contenuti in un oggetto chiamato *Event Enrollment*, piuttosto che effettivamente nell'oggetto di riferimento, il che permette l'attuazione di diversi algoritmi alla stessa proprietà, se doveroso.

Sia il prospetto delle modifiche caratteristico che quello algoritmico possono utilizzare l'oggetto *Notification Class* che contiene informazioni su come devono essere distribuite le notifiche degli eventi, confermate e riconosciute o non confermate e dunque non riconosciute. Questo accostamento di funzionalità consente un riconoscimento e una distribuzione di allarmi ed eventi enormemente importanti: le notifiche possono essere adattate a diversi destinatari in diversi momenti della giornata o della settimana, possono essere assegnate priorità diverse e così via. Un allarme salvavita, ad esempio, potrebbe essere indirizzato a postazioni di lavoro specifiche durante la giornata lavorativa, ma causare l'attivazione di una procedura di chiamata in uscita dopo l'orario di lavoro o durante il fine settimana.

Il terzo metodo di segnalazione eventi è chiamato *Change of Value (COV)*. Questo fa in modo che venga trasmessa una notifica quando una particolare proprietà cambia di un'entità predefinita oppure, a libero arbitrio del server COV, a intervalli di tempo predefiniti. Le notifiche COV non sottoscritte, generalmente comunque trasmesse, procurano un meccanismo per distribuire il valore corrente di informazioni significative a livello complessivo, come la temperatura dell'aria esterna di un determinato luogo o il suo stato di occupazione, a una cadenza di ripetizione fissata dal server.

I consumatori possono avvalersi del servizio *SubscribeCOV* per registrarsi al fine di ricevere le notifiche delle proprietà predefinite degli oggetti standard o il servizio

SubscribeCOVProperty per richiedere la notifica COV per qualsiasi proprietà di qualsiasi oggetto con qualsiasi incremento COV desiderato.

Dei vari servizi BACnet dedicati alla gestione di allarmi ed eventi (mostrati in Tabella 3), solamente il servizio *AcknowledgeAlarm* è rivolto all'operatore umano. Questo, infatti, rappresenta un mezzo di comunicazione per un dispositivo che concepisce l'allarme che un essere umano ha, a tutti gli effetti, notato e al quale sta rispondendo. Il dispositivo di origine può utilizzare la ricezione di tale conferma, o la sua mancanza, entro un intervallo di tempo specificato, per invocare un'altra logica specifica dell'applicazione per gestire la condizione di allarme: l'avvio di un arresto precauzionale del sistema, l'esecuzione di una notifica di chiamata in uscita ad alcuni destinatari aggiuntivi e così via.

Tabella 3: Servizi di allarme ed eventi in un dispositivo BACnet con relativa descrizione. (Swan, 1996)

SERVICE	BACnet	DESCRIPTION
AcknowledgeAlarm	C	Used to tell sender of alarm that a human has seen the alarm.
ConfirmedCOVNotification	C	Tells subscribing devices of the COV that occurred in a property.
ConfirmedEventNotification	C	Used to tell sender of a possible error condition.
GetAlarmSummary	C	Requests from a device a list of "active alarms," if any.
GenEnrollmentSummary	C	Requests a list of "event" (possible error) generating objects.
SubscribeCOV	C	Sent by a device to request that it be told of COVs in an object.
UnconfirmedCOVNotification	U	Tells subscribing devices that a change has occurred to one or more properties of a particular object.

I servizi di accesso ai file in BACnet, si veda la Tabella 4, vengono utilizzati per leggere e manipolare i file nei dispositivi BACnet. In BACnet, i file rappresentano byte di dati di gruppo di lunghezza e significato arbitrari; non si riferiscono necessariamente a nessun tipo di dispositivo di archiviazione di massa. Ad ogni file accessibile da BACnet è associato un oggetto file.

La parola "Atomic" nel nome del servizio significa semplicemente che può avvenire una sola operazione di lettura o scrittura alla volta. Altri tentativi di accedere al file durante un accesso falliranno o verranno sospesi.

Al momento non ci sono file standard definiti da BACnet. Per definizione, tutti gli accessi ai file sono di proprietà del fornitore sebbene BACnet abbia riservato i tipi di file *EventLog* e *ValueLog* per strutture di file da definire in un secondo momento.

Tabella 4: Servizi di accesso ai file BACnet con relativa descrizione. (Swan, 1996)

SERVICE	BACnet	DESCRIPTION
AtomicReadFile	C	Requests part or all of a File object's file.
AtomicWriteFile	C	Writes to part or all of a File object's file.

I servizi di accesso agli oggetti, elencati in Tabella 5, forniscono i mezzi per leggere, condizionare e scrivere proprietà e per aggiungere o eliminare oggetti e gruppi di valori di proprietà. I servizi di tipo Multiple, nello specifico, sono forniti per la lettura e la scrittura di proprietà e, in generale, lo scopo dei servizi più complessi, come *ReadPropertyMultiple* e *WritePropertyMultiple*, è quello di combinare tutte le letture o le scritture di proprietà di oggetti all'interno di un dispositivo BACnet in un singolo messaggio, riducendo così il sovraccarico di rete.

Il servizio *ReadPropertyConditional* va oltre quanto appena detto: il dispositivo che elabora la richiesta, infatti, verifica ciascuna proprietà referenziata in base ai criteri inclusi nella richiesta e restituisce il valore solo se i criteri sono soddisfatti.

Tabella 5: Servizi di accesso agli oggetti BACnet con relativa descrizione. (Swan, 1996)

SERVICE	BACnet	DESCRIPTION
AddListElement	C	Adds one or more items to a property that is a list.
RemoveListElement	C	Removes one or more items from a property that is a list.
CreateObject	C	Used to create a new instance of an object in the serving device.
DeleteObject	C	Used to delete a particular object in the serving device.
ReadProperty	C	Returns a value of one property of one object.
ReadPropertyConditional	C	Returns the values of multiple properties in multiple objects.
ReadPropertyMultiple	C	Returns the values of multiple properties of multiple objects.
WriteProperty	C	Writes a value to one property of one object.
WritePropertyMultiple	C	Writes values to multiple properties of multiple objects.

I servizi di gestione dei dispositivi remoti, consultabili in Tabella 6, forniscono una serie di funzioni disparate, tra cui il controllo dell'operatore, il trasferimento di messaggi specializzati e le funzioni di indirizzamento/configurazione automatica.

I servizi *DeviceCommunicationControl* e *ReinitializeDevice* hanno lo scopo di fornire a un operatore umano gli strumenti diagnostici che possono essere richiamati in remoto. Con essi è possibile ordinare a un dispositivo BACnet di ignorare tutti i messaggi BACnet, ad eccezione dei due citati, oppure di essere avviato a caldo o a freddo.

I servizi *ConfirmedPrivateTransfer* e *UnconfirmedPrivateTransfer* vengono utilizzati per trasmettere messaggi al di fuori dello standard BACnet, richiamando di fatto servizi non standard. Questi riportano sia il *Vendor ID Code* sia il *Service Code* in formato standard BACnet, mentre il resto del contenuto dipende interamente dal fornitore e spesso può capitare che non sia interpretabile da dispositivi di altri produttori.

I servizi *ConfirmedTextMessage* e *UnconfirmedTextMessage* trasportano i messaggi di testo ad altri dispositivi, come le stampanti.

Il servizio *TimeSynchronization* viene trasmesso da dispositivi designati che dispongono di orologi per altri dispositivi o broadcast, in modo che i dispositivi possano essere sincronizzati.

I servizi *Who-Is* e *I-Am* sono utilizzati per ottenere gli indirizzi di rete dei dispositivi BACnet su una rete BACnet. Possono facilitare ampiamente l'operato dell'installatore riducendo o eliminando la necessità di programmare gli indirizzi internet di altri dispositivi in ogni dispositivo BACnet. Un dispositivo BACnet che necessita di conoscere l'indirizzo di uno o più dispositivi, invece, può trasmettere una richiesta *Who-Is*, che nella pratica è un messaggio, sulla rete internet specificando un *Device Object* con un *Instance Number* o un intervallo di *Instance Numbers*. Le risposte non tornano come risposta.

Contrariamente, i dispositivi che hanno gli *Object Device* specificati trasmettono una richiesta di servizio *I-Am* sulla rete locale, su una rete remota o sull'intera rete, in modo che il dispositivo richiedente veda la risposta, che porta con sé le informazioni dell'indirizzo di colui che risponde. Ciò consente ad altri dispositivi, che potrebbero aver bisogno di conoscere i risponditori, di acquisire le informazioni sull'indirizzo senza creare ulteriore traffico di rete.

I servizi *Who-Has* e *I-Have* sono simili a *Who-Is* e *I-Am*, ma *Who-Has* aggiunge un *Object Identifier*, o *Object Nam*. I dispositivi di ricezione che contengono un oggetto corrispondente alla domanda trasmettono la richiesta del servizio *I-Have*.

Tabella 6: Servizi di gestione dei dispositivi remoti con relativa descrizione. (Swan, 1996)

SERVICE	BACnet	DESCRIPTION
DeviceCommunicationControl	C	Tells a device to stop (and start!) accepting network messages.
ConfirmedPrivateTransfer	C	Sends a vendor-proprietary message to a device.
UnconfirmedPrivateTransfer	U	Sends a vendor-proprietary message to one or more devices.
ReinitializeDevice	C	Orders the receiving device to cold- or warm-boot itself.
ConfirmedTextMessage	C	Conveys a text message to another device.
UnconfirmedTextMessage	U	Sends a text message to one or more devices.
TimeSynchronization	U	Sends the current time to one or more devices.
Who-Has	U	Asks which BACnet devices contain a particular Object.
I-Have	U	Affirmative response to Who-Has, broadcast.
Who-Is	U	Asks about the presence of specified BACnet devices.
I-Am	U	Affirmative response to Who-Is, broadcast.

Infine, i servizi *Virtual Terminal* (VT) possono essere utilizzati da un operatore per stabilire una connessione bidirezionale basata su un testo con un programma applicativo in esecuzione in un dispositivo remoto. Per l'intera durata di una sessione VT stabilita con il dispositivo remoto, il dispositivo dell'operatore sembra a tutti gli effetti un terminale collegato al programma applicativo remoto.

Tabella 7: Servizi di Virtual Terminal BACnet con relativa descrizione. (Swan, 1996)

SERVICE	BACnet	DESCRIPTION
VT-Open	C	Establishes a virtual terminal session with a remote BACnet device.
VT-Close	C	Closes an established virtual terminal session.
VT-Data	C	Sends text from one device to the other participating in a session.

3.5 PICS and BIBB's

Nonostante lo standard BACnet circoscriva un insieme considerevole di servizi, per la maggior parte dei dispositivi è sufficiente specificarne solo un sottoinsieme, poiché l'implementazione di tutti i servizi aumenterebbe invano le complessità e i costi senza fornire alcun vantaggio particolare. A tal scopo, per poter descrivere in modo sintetico le capacità offerte o richieste da un particolare dispositivo, è stato appoggiato il concetto di BIBBs (BACnet Interoperability Building Blocks) nello standard a partire dell'inizio degli anni 2000. Un BIBB descrive una tipica capacità funzionale in una delle cinque aree di interoperabilità dei dispositivi definite come:

1. condivisione dei dati;
2. gestione di allarmi ed eventi;
3. programmazione;
4. trend;
5. gestione di dispositivi e reti.

Sono generalmente fruibili in coppie client/server, i quali consentono di specificare con accuratezza se un determinato dispositivo funziona come iniziatore di una richiesta di servizio, risponditore a una richiesta di servizio o entrambi. Un BIBB può anche esigere la presenza di uno o più oggetti o il supporto di specifiche proprietà. Ad esempio, il BIBB *Trending-Viewing and Modifying Trends Internal-B* richiede che sia implementato il lato server del *ReadRange-Service* e che sia fornito un oggetto *Trend Log*.

	B-AWS	B-OWS	B-OD	B-BC	B-AAC	B-ASC	B-SA	B-SS
Data Sharing	DS-RP-A,B DS-RPM-A DS-WP-A DS-WPM-A DS-AV-A DS-AM-A	DS-RP-A,B DS-RPM-A DS-WP-A DS-WPM-A DS-V-A DS-M-A	DS-RP-A,B DS-WP-A DS-V-A DS-M-A	DS-RP-A,B DS-RPM-A,B DS-WP-A,B DS-WPM-B	DS-RP-B DS-RPM-B DS-WP-B DS-WPM-B	DS-RP-B DS-WP-B	DS-RP-B DS-WP-B	DS-RP-B
Alarm, Event Management	AE-N-A AE-ACK-A AE-AS-A AE-AVM-A AE-AVN-A AE-ELVM-A	AE-N-A AE-ACK-A AE-AS-A AE-VM-A AE-VN-A	 AE-VN-A	AE-N-I-B AE-ACK-B AE-INFO-B AE-ESUM-B	AE-N-I-B AE-ACK-B AE-INFO-B			
Scheduling	SCHED-AVM-A	SCHED-VM-A		SCHED-E-B	SCHED-I-B			
Trending	T-AVM-A	T-V-A		T-VMT-I-B T-ATR-B				
Device & Network Management	DM-DOB-A,B DM-ANM-A DM-ADM-A DM-DOB-B DM-DCC-A DM-MTS-A DM-OCD-A DM-RD-A DM-BR-A	DM-DOB-A,B DM-DOB-B DM-MTS-A	DM-DOB-A,B DM-DOB-B	DM-DOB-A,B DM-DOB-B DM-DCC-B DM-TS-B or DM-UTC-B DM-RD-B DM-BR-B	DM-DOB-B DM-DOB-B DM-DCC-B DM-TS-B or DM-UTC-B DM-RD-B	DM-DOB-B DM-DOB-B DM-DCC-B	DM-DOB-B	DM-DOB-B

Figura 2.19: Elenco dei BIBBs per i singoli dispositivi standard BACnet.

Per facilitare il lavoro degli specificatori, sono delineati diversi profili di dispositivi BACnet standard (fig. 2.19). Ogni profilo è una raccolta di BIBB che ha lo scopo di mappare le apparecchiature di building automation comunemente disponibili: postazioni di lavoro dell'operatore, controller di edifici avanzati e specifici dell'applicazione, attuatori intelligenti, e sensori intelligenti.

I BIBB sono vagliati per rappresentare un riferimento per il dato tipo di dispositivo. Per dichiarare la conformità a un determinato profilo, un produttore deve offrire al minimo le funzionalità contenute nel profilo, ma è totalmente libero di aggiungere qualsiasi funzionalità aggiuntiva appropriata all'applicazione prevista del dispositivo. I dettagli sulle parti di BACnet implementate in un dispositivo vengono documentati nella sua dichiarazione di conformità dell'implementazione del protocollo, detta PICS (Protocol Implementation Conformance Statement), che al suo interno contiene i BIBBs.

Se si prevede di seguire un determinato progetto in BACnet dove collaborano più costruttori, è necessario ottenere i PICS per verificare che i due sistemi offrono gli stessi servizi compatibili.

A tal proposito nel sito BACnet International è presente un elenco con PICs&BIBB's²³ per ogni casa produttrice, come ad esempio Schneider Electric, Mitsubishi Electric Corporation, Siemens e altri, per un totale di 1311 prodotti di 226 produttori, ma in continuo aggiornamento. Nella seguente immagine (fig. 2.20) è riportato un esempio della pagina web del sito BACnet International dedicata ai prodotti testati e certificati di Schneider.

L'elenco è diviso per macrogruppi di standard e profili di dispositivi; per ognuno poi è specificato il prodotto, il modello specifico, la versione, i PICS i quali non sempre sono disponibili, l'elenco BTL e la certificazione BTL.

Cliccando nel nome del prodotto che si vuole indagare si possono leggere le informazioni essenziali su di esso fornite direttamente dalla casa produttrice e non convalidate dal BACnet Testing Laboratories.

Poi, proseguendo, alla voce PICS di un prodotto si trova un documento, fornito dal produttore, che dichiara la conformità dell'implementazione del protocollo, che appunto corrisponde al PICS, dove sono chiarite le specifiche del prodotto come la data di fabbricazione, il nome di mercato, la versione del software associato, la versione del protocollo BACnet e una piccola descrizione.

²³ <https://www.bacnetinternational.net/btl/>




























Product	Model	Version	PICS	BTL Listing	Certificate
EcoStruxure Building Operation	AS-P, AS-B, ES, AS-P-NL	4.0.3			
bCX1 Controller/Router	bCX1-CR	4.5			
StruxureWare Building Operation Automation Server	AS	1.3.0.20011			
StruxureWare Building Operation Enterprise Server	SW-ES	1.3.0.20011			
StruxureWare Building Operation Enterprise Server	SW-ES	1.3.0.20011			
I/A Series® G3 ENC, TRD-T, TRD-J, TRD-NXS	ENC-410-1, ENC-410-1-N, ENC-520-2, ENC-520-2-N, ENC-SW-1M TRD-T-603, TRD-T-603-N, TRD-T-645, TRD-T-645-N TRD-J-200, TRD-J-600, TRD-J-600-USA, TRD-J-600E, TRD-J-600E-USA TRD-NXS-FL-G3, TRD-NXS-HD-G3	3.6			
I/A Series® G3 TRD-NXT	T-NXT-FL-G3, T-NXT-HD-G3	3.6			
bCX1 Controller/Router	bCX1-CR	4.5			
Enterprise Network Controller	ENC-4XX-XXX, ENC-5XX-XXX	3.2.20 or higher			

Figura 2.20: Interfaccia della pagina web di BACnet International dedicata ai PICs&BIBB's per Schneider Electric. (Fonte: <https://www.bacnetinternational.net/btl/>)

Seguono pagine dedicate a:

- descrizione dei BIBBs;
- dispositivi e oggetti supportati dal soggetto in analisi;
- capacità di segmentazione dei messaggi;
- opzioni di collegamento dati (BACnet IP, Ethernet, ARCNET, MS/TP master or slave, Point-To-Point EIA 232, LonTalk);
- opzioni di rete;
- capacità di binding;
- set di caratteri supportati (ANSI X3.4, ISO 8859-1, ISO 10646 (USC-2 e UCS-4), IBM Microsoft DBCS, JIS C 6226);
- proprietà supportate (S), scrivibili (W) e facoltative (O) per ogni singolo tipo di oggetto citato precedentemente (fig. 2.21);
- interpretazione di alcuni caratteri speciali all'interno di alcuni dati;
- restrizioni.

Analog Input			
Property	S	W	O
COV Increment	X	X	
Description	X	X	X
Event State	X		
Object Identifier	X		
Object Name	X		
Object Type	X		
Out Of Service	X	X	
Present Value	X	X	
Status Flags	X		
Units	X	X	

Binary Input			
Property	S	W	O
Active Text	X	X	X
Description	X	X	X
Event State	X		
Inactive Text	X	X	X
Object Identifier	X		
Object Name	X		
Object Type	X		
Out Of Service	X	X	
Polarity	X	X	
Present Value	X	X	
Status Flags	X		

Analog Output			
Property	S	W	O
COV Increment	X	X	
Description	X	X	X
Event State	X		
Object Identifier	X		
Object Name	X		
Object Type	X		
Out Of Service	X	X	
Present Value	X	X	
Priority Array	X		
Relinquish Default	X	X	
Status Flags	X		
Units	X	X	

Binary Output			
Property	S	W	O
Active Text	X	X	X
Description	X	X	X
Event State	X		
Inactive Text	X	X	X
Object Identifier	X		
Object Name	X		
Object Type	X		
Out Of Service	X	X	
Polarity	X	X	
Present Value	X	X	
Priority Array	X		
Relinquish Default	X	X	
Status Flags	X		

Analog Value			
Property	S	W	O
COV Increment	X	X	
Description	X	X	X
Event State	X		
Object Identifier	X		
Object Name	X	X	
Object Type	X		
Out Of Service	X	X	
Present Value	X	X	
Priority Array	X		
Relinquish Default	X	X	
Status Flags	X		
Units	X	X	

Figura 2.21: Proprietà supportate da oggetti appartenenti ad un controller B-AAC. (Schneider Electric, 2013)

Continuando l'analisi dell'interfaccia web, alla voce corrispondente all'elenco BTL, si trova un documento fornito direttamente dal laboratorio per i test BACnet che riporta sostanzialmente le medesime informazioni descritte pocanzi nella dichiarazione di conformità di implementazione del protocollo, con particolare attenzione ai BIBBs implementati dal dispositivo osservato (figura 2.22).

Il documento accerta che il prodotto sia stato esaminato presso un laboratorio di test BACnet qualificato e che sia risultato conforme a tutti i requisiti di interoperabilità necessari e in vigore alla data di test pubblicata, inoltre, rappresenta le capacità testate del soggetto. Per tutte le informazioni su funzionalità aggiuntive non trattate nel processo di test, lo stesso laboratorio invita a fare riferimento alla dichiarazione PICS del produttore sul sito web del protocollo associato.

BACnet Interoperability Building Blocks (BIBBs) Supported

BIBB	Name	BACnet Service	Support	Init	Exec
DS-RP-A ¹	Data Sharing - ReadProperty-A	ReadProperty	X	X	
DS-RP-B	Data Sharing - ReadProperty-B	ReadProperty	X		X
DS-RPM-B	Data Sharing - ReadPropertyMultiple-B	ReadPropertyMultiple	X		X
DS-RPC-B	Data Sharing - ReadPropertyConditional-B	ReadPropertyConditional	X		X
DS-WP-A ¹	Data Sharing - WriteProperty-A	WriteProperty	X	X	
DS-WP-B	Data Sharing - WriteProperty-B	WriteProperty	X		X
DS-WPM-B	Data Sharing - WritePropertyMultiple-B	WritePropertyMultiple	X		X
DS-COV-A	Data Sharing - COV-A	SubscribeCOV	X	X	
		ConfirmedCOVNotification	X		X
		UnconfirmedCOVNotification	X		X
DS-COV-B	Data Sharing - COV-B	SubscribeCOV	X		X
		ConfirmedCOVNotification	X	X	
		UnconfirmedCOVNotification	X	X	
AE-N-I-B	Alarm and Event-Notification-B	ConfirmedEventNotification	X	X	
		UnconfirmedEventNotification	X	X	
AE-ACK-B	Alarm and Event-ACK-B	AcknowledgeAlarm	X		X
AE-INFO-B	Alarm and Event-Information-B	GetEventInformation	X		X
SCHED-I-B	Scheduling-Internal-B		X		
T-VMT-I-B	Trending - Viewing and Modifying Trends Internal-B	ReadRange	X		X
T-ATR-B ¹	Trending – Automated Trend Retrieval-B	ConfirmedEventNotification	X	X	
		ReadRange	X		X
DM-DDB-A	Device Management-Dynamic Device Binding-A	Who-Is	X	X	
		I-Am	X		X
DM-DDB-B	Device Management-Dynamic Device Binding-B	Who-Is	X		X
		I-Am	X	X	
DM-DOB-A	Device Management-Dynamic Object Binding-A	Who-Has	X	X	
		I-Have	X		X
DM-DOB-B	Device Management-Dynamic Object Binding-B	Who-Has	X		X
		I-Have	X	X	
DM-DCC-B	Device Management-DeviceCommunicationControl-B	DeviceCommunicationControl	X		X
DM-TS-B	Device Management-TimeSynchronization-B	TimeSynchronization	X		X
DM-UTC-B	Device Management-UTCTimeSynchronization-B	UTCTimeSynchronization	X		X
DM-RD-B	Device Management-ReinitializeDevice-B	ReinitializeDevice	X		X
DM-BR-B ²	Device Management-Backup and Restore-B	AtomicReadFile	X		X
		AtomicWriteFile	X		X
		ReinitializeDevice	X		X
DM-OCD-B	Device Management-Object Creation and Deletion-B	CreateObject	X		X
		DeleteObject	X		X

Figura 2.22: BIBBs supportati da un Advanced application Controller B-AAC.(Schneider Electric, 2013)

Infine, come si può vedere nella figura 2.20, è presente una sezione dedicata alla certificazione BTL, per la quale sarà ora dedicato un paragrafo specifico.

Date queste premesse, per facilitare la comprensione del concetto PICS and BIBB's, è ora illustrato un esempio operativo. Si ipotizzi di disporre in una stanza un dispositivo avente un sensore di temperatura la cui temperatura è accessibile come proprietà di un oggetto BACnet. Nella stessa stanza in esame un altro dispositivo deve integrare questa temperatura. In questo caso il dispositivo che ha il sensore è il Server, il quale deve essere in grado di ricevere una richiesta di servizio *ReadProperty* ed eseguirlo e restituire un risultato. Il BIBB per questo tipo di funzionalità è chiamato "DS RP-B", che corrisponde a Data Sharing Read Property B, per cui fornisce un servizio.

Dall'altra parte il dispositivo che deve integrare la temperatura è il Client, il quale deve essere in grado di avviare la richiesta di servizio *ReadProperty* e di accettare la risposta al suo arrivo. Il BIBB per questo tipo di funzionalità è chiamato "DS RP-A", che significa Data Sharing Read Property A, specificata per il colui che richiede un servizio. Per avere un'interazione come questa, sono definiti due BIBB, una per la chiamata ed uno per risposta

Seguendo sempre lo stesso concetto, può capitare che lo stesso client abbia bisogno di accendere una luce, la quale è controllata fisicamente da un altro dispositivo BACnet. In questo esempio, il dispositivo che controlla la luce è anche un server BACnet. Piuttosto che chiedere informazioni, il nostro client vuole ora richiedere l'esecuzione di qualche azione e cioè, accendere una luce. D'altra parte, il client vuole sapere, come risposta, se la richiesta è riuscita o fallita. In BACnet questo genere di richiesta è di solito eseguito scrivendo o modificando il valore di una proprietà di un oggetto che rappresenta la luce, ad esempio una proprietà *Present_Value* di un oggetto Binary Output. Il client accende la luce usando un servizio *WriteProperty* per modificare il valore *Present_Value* su "ON". In questo caso i BIBB DS-WP-Ae DS-WP-B sono richieste rispettivamente il primo dal Cliente e il secondo dal Server.

3.6 Certificazione BTL

I prodotti BTL Certified possono essere identificati dal marchio BTL (fig. 2.23) posto fisicamente nel prodotto o nel database online BTL Listing of Tested Products²⁴. Per ottenere la certificazione BTL, i prodotti devono essere testati con successo da un'organizzazione riconosciuta (RBTO) e indipendente e sottoposti a certificazione formale. I laboratori di prova BACnet gestiscono il programma di certificazione e stabiliscono i requisiti di prova²⁵.



Figura 2.23: Marchio BTL. (Fonte: <https://btl.org/>)

²⁴ Elenco consultabile nel sito web di certificazione dei prodotti BACnet. (<https://www.bacnetinternational.net/btl/>)

²⁵ Organizzazioni riconosciute, test di prova, programmi di certificazione e documentazione specifica consultabili al <https://btl.org/btl-certification-program/>

La certificazione BACnet Testing Laboratories (BTL), di cui si riposta un esempio in figura 2.24, dona ai fruitori la garanzia che i prodotti BACnet siano testati in modo indipendente e in conformità con il pacchetto di test accettato dal settore. Questo aiuta a ridurre al minimo costi e rischi associati al sistema.



Figura 2.24: Esempio di certificazione BTL per un controller B-AAC.(BACnet International, 2021)

Il marchio BTL è un segno distintivo e, ad oggi, rappresenta un alto livello di qualità e conformità fondato su test indipendenti e meticolosi. I test sono progettati in modo tale da convalidare la corretta implementazione di un prodotto del set specificato di funzionalità BACnet. Gli scrupolosi test associati all'acquisizione del diritto di utilizzo del marchio BTL sono

un autorevole espediente per scorgere e rimuovere gli errori di implementazione prima che un prodotto raggiunga il mercato. (BACnet International, 2022)

Il completamento con successo del processo di certificazione BTL comporta quanto segue:

- La possibilità di apporre il marchio BTL su un proprio prodotto e sul materiale promozionale ad esso associato, il tutto in conformità al documento sulla politica di utilizzo del marchio BTL.
- L'inclusione del prodotto nell'elenco ufficiale BTL del database dei prodotti testati, insieme al suo PICS e alle informazioni concernenti il rapporto di prova.
- La creazione di un certificato di conformità BTL per il prodotto, il quale viene incluso nel sito Web dell'elenco BTL.

3.7 Architettura BACnet

BACnet ha un'architettura di protocollo a strati basata su una versione compressa dell'Open Systems Interconnection (OSI) - Basic Reference Model. I livelli 1, 2, 3 e 7 del modello OSI, e dunque i rispettivi livelli Fisico, Collegamento Dati, Rete, e Applicazione vengono utilizzati come mostrato nella Figura 2.25.

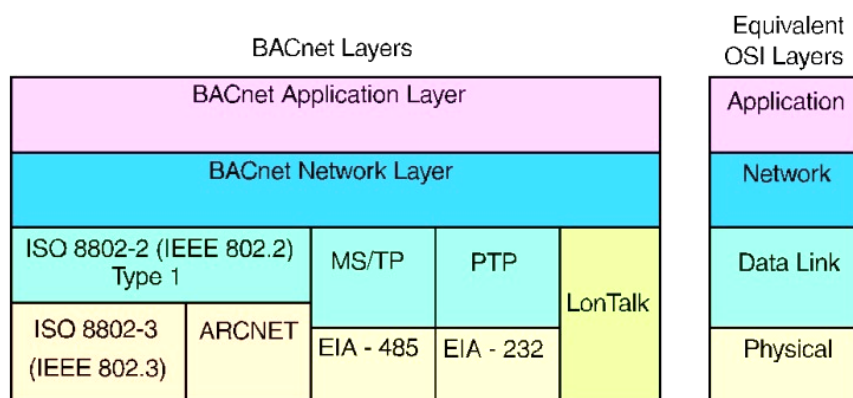


Figura 2.25: Lo stack di BACnet e gli strati equivalenti dello stack ISO-OSI. (Bushby T, 1997)

I protocolli a livello di applicazione e di rete comuni vengono utilizzati con una qualsiasi delle quattro opzioni per le tecnologie di rete locale (LAN) o un protocollo Point-To-Point adatto per le comunicazioni telefoniche dial-up. Il livello di rete fornisce un modo per interconnettere LAN diverse per formare una rete.

Così come succede anche in altri modelli di protocolli a pila, un ruolo considerevole è affidato al messaggio, il quale viene trasmesso da un dispositivo a un altro. In generale, si prevede una struttura ordinata in due parti principali:

- L'Intestazione, usata per conservare le informazioni, che spesso sono

- pletoriche, per garantire la correttezza e per gestire il routing;
- Il Body, usato per moderate quelle informazioni che devono, a tutti gli effetti, essere comunicate (PayLoad).

Lo strato Application, su quasi tutti i protocolli a pila, si impegna a saturare il Body ed eventualmente anche parti dell'Intestazione. Tutti gli altri strati sottostanti l'Application si occupano di inserire nell'intestazione un insieme di informazioni essenziali per gestire le esigenze dello strato, con lo scopo di garantire le funzionalità più utili agli strati superiori.

3.7.1 Application Layer

Questo layer include in sé sia lo schema astratto dell'informazione e dunque gli oggetti, sia i servizi utilizzati per il dialogo tra i dispositivi, i cui dettagli sono già stati visti in precedenza.

3.7.2 Network Layer

In questo strato si forniscono i procedimenti mediante i quali i messaggi possono essere trasportati da una rete BACnet a un'altra, a prescindere dalla tecnologia usata. Qui il soggetto principale è il router.

A causa della gestione di diversi media, ampiamente distinti tra loro, la segmentazione e il riassetto dei messaggi diventa una restrizione, poiché questi appaiono a carico del precedente strato di applicazione. Inoltre, BACnet istituisce un vero e proprio limite alla lunghezza dei messaggi che attraversano i router. In particolare, la massima lunghezza ammissibile è da comparare con la capacità massima di un data-link casuale che si trova lungo il path da mittente a destinatario, questa infatti non deve mai essere superata.

In questo layer è necessario adempiere ad alcuni compiti ben definiti dal protocollo e perciò si rivela necessario: gestire la propria tabella di routing, ricercare il path al fine di arrivare al dispositivo di destinazione, coordinare la presenza di un solo path attivo tra i due dispositivi che scambiano informazioni e indicare al dispositivo client quale sia il path richiesto nel caso in cui non sia localizzabile alcun ricevente.

Per di più, considerando che la rete BACnet è piuttosto eterogenea, sono definite anche le istruzioni per un router al fine di effettuare il tunneling di un pacchetto dati, nel momento in cui questo si trova ad attraversare reti IP oppure IPX²⁶.

3.7.3 Data Link Layer and Physical Layer

I messaggi BACnet, in linea di principio, sono trasmissibili su qualsiasi tipo di rete, ma, nonostante ciò, un numero circoscritto di tipi di rete è stato standardizzato per l'uso del

²⁶ Internetwork Packet Exchange è un insieme di protocolli di commutazione di pacchetto e sequenziamento di pacchetti progettati per funzionare in reti di piccole e grandi dimensioni. Nel modello OSI, è il protocollo di livello di rete nello stack di protocollo Internetwork Packet Exchange / Sequenced Packet Exchange (IPX / SPX). IPX fornisce connettività di supporto peer-to-peer.

protocollo, allo scopo di massimizzare le possibilità che due dispositivi comparabili aventi una qualsiasi funzionalità utilizzino lo stesso tipo. I tipi di rete scelti coprono una gamma di velocità e throughput.

Come mostrato nella figura 1.16, il protocollo BACnet specifica 4 soluzioni di tecnologia LAN: Ethernet, ARCNET, Master-Slave/Token-Passing (MS/TP), LonTalk e una tecnologia che prevede il protocollo Point-to-Point (PTP). Ogni tipo di rete locale, ad eccezione di MS/TP e PTP, è una soluzione standard.

La scelta di una determinata tecnica dipende da svariate ragioni come la scelta della velocità desiderata, la presenza di chip e di schede disponibili a implementare il protocollo, il grado di compatibilità con i sistemi adoperati nello specifico progetto, il costo che si è disposti ad affrontare, ecc. L'elasticità in questo determinato layer permette al progettista di prediligere la soluzione più adeguata ai particolari sfondi di applicazione che si presentano. Nei grandi sistemi, ad esempio, si fa abitualmente uso di soluzioni miste ordinate in strutture di gerarchia, per cui i controller di campo, risiedendo agli estremi, saranno collegati direttamente a reti a basso costo e a bassa velocità, e saranno supervisionati da altri controllori che sono interconnessi da reti ad alta velocità come la LAN.

L'Ethernet, o anche ISO 8802-3 secondo lo standard che lo caratterizza, rappresenta la soluzione con le prestazioni più alte. Offre velocità fino a 100 Mbps e si trova tra le tecnologie più ricorrenti. Usufruisce del protocollo CSMA/CD²⁷ (Carrier Sense Multiple Access with Collision Detection) per gestire il mezzo fisico in comune. Una peculiarità propria di questa tecnologia è che in occasione di collisione viene ridotta la larghezza della banda di trasmissione dati e per questo motivo non è possibile intuire, con sufficiente sicurezza, quanto tempo sia necessario ad un dispositivo affinché questo abbia la possibilità di consegnare con successo un messaggio al mittente.

ARCNET, che letteralmente significa Attached-Resource Computing architecture NETwork, è un protocollo per reti locali sviluppato nel 1976, e pubblicato l'anno successivo, dalla Datapoint Corporation. Attualmente è uno standard nazionale americano ATA/ANSI 878.1. Si presenta come un'alternativa a basso costo funzionante alla velocità di 2.5 Mbps e dichiara di essere la più antica LAN commercialmente fruibile, con quasi 3 milioni di nodi installati. ARCNET è un protocollo a token²⁸ circolante e per questo risulta deterministico, ovvero è possibile massimizzare un tempo certo per il quale il dispositivo dovrà aspettare al

²⁷ Si tratta di un protocollo di accesso multiplo, evoluzione del protocollo di livello MAC di CSMA, creato per la risoluzione dei problemi di trasmissione, come le collisioni, dovuti al CSMA puro, presenti in un certo dominio di collisione su reti locali cablate di tipo broadcast.

²⁸ Si tratta di una LAN nella quale i dispositivi sono connessi in topologia ad anello o a stella che, per evitare la collisione di dati trasmessi simultaneamente da due device della rete, usa un protocollo chiamato esattamente Token Ring. Il protocollo Token ring è il secondo protocollo di rete per frequenza di impiego, subito dopo il protocollo Ethernet.

fine di poter trasmettere un messaggio.

EIA-485 è uno standard definito per lo strato fisico ed è ordinariamente utilizzato nei sistemi di controllo degli edifici, soprattutto per quanto riguarda le reti di sensori per applicazioni specifiche. Il mezzo MS/TP rappresenta un mezzo consolidato e a basso costo per la comunicazione fino ad un massimo di 78,4 kb/s ed è utile con carichi di traffico come quelli che normalmente si verificherebbero con controller unitari o specifici per l'applicazione. In una rete di questo tipo si possono trovare uno o più nodi master equiparabili e dei nodi slave che non sono abilitati a trasmettere messaggi fino al momento in cui non viene richiesto loro da parte di un nodo master. L'accesso al mezzo viene concesso con la tecnica del Token-Ring.

LonTalk è un protocollo proprietario sviluppato dalla Echelon Corporation, già trattato al paragrafo 2.3 del Capitolo Secondo. I dispositivi basati sul protocollo LonTalk non sono, in generale, interoperabili perché il protocollo non impone loro nessuna particolare funzionalità applicativa. L'uso di tale protocollo in BACnet è arginato al solo trasporto di messaggi specifici, non utilizzando questo il concetto di tipo di variabile di rete standard LON (SNVT). L'interoperabilità viene migliorata usando lo stesso strato Application e Network come avviene per tutti gli altri dispositivi BACnet.

Il protocollo Point-To-Point, o anche PTP, viene utilizzato per l'accesso al mezzo condiviso attraverso la porta EIA-232²⁹, una sua applicazione comune potrebbe essere connettere a un modem un sistema domotico remoto. Il protocollo PTP non specifica come si stabilisce una connessione fisica, ma una volta che si è stabilita la connessione fisica, definisce in che modo una comunicazione BACnet è inizializzata, conservata e conclusa.

In particolare, il MS/TP indirizza la connettività su doppini intrecciati utilizzando la segnalazione EIA-485 mentre il PTP supporta le comunicazioni dial-up e altre applicazioni Point-To-Point utilizzando EIA-232 e, all'occasione, modem o altre apparecchiature di comunicazione dei dati.

Dalla creazione dello standard il desiderio era quello di poter utilizzare il protocollo Internet (IP) e, infatti, a partire dal 1999 è stato finalizzato "BACnet/IP". Lo stack di protocollo è stato esteso con un *BACnet Virtual Link Layer* (BVLL) il quale acconsente all'utilizzo dei protocolli sottostanti, ad esempio *User Datagram Protocol over IP*, come se fossero di per sé un livello di collegamento dati. Di conseguenza, le reti IP sono così supportate nativamente dal livello di rete BACnet esistente che consente ai dispositivi BACnet di comunicare utilizzando direttamente l'IP, piuttosto che tramite router di tunneling, come accadeva nello standard originale.

BACnet/IP e BACnet over Ethernet sono più adatti in caso di comunicazioni che comportano entità di dati più rilevanti. Anche ARCNET viene largamente utilizzato per la

²⁹ Gli Standard EIA-232 e EIA-485 sono dedicati ad un tipo di comunicazione seriale.

comunicazione da controller a controller, soprattutto negli Stati Uniti e in Asia a causa dell'avvento della versione a doppino intrecciato a basso costo e velocità relativamente elevata (156 kb/s). Il sistema Point-To-Point è ancora oggi usato saltuariamente ma è stato ampiamente sostituito da Internet, almeno per il traffico delle workstation.

Per quanto riguarda le comunicazioni wireless, è stata recentemente presentata una soluzione di bridging trasparente³⁰ basata su IEEE 802.11³¹, mentre BACnet su Ethernet wireless esiste da anni. Inoltre, BACnet fornisce anche supporto per connessioni intermittenti (come PTP) gestite da mezzi router.

Nella topologia di rete propria di BACnet, l'elemento che sta alla base di tutto è il segmento, il quale rappresenta un percorso fisico di cavi che si possono appaiare tramite ripetitori e bridge al fine di creare una rete. Queste ultime si collegano ai router formando una rete BACnet. Considerando due dispositivi qualsiasi all'interno di una rete, esiste un unico percorso praticabile che collega uno e l'altro dispositivo.

L'indirizzo di rete BACnet si compone di un numero di rete a soli 2 byte e un indirizzo locale che arriva ad un massimo di 255 byte. Tale indirizzo è caratteristico del mezzo del livello di collegamento, come ad esempio un indirizzo MAC per LAN o un indirizzo IP per BACnet/IP.

I router di BACnet che collegano le molteplici reti incanalano i pacchetti di informazioni a seconda dei numeri di rete e usano un sistema di gestione basato sull'autoapprendimento. Ciò significa che, dotati di numero di rete per le singole porte che gli appartengono, questi sono capaci di acquisire e comprendere la topologia di rete avvalendosi dei servizi di gestione della rete del router confacenti, come può essere il Who-Is-Router-To-Network.

Recentemente è stata aggiunta a BACnet la possibilità di allegati che descrivono l'uso di servizi Web e XML allo scopo di integrare i BACS con altri sistemi aziendali a livello di gestione (BACnet/WS). BACnet/WS sarà imparziale dal punto di vista del protocollo e pertanto coerentemente applicabile a sistemi non BACnet, malgrado nella bozza dello standard sia inclusa una mappatura completa tra i servizi BACnet e BACnet/WS.

³⁰ Tecnica per il collegamento di più segmenti di una rete Ethernet.

³¹ Definisce un insieme di standard di trasmissione per reti WLAN, con particolare riguardo al livello fisico e MAC del modello ISO/OSI, specificando sia l'interfaccia tra client e base station sia le specifiche tra client wireless.

Capitolo Terzo

LO SCAMBIO DI INFORMAZIONI TRA BACNET E IFC

Ad oggi, nonostante lo sviluppo delle varie tecnologie di progettazione, si vede raramente progettare i sistemi di automazione tramite l'utilizzo di strumenti BIM, o anche scambiare informazioni BAS con IFC in diverse fasi del progetto utilizzando strumenti BIM. L'attuale progettazione del sistema BAS, infatti, utilizza ancora il disegno bidimensionale basato su CAD o strumenti personalizzati.

A differenza degli altri sistemi di costruzione, infatti, l'automazione di rado è inclusa nel ciclo BIM di progettazione-costruzione e si fonde con il Facility Management solamente nella fase successiva. Così facendo si verificano complicazioni che portano alla perdita di informazioni, alla collaborazione inefficiente e alla costruzione sottoposta a errori a causa della complessità degli stessi sistemi di costruzione.

I sistemi di automazione e controllo dell'edificio, però, si possono dichiarare nel BIM fin dall'inizio del processo di progettazione di ogni singolo sottosistema, consentendo l'integrazione delle conoscenze e dell'esperienza di ciascun subappaltatore e sfruttando così a pieno il potenziale delle due tecnologie. Il modello conterrà allora dispositivi di input come sensori di temperatura, rilevatori di movimento, contatti magnetici e dispositivi di output come valvole o attuatori luminosi.

In tale occasione BACnet entra in gioco grazie ad una sua caratteristica fondamentale: lo standard permette la modellazione del sistema BMS mediante la selezione dei dispositivi in base al loro profilo standard e in base alle caratteristiche di interoperabilità da implementare. Pertanto, consente la progettazione senza la necessità di dichiarare specifici dispositivi reali. Il modello creato è coerente con i dispositivi che espongono il profilo desiderato a condizione che questi supportino i BIBBs e che soddisfino le specifiche dimensionali appropriate in termini di input/output (I/O), oggetti e tutti gli altri dettagli che possono essere utili.

Seguendo questa logica, in fase di costruzione, affidandosi al BIM, l'automazione non è più l'ultimo sistema ad essere realizzato perché è già stato realizzato almeno una volta nell'edificio virtuale, e quindi non risente più degli errori e delle correzioni apportate durante la messa a punto dell'edificio in altri sistemi. Tutti i potenziali problemi sono già stati risolti durante la fase di costruzione virtuale, condividendo l'esperienza del progettista BMS, di tutti i subappaltatori e dell'appaltatore generale dell'edificio.

BACnet, anche in questo caso, si dimostra prezioso: identifica univocamente gli oggetti sulla rete e consente l'interconnessione tra oggetti senza particolari opzioni nel livello fisico che incidono sulla comunicazione. In questo modo, un dispositivo indicato nel modello come dotato

di determinate caratteristiche può essere sostituito, dopo aver opportunamente corretto il modello, con uno o più reali dispositivi che implementano almeno lo stesso set di funzionalità.

Mentre l'edificio è ancora in costruzione, il modello può essere utilizzato per contenere tutti i dispositivi specifici informazioni, inclusi link a manuali, credenziali di accesso per la riprogrammazione, informazioni sull'interconnessione dei dispositivi, e non solo quelli specificamente appartenenti al BMS, ma anche quelli appartenenti alle relative interfacce.

Una volta che l'edificio viene consegnato al proprietario, il modello BIM rappresenta il primo strumento per aiutare i gestori delle strutture, poiché contiene in sé tutte le informazioni sulla costruzione e anche tutte le conoscenze relative al funzionamento e alla manutenzione delle strutture, compresi i sistemi di automazione.

Come si può dedurre, i modelli BIM sono in grado di contenere una pletora di informazioni, generando di conseguenza dei files enormi, ma un modello completamente popolato non è strettamente necessario. D'altra parte, anche le informazioni BAS non richiedono l'intero schema IFC per la rappresentazione dei dati, è sufficiente un sottoschema IFC che corrisponde al protocollo di comunicazione BAS per lo scambio di informazioni BAS tra varie parti, software e fasi del progetto.

Per risolvere il problema di proliferazione dei dati, buildingSMART International ha creato gli approcci IDM (*Information Delivery Manual*) e MVD (*Model View Definitions*) allo scopo di definire sottoinsiemi dello schema IFC per determinati ER (*Exchange Requirements* o requisiti di scambio informativo). Gli IDM sono normati dalla ISO 29481-1 e ISO 29481-2.

Nel 2012, la buildingSMART ha pubblicato un approccio integrato IDM/MVD denominato "*An Integrated Process for Delivering IFC Based Data Exchange*" il quale congloba il precedente approccio IDM e l'approccio MVD in uno.

Un IDM, letteralmente "manuale di consegna informativa", mira a specificare i processi che sussistono durante il ciclo di vita di una certa opera e la sequenza di informazioni necessarie a svilupparli. Si occupa, dunque, di standardizzare i processi di scambio BIM e di migliorare la qualità della comunicazione tra i vari specialisti coinvolti nel progetto.

Un MDV, letteralmente "definizione della vista di modello", è un livello di implementazione specifico di IFC che si occupa di descrivere e semplificare un determinato utilizzo o un flusso operativo specifico. Raffigura un filtro dello schema generale in risposta a indicati requisiti di scambio, i quali sono precedentemente identificati dall'IDM stesso. Così facendo, è possibile filtrare e ridimensionare i modelli secondo un dato modello per soddisfare processi aziendali specifici.

In via generale, gli MVD possono essere ricchi di informazioni oppure specifici per poche tipologie di elementi e dati associati. Questo perché lo schema IFC è progettato per adattarsi a molte configurazioni diverse, ad esempio, un muro può essere rappresentato: come una linea tra due punti, oppure come una geometria tridimensionale, con forme semplici o con dettagli

costruttivi esclusivi, associando dati riguardanti proprietà ingegneristiche, parti responsabili, pianificazione e informazioni sui costi. Tale pletora di informazioni non è però indispensabile a tutti gli attori coinvolti nei vari processi del progetto; dunque, allora un MVD diventa fondamentale per restringere l'ambito a seconda della necessità del destinatario delle informazioni per un flusso di lavoro specifico.

Consultando l'MVD database ufficiale degli di buildingSMART si ottiene una panoramica completa dei vari MVD, con specificato il loro status e lo schema IFC di riferimento³².

BuildingSMART ha pubblicato MVD ufficiali tra cui emerge COBie (Construction Operation Building Information Exchange), il quale è impiegato per lo scambio delle specifiche per l'acquisizione e la consegna delle informazioni necessarie a coloro che si occupano della gestione delle strutture. Nonostante COBie sia capace di acquisire le informazioni riguardanti il ciclo di vita necessarie ai facility manager, le entità di dati nella documentazione COBie MVD si sono concentrate in particolare sul dominio di gestione della costruzione.

Inoltre, COBie non pone alcuna attenzione sullo scambio di informazioni BAS che seguono il protocollo BACnet. Alcuni hanno suggerito che esso racchiuda in sé parametri universali per la gestione delle strutture e che non riesca a filtrare selettivamente i dati più rilevanti per i requisiti di funzionamento e manutenzione su misura. C'è una lacuna nella comprensione dei dati semantici necessari da scambiare per quanto riguarda la progettazione, la costruzione e il funzionamento del BAS seguendo uno standard internazionale come BACnet.

1.1 Riferimenti BACnet nello standard IFC

Il primo approccio allo schema IFC da parte di un principiante risulta sicuramente arduo: se a livello contenutistico si può comprendere in modo pressoché adeguato la struttura dei dati proposta, mancano delucidazioni chiare su molti aspetti grafici e visivi (diagrammi e colori, classi astratte e concrete ecc.). Si genera dunque, di conseguenza, una particolare confusione dovuta alla frammentarietà delle informazioni di correlazione tra diverse sezioni specifiche. In generale, sono presenti esemplificazioni dei concetti che si focalizzano ad un tipo di dati geometrici e spaziali rispetto che ad altri tipi di relazione tra oggetti, e la documentazione che fornisce buildingSMART aspira a chiarire puramente i dubbi riguardanti l'ambito informatico dello standard ed è dunque rivolto a implementatori dello schema stesso e di software di gestione dei dati del database IFC piuttosto che ad altre utenze del settore costruttivo che operano mediante gli applicativi stessi, quali i progettisti.

Queste ambiguità e le lacune informative interne alla specifica stessa e ai suoi strumenti principali di implementazione (IfcDoc, IDM e MVD), in relazione ai lunghi tempi richiesti per l'estensione dello schema al fine di sopperire a queste mancanze, risultano un ostacolo per lo sviluppo di una interoperabilità completa tra discipline e applicativi differenti. Focalizzando lo

³² Consultabile al <https://technical.buildingsmart.org/standards/ifc/mvd/mvd-database/>

sguardo al caso dello standard BACnet, tutt'ora in continua evoluzione e che vede la pubblicazione di suoi aggiornamenti ogni 2-4 anni in media (Tang et al., 2020), è chiaro che questo contribuisce alla difficoltà di portare a compimento una completa definizione delle informazioni sugli impianti di automazione e controllo all'interno del modello di dati IFC.

Durante il processo di mappatura tra specifiche di standard per impianti BACS e proprietà e attributi contenuti nello schema IFC si intercorre in numerose mancanze informative nella struttura dei dati descritte e supportate da quest'ultimo; tali lacune riguardano ad esempio entità e parametri previsti negli standard di progettazione esistenti, quali le caratteristiche prestazionali energetiche e acustiche. Tali mancanze portano l'utente alla creazione ad hoc di concept e proprietà user-defined, all'uso e al riferimento di entità generiche come `IfcBuildingElementProxy` o all'inserimento di `PredefinedType` personalizzati per la classificazione degli oggetti di un progetto. Questi procedimenti manuali spesso comportano una perdita di informazioni, ridondanze o differente gestione degli stessi dati tra diverse parti interessate che complicano e spesso prolungano inefficientemente il processo di interoperabilità.

Con il passaggio dalla versione IFC2x3 alla release IFC4, si sono inoltre perse alcune indicazioni predefinite da `buildingSMART`, riportate direttamente nello schema, per una possibile mappatura tra oggetti di protocolli BACnet, OPC, ecc. e entità IFC. Tali indicazioni, non più fornite nelle nuove release, sono di seguito riportate in figura 3.1.

Mapping to Building Automation Standards

The `IfcBuildingControlsDomain` schema does not specify building automation protocols, but may be mapped to standard protocols or vendor implementations for commissioning and operations interoperability. Several mappings are illustrated:

- **BACnet:** www.bacnet.org
 - **Accumulator:** `IfcController` having `PredefinedType=FLOATING`, `FloatingType=ACCUMULATOR`, and aggregated inside the `IfcController Device`.
 - **Analog Input:** `IfcController` having `PredefinedType=VARIABLE`, `ValueType=ANALOGINPUT`, and aggregated inside the `IfcController Device`.
 - **Analog Output:** `IfcController` having `PredefinedType=VARIABLE`, `ValueType=ANALOGOUTPUT`, and aggregated inside the `IfcController Device`.
 - **Analog Value:** `IfcController` having `PredefinedType=VARIABLE`, `ValueType=ANALOGVALUE`, and aggregated inside the `IfcController Device`.
 - **Averaging:** `IfcController` having `PredefinedType=FLOATING`, `FloatingType=AVERAGING`, and aggregated inside the `IfcController Device`.
 - **Binary Input:** `IfcController` having `PredefinedType=VARIABLE`, `ValueType=BINARYINPUT`, and aggregated inside the `IfcController Device`.
 - **Binary Output:** `IfcController` having `PredefinedType=VARIABLE`, `ValueType=BINARYOUTPUT`, and aggregated inside the `IfcController Device`.
 - **Binary Value:** `IfcController` having `PredefinedType=VARIABLE`, `ValueType=BINARYVALUE`, and aggregated inside the `IfcController Device`.
 - **Calendar:** `IfcController` having `PredefinedType=TWOPOSITION`, `TwoPositionType=CALENDAR`, and assigned to `IfcWorkCalendar` indicating times.
 - **Command:** `IfcProcedure` having `PredefinedType=CALIBRATION`, and assigned to `IfcController` having `PredefinedType=VALUE`.
 - **Device:** `IfcController` having `PredefinedType=PROGRAMMABLE`.
 - **Event Enrollment:** `IfcEvent` assigned to `IfcController`.
 - **File:** `IfcProxy` having associated file using `IfcRelAssociatesDocument` and `IfcDocumentInformation`.
 - **Group:** `IfcDistributionSystem` having `PredefinedType=CONTROL`.
 - **Life Safety Point:** `IfcSensor` having `PredefinedType=FIRESSENSOR`.
 - **Life Safety Zone:** `IfcZone` having `PredefinedType=FIRESAFETY`.
 - **Loop:** `IfcController` having `PredefinedType=PROPORTIONAL`, and aggregated inside the `IfcController Device`.
 - **Multi-state Input:** `IfcController` having `PredefinedType=VARIABLE`, `ValueType=MULTISTATEINPUT`, and aggregated inside the `IfcController Device`.
 - **Multi-state Output:** `IfcController` having `PredefinedType=VARIABLE`, `ValueType=MULTISTATEOUTPUT`, and aggregated inside the `IfcController Device`.
 - **Multi-state Value:** `IfcController` having `PredefinedType=VARIABLE`, `ValueType=MULTISTATEVALUE`, and aggregated inside the `IfcController Device`.
 - **Notification Class:** `IfcProcedure` assigned to `IfcController`.
 - **Program:** `IfcTask` assigned to `IfcController`.
 - **Pulse Converter:** `IfcController` having `PredefinedType=FLOATING`, `FloatingType=PULSECONVERTER`, and aggregated inside the `IfcController Device`.
 - **Schedule:** `IfcPerformanceHistory` having `PredefinedType=OPERATION`, and `IfcTimeSeries` having `DataOrigin=SIMULATED`.
 - **Trend Log:** `IfcPerformanceHistory` having `PredefinedType=OPERATION`, and `IfcTimeSeries` having `DataOrigin=MEASURED`.
 - **OPC:** www.opcfoundation.org
 - **Server:** `IfcController` having `PredefinedType=PROGRAMMABLE` and associated `IfcClassification` indicating address space.
 - **Item:** `IfcController` having `PredefinedType=VARIABLE`, having associated `IfcClassificationReference` indicating item address, and aggregated inside the `IfcController Server`.
 - **Event Category:** `IfcEventType` having associated `IfcClassification` indicating conditions and subconditions, and `IfcPropertySet` indicating custom properties.
 - **Event Source:** `IfcEvent` defined by `IfcEventType` Category, having associated `IfcClassificationReference` indicating condition, and assigned to an `IfcController Item`.
- HISTORY: New schema in IFC 2x2.

Figura 3.1: Indicazioni dalla versione IFC2x3 per una mappatura tra entità dello standard IFC e tipi di oggetti definiti negli standard per l'automazione degli edifici. (BuildingSMART International, 2020)

1.1.1 `IfcBuildingControlsDomain`

Nella pagina ufficiale dello schema in versione IFC4ADD2, dunque quello di comune utilizzo ad oggi, si trovano informazioni minime e frammentarie riguardo l'automazione. In

particolare, al capitolo 7, entrando nell'*IfcBuildingControlsDomain*, si vede come esso sia destinato alla definizione dei concetti di building automation, controllo, strumentazione e allarme. Tale sezione supporta idee che includono tipi e occorrenze di:

- Attuatore;
- Allarme;
- Controllore;
- Sensore;
- Strumenti di flusso;
- Elementi di controllo unitario.

Continuando la lettura della pagina, però, si legge come questo capitolo non si occupi di specificare i protocolli di automazione degli edifici, ma piuttosto può essere utilizzato per mappare di protocolli standard o implementazioni del fornitore per la messa in servizio e l'interoperabilità delle operazioni, si veda la figura 3.2.

The *IfcBuildingControlsDomain* schema does not specify building automation protocols, but may be mapped to standard protocols or vendor implementations for commissioning and operations interoperability. Common applicable entities are described as follows:

- *IfcPerformanceHistory*: Captures realtime device data in the form of property sets.
- *IfcPropertySet*: Captures a set of realtime device data, either using predefined data structures or custom information.
- *IfcPropertySetTemplate*: Captures metadata about custom properties, such as names, descriptions, data types, units, and ranges.
- *IfcRelAssociatesClassification*: Associates addresses of devices and control points to uniquely identify within a control system.
- *IfcController*: Captures non-realtime occurrence information for hardware or software-based arbitrary analog and digital data.
- *IfcAlarm*: Captures non-realtime occurrence information for hardware or software-based alarm sources.
- *IfcEvent*: Captures alarm event handling information.
- *IfcProcedure*: Captures device procedures that may be run arbitrarily or in response to events.
- *IfcTask*: Captures device programs that may be scheduled at various times.

Figura 3.2: Specificazioni dell'*IfcBuildingControlsDomain* e descrizione delle entità riguardanti l'automazione nella versione IFC4ADD2 al capitolo 7.2. (BuildingSMART International, 2022)

Le entità comunemente utilizzate, come si può vedere, sono così descritte:

- *IfcPerformanceHistory*: acquisisce i dati del dispositivo in tempo reale sotto forma di insiemi di proprietà.
- *IfcPropertySet*: acquisisce un set di dati del dispositivo in tempo reale, utilizzando strutture di dati predefinite o informazioni personalizzate.
- *IfcPropertySetTemplate*: acquisisce i metadati sulle proprietà personalizzate, ad esempio nomi, descrizioni, tipi di dati, unità e intervalli.
- *IfcRelAssociatesClassification*: associa gli indirizzi dei dispositivi e dei punti di controllo per identificarli in modo univoco all'interno di un sistema di controllo.
- *IfcController*: acquisisce informazioni sugli eventi non in tempo reale per dati analogici e digitali arbitrari basati su hardware o software.
- *IfcAlarm*: acquisisce informazioni sull'occorrenza non in tempo reale per sorgenti di allarme basate su hardware o software.
- *IfcEvent*: cattura le informazioni sulla gestione degli eventi di allarme.
- *IfcProcedure*: acquisisce le procedure del dispositivo che possono essere eseguite arbitrariamente o in risposta a eventi.

- IfcTask: acquisisce i programmi del dispositivo che possono essere programmati in momenti diversi.

1.1.2 IfcDistributionControlElement

Continuando con la mappatura degli elementi, un altro rimando ai concetti della building automation lo si può trovare al capitolo 6, nell'*IfcSharedBldgServiceElements*. Navigando in esso si legge come tale pagina definisca i concetti di base richiesti per l'interoperabilità principalmente tra le estensioni del dominio che riguarda la Building Service, e in particolare IfcHvacDomain, IfcPlumbingFireProtectionDomain, IfcElectricalDomain e IfcBuildingControlsDomain, prima analizzato. Questo schema include concetti quali definizioni di tipo e occorrenza di base per sistemi di flusso e distribuzione, e insiemi di proprietà per scenari comuni di servizi di costruzione come proprietà di flusso dei fluidi, proprietà elettriche e proprietà termiche dello spazio.

Se si consultano le entità di questo capitolo, tra di esse, alla voce 6.2.3.4, si trova l'*IfcDistributionControlElement* il quale chiarisce gli elementi di occorrenza di un sistema di controllo dell'automazione degli edifici. Questi sono tipicamente impiegati per controllare gli elementi del sistema di distribuzione per mantenere variabili come temperatura, umidità, pressione, flusso, potenza o livelli di illuminazione, attraverso la modulazione, la messa in scena o la sequenza di dispositivi meccanici o elettrici. (BuildingSMART International, 2020)

Le tre categorie funzionali generali degli elementi di controllo sono le seguenti:

- Impartire il controllo sugli elementi di controllo del flusso (IfcFlowController) in un sistema di distribuzione come serrande, valvole o relè, in genere attraverso l'uso dell'azionamento (IfcActuator).
- Elementi sensibili (IfcSensor) che misurano i cambiamenti nella variabile controllata come temperatura, umidità, pressione o flusso.
- Controller (IfcController) tipicamente classificati in base all'azione di controllo che cercano di eseguire e generalmente responsabili di prendere decisioni sugli elementi sotto controllo.

1.1.3 IfcClassification

Scorrendo la pagina dedicata all'entità precedentemente nominata si arriva alla voce *Object Classification*, dove appare la seguente tabella:

Tabella 8: Classificazioni di indirizzo per IfcDistributionControlElement. (BuildingSMART International, 2020)

Name	Source	Tokens	Description
BACnet	ASHRAE	{(D:0..65535);(D:0..65535)}	32-bit decimal BACnetObjectIdentifier indicating type ID and instance ID (e.g. '12.15' for Digital Input #15).
IPv4	IETF	{(D:0..255);(D:0..255);(D:0..255);(D:0..255)}	32-bit decimal address for an IPv4 network (e.g. '192.168.1.1').
IPv6	IETF	{(X:0000..FFFF);(X:0000..FFFF);(X:0000..FFFF);(X:0000..FFFF);(X:0000..FFFF);(X:0000..FFFF);(X:0000..FFFF);(X:0000..FFFF)}	128-bit hexadecimal address for an IPv6 network.
MAC	IETF	{(X:00..FF);(X:00..FF);(X:00..FF);(X:00..FF);(X:00..FF);(X:00..FF)}	48-bit hexadecimal form of MAC address.
OPC	OPC Foundation	{(S);(I)}	Hierarchical ItemID in alphanumeric form (i.e. 'B204.Tank2.Temperature')
Insteon	SmartLabs	{(X:00..FF);(X:00..FF);(X:00..FF)}	24-bit hexadecimal instance address.
LonTalk	ISO/IEC	{(X:00..FF);(X:00..FF);(X:00..FF);(X:00..FF);(X:00..FF);(X:00..FF);(X:00..FF);(X:00..FF)}	48-bit hexadecimal neuron ID.

Si vedono in essa nominati diversi indirizzi, tra i quali si notano BACnet, Insteon e LonTalk come indirizzi per la comunicazione dei sistemi di automazioni, ai quali sono associati gli enti, le associazioni e i consorzi di riferimento per la corretta definizione di norme tecniche sotto la voce "Source". Seguono poi i Token, termine con cui si indica un segnalatore univoco catalogato in un registro condiviso, in gergo *blockchain*, con funzione di rappresentare un oggetto digitale, di certificare la proprietà di un bene o di consentire l'accesso a un servizio. Infine, si trova la descrizione, che nel caso di BACnet corrisponde all'*Object Identifier*, descritto al paragrafo 3.2 del Capitolo Secondo.

Cliccando alla voce "Classification" al di sopra della tabella appena esaminata, si viene rimandati alla pagina dedicata al paragrafo 4.1 del capitolo 4, dedicato ai concetti fondamentali e ai presupposti. Qui si può capire come il concetto di classificazione descriva in che modo oggetti e tipi di oggetti possano essere ulteriormente descritti associando riferimenti a fonti esterne di informazioni. La fonte delle informazioni può essere:

- un sistema di classificazione;
- un server di dizionario;
- qualsiasi catalogo esterno che classifichi ulteriormente l'oggetto;
- qualsiasi servizio che combini le caratteristiche di cui sopra.

È possibile selezionare un singolo elemento all'interno della fonte esterna di informazioni e applicare, di conseguenza, il significato intrinseco dell'elemento all'oggetto o alla proprietà.

Gli attributi principali da fornire per un'associazione di classificazione sono:

- *Identification*: contiene la chiave prevista per un riferimento specifico agli elementi di classificazione (o tabelle);
- *Name*: consente una designazione interpretabile dall'uomo di una notazione di classificazione.

Il diagramma raffigurato di seguito (figura 3.3) mostra le classi e le relazioni generiche utilizzate durante l'applicazione di questo concetto. Inoltre, concetti simili possono avere particolare importanza per pratiche e scenari di settore comuni o standardizzati. Per questi scenari di utilizzo specifici, le tabelle seguenti mostrano un elenco consigliato di modelli di utilizzo generali che gli utenti possono adottare.

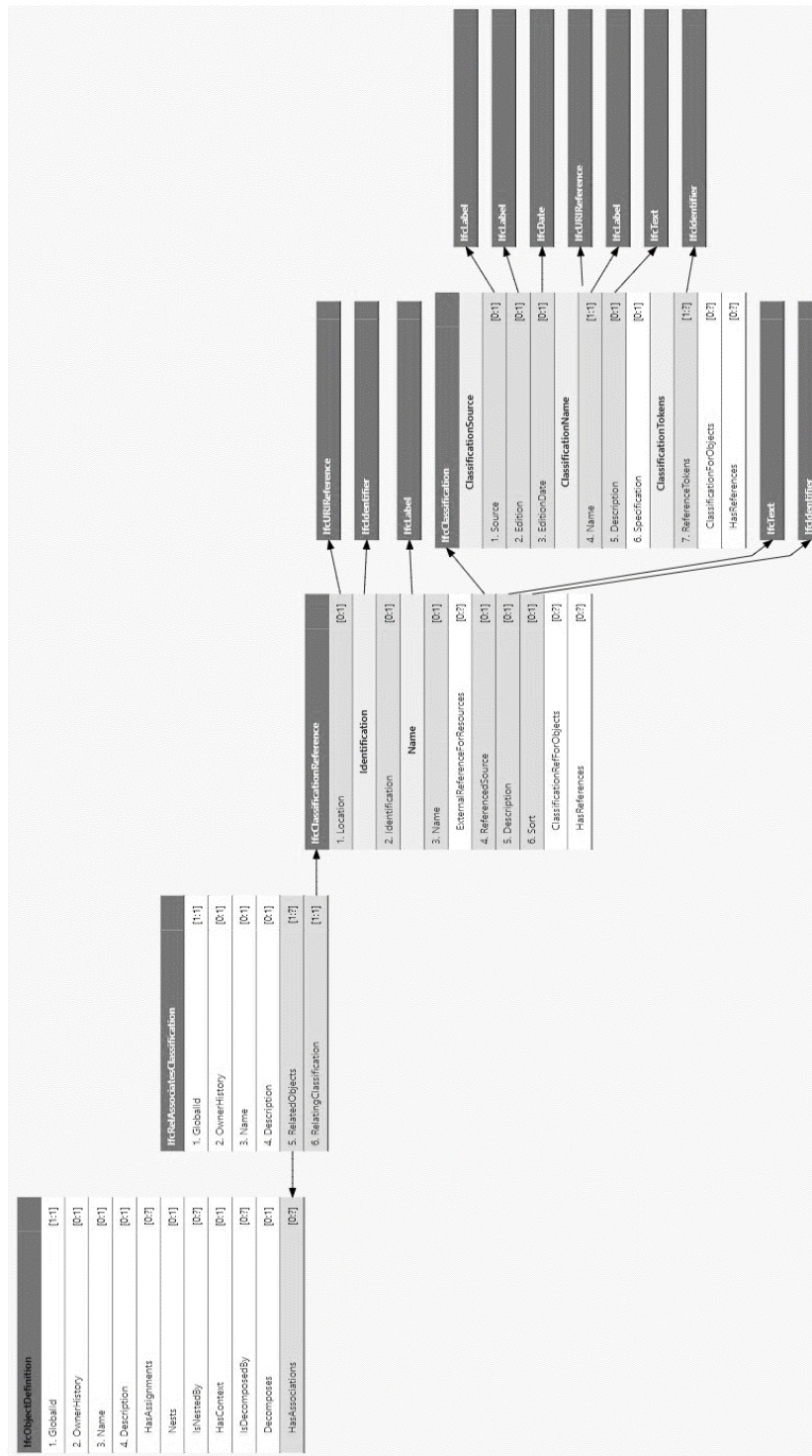


Figura 3.3: Diagramma di istanza per la classificazione di un IfcObjectDefinition. (BuildingSMART International, 2020)

Saranno esaminate, nei paragrafi che seguono, le voci principali delle entità presenti nello schema visualizzato, alle quali si può accedere direttamente dallo schema stesso cliccando sopra la voce indagata, al fine di consentire una corretta comprensione dei concetti.

1.1.4 *IfcObjectDefinition*

Al capitolo 5, cliccando al 5.1.3.7, si trova la definizione di *IfcObjectDefinition*, il quale è la generalizzazione di qualsiasi oggetto o processo trattato semanticamente, sia che si tratti di un tipo o di un'occorrenza. Le definizioni degli oggetti possono essere denominate utilizzando l'attributo *Name* ereditato, il quale dovrebbe essere un'etichetta riconoscibile dall'utente per la ricorrenza dell'oggetto. Ulteriori spiegazioni sull'oggetto possono essere fornite utilizzando l'attributo *Description* ereditato. Un contesto è un tipo specifico di definizione dell'oggetto in quanto fornisce il contesto del progetto o della libreria in cui vengono definiti i tipi di oggetto e le occorrenze dell'oggetto.

Gli oggetti sono informazioni indipendenti che potrebbero contenere o fare riferimento ad altre informazioni. Esistono quattro tipi essenziali di relazioni in cui possono essere coinvolte definizioni di oggetti (dai loro sottotipi istanziabili):

- Assegnazione di altri oggetti: si tratta di una relazione di assegnazione (*IfcRelAssigns*) che fa riferimento ad altri tipi di oggetti e crea un'associazione bidirezionale. La semantica dell'assegnamento è stabilita a livello dei sottotipi della relazione generale *IfcRelAssigns*. Non vi è alcuna dipendenza implicita a priori dall'assegnazione.
- Associazione a risorse esterne: si tratta di una relazione di associazione (*IfcRelAssociates*) che fa riferimento a fonti esterne di informazioni, in particolare una classificazione o un documento, e crea un'associazione unidirezionale. Non vi è alcuna dipendenza implicita dall'associazione.
- Aggregazione di altri oggetti: si tratta di una relazione di aggregazione (*IfcRelAggregates*) che stabilisce una relazione spaziale intera/parte non ordinata e crea una relazione bidirezionale. È stata stabilita una dipendenza implicita.
- Annidamento di altri oggetti: si tratta di una relazione di annidamento (*IfcRelNests*) che stabilisce una relazione intero/parte ordinata e non spaziale e crea una relazione bidirezionale. È stata stabilita una dipendenza implicita.
- Dichiarazione all'interno di un contesto: si tratta di una relazione (*IfcRelDeclares*) della definizione di oggetto superiore all'interno dell'albero di definizione dell'oggetto, come ad esempio l'oggetto di riepilogo all'interno di un albero di nidificazione degli oggetti, al contesto, come un progetto o una libreria di progetto. Applica le unità, il contesto di rappresentazione e altre informazioni di contesto a questa definizione di oggetto e a tutte quelle dipendenti.

1.1.5 *IfcRelAssociatesClassification*

Al paragrafo 5.1.3.31 si trova la definizione di *IfcRelAssociatesClassification*, relazione oggettivata che gestisce l'assegnazione di un elemento di classificazione, gli elementi della selezione *IfcClassificationSelect*, alle occorrenze degli oggetti, dunque i sottotipi di *IfcObject*, oppure ai tipi di oggetto, cioè i sottotipi di *IfcTypeObject*. Tale relazione viene utilizzata per assegnare un elemento di classificazione o un sistema di classificazione stesso agli oggetti. A seconda del tipo di *RelatingClassification* si tratta di:

- un riferimento a un elemento di classificazione all'interno di un sistema di classificazione esterno;
- un riferimento al sistema di classificazione stesso.

Si fa notare che il riferimento a un elemento di classificazione include un collegamento al sistema di classificazione all'interno del quale l'elemento è dichiarato e assegna il significato dell'elemento di classificazione all'oggetto, sia questo di occorrenza o di tipo. Il riferimento al sistema di classificazione fornisce l'informazione che l'oggetto è disciplinato dal sistema di classificazione, ma non è stata ancora effettuata alcuna assegnazione di un particolare elemento.

L'attributo ereditato *RelatedObjects* definisce gli oggetti o i tipi di oggetto a cui viene applicata la classificazione; l'attributo *RelatingClassification* è il riferimento a una classificazione, applicata all'oggetto o agli oggetti. Si conclude che una singola *RelatingClassification* può quindi essere applicata a uno o più oggetti.

1.1.6 *IfcClassificationReference*

Passando ora al capitolo 8 dello standard IFC, al paragrafo 8.6.3.2 si può leggere la definizione di *IfcClassificationReference*, il quale è intendibile come riferimento in un sistema di classificazione o in una fonte (per la quale si rimanda a *IfcClassification*) per una specifica chiave di classificazione o notazione. Gli attributi ereditati hanno il seguente significato:

- *Identification*: contiene la chiave prevista per un riferimento specifico agli elementi di classificazione o alle tabelle.
- *Name*: consente una designazione interpretabile dall'uomo di una notazione di classificazione.
- *Position*: facoltativamente contiene un collegamento URI diretto nel sistema di classificazione, oppure anche alla fonte, per collegare ipertestualmente la chiave di classificazione.

IfcClassificationReference può essere assegnato direttamente ad una *IfcClassification*, nei casi in cui non si riveli necessario includere alcuna gerarchia di classificazione, oppure se si fa riferimento alla notazione di classificazione principale, se la gerarchia di classificazione completa è inclusa nel set di dati. La prima è denominata "classificazione leggera" mentre la seconda "classificazione completa". L'attributo *ReferencedSource* contiene quindi le seguenti

informazioni (scelte da *IfcClassificationReferenceSelect*):

- essere di tipo *IfcClassification*: riferimento diretto al sistema di classificazione (con informazioni riguardo i metadati);
- essendo di tipo *IfcClassificationReference*: riferimento alla notazione di classificazione principale all'interno della gerarchia di classificazione.

Si vede dunque come il dialogo tra BuildingSMART e le tecnologie di automazione sia ancora in fase di sviluppo, lo standard IFC infatti presenta al suo interno tutte le informazioni necessarie all'implementazione di sistemi di automazione, ma manca totalmente di riferimenti aggiornati e ufficiali a standard per la comunicazione riconosciuti come lo è, appunto, BACnet. L'interoperabilità è un argomento ancora in via di sviluppo e, per ora, non c'è alcuna novità a riguardo nella nuova versione, ancora fase di approvazione, di IFC4.3.1.x.

1.2 Gli oggetti BACnet nello standard IFC

Dati gli oggetti standard BACnet analizzati al 3.2 del Capitolo Secondo, vengono di seguito studiate le analogie con gli stessi presenti nello standard IFC.

Per far ciò, si sono indagate le papabili classi Ifc di appartenenza dei vari BACnet Object nella versione IFC4ADD2, con i loro TypeEnum e, quando presenti, i Property Set al fine di descriverli il più approfonditamente possibile. Per permettere un confronto diretto nel sito di buildingSMART sono specificati anche i riferimenti numerici di ognuno. (Tabella 9)

Di seguito vengono descritte le singole voci presenti in tabella (BuildingSMART International, 2020).

Tabella 9: Mappatura degli oggetti BACnet nello standard IFC.

BACnet Object	IfcEntities	IfcTypeEnum	IfcPropertySet
Accumulator	7.2.3.5 IfcController	7.2.2.3 FLOATING	7.2.4.12 ACCUMULATOR
Alert Enrollment	5.3.3.3 IfcProcedure	5.3.2.3. DIAGNOSTIC	
Analog Input	7.2.3.5 IfcController	7.2.2.3 FLOATING	7.2.4.12 INPUT
Analog Output	7.2.3.5 IfcController	7.2.2.3 FLOATING	7.2.4.12 OUTPUT
Analog Value	7.2.3.5 IfcController	7.2.2.3 FLOATING	7.2.4.12 VARIABLE
Averaging	7.2.3.5 IfcController	7.2.2.3 FLOATING	7.2.4.12 AVERAGE

			7.2.4.12 MAXIMUM
			7.2.4.12 MINIMUM
Binary Input	7.2.3.5 IfcController	7.2.2.3 TWOPOSITION	7.2.4.16 INPUT
Binary Output	7.2.3.5 IfcController	7.2.2.3 TWOPOSITION	7.2.4.16 OUTPUT
Binary Value	7.2.3.5 IfcController	7.2.2.3 TWOPOSITION	7.2.4.16 VARIABLE
Calendar	7.2.3.5 IfcController	7.2.2.3 TWOPOSITION	7.2.4.16 CALENDAR
	5.3.3.8 IfcWorkCalendar	5.3.2.6 USERDEFINED	
Command	5.3.3.3 IfcProcedure	5.3.2.3 SHUTDOWN	
		5.3.2.3 STARTUP	
Date Value	8.5.2.2 IfcDate		
DateTime Value	8.5.2.3 IfcDateTime		
Device	7.2.3.5 IfcController	7.2.2.3 PROGRAMMABLE	7.2.4.14 PRIMARY
			7.2.4.14 SECONDARY
Event Enrollment	5.3.3.1 IfcEvent	5.3.2.1 EVENTMESSAGE	
Event Log	8.5.3.1 IfcEventTime		
Global Group	6.2.3.9 IfcDistributionSystem	6.2.2.3 COMMUNICATION	
		6.2.2.3 CONTROL	
Group	6.2.3.9 IfcDistributionSystem	6.2.2.3 COMMUNICATION	
		6.2.2.3 CONTROL	
Integer Value	8.16.3.12 IfcDescriptiveMeasure		
Large Analog Value	8.16.3.12 IfcDescriptiveMeasure		

Life Safety Point	7.2.3.9 IfcSensor	7.2.2.5 FIRESENSOR	
Life Safety Zone	5.4.3.70 IfcSpatialZone	5.4.2.23 FIRESAFETY	
Lighting Output	8.14.3.2 IfcLightIntensityDistribution		
Loop	7.2.3.5 IfcController	7.2.2.3 PROPORTIONAL	7.2.4.15 PROPORTIONAL
			7.2.4.15 PROPORTIONALINTEGRAL
			7.2.4.15 PROPORTIONALINTEGRALDERIVATIVE
Multi-state Input	7.2.3.5 IfcController	7.2.2.3 MULTIPOSITION	7.2.4.13 INPUT
Multi-state Output	7.2.3.5 IfcController	7.2.2.3 MULTIPOSITION	7.2.4.13 OUTPUT
Multi-state Value	7.2.3.5 IfcController	7.2.2.3 MULTIPOSITION	7.2.4.13 VARIABLE
Network Security	7.4.3.11 IfcCommunicationsAppliance	7.4.2.6. NETWORKAPPLIANCE	
Notification Class	5.3.3.3 IfcProcedure	5.3.2.3. ADVICE_NOTE	
Notification Forwarder	5.3.3.3 IfcProcedure	5.3.2.3. ADVICE_NOTE	
Positive Integer Value	8.11.2.12 IfcCountMeasure		
Program	5.3.3.6 IfcTask	5.3.2.5. CALIBRATION	
Pulse Converter	7.2.3.5 IfcController	7.2.2.3 FLOATING	7.2.4.12 PULSECONVERTER
Schedule	5.2.3.1 IfcPerformanceHistory	5.2.3.1.3 OPERATION	
		8.5.2.1 SIMULATED	
	8.5.3.8 IfcSchedulingTime	8.5.2.1 SIMULATED	
Structured View	6.2.3.9 IfcDistributionSystem	6.2.2.3 COMMUNICATION	
		6.2.2.3 CONTROL	

Time Value	8.5.2.10 IfcTime		
Trend Log	5.2.3.1 IfcPerformanceHistory	5.2.3.1.3 OPERATION	

1.2.1 IfcController

Si può definire un controller come un dispositivo che monitora gli ingressi e le uscite all'interno di un sistema di automazione. Esso può essere fisico, cioè avere un posizionamento all'interno di una struttura spaziale, o logico, ossia dotato di un'interfaccia software o aggregato all'interno di un controller fisico programmabile.

1.2.1.1 IfcControllerTypeEnum

- FLOATING: L'uscita aumenta o diminuisce ad un tasso costante o accelerato.
- PROGRAMMABLE: L'uscita è programmabile come Discrete Digital Control (DDC).
- PROPORTIONAL: L'uscita è proporzionale all'errore di controllo e opzionalmente all'integrale e alla derivata del tempo.
- MULTIPOSITION: L'uscita è un valore discreto, può essere uno di tre o più valori.
- TWOPOSITION: L'uscita può essere attivata (on) o disattivata (off).
- USERDEFINED: Tipo definito dall'utente.
- NOTDEFINED: Tipo non definito.

1.2.1.2 Pset_ControllerTypeFloating

- CONSTANT: Nessun ingresso e SignalOffset viene scritto nel valore di uscita.
- MODIFIER: il singolo ingresso analogico viene letto, aggiunto a SignalOffset, moltiplicato per SignalFactor e scritto nel valore di uscita.
- ABSOLUTE: viene letto un singolo ingresso analogico e il valore assoluto viene scritto nel valore di uscita.
- INVERSE: Viene letto un singolo ingresso analogico, 1.0 viene diviso per il valore di ingresso e scritto nel valore di uscita.
- HYSTERISIS: il singolo ingresso analogico viene letto, ritardato in base a SignalTime e scritto nel valore di uscita.
- RUNNINGAVERAGE: viene letto un singolo ingresso analogico, calcolata la media su SignalTime e scritto nel valore di uscita.
- DERIVATIVE: viene letto un singolo ingresso analogico e il tasso di variazione durante il SignalTime viene scritto nel valore di uscita.

- INTEGRAL: Viene letto un singolo ingresso analogico e il valore medio durante il SignalTime viene scritto nel valore di uscita.
- BINARY: Viene letto un singolo ingresso binario e SignalOffset viene scritto nel valore di uscita se True.
- ACCUMULATOR: viene letto un singolo ingresso binario e per ogni impulso SignalOffset viene aggiunto all'accumulatore e, mentre l'accumulatore è maggiore di SignalFactor, l'accumulatore viene decrementato di SignalFactor e il risultato intero viene incrementato di uno.
- PULSECONVERTER: viene letto un singolo ingresso intero e per ogni incremento viene aggiunto SignalMultiplier e scritto nel valore di uscita.
- SUM: due ingressi analogici vengono letti, sommati e scritti nel valore di uscita.
- SUBTRACT: due ingressi analogici vengono letti, sottratti e scritti nel valore di uscita.
- PRODUCT: due ingressi analogici vengono letti, moltiplicati e scritti nel valore di uscita.
- DIVIDE: due ingressi analogici vengono letti, divisi e scritti nel valore di uscita.
- AVERAGE: Vengono letti due ingressi analogici e la media viene scritta nel valore di uscita.
- MAXIMUM: vengono letti due ingressi analogici e il valore massimo viene scritto nel valore di uscita.
- MINIMUM: vengono letti due ingressi analogici e il valore minimo viene scritto nel valore di uscita.
- INPUT: l'elemento di controllo è un ingresso dedicato.
- OUTPUT: l'elemento controller è un'uscita dedicata.
- VARIABLE: l'elemento controller è una variabile in memoria.

1.2.1.3 Pset_ControllerTypeMultiPosition

- INPUT: l'elemento di controllo è un ingresso dedicato.
- OUTPUT: l'elemento controller è un'uscita dedicata.
- VARIABLE: l'elemento controller è una variabile in memoria.

1.2.1.4 Pset_ControllerTypeProgrammable

- PRIMARY: il controller dispone di un'interfaccia di comunicazione integrata per la connessione al PC, può gestire controller secondari.

- SECONDARY: il controller comunica con il controller primario e i propri dispositivi gestiti.

1.2.1.5 Pset_ControllerTypeProportional

- PROPORTIONAL: l'uscita è proporzionale all'errore di controllo. Il guadagno di un controllo proporzionale (K_p) avrà l'effetto di ridurre il tempo di salita e di ridurre, ma mai eliminare, l'errore di regime della variabile controllata.
PROPORTIONALINTEGRAL: Parte dell'uscita è proporzionale all'errore di controllo e parte è proporzionale all'integrale nel tempo dell'errore di controllo. L'aggiunta del guadagno di un controllo integrale (K_i) avrà l'effetto di eliminare l'errore di stato stazionario della variabile controllata, ma potrebbe peggiorare la risposta transitoria.
- PROPORTIONALINTEGRALDERIVATIVE: Parte dell'uscita è proporzionale all'errore di controllo, parte è proporzionale all'integrale temporale dell'errore di controllo e parte è proporzionale alla derivata temporale dell'errore di controllo.

1.2.1.6 Pset_ControllerTypeTwoPosition

- LOWERLIMITSWITCH: viene letto un singolo ingresso analogico e, se minore di Value.LowerBound, True viene scritto nel valore di uscita.
- UPPERLIMITSWITCH: viene letto un singolo ingresso analogico e, se maggiore di Value.UpperBound, True viene scritto nel valore di uscita.
- LOWERBANDSWITCH: viene letto un singolo ingresso analogico e, se minore di Value.LowerBound+BandWidth, True viene scritto nel valore di uscita.
- UPPERBANDSWITCH: viene letto un singolo ingresso analogico e, se maggiore di Value.UpperBound-BandWidth, True viene scritto nel valore di uscita.
- NOT: Viene letto un singolo ingresso binario e il valore opposto viene scritto nel valore di uscita.
- AND: Vengono letti due ingressi binari e se entrambi sono True allora True viene scritto nel valore di uscita.
- OR: vengono letti due ingressi binari e se uno dei due è Vero, Vero viene scritto nel valore di uscita.
- XOR: Vengono letti due ingressi binari e se uno è vero allora Vero viene scritto nel valore di uscita.
- CALENDAR: Nessun input; l'ora corrente viene confrontata con l'IfcWorkCalendar a cui viene assegnato l'IfcController e viene scritto True se attivo.

- INPUT: l'elemento controller è un ingresso dedicato.
- OUTPUT: l'elemento controller è un'uscita dedicata.
- VARIABLE: l'elemento controller è una variabile in memoria.

1.2.2 *IfcProcedure*

Si definisce come un insieme logico di azioni da intraprendere in risposta a un evento o per causare il verificarsi di un evento.

IfcProcedure viene utilizzato per acquisire informazioni sui processi a fasi come la calibrazione, le procedure di avvio e arresto per gli elementi di un dispositivo, le azioni designate da intraprendere in caso di emergenza, ecc. Una procedura non è un'attività, ma può descrivere una serie di attività e il loro ordine di occorrenza in risposta a o per causare un evento. In generale, non limita l'elemento esterno nel suo complesso, ma descrive passaggi specifici di come qualcosa dovrebbe accadere.

1.2.2.1 *IfcProcedureTypeEnum*

- **ADVICE_CAUTION**: Un'avvertenza che dovrebbe essere presa in considerazione come procedura o durante l'esecuzione di una procedura.
- **ADVICE_NOTE**: Ulteriori informazioni o consigli che dovrebbero essere presi in considerazione come procedura o durante l'esecuzione di una procedura.
- **ADVICE_WARNING**: Un avviso di potenziale pericolo che dovrebbe essere preso in considerazione come procedura o durante l'esecuzione di una procedura.
- **CALIBRATION**: Una procedura intrapresa per calibrare un artefatto.
- **DIAGNOSTIC**: Diagnostico.
- **SHUTDOWN**: Una procedura intrapresa per arrestare l'operazione un artefatto.
- **STARTUP**: Una procedura intrapresa per avviare l'operazione un artefatto.
- **USERDEFINED**: Definito dall'utente.
- **NOTDEFINED**: Non definito.

1.2.3 *IfcWorkCalendar*

Tale oggetto definisce periodi di tempo lavorativi e non lavorativi per attività e risorse. Consente di definire sia periodi di tempo specifici, ad esempio dalle 12:00 alle 14:00 del 25 gennaio 2022, sia periodi di tempo ripetitivi basati su schemi di ricorrenza e utilizzati di frequente, ad esempio ogni lunedì dalle 9:00 alle 12:00 tra 15 marzo 2022 e 20 dicembre 2022.

Un calendario lavorativo è un sottotipo di *IfcControl* e quindi eredita la funzionalità per

il controllo di altri oggetti tramite *IfcRelAssignsToControl*, che viene utilizzato per definire un calendario lavorativo per attività e risorse. Eredita anche un nome e un attributo di descrizione, mentre un nome deve essere dato e una descrizione può essere data come indicazione del suo contenuto e utilizzo.

La definizione dei periodi di tempo è derivabile da un calendario di base e modificata da un insieme di orari di lavoro e orari di eccezione non lavorativi. Tutti i periodi di tempo definiti da *IfcWorkCalendar.ExceptionTimes* sovrascrivono i periodi di tempo ereditati dal calendario di base. Pertanto, gli orari di eccezione sostituiscono gli orari di lavoro del calendario di base.

1.2.3.1 IfcWorkCalendarTypeEnum

Un *IfcWorkCalendarTypeEnum* è un tipo di dati di enumerazione che specifica i tipi di calendario lavorativo da cui è possibile selezionare il controllo pertinente. Se fornito, dovrebbe aiutare a identificare i calendari di base.

- FIRSTSHIFT: Appartiene al primo turno.
- SECONDSHIFT: Appartiene al secondo turno.
- THIRDSHIFT: Appartiene al terzo turno.
- USERDEFINED: Definito dall'utente.
- NOTDEFINED: Non definito.

1.2.4 IfcDate

IfcDate identifica un particolare giorno di calendario, espresso per anno, mese e giorno. È espresso da una stringa di valori che segue la seguente rappresentazione lessicale: AAAA-MM-GG, dove AAAA rappresenta l'anno, MM il numero ordinale del mese e GG il numero ordinale del giorno.

1.2.5 IfcDateTime

IfcDateTime identifica un particolare momento temporale, espresso in ore, minuti e secondi facoltativi trascorsi in un giorno di calendario, espresso in anno, mese e giorno nel mese. È espresso da una stringa di valori che segue la seguente rappresentazione lessicale: AAAA-MM-GGThh:mm:ss dove "AAAA" rappresenta l'anno, "MM" il mese e "GG" il giorno, la lettera "T" è il separatore di data/ora e "hh", "mm", "ss" rappresentano rispettivamente l'ora, i minuti e i secondi. Se lo si desidera, è possibile utilizzare cifre aggiuntive per aumentare la precisione delle frazioni di secondo, ad esempio è supportato il formato ss.ss con qualsiasi numero di cifre dopo la virgola decimale. La parte dei secondi frazionari è facoltativa; altre parti della forma lessicale non sono facoltative. Per accogliere i valori dell'anno maggiori di 9999 cifre aggiuntive possono essere aggiunte a sinistra di questa rappresentazione. Gli zeri iniziali sono obbligatori se il valore dell'anno avrebbe altrimenti meno di quattro cifre.

1.2.6 *IfcEvent*

Un *IfcEvent* rappresenta qualcosa che accade e che, di conseguenza, attiva un'azione o una risposta. Viene utilizzato per acquisire informazioni su cose particolari che accadono o che potrebbero accadere. Tale oggetto viene particolarmente impiegato nei piani di lavoro poiché identificano, ad esempio, un punto in cui un messaggio contenente informazioni può essere emesso o in cui viene invocata una regola o un vincolo.

1.2.6.1 *IfcEventTriggerTypeEnum*

IfcEventTriggerTypeEnum definisce l'intervallo di diversi tipi di inneschi di evento che possono essere specificati. La loro definizione è stata adottata dalla Business Process Modeling Notation (BPMN), utilizzata anche nell'Information Delivery Manual (IDM) per la definizione dei processi aziendali.

- EVENTCOMPLEX: Un trigger di evento che è una combinazione complessa di cose.
- EVENTMESSAGE: Un trigger di evento che è un messaggio o un insieme di informazioni.
- EVENTRULE: Un trigger di evento che è una regola o un vincolo.
- EVENTTIME: Un trigger di evento che si trova o si verifica dopo un particolare momento o periodo di tempo.
- USERDEFINED: Definito dall'utente.
- NOTDEFINED: Non definito.

1.2.7 *IfcEventTime*

IfcEventTime acquisisce le informazioni relative all'ora di un evento, inclusi i diversi tipi di date dell'evento ossia quelle effettive, programmate, in anticipo o in ritardo. Tutti i valori dati devono essere forniti dall'applicazione, poiché lo schema IFC non si occupa delle dipendenze tra i valori del tempo di processo.

Tale oggetto fornisce inoltre un meccanismo generico per distinguere tra valori temporali dati dall'utente, valori temporali derivati dai primi e altri vincoli come calendari di lavoro e risorse assegnate.

1.2.8 *IfcDistributionSystem*

Tale oggetto rappresenta un sistema di distribuzione, ovvero una rete progettata per ricevere, archiviare, mantenere, distribuire o controllare il flusso di un supporto di distribuzione. Un esempio comune è un sistema di riscaldamento dell'acqua calda costituito da una pompa, un serbatoio e un sistema di tubazioni interconnesse per la distribuzione dell'acqua calda ai terminali.

Il gruppo IfcDistributionSystem definisce l'occorrenza di un sistema specializzato per l'uso nel contesto di servizi di costruzione o utilità per strutture costruite.

1.2.8.1 IfcDistributionSystemEnum

Con questo TypeEnum si identificano i diversi tipi di sistemi di distribuzione. Viene utilizzato per designare i sistemi in base alla loro funzione, nonché le porte dei dispositivi all'interno di tali sistemi per limitare la connettività a connessioni compatibili.

- AIRCONDITIONING: Sistema di distribuzione dell'aria condizionata al fine di mantenere un intervallo di temperatura all'interno di uno o più ambienti.
- AUDIOVISUAL: Un trasporto di una singola sorgente multimediale, con flussi audio e/o video.
- CATENARY_SYSTEM: Un sistema di distribuzione longitudinale che supporta i fili di contatto, inclusi contagocce per catenaria e fili di cucitura.
- CHEMICAL: Sostanza chimica arbitraria ulteriormente qualificata dal set di proprietà, ad esempio per uso medico o industriale.
- CHILLEDWATER: Acqua refrigerata non potabile, ad esempio fatta circolare attraverso un evaporatore.
- COMMUNICATION: Comunicazione.
- COMPRESSED AIR: Sistema ad aria compressa.
- CONDENSERWATER: Acqua non potabile, ad esempio fatta circolare attraverso un condensatore.
- CONTROL: Un trasporto o una rete dedicata al controllo dell'utilizzo del sistema.
- CONVEYING: Fornitura arbitraria di sostanze.
- DATA: Una rete con utilizzo generico.
- DISPOSAL: Smaltimento arbitrario di sostanze.
- DOMESTIC COLD WATER: Sistema di distribuzione dell'acqua potabile non riscaldata.
- DOMESTIC HOT WATER: Sistema di distribuzione dell'acqua potabile riscaldata.
- DRAINAGE: Sistema di raccolta dei drenaggi.
- EARTHING: Un percorso per il collegamento equipotenziale, che conduce corrente a terra.
- ELECTRICAL: Un circuito per fornire energia elettrica.
- ELECTROACOUSTIC: Un segnale audio amplificato come per gli altoparlanti.

- EXHAUST: Sistema di raccolta dell'aria di scarico per rimuovere l'aria viziata o nociva da uno o più ambienti.
- FIREPROTECTION: Sistema antincendio a sprinkler.
- FIXEDTRANSMISSIONNETWORK: Rappresenta tutte le reti cablate che forniscono un canale di trasmissione dati utilizzando cavi in fibra ottica, cavi in rame o entrambi. Aggrega molte tecnologie basate sul metodo multiplexing.
- FUEL: Fornitura arbitraria di carburante.
- GAS: Materiali in fase gassosa come metano o gas naturale.
- HAZARDOUS: Sistema di raccolta di materiali o fluidi pericolosi.
- HEATING: Acqua o vapore riscaldato da una caldaia e fatto circolare attraverso i radiatori.
- LIGHTING: Un circuito dedicato all'illuminazione, come un apparecchio con prese per lampade.
- LIGHTNINGPROTECTION: Un percorso per condurre la corrente del fulmine a terra.
- MOBILENETWORK: La rete mobile assicura la comunicazione wireless fornendo una piattaforma sicura per la comunicazione voce e dati tra gli operatori dell'infrastruttura, inclusi autisti, spedizionieri, membri del team di manovra e controllori di stazione.
- MONITORINGSYSTEM: Sistema basato su sensori per il monitoraggio e il controllo ambientale di edifici e infrastrutture.
- MUNICIPALSOLIDWASTE: Articoli consumati e scartati, comunemente noti come spazzatura o immondizia.
- OIL: Sistema di distribuzione dell'olio.
- OPERATIONAL: Sistema di forniture operative.
- OPERATIONALTELEPHONYSYSTEM: Un sistema che consente le comunicazioni tra gli operatori (es. switchtender, regolatore del traffico, agenti operativi, ecc.) nei centri operativi e sul sito dell'infrastruttura (es. ferrovia, galleria o strada).
- OVERHEAD_CONTACTLINE_SYSTEM: Un sistema di linea aerea di contatto al di sopra del limite superiore del treno che utilizza una linea aerea di contatto e un sistema di catenaria per fornire corrente alle unità di trazione.
- POWERGENERATION: Un percorso per la produzione di energia.
- RAINWATER: Acqua piovana derivante da precipitazioni che cadono

direttamente su un lotto.

- REFRIGERATION: Sistema di distribuzione del refrigerante allo scopo di completare in tutto o in parte un ciclo di refrigerazione.
- RETURN_CIRCUIT: Un sistema di distribuzione che costituisce il percorso previsto per la corrente di ritorno della trazione e la corrente in condizioni di guasto.
- SECURITY: Un trasporto o una rete dedicata all'utilizzo del sistema di sicurezza.
- SEWAGE: Sistema di raccolta delle acque reflue.
- SIGNAL: Un segnale analogico grezzo, come dati modulati o misurazioni dai sensori.
- STORMWATER: Acqua piovana derivante da precipitazioni che scorrono via o viaggiano sulla superficie del suolo.
- TELEPHONE: Un trasporto o una rete dedicata all'utilizzo del sistema telefonico.
- TV: Un trasporto di più fonti multimediali come TV analogica via cavo, TV satellitare o TV via etere.
- VACUUM: Sistema di distribuzione del vuoto.
- VENT: Sistema di sfiato per sistemi di tubazioni per acque reflue.
- VENTILATION: Sistema di distribuzione dell'aria di ventilazione coinvolto sia nel ricambio dell'aria verso l'esterno che nella circolazione dell'aria all'interno di un edificio.
- WASTEWATER: Acqua compromessa in termini di qualità dall'influenza antropica, possibilmente originata da acque reflue, drenaggio o altra fonte.
- WATERSUPPLY: Approvvigionamento idrico arbitrario.
- USERDEFINED: Nessuna descrizione disponibile.
- NOTDEFINED: Nessuna descrizione disponibile.

1.2.9 IfcSensor

Un sensore rappresenta un dispositivo capace di misurare una quantità fisica e convertirla in un segnale leggibile dall'utente, come anche da un'ulteriore strumentazione. Per identificare al meglio il dispositivo, arricchendolo di dettagli, è specificato dallo standard un elenco di tipi, l'*IfcSensorTypeEnum*, il quale definisce l'intervallo di diversi tipi di sensore che possono essere specificati.

1.2.9.1 IfcSensorTypeEnum

- COSENSOR: Sensore che rivela monossido di carbonio (CO).

- CO2SENSOR: Sensore che rivela diossido di carbonio (CO₂).
- CONDUCTANCESENSOR: Sensore che rileva la conduttanza elettrica.
- CONTACTSENSOR: Sensore che percepisce il contatto, come per esempio per rilevare se una porta è chiusa.
- FIRESENSOR: Sensore che rivela il fuoco.
- FLOWSSENSOR: Sensore che rileva il flusso in un fluido.
- FROSTSENSOR: Sensore che rileva la brina su una finestra.
- GASSENSOR: Sensore che rivela una concentrazione di gas, diverso da CO₂.
- HEATSENSOR: Sensore che rileva il calore.
- HUMIDITYSENSOR: Sensore che rileva l'umidità.
- IDENTIFIERSENSOR: Sensore capace di leggere un "tag", ad esempio per avere accesso ad una porta o ad un ascensore.
- IONCONCENTRATIONSENSOR: Sensore che rivela la concentrazione di ioni, ad esempio per valutare la durezza dell'acqua.
- LEVELSENSOR: Sensore che rileva il livello di riempimento, come per un serbatoio.
- LIGHTSENSOR: Sensore che rileva la luce.
- MOISTURESENSOR: Sensore che rileva il tasso di idratazione o di umidità.
- MOVEMENTSENSOR: Sensore che rivela il movimento.
- PHSENSOR: Sensore che rileva l'acidità (pH) di gas e liquidi.
- PRESSURESENSOR: Sensore che rileva la pressione.
- RADIATIONSENSOR: Sensore che rivela le radiazioni.
- RADIOACTIVITYSENSOR: Sensore che rileva il decadimento atomico (radioattività).
- SMOKESENSOR: Sensore che rivela la presenza di fumo.
- SOUNDSSENSOR: Sensore che rileva il suono.
- TEMPERATURESENSOR: Sensore che rileva la temperatura.
- WINDSENSOR: Sensore che rileva direzione e velocità del vento.
- EARTHQUAKESENSOR: Dispositivo che rileva un'onda sismica e misura l'intensità sismica in caso di terremoto.
- FOREIGNOBJECTSENSOR: Dispositivo che rileva la presenza di oggetti estranei

che possono interrompere la rete elettrica, in tal caso facendo scattare un allarme.

- OBSTACLESENSOR: Dispositivo che rileva la presenza di ostacoli (rivela ad esempio la caduta di oggetti da un ponte, la caduta di massi, etc.).
- RAINSENSOR: Dispositivo che raccoglie informazioni relative alla quantità di pioggia caduta.
- SNOWDEPTHSENSOR: Dispositivo che rileva la profondità del manto nevoso.
- TRAINSENSOR: Dispositivo generalmente installato nella parte posteriore dell'ultima carrozza di un convoglio ferroviario e che agisce su un'altra apparecchiatura fissa, dando l'indicazione che il treno è completo.
- TURNOUTCLOSURESENSOR: Dispositivo che rileva la posizione della lama di un deviatoio ferroviario.
- WHEELSENSOR: Dispositivo che rivela il passaggio di una ruota.
- USERDEFINED: Tipo definito dall'utente.
- NOTDEFINED: Tipo non definito.

1.2.10 IfcSpatialZone

Una zona spaziale è una scomposizione non gerarchica e potenzialmente sovrapposta del progetto sotto qualche considerazione funzionale. Una zona spaziale potrebbe essere utilizzata per rappresentare una zona termica, una zona di costruzione, una zona di illuminazione, una zona di area utilizzabile. Una zona spaziale potrebbe avere il suo posizionamento indipendente e la sua rappresentazione della forma.

IfcSpatialZone eredita e dichiara questi attributi che avranno il seguente significato:

- Nome: un numero o un designatore fornito dall'utente o dal sistema per l'elemento territoriale
- LongName: nome dell'elemento spaziale fornito dall'utente
- Descrizione: qualsiasi descrizione aggiuntiva fornita dall'utente
- ObjectType: riservato per la tipizzazione di elementi spaziali in caso di PredefinedType = .USERDEFINED., le restrizioni sui valori applicabili potrebbero essere pubblicate nelle definizioni della vista o negli accordi dell'implementatore.

1.2.10.1 IfcSpatialZoneTypeEnum

Questa enumerazione definisce l'intervallo di diversi tipi di zone spaziali che possono specificare ulteriormente un *IfcSpatialZoneTypeEnum*.

- CONSTRUCTION: La zona spaziale è utilizzata per rappresentare una zona di

costruzione per il processo di produzione.

- FIRESAFETY: La zona spaziale è utilizzata per rappresentare una zona di sicurezza antincendio, o compartimento antincendio.
- INTERFERENCE: La zona spaziale viene utilizzata per definire un'interferenza tra le occorrenze di IfcSpatialElement.
- LIGHTING: La zona spaziale viene utilizzata per rappresentare una zona di illuminazione; una zona diurna o una zona di illuminazione artificiale.
- OCCUPANCY: La zona spaziale viene utilizzata per rappresentare una zona di particolare occupazione.
- RESERVATION: Una zona spaziale che segna una sorta di riserva all'interno dell'estensione del progetto.
- SECURITY: La zona spaziale viene utilizzata per rappresentare una zona per la pianificazione della sicurezza e per i lavori di manutenzione.
- THERMAL: La zona spaziale è usata per rappresentare una zona termica.
- TRANSPORT: Nessuna descrizione disponibile.
- VENTILATION: La zona spaziale viene utilizzata per rappresentare una zona di ventilazione.
- USERDEFINED: Zona spaziale di tipo definito dall'utente.
- NOTDEFINED: Zona spaziale di tipo non definito.

1.2.11 IfcCommunicationsAppliance

Un apparecchio di comunicazione trasmette e riceve informazioni elettroniche o digitali sotto forma di dati o suoni.

Gli apparecchi di comunicazione possono essere fissati sul posto o possono essere spostati da uno spazio all'altro. Gli apparecchi di comunicazione richiedono un'alimentazione elettrica che può essere fornita da un circuito elettrico o fornita da una fonte di batteria locale.

1.2.11.1 IfcCommunicationsApplianceTypeEnum

- ANTENNA: Un trasduttore progettato per trasmettere o ricevere onde elettromagnetiche.
- AUTOMATON: Un dispositivo artificiale auto-agente, il cui comportamento è governato o in modo graduale da determinate regole decisionali o continuamente nel tempo da relazioni definite, mentre le variabili di output del quale sono create dalle sue variabili di input e di stato. Tale definizione proviene da IEC 60050-351-42-32.

- COMPUTER: Un desktop, un laptop, un tablet o un altro tipo di computer che può essere spostato da un luogo all'altro e collegato a un'alimentazione elettrica tramite una presa collegata.
- FAX: Una macchina che ha la funzione primaria di trasmettere una copia facsimile di stampati utilizzando una linea telefonica.
- GATEWAY: Un gateway collega più segmenti di rete con protocolli diversi a tutti i livelli (livelli 1-7) del modello Open Systems Interconnection (OSI).
- INTELLIGENTPERIPHERAL: Una periferica intelligente è un dispositivo che offre una varietà di risorse specializzate secondo il corrispondente programma logico di servizio sotto il controllo di SCP. Queste risorse contengono il ricevitore di DTMF (Dual-Tone Multi-Frequency, generatore di segnale, avviso di registrazione, ecc.). Una periferica intelligente fornisce funzioni di risorse dedicate nella rete intelligente, alloca, controlla e gestisce varie risorse dedicate, comunica con altre entità nella rete e completa le funzioni di risorse SRF nonché le funzioni di manutenzione, gestione e statistiche delle risorse.
- IPNETWORKEQUIPMENT: Un'apparecchiatura di rete IP è un dispositivo che fornisce un canale di trasmissione dati IP per sottosistemi di telecomunicazioni o altri sottosistemi, ad esempio router, switch di rete o firewall.
- LINESIDEELECTRONICUNIT: L'unità elettronica lungo la linea (LEU) è l'interfaccia tra la balise e l'interblocco ferroviario. Il LEU acquisisce le informazioni dall'interlocking, e invia le opportune informazioni alle balise in concordanza con la segnalazione di linea (se disponibile).
- MODEM: Un modem (da modulatore-demodulatore) è un dispositivo che modula un segnale portante analogico per codificare informazioni digitali e demodula anche tale segnale portante per decodificare le informazioni trasmesse.
- NETWORKAPPLIANCE: Un'apppliance di rete esegue una funzione dedicata come la protezione del firewall, il filtraggio dei contenuti, il bilanciamento del carico o la gestione delle apparecchiature.
- NETWORKBRIDGE: Un bridge di rete collega più segmenti di rete al livello di collegamento dati (livello 2) del modello OSI e il termine switch di livello 2 è molto spesso utilizzato in modo intercambiabile con bridge.
- NETWORKHUB: Un hub di rete collega più segmenti di rete al livello fisico (livello 1) del modello OSI.
- OPTICALLINETERMINAL: Un terminale di linea ottica è un endpoint del fornitore di servizi di una rete ottica passiva o attiva. È l'apparecchiatura terminale per il collegamento dei tronchi in fibra ottica.

- OPTICALNETWORKUNIT: Un'unità di rete ottica è un tipo di apparecchiatura di connessione alla rete di trasmissione ottica che viene installata sul lato utente.
- PRINTER: Una macchina che ha la funzione primaria di stampare testo e/o grafica su carta o altri supporti.
- RADIOBLOCKCENTER: Un centro di blocco radio è un dispositivo informatico specializzato nelle ferrovie con specifiche per generare autorità di movimento (MA) e trasmetterle ai treni. Riceve informazioni dal controllo del segnalamento e dai treni nella sua sezione.
- REPEATER: Un ripetitore è un dispositivo elettronico che riceve un segnale e lo ritrasmette a un livello superiore e/oa una potenza maggiore, o dall'altra parte di un ostacolo, in modo che il segnale possa coprire distanze maggiori senza degrado.
- ROUTER: Un router è un dispositivo di rete il cui software e hardware sono generalmente adattati alle attività di instradamento e inoltramento delle informazioni. Ad esempio, su Internet, le informazioni vengono indirizzate a vari percorsi dai router.
- SCANNER: Una macchina che ha la funzione primaria di scansionare il contenuto del materiale stampato e convertirlo in un formato digitale che può essere memorizzato in un computer.
- TELECOMMAND: Un sistema che invia il comando per controllare e monitorare gli interruttori e gli interruttori o sistemi direttamente o non collegati (ad esempio tramite fili) all'interno del sistema di alimentazione a distanza.
- TELEPHONYEXCHANGE: Una centrale telefonica è un dispositivo che garantisce l'instradamento delle chiamate e delle comunicazioni telefoniche.
- TRANSITIONCOMPONENT: Un componente di transizione è un dispositivo attivo minore che converte i segnali elettrici in segnali ottici al mittente e converte i segnali ottici in segnali elettrici al ricevitore.
- TRANSPONDER: Un transponder è un dispositivo di comunicazione, monitoraggio o controllo che, alla ricezione di un segnale, emette un segnale diverso in risposta. I transponder possono essere passivi o attivi (ad es. beacon elettronico, balise).
- TRANSPORTEQUIPMENT: Un'apparecchiatura di trasporto è un elemento di rete responsabile della fornitura di funzionalità di trasporto, multiplexing, commutazione, gestione e supervisione dei canali di trasmissione tra diversi host. Il servizio di trasporto dati utilizza tre metriche specifiche: la larghezza di banda, il jitter e il tasso di perdita di pacchetti.

- USERDEFINED: Tipo definito dall'utente.
- NOTDEFINED: Tipo non definito.

1.2.12 *IfcTask*

Un *IfcTask* è un'unità di lavoro identificabile da svolgere in un progetto di costruzione. Viene generalmente utilizzata per descrivere un'attività per la costruzione o l'installazione di prodotti, ma non è limitata unicamente a questi tipi. Ad esempio, potrebbe essere utilizzato per descrivere i processi di progettazione, le operazioni di spostamento e anche altre attività correlate alla progettazione, costruzione e funzionamento.

1.2.12.1 *IfcTaskTypeEnum*

- ADJUSTMENT: Apportare modifiche alla configurazione fisica di qualcosa.
- ATTENDANCE: Partecipazione o attesa di altre cose che accadono.
- CALIBRATION: Apportare modifiche alla configurazione operativa di qualcosa.
- CONSTRUCTION: Costruire o costruire qualcosa.
- DEMOLITION: Demolire o abbattere qualcosa.
- DISMANTLE: Smontare qualcosa con cura in modo che possa essere riciclato o riutilizzato.
- DISPOSAL: Smaltire o sbarazzarsi di qualcosa.
- EMERGENCY: Compiti richiesti quando si risponde o si garantisce la capacità di rispondere a una situazione di emergenza. Ad esempio, spazi, zone e sistemi possono avere compiti di emergenza in caso di diverse condizioni di emergenza. Un compito può essere destinato a garantire la configurazione sicura di un'apparecchiatura durante una situazione imprevista.
- INSPECTION: Controlla se qualcosa è installato e funziona entro i parametri previsti.
- INSTALLATION: Installare qualcosa (equivalente alla costruzione ma più comunemente usato per attività di ingegneria).
- LOGISTIC: Trasporto o consegna di qualcosa.
- MAINTENANCE: Compiti necessari per mantenere un oggetto in buone condizioni di funzionamento.
- MOVE: Spostare le cose da un posto all'altro.
- OPERATION: Una procedura intrapresa per avviare l'operazione un artefatto.
- REMOVAL: Rimozione di un oggetto dall'uso e sottrazione dal suo luogo di utilizzo.

- RENOVATION: Portare qualcosa a uno stato "come nuovo".
- SAFETY: Attività necessarie per garantire un uso sicuro dell'oggetto. Ad esempio, le istruzioni di "blocco" elettrico.
- SHUTDOWN: L'insieme di attività necessarie per uno spegnimento ordinato senza impatti negativi, tipicamente applicato ai sistemi.
- STARTUP: L'insieme di attività necessarie per avviare o riavviare il funzionamento senza impatti negativi, in genere applicato ai sistemi.
- TESTING: L'insieme di attività necessarie per valutare le prestazioni di un oggetto, per garantire che qualcosa sia installato e funzioni entro i parametri previsti. Ad esempio, un sistema o un tipo di elemento può avere attività di test come parte della manutenzione ordinaria.
- TROUBLESHOOTING: L'insieme di attività necessarie per diagnosticare problemi di prestazioni comunemente riscontrati, tipicamente applicati a tipi di elementi e sistemi.
- USERDEFINED: Definito dall'utente.
- NOTDEFINED: Non definito.

1.2.13 *IfcPerformanceHistory*

IfcPerformanceHistory viene utilizzato per documentare le prestazioni effettive di un'istanza di occorrenza nel tempo. Include dati misurati dai sistemi di automazione degli edifici e dati specificati dall'uomo come attività e utilizzo delle risorse. Questi dati possono rappresentare condizioni reali, previsioni o simulazioni.

In generale, i dati in tempo reale tracciati dalla cronologia delle prestazioni assumono la forma di insiemi di proprietà, le quali sono basate su serie temporali. A differenza dei dati basati sulla progettazione in corrispondenza di occorrenze e tipi, i dati che si fondano sulle prestazioni sono sensibili al tempo e possono cambiare in tempo reale da alcuni dispositivi di misurazione. I valori misurati possono essere acquisiti a intervalli irregolari, ad esempio quando i valori cambiano oltre le soglie stabilite o a intervalli regolari di durata specificata.

Il `TypeEnum` di questa entità non fornisce ulteriore descrizione in quanto può essere solamente `USERDEFINED` oppure `NOTDEFINED`. Può tornare utile però, definire la "Fase del ciclo di vita" tramite *IfcLabel*. Tale etichetta descrive la fase del ciclo di vita dell'edificio applicabile e i valori tipici sono a scelta dell'utente; quelli indicati dallo standard sono: `DESIGNDEVELOPMENT`, `SCHEMATICDEVELOPMENT`, `CONSTRUCTION`, `OPERATION`, `ASBUILT`, `CONSTRUCTIONDOCUMENT`, `COMMISSIONING`.

1.2.13.1 IfcDataOriginEnum

Con IfcDataOriginEnum si identifica l'origine dei dati temporali.

- MEASURED: L'origine dei dati temporali è un dispositivo di misurazione.
- PREDICTED: I dati temporali sono una previsione.
- SIMULATED: L'origine dei dati temporali è una simulazione.
- USERDEFINED: Definito dall'utente.
- NOTDEFINED: L'origine dei dati temporali non è definita.

1.2.14 IfcSchedulingTime

IfcSchedulingTime è il supertipo astratto di entità che acquisiscono informazioni relative al tempo dei processi. Uno degli attributi associabili a tale classe è l'IfcDataOriginEnum.

1.2.14.1 IfcDataOriginEnum

Tale proprietà identifica l'origine dei dati temporali.

- MEASURED: L'origine dei dati temporali è un dispositivo di misurazione.
- PREDICTED: I dati temporali sono una previsione.
- SIMULATED: L'origine dei dati temporali è una simulazione.
- USERDEFINED: Definito dall'utente.
- NOTDEFINED: L'origine dei dati temporali non è definita.

1.2.15 IfcTime

Tale oggetto identifica un tempo all'interno di una giornata, espresso in ore, minuti e secondi. È espresso da una stringa di valori che segue la seguente rappresentazione lessicale: hh:mm:ss dove hh rappresenta le ore, mm minuti e ss secondi. Se lo si desidera, è possibile utilizzare cifre aggiuntive per aumentare la precisione delle frazioni di secondo, ad esempio il formato ss.ss. È possibile indicare anche un fuso orario tramite l'utilizzo di un segno [+/-] seguito da hh e opzionalmente: mm. Ad esempio, 18:00:00-02:00 rappresenta le 18:00 per un fuso orario che è 8 ore indietro rispetto al Coordinated Universal Time.

1.3 Proposta di un'integration system tra BACnet e IFC mediante una BACnet MVD

Usufruendo delle metodologie IDM/MVD è possibile definire uno schema di sottoinsieme IFC, come un BACnet MVD, in modo da rappresentare le informazioni della building automation conformi al protocollo di scambio BACnet nel modello di dati IFC, allo scopo di scambiare informazioni tra strumenti BIM e di facility management durante le fasi del progetto (Tang et al., 2020). Il lavoro proposto da Tang, Shelden e al. segue appunto questa logica, sfruttando al meglio le metodologie citate tramite un processo ben scandito e dividibile

in tre parti, come mostrato in figura seguente.

Nei paragrafi che seguono sarà illustrato il percorso che gli stessi hanno seguito, partendo dallo sviluppo di un modello di processo, di un MDV e infine di un test prototipo, illustrando anche le limitazioni emerse.

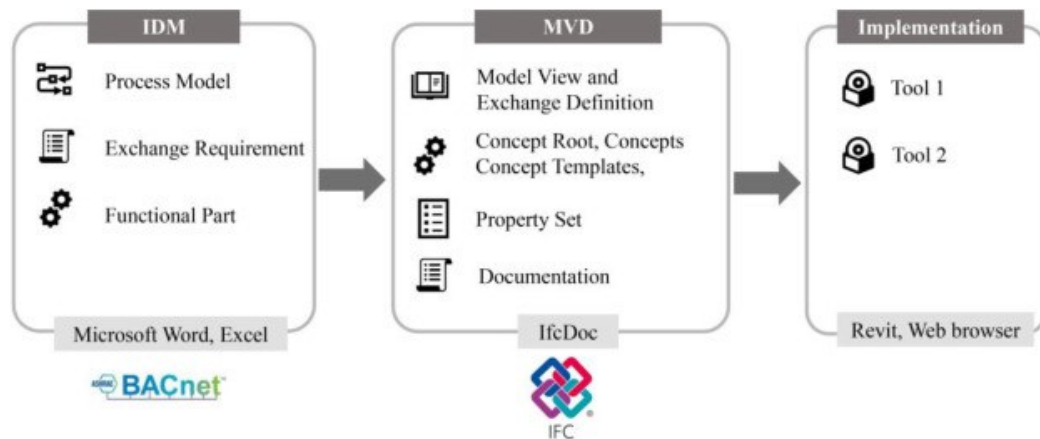


Figura 3.4: Processo di implementazione del BACnet MVD. (Tang et al., 2020)

1.3.1 Parte prima: sviluppo del modello di processo

Seguendo l'approccio IDM, è stato creato un modello di processo per catturare lo scopo e le informazioni BAS di scambio dalla fase di progettazione alla fase operativa. La formalizzazione del modello di processo descrive infatti le attività, le informazioni correlate, la sequenza logica delle attività e i ruoli coinvolti per un determinato obiettivo. L'Object Management Group (OMG) ha sviluppato il Business Process Model and Notation (BPMN) per la modellazione dei processi. Esso fornisce un linguaggio di diagrammi standard per mappare rappresentazioni orientate al flusso del processo aziendale, facilitando l'identificazione degli Exchange Requirements. Questi ultimi, da scambiare a livello di utente per impianti di building automation, sono stati identificati all'interno dello standard ANSI/ASHRAE 135-2016.

Il modello di processo così creato per illustrare il processo di progettazione BAS e lo scambio di informazioni operative attraverso le fasi del progetto presenta codici e fasi di questo processo sono basati sul sistema di classificazione delle costruzioni OmniClass. Durante la fase di progettazione vengono sviluppati determinati modelli BIM preliminari, contenenti la creazione di dati contestuali, i quali sono poi presentati ai progettisti BAS per procedere con l'adeguata progettazione preliminare. Il modello di automazione finale contenente i metadati BAS può essere importato in strumenti di Facility Management per BAS nella fase operativa.

Il processo IDM ha facilitato lo scambio di informazioni dal livello utente al livello dello schema tecnico. Gli autori hanno identificato 395 unità di informazioni tra cui 25 BACnet Object Types e 370 identificatori di proprietà, che sono stati riportati in elenco all'interno di un file excel ER_BACnet MVD.xlsx. Sono stati aggiunti i requisiti di scambio relativi alla rappresentazione geometrica, all'ubicazione, alle relazioni spaziali e alla connettività,

trattandosi di informazioni fondamentali per la progettazione BAS mediante la modellazione BIM di istanze virtuali. Tra queste unità di informazioni sono dunque incluse: Object Geometric Representation, Object Placement, Object Contained In Space, Object Connected From e Decomposes. Le unità informative ricavate sono state poi ulteriormente suddivise in Functional Parts (FP).

Infine, si è effettuata una mappatura concettuale e si sono riportate le informazioni aggiuntive relative ai requisiti di importazione ed esportazione. Un esempio è riportato in figura 3.5.

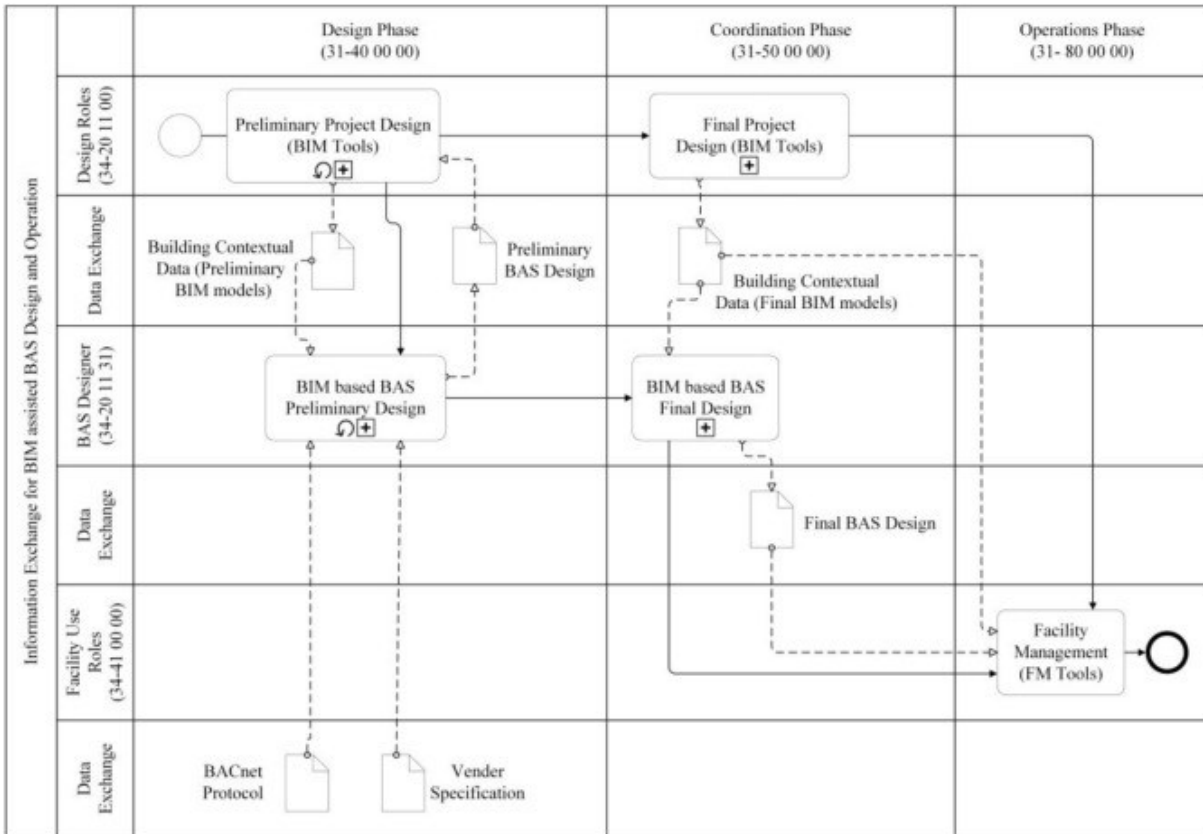


Figura 3.5: Mappatura dei processi per la progettazione di BACS su BIM e rappresentazione dello scambio di informazioni operative. (Tang et al., 2020)

1.3.2 Parte seconda: sviluppo dell'MDV

Successivamente da questi dati si è sviluppata una apposita MVD utilizzando lo strumento IfcDoc.

L'MVD BACnet creato si è concentrato sull'ultima versione dello schema IFC, ossia la IFC4 Addendum 2 (IFC4.2) e ha consentito alle uscite IDM di essere tradotte in entità, attributi e proprietà IFC al fine di facilitare l'interoperabilità a livello di software. Ogni tipo di oggetto, identificatore di proprietà, tipo di dato di proprietà e codice di conformità nel protocollo BACnet è stato così trasformato rispettivamente in entità, attributo, tipo di dato IFC e requisiti di importazione/esportazione.

La figura 3.6 mostra una matrice di utilizzo delle entità e l'utilizzo dei concetti nella MVD BACnet. Le caselle colorate rappresentano i requisiti di importazione/esportazione per i concetti applicati a ciascuna entità, i quali hanno seguito il codice di conformità BACnet. Ad esempio, le unità di informazione con codice di conformità "R" (obbligatorio) e "O" (opzionale) sono state impostate rispettivamente su importazione/esportazione obbligatoria (verde) e opzionale (gialla).

	UnitAssignment	FootPrintAnnotation Geometry	FootPrint Geometry	Box Geometry	Product Local Placement	Path Connectivity	Spatial Containment	Spatial Structure	BACnet-Task Assign to Controller	BACnet-Procedure Assign to Event	BACnet-Event Assign to Controller	BACnet-Controller Assign to Work Calendar	BACnet-Procedure Assign to Controller	Product Assignment	Control Assignment	Object Nesting	Object Definition Decomposition	Object Definition Composition	Document Association	Work Times	Element Occurrence Attributes	Object Predefined Type	Object User Identity	Software Identity	BACnet Performance History Predefined	BACnet Sensor Predefined Type	BACnet Distribution System Predefined Type	BACnet Procedure Predefined Type	BACnet Controller Predefined Type	Quantity Set	Property Sets for Objects	Object Typing
IfcController																																
IfcDistributionSystem																																
IfcEvent																																
IfcPerformanceHistory																																
IfcProcedure																																
IfcProxy																																
IfcSensor																																
IfcTask																																
IfcTimeSeries																																
IfcUnitAssignment																																
IfcWorkCalendar																																
IfcZone																																

■ Import/Export Mandatory
 ■ Incompatible
 ■ Import/Export Required
 ■ Within Scope but not defined
 ■ Import/Export Optional
 Defined but not relevant

Figura 3.6: Requisiti di esportazione e importazione per sistemi BACS nella MVD BACnet (Tang et al., 2020)

Per la determinazione degli entity usage, dei concept usage e dei requisiti per la vista BACnet, si è seguita la convenzione:

- Tipi di oggetti BACnet → IfcEntities → Entities Usage
- Identificatori delle proprietà degli oggetti BACnet → IfcEntity.Attributes → Concept Usage

I concept usage ricavati hanno contenuto sia modelli concettuali (concept templates) predefiniti dal file di base sia modelli concettuali personalizzati.

Nonostante il file di base contenga un set predefinito di concept templates che può esprimere determinate relazioni tra alcuni IfcEntities, sono tuttora presenti delle limitazioni per esprimere al meglio le relazioni tra i tipi di oggetto BACnet e i loro identificatori di proprietà. I concept templates predefiniti, come l'identità del software (software identity), il contenimento spaziale (spatial containment) e l'identità dell'utente dell'oggetto (object user identity), soddisfano la necessità di rappresentare le relazioni tra alcuni tipi di oggetti BACnet e i loro identificatori di proprietà; tuttavia, alcune relazioni non possono essere rappresentate. Ad

esempio, il tipo di oggetto BACnet Calendar Object Type è stato mappato come IfcController, mentre alcuni suoi identificatori di proprietà sono stati mappati come attributi in IfcWorkCalendar. È stata necessaria quindi una relazione di assegnazione tra IfcController e IfcWorkCalendar.

Tabella 10: Elenco completo delle Property Set e delle Property Enumeration user-defined. (Tang et al., 2020)

Property set	Property enumeration
Pset_BACnetAnalogInput Object	PEnum_BACnetAction
Pset_BACnetAnalogOutput Object	PEnum_BACnetDestination
Pset_BACnetAnalogValueObject	PEnum_BACnetDeviceObjectPropertyReference
Pset_BACnetAveragingObject	PEnum_BACnetDeviceStatus
Pset_BACnetBinaryInputObject	PEnum_BACnetEventParameter
Pset_BACnetBinaryOutputObject	PEnum_BACnetEventState
Pset_BACnetBinaryValueObject	PEnum_BACnetEventTimeStamp
Pset_BACnetCommandObject	PEnum_BACnetEventTransitionStamp
Pset_BACnetDeviceObject	PEnum_BACnetEventTransitionBits
Pset_BACnetEventEnrollmentObject	PEnum_BACnetEventType
Pset_BACnetFileObject	PEnum_BACnetFileAccessMethod
Pset_BACnetLifeSafetyPointObject	PEnum_BACnetLifeSafetyMode
Pset_BACnetLifeSafetyZoneObject	PEnum_BACnetLifeSafetyOperation
Pset_BACnetLoopObject	PEnum_BACnetLifeSafetyState
Pset_BACnetMulti-stateInputObject	PEnum_BACnetLoggingType
Pset_BACnetMulti-stateOutputObject	PEnum_BACnetLogRecord
Pset_BACnetMulti-stateValueObject	PEnum_BACnetNotifyType
Pset_BACnetNotificationClassObject	PEnum_BACnetPolarity
Pset_BACnetProgramObject	PEnum_BACnetPolarityArray
Pset_BACnetPulseConverterObject	PEnum_BACnetProgramState
Pset_BACnetScheduleObject	PEnum_BACnetReliability
Pset_BACnetTrendLogObject	PEnum_BACnetSegmentation
	PEnum_BACnetServiceSupported
	PEnum_BACnetSilencedState
	PEnum_BACnetStatusFlag

Tutti i tipi di oggetti BACnet e i loro identificatori di proprietà sono stati mappati come unità di informazione, tuttavia, parte di queste unità di informazione non può essere rappresentata utilizzando lo schema IFC e i set di proprietà predefiniti. A tal proposito, sono stati creati property set e proprietà definite dall'utente (user-defined) per ricalcare il protocollo BACnet in osservanza alla descrizione fornita per i Property Identifier del protocollo BACnet, i quali sono riassunti in Tabella 9.

Tra le proprietà user-defined vi sono per esempio lo stato di accesso (access state), il tipo di proprietà (property type), il tipo di dati principale (property type) e il tipo di dati secondario (secondary property Type). Ad esempio, nel protocollo BACnet, l'identificatore di proprietà BACnet "Status Flag", che rappresenta una spunta booleana per indicare lo stato dell'oggetto, è del tipo di dato BACnet_StatusFlags, che è una enumerazione di IN ALARM, FAULT, OVERRIDEN e OUT OF SERVICE. In accordo al protocollo BACnet, la proprietà personalizzata Status Flag in Pset_BACnetAnalogInputObject è stata impostata come proprietà del tipo P_ENUMERATEDVALUE e il tipo di dato come IfcLabel. Il tipo di dato secondario era un valore di enumerazione personalizzato chiamato PEnum_BACnetStatusFlags che ha ereditato i valori dal tipo di dati BACnetStatusFlag.

Alla fine del processo, IfcDoc ha permesso di generare una documentazione HTML contenente gli entity usage, i concept usage e le proprietà per la progettazione BAS e lo scambio

di informazioni sull'operazione.

1.3.3 Parte terza: sviluppo di un test prototipo

Nell'ultima fase del progetto è stato eseguito un test prototipo utilizzando Revit e il browser Web rispettivamente come strumenti di esportazione e importazione per lo scambio di informazioni sui sistemi BAS definite dalla MVD.

Gli oggetti tipo BACnet sono stati modellati virtualmente come famiglie Revit, per le quali è stato necessario costruire proprietà personalizzate (user-defined property) per riportare le specifiche relative agli identificatori di proprietà BACnet, in aggiunta a quelle automaticamente ottenute dalla modellazione e relative alla rappresentazione geometrica di base, al posizionamento e alla connessione con altri dispositivi. Per ognuna di queste famiglie di oggetti BACnet, sono inoltre stati definiti i parametri di condivisione Revit denominati "IFCExportAs" e "IFCExportType" per specificare le entità di esportazione IFC di destinazione in accordo con l'MVD BACnet proposta.

Nel caricamento di queste famiglie di oggetti BACnet nei progetti Revit in base a vari casi di test, è stata necessaria un'installazione di esportazione "In-session" per seguire i parametri di condivisione precedentemente assegnati.

Il file di istanza IFC esportato dal programma è stato poi sottoposto a screening per verificare se erano inclusi i dati BACnet obbligatori.

Si è poi impiegato IfcDoc per la conversione del file .ifc nel file .ifcXML prima di specificare le entità IFC di importazione. Questa conversione è stata giustificata dalla decisione di utilizzare un browser Web come strumento di importazione. I tipi di oggetti BACnet e gli identificatori di proprietà espressi in IfcXML sono stati dunque importati e visualizzati nel browser Web.

Lo strumento di importazione ha raggiunto l'obiettivo di visualizzare le informazioni BAS desiderate tra cui entità IFC, identificazione univoca globale (GUID), nome, tipo predefinito (PredefinedType) e altre proprietà richieste nel protocollo BACnet, confermando la possibilità di scambiare informazioni BAS usando il modello di dati IFC.

I modelli ifcXML o IFC esportati possono facilitare lo scambio di informazioni BAS nelle situazioni di:

- integrazione con strumenti BAS come Metasys e Niagara;
- connessione con dati di altri domini come letture di sensori in tempo reale;
- integrazione con diversi modelli di dati come Linked Data e JSON;
- condivisione delle informazioni tra le varie fasi del progetto, inclusi progettazione, costruzione, funzionamento, ecc. per i sistemi BAS.

Il lavoro appena descritto può raffigurare un primo passo per facilitare lo scambio di dati in altri settori come l'Electronic Computer Engineering con l'industria AEC e per lo sviluppo

dell'integrazione IoT e BIM.

1.3.4 Limitazioni riscontrate nel caso studio

Nonostante siano stati mappati in questa ricerca 25 tipi di oggetti BACnet e 370 identificatori di proprietà delle entità, attributi e proprietà dello schema IFC, solamente una parte dei BACnet Object Type e dei Property Identifier richiesti sono stati coinvolti per dimostrare la possibilità di rappresentare i dati BACnet nei dati IFC di un modello.

Sono state trovate informazioni ufficiali, seppur limitate, sulla mappatura tra protocollo BACnet e IFC solamente in IFC 2 x 4 senza alcun aggiornamento negli ultimi anni. ASHRAE continua ad aggiungere allegati ogni 2-4 anni per aggiornare i tipi di oggetti BACnet e gli identificatori di proprietà per stare al passo con il mondo BAS in evoluzione. Poiché questo studio ha parzialmente mappato i tipi di oggetti BACnet e gli identificatori di proprietà a causa delle informazioni ufficiali limitate sulla mappatura fornite in IFC 2 x 4, è necessario uno sforzo aggiuntivo per tutti i tipi di oggetti e gli identificatori di proprietà.

I tipi di oggetti BACnet mappati hanno indicato 12 Entity Usage nella MVD BACnet e, in generale, non sono stati in grado di fornire alcuna informazione sugli identificatori e sui servizi delle proprietà BACnet sul sito web di IFC. I Concept Usage sono stati applicati agli Entity Usage con requisiti di importazione ed esportazione, e hanno ospitato informazioni sui tipi di oggetti BACnet e sugli identificatori di proprietà. Tuttavia, i modelli di concetto predefiniti presentavano limitazioni per esprimere pienamente la relazione tra le entità IFC target. Per ovviare a queste limitazioni, nella MVD sono stati definiti diversi concept templates user-defined.

Poiché alcuni identificatori di proprietà BACnet non possono essere rappresentati con proprietà IFC predefinite, sono stati creati set di proprietà, proprietà ed enumerazioni di proprietà personalizzati al fine di risolvere questo problema. Nel complesso sono stati definiti 22 set di proprietà e 362 proprietà. La maggior parte della mappatura dei dati tra i tipi di oggetti BACnet/identificatori di proprietà e entità/attributi IFC è stata raggiunta manualmente, processo che pecca di attendibilità e richiede molto tempo, vale dunque la pena esplorare modi automatici per la mappatura dei dati al fine di garantire un processo di convalida dei dati affidabile, velocizzato e semplificato.

Un'altra limitazione, sempre in termini di mappatura dei dati, è la mancanza di rappresentanza per i servizi BACnet nel modello di dati IFC. I servizi BACnet consentono agli oggetti e ai dispositivi BACnet di inviare comandi per l'accesso e la manipolazione delle informazioni, oltre a fornire funzioni aggiuntive per le applicazioni. I tali, che contengono informazioni sul controllo e la comunicazione tra oggetti BACnet, durante questa ricerca non sono stati rappresentati utilizzando il modello di dati IFC.

Il risultato di questo studio ha dimostrato che lo standard IFC è adatto a rappresentare i metadati BAS, mentre rimane un dubbio riguardo la rappresentazione del controllo BAS e la

comunicazione mediante il modello di dati IFC. Inoltre, questo studio ha una limitazione nel rappresentare pienamente i vincoli delle relazioni tra i tipi di oggetti BACnet e gli identificatori di proprietà. Inoltre, le entità IFC esistenti non soddisfano la rappresentazione di tutti i tipi di oggetti BACnet, per cui ad oggi è necessario creare nuove entità IFC oppure è possibile utilizzare IfcBuildingElementProxy. Tuttavia, il processo di certificazione per l'aggiunta di nuove entità IFC è complesso e l'entità IfcBuildingElementProxy può causare problemi di esportazione per oggetti simili, sono necessarie nuove soluzioni per ulteriori esplorazioni.

Nel corso dei capitoli precedenti è stato dimostrato quanto modellazione delle informazioni di costruzione e sistemi di automazione e controllo degli edifici siano due campi tanto vasti quanto poco definiti in modo univoco. È evidente la mancanza di dialogo aperto e standardizzato tra i due campi e dunque di un metodo che consenta la modellazione di un progetto al più possibile completo, celere e non fraintendibile, senza alcuna perdita di informazioni tra software.

Ad oggi, infatti, lo scenario tipico di un progetto di automazione vede diversi problemi di implementazione. In particolare, quando si sviluppa tale disciplina si crea, attraverso le appropriate piattaforme, una collezione anagrafica di dispositivi non ordinata gestibile generalmente tramite un PLC. Ciò significa che non si avrà una visualizzazione bidimensionale (come planimetrie, sezioni, prospetti) oppure tridimensionale del progetto con gli oggetti in esso contenuti, ma un listato particolarmente ricco di informazioni che risulta poco immediato alla comprensione.

Non solo, la presenza di un modello informativo non è utile alla causa, non è di fatto ancora possibile esportare un file IFC ed importarlo in una piattaforma abilitata all'IoT al fine di avere la compilazione spontanea di tutte le informazioni anagrafiche necessarie. Tali piattaforme, infatti, non supportano questo tipo di estensione file.

Allo scopo di sopperire a tale mancanza, grazie ad una collaborazione avviata con Schneider Electric, si è deciso di implementare un caso studio base con l'utilizzo del software BIM Revit2023 di Autodesk e il software integrato EcoStruxure™ Building Operation di Schneider Electric. Si propone pertanto di esportare un file IFC da Revit e manipolarlo in Visual Studio Code al fine di creare un file innanzitutto compatibile con il software citato e che aspira ad essere un trampolino di lancio per l'automazione del processo progettuale di un sistema di controllo e automazione dell'edificio.

1 ECOSTRUXURE™

EcoStruxure è un'architettura e una piattaforma creata e promossa da Schneider Electric. Essa è abilitata per le tecnologie IoT, pronta all'uso, aperta e promuove l'interoperabilità dei software, è configurata per le abitazioni, gli edifici, fino a data center, infrastrutture e industrie.

L'obiettivo di Schneider è migliorare sicurezza, affidabilità, efficienza, sostenibilità e connettività delle aziende. Grazie a EcoStruxure, unico sistema aperto e abilitato all'IoT

l'azienda è in grado di introdurre l'innovazione ad ogni livello: progetti connessi, sistemi e app di controllo perimetrale, strumenti di analisi e servizi per le aziende. Sfruttando le conoscenze e l'esperienza in quattro settori del mercato (Building, Data Center, Industry, Grid³³) si offrono soluzioni basate su sei architetture di riferimento comprovate: Building, Power, Data Centre, Machine & Plant³⁴ and Grid (figura 4.1). Al centro dell'architettura per i sistemi di IoT c'è EcoStruxure che è capace di connettere qualsiasi cosa all'interno di un'azienda, dall'officina agli uffici, di raccogliere dati importanti dai sensori e dal cloud, analizzare dati per individuare informazioni importanti e di agire in modo efficace chiudendo il ciclo attraverso informazioni in tempo reale e logica di business.

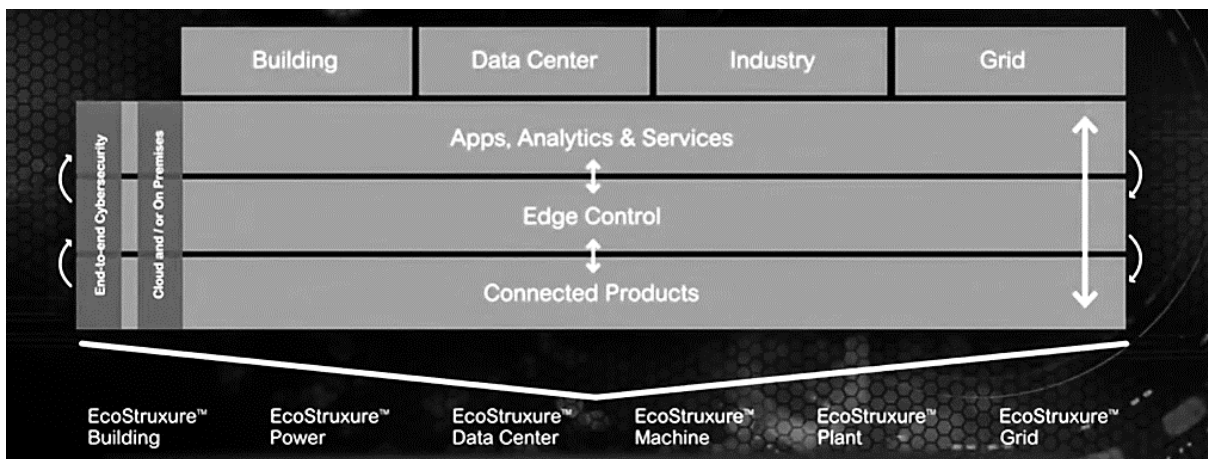


Figura 4.1: Struttura del sistema EcoStruxure di Schneider Electric. (Schneider Electric, 2022)

La piattaforma è la colonna portante tecnologica su cui sono create e basate le soluzioni di Schneider. Questa supporta caratteristiche di connettività e funzioni intelligenti incorporate tramite protocolli di comunicazione basati su standard e la possibilità dei dispositivi intelligenti di eseguire analisi su dati nativi e prendere decisioni di controllo. Aiuta gli sviluppatori, gli integratori di sistemi e il personale tecnico a creare app quali sistemi di monitoraggio, visualizzazione e controllo per l'implementazione di meccanismi intelligenti in tutta l'azienda.

Sfruttando la tecnologia IoT scalabile di Microsoft Azure, la piattaforma EcoStruxure consente a Schneider Electric di garantire servizi digitali a valore aggiunto per ottimizzare le operazioni aziendali, in termini di mobilità, tecnologia cloud, rilevamento, analisi e sicurezza informatica, garantendone la rapidità di adattamento e distribuzione in ambienti di elaborazione dei dati di importanza strategica.

³³ Con il termine Smart Grid si intende la rete intelligente, o meglio un insieme di reti elettriche e di tecnologie che, grazie allo scambio reciproco di informazioni, consentono di gestire e monitorare la distribuzione di energia elettrica da tutte le fonti di produzione e, inoltre, di soddisfare le differenti richieste di elettricità degli utenti collegati, produttori e consumatori in maniera più efficiente, razionale e protetta.

³⁴ La costruzione delle industrie del futuro richiede soluzioni innovative che garantiscano apertura, flessibilità e connettività ineguagliabili per la sicurezza e la sostenibilità delle operazioni. Per questo Schneider garantisce soluzioni innovative degli impianti e per i costruttori di macchinari.

Con tutto ciò si vuole ottenere la massima efficienza energetica e la massima sostenibilità dei risultati, ottimizzare la disponibilità e le prestazioni delle risorse, implementare attività intelligenti, produttive e proficue, ottenere dati sulla mobilità, garantire la riduzione proattiva dei rischi e promuovere l'innovazione e l'interoperabilità.

1.1 EcoStruxure™ Building

EcoStruxure™ Building fa parte di EcoStruxure di Schneider Electric. Con tale piattaforma è possibile modernizzare e rendere scalabile l'infrastruttura di gestione degli edifici esistente e futura. Essa offre una soluzione Internet of Things collaborativa che presenta un'architettura scalabile, sicura e globale capace di rendere intelligenti gli edifici di tutti i tipi.

Tale app collega in modo sicuro hardware, software e servizi su una dorsale di rete Ethernet/IP al fine di massimizzare l'efficienza di un determinato edificio, ottimizzarne il comfort e la produttività e, infine, aumentarne il valore.

EcoStruxure Building include un'offerta completa di software e piattaforme, nello specifico comprende:

- EcoStruxure Building Operation: è un software di building management che fonde sistemi e dati delle applicazioni al fine di semplificare il monitoraggio, l'amministrazione e l'ottimizzazione della vita esecutiva dell'edificio;
- EcoStruxure Building Operation - Energy Expert: è uno schema dedicato alla gestione energetica nell'edificio che traccia i consumi elettrici, ne ripartisce i costi e si dedica a conservare energia;
- EcoStruxure Security Expert: è una soluzione volta alla sicurezza che produce efficienze creando ambienti di business sicuri, raggruppando il controllo degli accessi e il rilevamento delle intrusioni;
- EcoStruxure Fire Expert: è una soluzione antincendio che permette l'accesso immediato ai dati sul sistema antincendio tramite il cloud, velocizzando l'operazione;
- EcoStruxure Building Advisor: è uno strumento che monitora perpetuamente le prestazioni di un edificio, diagnostica in modo proattivo gli eventuali problemi e suggerisce azioni che puntano ad affinare il comfort per gli utenti;
- EcoStruxure Workplace Advisor: si dedica a migliorare l'utilizzo degli edifici permettendo di cogliere al meglio l'utilizzo degli spazi e di prendere decisioni sulla base dei dati raccolti;
- EcoStruxure Apps Studio: è una piattaforma volta allo sviluppo applicativo che permette di creare applicazioni mobili per ispezionare facilmente ogni dettaglio dell'ambiente.

2 ECOSTRUXURE BUILDING OPERATION

Il software di gestione degli edifici, EcoStruxure Building Operation (EBO), è il cuore del controllo periferico del sistema EcoStruxure Building. Consente alle aziende di facilitare lo scambio di dati in sicurezza e l'analisi sia da Schneider Electric che da altri sistemi di terze parti per la gestione di energia, illuminazione, climatizzazione, la sicurezza antincendio, la sicurezza e la gestione del posto di lavoro, sfruttando al contempo la digitalizzazione e i big data con lo scopo di creare edifici intelligenti e orientati al futuro.

Con un framework di integrazione software aperto e sicuro, EcoStruxure Building Operation consente la collaborazione tra i complessi sistemi dei vari fornitori per creare soluzioni personalizzate innovative. La piattaforma EcoStruxure supporta nativamente i protocolli aperti standard del settore, tra cui BACnet, ModBus, LON e ZigBee (figura 4.2).

La stessa piattaforma, inoltre, include anche le certificazioni BACnet Testing Laboratories BTL³⁵ per i componenti hardware e software SmartX del sistema. L'integrazione di prodotti connessi utilizzando protocolli standard del settore garantisce che le nuove espansioni tecnologiche saranno più facili da adottare e implementare.

All'interno della piattaforma EcoStruxure Building, come si può vedere in figura 4.2, sono presenti diversi dispositivi, i quali saranno di seguito descritti.

L'Enterprise Central è il server di supervisione dedicato che si trova più in alto nell'architettura del sistema EBO, in particolare nella zona superiore dell'Edge Control. Può integrare fino ad un massimo di 10 Enterprise Server e assicurare l'hosting di fino a 2.500 SpaceLogic Edge Server (AS-P e AS-B) per consentire una gestione elastica di grandi edifici, campus e complessi multi-sito da una sola postazione.

L'Enterprise Server è la versione per applicazioni Windows di Building Operation Server e si caratterizza come il punto di amministrazione centralizzato delle applicazioni WorkStation, WebStation e Smart-phone. L'Enterprise Server rastrella i dati a livello di sito e configura, ispeziona e monitora il sistema nella sua totalità. Restituisce report aggregati, uniforma le politiche di sicurezza critica, normalizza la presentazione e la priorità degli allarmi e si accerta dell'operatività nell'intera organizzazione.

SpaceLogic ed Edge Server sono i server di automazione a livello periferico dell'architettura, usufruibili per qualsivoglia applicazione. Sono declinati nelle versioni AS-P, ideale per installazioni di tipo medio-piccolo, e AS-B, per le applicazioni di controllo di piccole dimensioni. Sono entrambi dotati di doppia porta Ethernet per portare a livello IP la comunicazione su bus di campo BACnet e modernizzare i sistemi di gestione edifici mantenendo in funzione bus di campo e dispositivi già esistenti.

Il Compliance Pack è un modulo aggiuntivo di EcoStruxure Building che consente ai

³⁵ Certificazione descritta al paragrafo 3.6 del capitolo Secondo.

settori regolamentati di soddisfare i requisiti di conformità delle firme elettroniche e dei registri elettronici. Questa funzionalità di EcoStruxure Building Operation offre la capacità di responsabilità e tracciabilità nell'intera struttura relativamente alle condizioni ambientali che possono influire sulla qualità e la sicurezza dei prodotti sviluppati in ambienti controllati.

La Workstation è un'interfaccia che permette a utenti e tecnici di accedere agli SpaceLogic, agli Edge Server e agli Enterprise Server per raffigurare e coordinare grafici, allarmi, programmazione oraria, tendenze e report. Per ottimizzare l'efficienza di sviluppo e accelerare le funzioni amministrative, la workstation mette a disposizione una nuova serie di strumenti e di funzioni, tra cui applicazioni standard, sostituzioni e aggiornamenti di massa, tipi e librerie personalizzati e molto altro.

La WebStation è un'interfaccia utente che garantisce l'accesso flessibile alle funzioni più frequentemente utilizzate di EcoStruxure Building Operation, su qualunque piattaforma e senza l'installazione di software aggiuntivo. L'area di lavoro si adatta automaticamente a desktop, tablet e smartphone, permettendo agli sviluppatori di risparmiare tempo e risorse.

Lo Smart Connector è un ambiente di sviluppo aperto, scalabile e configurabile che consente agli sviluppatori di creare soluzioni, applicazioni e funzioni innovative che estendono e migliorano EcoStruxure™ Building. Esso apporta un nuovo livello di flessibilità e apertura consentendo un'integrazione rapida, replicabile ed elastica con i sistemi di terze parti o altre fonti di dati.

Uno Smart Driver è un driver personalizzato per la connessione ad altri dispositivi intelligenti di edificio che utilizzano protocolli non nativi. È disponibile con versioni di Building Operation v1.8.1 o superiori.

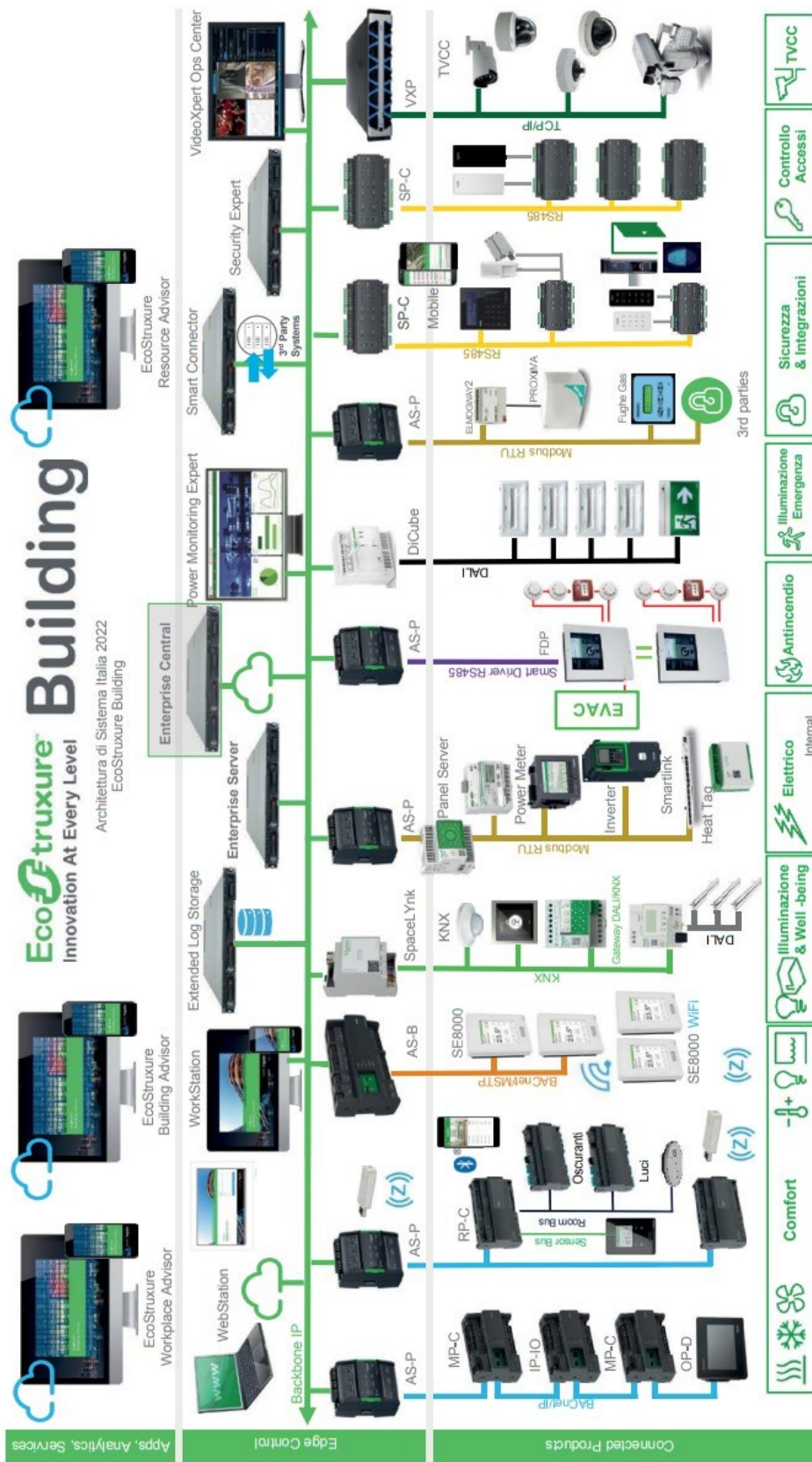


Figura 4.2: Architettura completa di EcoStruxure™ Building. (Schneider Electric, 2022)

2.1 Concetto di Binding in EcoStruxure Building Operation

I programmi sono caratterizzati da svariati elementi: variabili, strutture di controllo e, nei linguaggi orientati ad oggetti, anche gli oggetti e i metodi. Gli elementi citati acquisiscono significato solamente nel momento in cui vengono collegati in maniera opportuna.

Si definisce, allora, binding il processo attraverso il quale viene realizzato un collegamento fra un oggetto ed il suo corrispettivo valore. Nella sua forma più semplice, un binding è, pertanto, costituito da un'origine e da una destinazione, come mostrato in Figura 4.3, dove la freccia mostra il modo in cui scorre la navigazione.

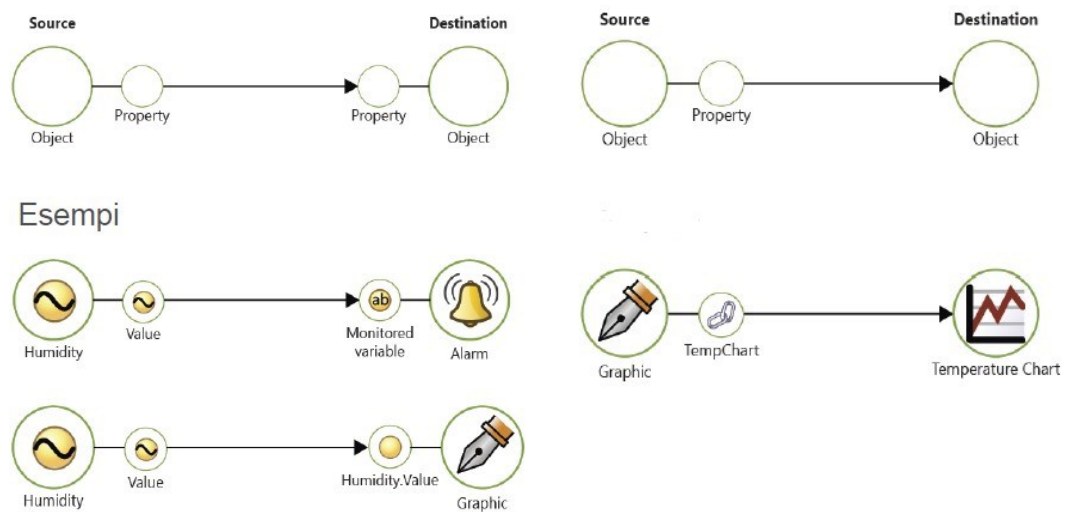


Figura 4.3: A sinistra concetto ed esempi di Binding da proprietà a proprietà, a destra concetto ed esempio di Link tra proprietà ed oggetto.

Le varie associazioni vengono utilizzate per il trasferimento di valori o dati da un'origine a una o più destinazioni. Le associazioni sono fondamentali per la funzionalità, ad esempio per visualizzare valori in tempo reale nella grafica, generare allarmi, gestire programmi di controllo e orari, oppure in generale per la navigazione in aree di lavoro, pannelli e grafica.

In EcoStruxure Building Operation si definiscono gli Oggetti come strutture di dati, mentre le Proprietà sono le informazioni memorizzate negli stessi oggetti. A tal proposito, EBO fa una distinzione tra l'associazione del tipo proprietà-proprietà e quella proprietà-oggetto, quest'ultima infatti non viene definita come binding ma come link (fig. 4.3) e nell'interfaccia del programma saranno raggruppati in liste differenti (fig. 4.4).

Bindings	
DateTime	~/DateTime
OutdoorAirTemperature	/Building A/CAT/Outdoor Air Temperature/Value
Links	
Home_Button.Link	../Buildings

Figura 4.4: Visualizzazione di Binding e Link nell'interfaccia di EBO.

Le proprietà supportate sono di diverso tipo e, all'interno del software, sono concessi binding solamente tra proprietà di tipo analogo. Nello specifico si distinguono le seguenti:

- Reale: come una temperatura o una percentuale di umidità (22.5 °C, 45%);
- Multistato: 0 = occupato, 1 = non occupato, 2 = bypass, 3 = standby;
- Digitale: acceso/spento, I|O;
- Testo: Text_123;
- Data: 22-10-2018.

È possibile collegare oggetti e proprietà appartenenti allo stesso server, come anche a server differenti. Nel primo caso i valori vengono aggiornati automaticamente nel programma senza necessità alcuna di configurazione da parte dell'operatore.

Dall'esempio in figura 4.5 si vede un esempio di questo caso: oltre ai valori di input e di output, si vedono nell'interfaccia di EBO delle frecce, le quali indicano la direzione di collegamento: quelle verso sinistra indicano un'associazione fatta al programma, mentre quelle verso destra un binding che proviene dal programma. È possibile anche scorgere frecce che indicano entrambe le direzioni, allora in quel caso l'associazione sarà a grafica e l'utente è in grado di visualizzare e modificare un valore dal grafico creato per l'occasione e trasferire le informazioni in ambedue i versi.

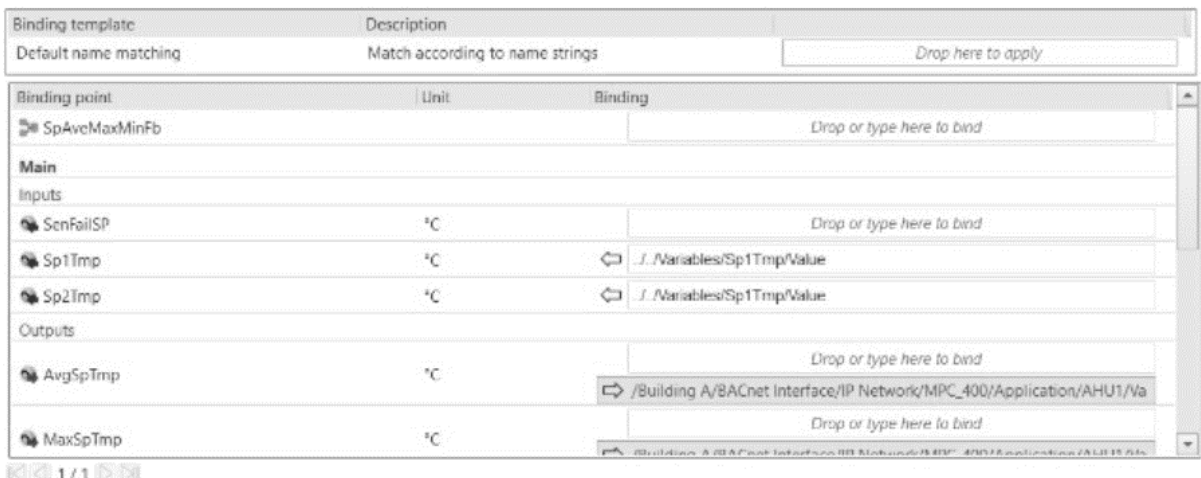


Figura 4.5: Esempio di binding di oggetto di invio e ricezione appartenenti a server differenti.

Nel caso in cui, piuttosto, sia doveroso operare associazioni tra server dissimili si rende indispensabile un'azione di configurazione delle attività di trasferimento. Nell'interfaccia del software, a tal proposito, comparirà il simbolo di un meccanismo a destra dei binding di questo genere (fig. 4.6), dal quale si può procedere con la programmazione dell'aggiornamento dei valori.

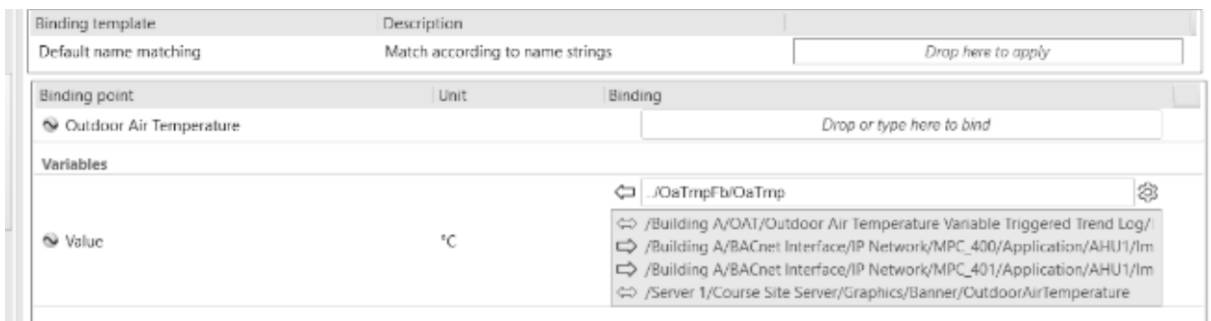


Figura 4.6: Esempio di binding di oggetti di invio e ricezione appartenenti al medesimo server.

Aprendo la finestra di dialogo riservata alle *Value Transfer Settings* si ha la possibilità di intervenire su vari valori (fig. 4.7). Il primo slot di modifica è destinato al tasso di trasferimento richiesto, il quale determina il ritmo di spostamento: nel caso in cui l'operatore scelga di non settare le impostazioni, il software aggiorna i dati ogni dieci secondi secondo logiche predefinite, ma è possibile regolare anche un intervallo manuale che varia da secondi, minuti, ore fino a interi giorni a seconda della corretta evenienza in questione.

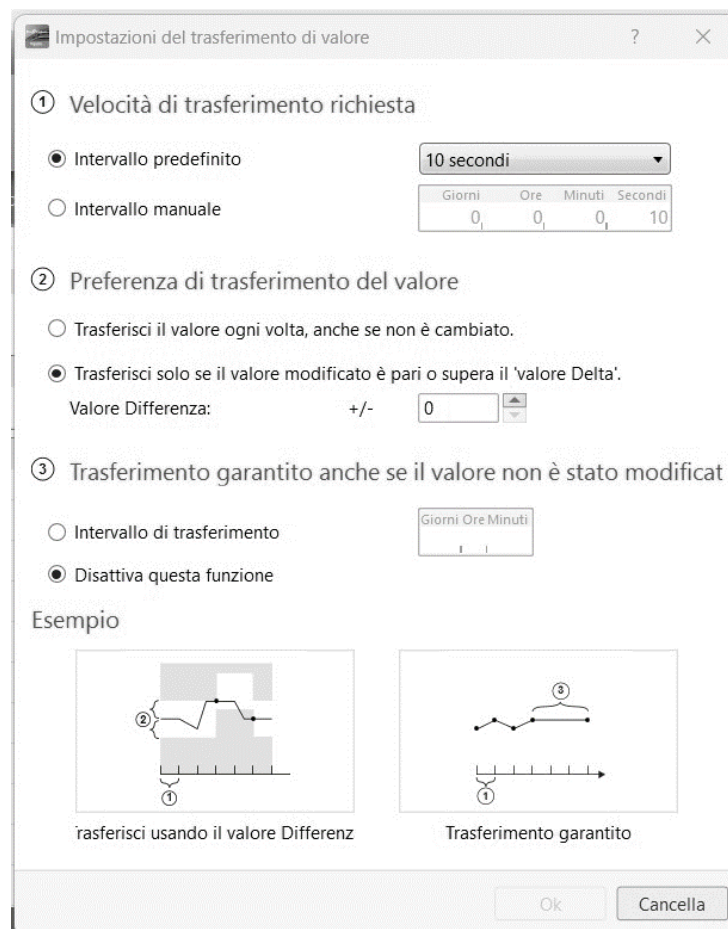


Figura 4.7: Finestra di dialogo dedicata alle impostazioni di trasferimento valori.

Seguendo ancora la figura 4.7, si procede poi con l'impostazione della preferenza di trasferimento del valore indagato; pertanto, si può scegliere se trasferirlo con cadenza regolare ad ogni intervallo scelto, anche se la quantità non è commutata, altrimenti trasmetterlo in caso di variazione di un determinato valore delta. Quest'ultimo è scelto convenientemente dall'operatore in questo frangente e l'aggiornamento verrà effettuato solo se la grandezza è superiore o inferiore a tale delta. Infine, indipendentemente dalle scelte fatte in precedenza e appena descritte, è consentito scegliere di attivare un intervallo di trasferimento garantito programmando minuti, ore e giorni.

È anche previsto un modello di binding predefinito che consente di automatizzare l'attività di associazione delle quantità, chiamato binding template. Grazie alla corrente impostazione diminuisce sensibilmente il numero di associazioni manuali e, di conseguenza, il volume di errori e imprecisioni derivanti da tale modalità di compilazione. Inoltre, è plausibile riutilizzare l'intero template per associazioni successive di egual tipo e associarlo a intere cartelle di file, velocizzando pertanto il processo.

2.2 Topologie di rete in EcoStruxure Building Operation

Si definisce una topologia di rete come il modello geometrico finalizzato a rappresentare le relazioni di connettività, sia essa fisica o logica, tra gli elementi costituenti, detti nodi, e la rete stessa. Il concetto di topologia si applica a qualsiasi tipo di rete di telecomunicazioni: telefonica, rete di computer, Internet.

Vanno innanzitutto distinte le due topologie, ovvero quella logica e quella fisica. Con la topologia logica si vuol descrivere un flusso di dati che fluisce attraverso una dislocazione spaziale già determinata, mentre quella fisica chiarisce come si configura tale dislocazione. A tal proposito, le opzioni sono diverse e prendono il nome solitamente dal modello geometrico che si viene a creare in seguito ai vari collegamenti tra nodi, si formano dunque topologie ad anello, a maglia, a bus, a stella, ad albero o miste dove cioè confluiscono più forme.

Le topologie di rete implementate in EBO, visualizzabili in figura 4.8, sono le seguenti:

- Star
- Daisy Chain
- Daisy Chain Ring

Nel primo caso si sta parlando di una topologia di rete in cui ciascuno dei terminali remoti o nodi è direttamente connesso al nodo centrale, le linee di bus corrono dal centro di una struttura a stella ai nodi periferici. In generale, tutte le linee sono del tipo point-to-point e vengono utilizzate per inviare un segnale, in una singola direzione, per una singola linea.

Man mano che il controller genera i segnali PWM³⁶, li invia sulle linee di trasmissione a ciascuno degli interruttori controllati, anche i valori rilevati, siano essi in forma analogica o digitale, vengono inviati uno per riga, ma in una direzione differente.

Anche quando i dati sono combinati condividendo la stessa capacità trasmissiva, però, questa pratica spreca le risorse utilizzate. Il numero di linee, infatti, è almeno uguale al numero di moduli di potenza e nel caso in cui la comunicazione in linea sia unidirezionale, il numero di linee sarebbe pari a due volte il numero di moduli di potenza. Un ulteriore svantaggio è che questo approccio rende l'interfaccia del controller dipendente dalla topologia del convertitore.

È tuttavia la tipologia di rete più affidabile tra quelle elencate, poiché il guasto ed il power off eventuali di un qualsiasi dispositivo non compromette il funzionamento di altri controller o dell'intera linea, ma piuttosto il nodo guasto rimane isolato, mentre gli altri continuano a funzionare attraverso la mediazione del nodo centrale. È inoltre una topologia di rete caratterizzata da un'espandibilità particolarmente agevole dato che è possibile il collegamento di più reti della stessa tipologia creando un collegamento a cascata, da cui si genera la rete a stella in espansione. Per ultimo, ogni dispositivo collegato tramite questo tipo di rete può inviare dati in qualsiasi momento al server centrale senza alcuna intermediazione.

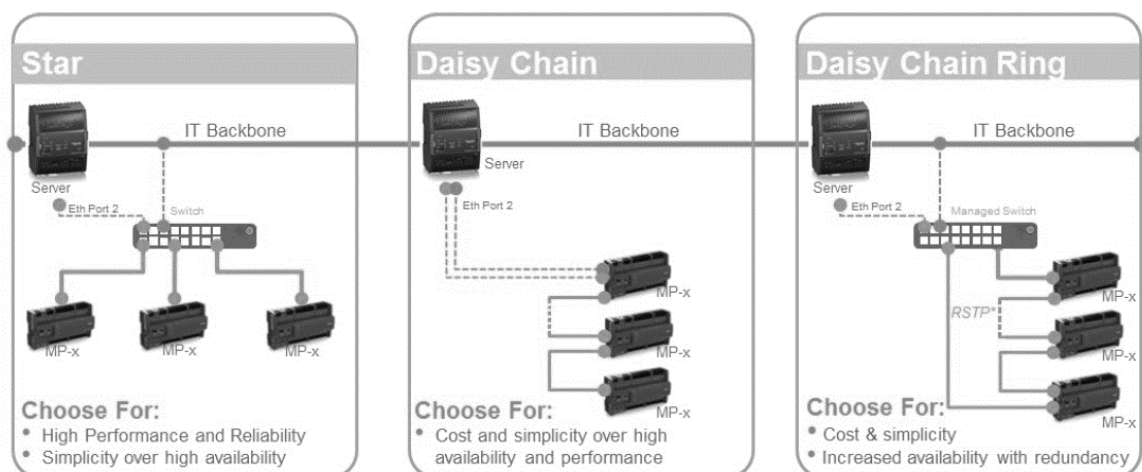


Figura 4.8: Le tre topologie di rete implementate in EBO.

In un collegamento del tipo daisy chain, un nodo di rete è collegato al successivo in una linea o in una catena. Nel primo caso, dove la topologia è lineare, il primo e l'ultimo nodo non sono collegati, mentre nel secondo, ad anello, il primo e l'ultimo nodo sono collegati e in questo caso si parla di daisy chain ring. Quest'ultima consente il passaggio bidirezionale dei dati,

³⁶ La modulazione PWM, Pulse Width Modulation, è un segnale capace di regolare la tensione in uscita a partire da una sorgente in corrente continua e, contemporaneamente, di ridurre notevolmente la potenza dissipata dal sistema elettrico. Il funzionamento di un circuito di questo tipo dipende da un interruttore posto tra sorgente e carico che esegue un'azione I|O ad alta velocità, generando un valore di tensione medio rispetto alla totale capacità del sistema.

mentre in una configurazione lineare un messaggio deve passare da una macchina all'altra in una singola direzione.

Generalmente meno versatile, una configurazione di rete lineare di questo tipo è simile a un circuito elettrico in serie, in cui un'interruzione influisce su altri elementi collegati e un nodo di rete compromesso può interrompere qualsiasi macchina oltre quel punto. Al contrario, una struttura ad anello può inviare dati in entrambe le direzioni, impedendo che il guasto di un nodo interrompa determinate parti della rete.

2.3 Estensioni file consentite da EcoStruxure Building Operation

Per impostazione predefinita, l'elenco delle estensioni di file consentite da EcoStruxure contiene le seguenti:

- avi
- dfx
- doc
- docx
- dwg
- gif
- htm
- html
- jpg
- mp4
- mpg
- odt
- PDF
- png
- rtf
- TXT
- vsd
- xls
- xlsx

La configurazione dell'elenco delle estensioni file consentite su un Enterprise, o Automation, Server viene distribuita a tutti i server di automazione in EcoStruxure. (Schneider Electric, 2021)

Se necessario, l'utente ha l'opportunità di aggiungere e di eliminare estensioni di file dall'elenco delle tipologie consentite. Se tale catalogazione è stata configurata e salvata, viene poi conservata quando si aggiorna la piattaforma.

Se un'estensione di file non è nell'elenco delle consentite e un utente esegue un'operazione di ripristino, l'operazione è correttamente svolta dalla piattaforma e il

documento in questione viene ripristinato, ma non è possibile aprirlo data la mancanza di associazione. Nel momento in cui l'estensione del documento in questione viene aggiunta all'elenco, è allora permesso aprire e salvare il file del modello ripristinato.

2.4 BACnet Interface in EcoStruxure Building Operation

EBO consente il controllo e la gestione centralizzati di più sistemi integrati da un'unica piattaforma. La spina dorsale dell'Edge Control di EBO sono i dispositivi server Smart Struxure, responsabili del monitoraggio e del controllo in tempo reale dei processi e dei sistemi autonomi. In questo caso i dispositivi Smart Struxure utilizzati sono server di automazione embedded³⁷ che eseguono funzionalità chiave, come la logica di controllo, la registrazione delle tendenze e la supervisione degli allarmi, e supportano la comunicazione e la connettività agli I/O e ai bus di campo. L'intelligenza distribuita della soluzione Smart Struxure garantisce la tolleranza ai guasti nel sistema e fornisce un'interfaccia utente completa tramite WorkStation e WebStation. Esistono due tipi di server Smart Struxure: Automation Server (AS-P) ed Enterprise Server (ES).

Un AS-P è un dispositivo che può fungere da server autonomo e anche controllare i moduli I/O e monitorare e gestire i dispositivi del bus di campo. In una piccola installazione, il dispositivo AS-P integrato funge da server autonomo, montato con i suoi moduli I/O in un ingombro ridotto. Nelle installazioni di medie e grandi dimensioni, la funzionalità è distribuita su più dispositivi server Smart Struxure che comunicano e scambiano dati su TCP/IP. Ne consegue che tutti i dispositivi BACnet nelle reti IP ed MS/TP possono scambiare dati tra di loro e dunque, devono ineluttabilmente avere un Instance Number diverso.

Come già accennato in precedenza e come si può vedere nell'immagine seguente, gli AS-P dispongono di numerose porte, le quali consentono al server di comunicare con un'ampia gamma di protocolli standard aperti comunemente utilizzati in BACS, in cui sono compresi ModBus RTU, LonWorks TP/FT e BACnet MS/TP, che semplificano l'accesso ad una pletera di dispositivi di campo (fig. 4.9). Inoltre, i server supportano la comunicazione con ModBus TCP, BACnet IP o servizi Web standard aperti sul cavo Ethernet.

In particolare, un generico AS-P dispone delle seguenti porte:

- Due porte Ethernet 10/100
- Due porte RS-485
- Una porta LonWorks TP/FT
- Una porta bus I/O integrata
- Una porta host USB
- Una porta per dispositivo USB

³⁷ È un sistema di elaborazione incorporato, che si integra all'interno dell'oggetto o del sistema informatico in cui è inserito per monitorarne e gestirne alcune funzionalità.

Le due porte Ethernet sono collegate a uno switch Ethernet integrato, dove una porta deve essere collegata alla rete del sito e l'altra può essere utilizzata per collegare una singola WorkStation o WebStation, un'unità ModBus TCP o un dispositivo BACnet/IP, ma non un altro server Smart Struxure. La porta del dispositivo USB consente di aggiornare e interagire con AS-P utilizzando il Device Administrator. La porta host USB può essere utilizzata per fornire alimentazione e comunicazioni per AD.

Un Enterprise Server, ES, è la versione dell'applicazione Windows di un server Smart Struxure che raccoglie dati a livello di sito per l'aggregazione e l'archiviazione, ma è sufficientemente flessibile per eseguire applicazioni autonome. L'Enterprise Server funge anche da singolo punto di amministrazione tramite WorkStation o WebStation per la soluzione Smart Struxure ed è il punto centrale da cui gli utenti possono configurare, controllare e monitorare l'intero sistema.

L'ES può eseguire più programmi di controllo utilizzando una varietà di protocolli. Può gestire allarmi, utenti, pianificazioni e registri delle tendenze. I dati dall'Enterprise Server possono essere consegnati direttamente all'utente o ad altri server Smart Struxure e dispositivi di campo in tutto il sito o l'azienda.

A differenza del caso precedente però, per rendere un ES un BACnet Gateway occorre una programmazione aggiuntiva e, soprattutto, non espone automaticamente su BACnet gli oggetti al suo interno, anzi è necessario creare tutti gli oggetti BACnet essenziali al progetto e collegarli via binding.

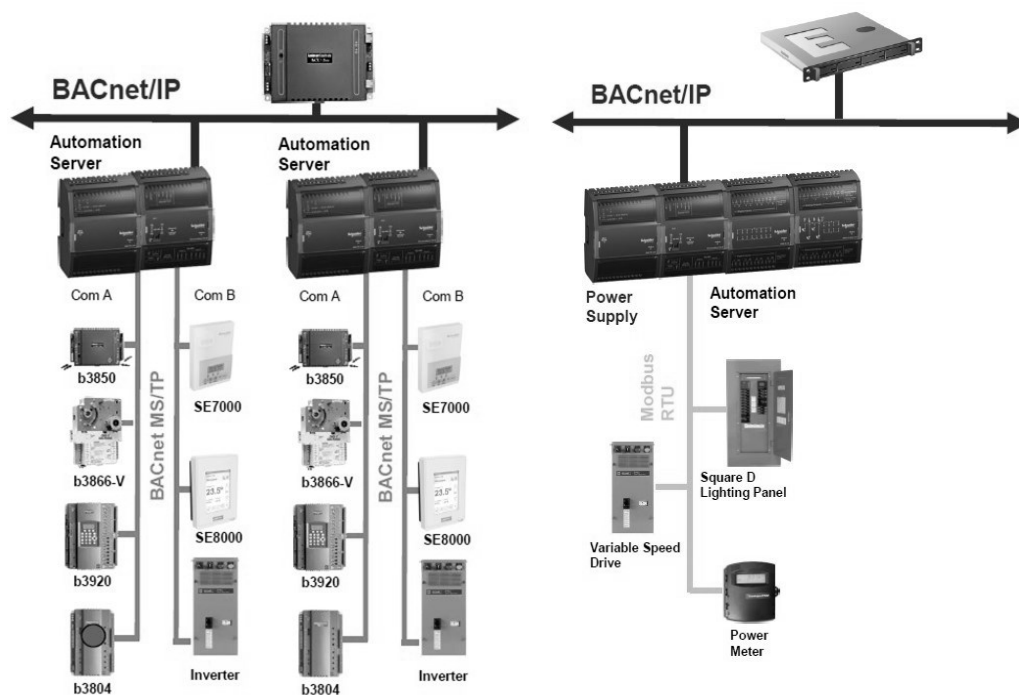


Figura 4.9: BACnet/IP con Automation Server (sinistra) ed Enterprise Server (destra).

2.4.1 Creare un'interfaccia BACnet in EBO

In EBO, che si voglia utilizzare un Automation Server oppure un Enterprise Server, è necessario creare per prima cosa una BACnet Interface. Per far ciò, nella facciata principale del programma, selezionando il server che si sta utilizzando, si procede così:

- New → Interface → BACnet Interface

A partire da questo momento inizia la sua programmazione come fosse un oggetto BACnet con la scelta delle impostazioni di base BACnet: Instance ID, BACnet Name, Network ID. Se non ci sono particolari richieste si può ricorrere alla generazione automatica di default di tali valori, altrimenti l'utente può personalizzarli a proprio piacimento. Si ricorda che l'Instance ID è il valore con cui il Server comunicherà su BACnet e che se si imposta manualmente deve inevitabilmente essere unico su tutte le reti BACnet, il BACnet Name è il nome del device su BACnet, che se non impostato è, ad esempio, AS_InstanceID e, infine, la Network ID è uguale ad 1, il numero standard della rete BACnet IP ed è modificabile.

Fatto ciò, la BACnet Interface è creata e viene visualizzata una finestra di dialogo come quella riportata in figura 4.10. Si vede come, entrando nelle proprietà avanzate dell'oggetto, l'unico parametro ancora modificabile sia il BACnet Name, attività possibile in qualsiasi momento del progetto, e contraddistinguibile da un carattere di scrittura color nero, mentre gli altri sono bloccati e di colore grigio.

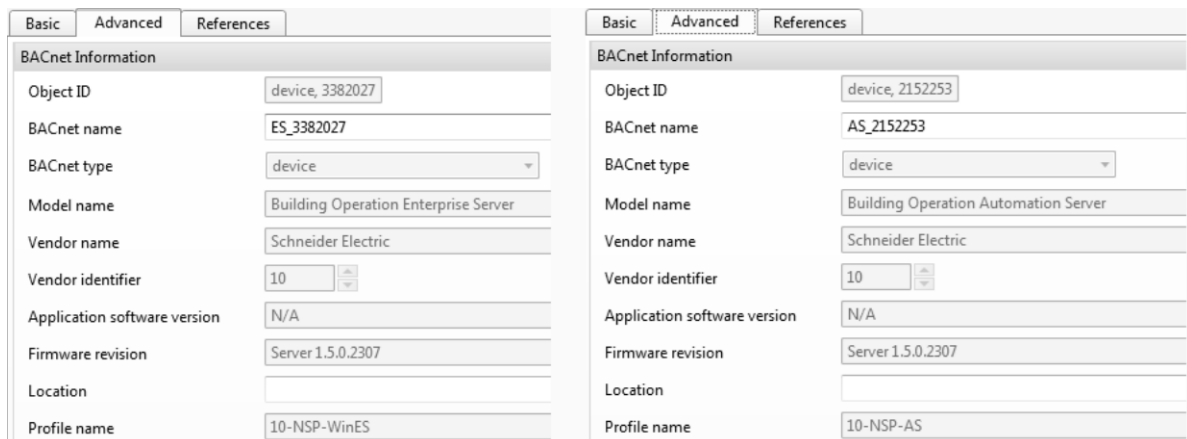


Figura 4.10: Finestra di dialogo in EBO dedicata alle proprietà di una BACnet Interface, a sinistra il caso con un Enterprise Server e a destra quello con un Automation Server.

Per modificare, invece, il valore della Network ID, occorre cliccare nella cartella indirizzata all'IP Network presente nel sistema ad albero riassuntivo a sinistra. Da qui è effettuabile la navigazione nelle proprietà della rete, siano esse di base o avanzate, e mutare le grandezze desiderate (fig. 4.11).

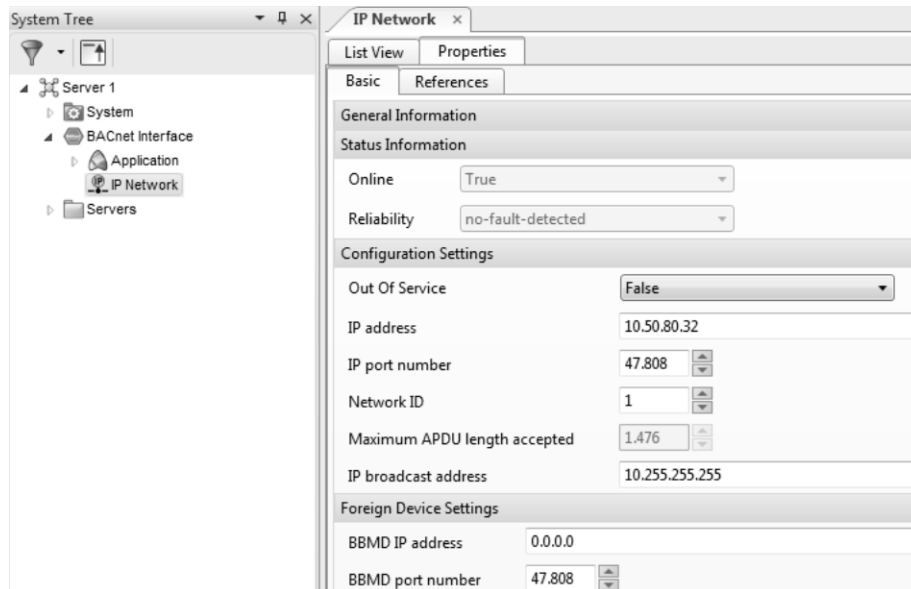


Figura 4.11: Finestra di dialogo in EBO dedicata alle proprietà dell'IP Network.

Dalla versione 1.5 di EBO sono attuabili più di una IP Network, a patto che queste abbiano Network Name, ID e Port Number diversi. Di default, in EBO, l'ID port number è 47808, anche scritta come "BAC0" nel linguaggio esadecimale hex. Tale concetto, assieme a quello di Device IP, sono concatenati e determinano il MAC Address, il quale identifica univocamente ogni nodo all'interno di reti comunicanti, indipendentemente dal mezzo fisico di trasporto.

2.4.2 Creare un dispositivo BACnet in EBO

Per creare in un server i dispositivi necessari al funzionamento del progetto desiderato si procede come con la creazione di una nuova interfaccia. In particolare, si interviene sull'IP Network, cliccando con il tasto destro del mouse e creando un nuovo dispositivo. Ad esempio, per creare un generico controller della serie SE8000:

- IP Network → New → Serie SE800 → SE8000 (generico)

Il device creato può essere un generico device BACnet oppure un dispositivo proprietario Schneider. Nel primo caso il dispositivo è interamente da programmare da parte dell'utente, specificando ogni oggetto BACnet al suo interno, seguendo la procedura descritta al 2.3.3 del medesimo capitolo; nel secondo invece, il dispositivo è creato dall'azienda per essere compatibile con l'interfaccia utente del software EcoStruxure Building Operation, dunque già arricchito di tutte le informazioni essenziali. Si veda la figura 4.12.

In EBO si può scegliere tra i dispositivi Schneider della serie SE8000 o della serie SmartX, per maggiori dettagli e per conoscere le specifiche tecniche si rimanda al sito ufficiale di Schneider Electric³⁸.

³⁸ Catalogo dispositivi Serie SE8000: <https://www.se.com/it/it/download/document/LEESCAT322DI/>
Ricerca dispositivi della serie SmartX: <https://www.se.com/it/it/search/SmartX%2520controller>

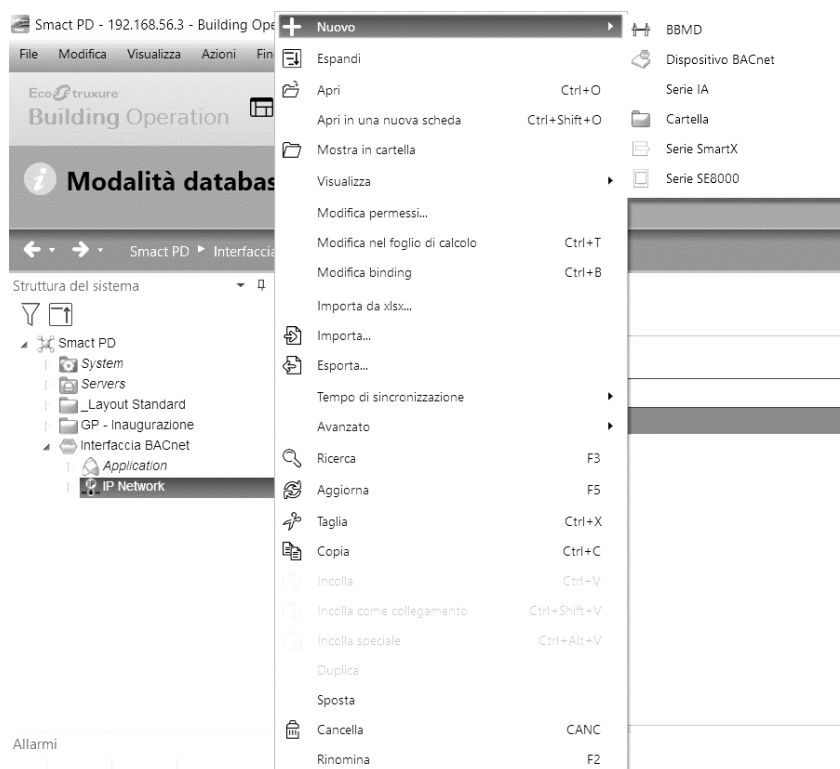


Figura 4.12: Finestra di dialogo in EBO dedicata alla creazione di dispositivi BACnet.

Per ogni dispositivo creato sono poi visualizzabili le proprietà di base, le proprietà avanzate e i riferimenti all'interno del progetto che il tale ha con altre entità presenti. Come anteriormente accennato, ogni strumento proprio di Schneider Electric ha, al suo interno, cartelle specifiche che facilitano l'utilizzo di EBO, mentre un dispositivo generico risulta vuoto.

2.4.3 Creare un oggetto BACnet in EBO

Per creare gli oggetti BACnet desiderati da associare ad un dispositivo BACnet, si accede alla cartella *Application*, dalla quale, seguendo il procedimento per la creazione di un nuovo oggetto, si aprirà l'elenco che li comprende tutti (fig. 4.13).

Per mappare ingressi e uscite del server su BACnet è opportuno utilizzare i valori elencati e spiegati di seguito, associati rispettivamente alle loro grandezze:

- BACnet Analog Input → ingresso analogico → 21 °C
- BACnet Analog Output → uscita analogica → 75 %
- BACnet Digital Input → ingresso digitale → True/False
- BACnet Digital Output → uscita digitale → On/Off
- BACnet Multistate Input → ingresso multistato → On/Off/Manual
- BACnet Multistate Output → uscita multistato → Stage1/Stage2/Off

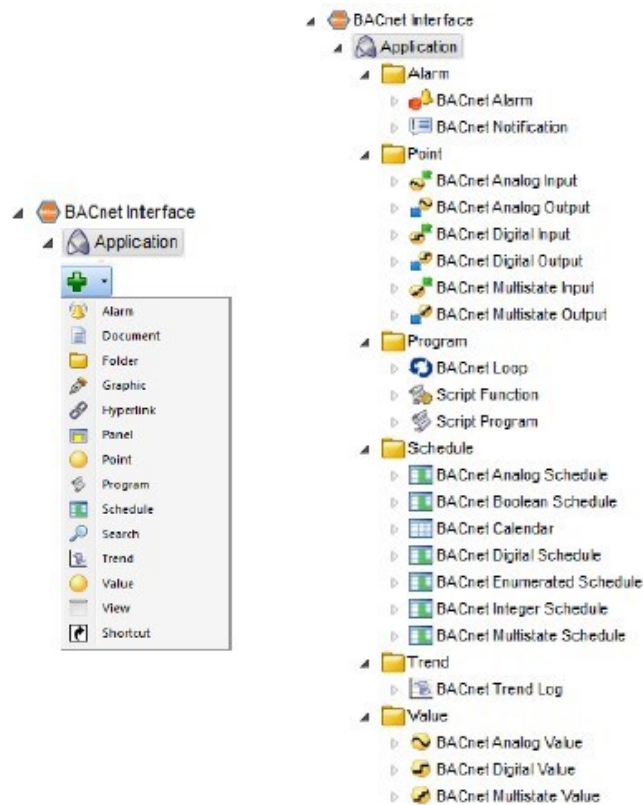


Figura 4.13: Elenco ad albero degli oggetti BACnet in EBO.

Nel caso generico, per creare un oggetto BACnet si procede come visto in precedenza con le impostazioni di base. Ad esempio, per un BACnet Analog Input:

- Application → New → Point → BACnet Analog Input

Seguono da qui i passaggi per settare a dovere l'elemento, scegliendo se ricorrere alla generazione automatica dei valori o a quella manuale. Un oggetto come questo in esame ha bisogno di un'associazione ad un ingresso fisico dell'AS/ES da cui leggere il valore da restituire, è pertanto fondamentale procedere con il binding nel campo "Associated Value"; se ciò non viene eseguito, il punto sarà sempre *Out of Service*, dunque non attivo e fuori servizio. Un oggetto in input è collegato ad un *Value*, mentre uno in output con un *RequestedValue*.

Gli Automation server e gli Enterprise server supportano entrambi il servizio COV, ovvero di Change of Value, sugli oggetti, ciò vuol dire che una volta che l'utente si registra sul server otterrà un aggiornamento di valore a seconda del *COV Increment* scelto.

3 CASO STUDIO APPLICATIVO

Il caso studio applicativo di seguito presentato nasce con lo scopo di testare l'integrazione sistematica di un unico file completo IFC nell'ambiente BACS, riducendo gli oneri, semplificando le procedure ed eliminando le ambiguità anagrafiche derivabili dall'elaborazione di più modelli in discipline diverse. Rappresenta, pertanto, il punto d'incontro delle due realtà analizzate fino ad ora.

Il progetto, sviluppato in Autodesk Revit2023, è composto da una stanza popolata di tutti quegli oggetti finalizzati all'accensione automatizzata di una lampadina: un comando luce, un sensore di rilevazione presenza, un attuatore e la lampadina stessa, il tutto collegato ad un quadro elettrico (fig. 4.14).

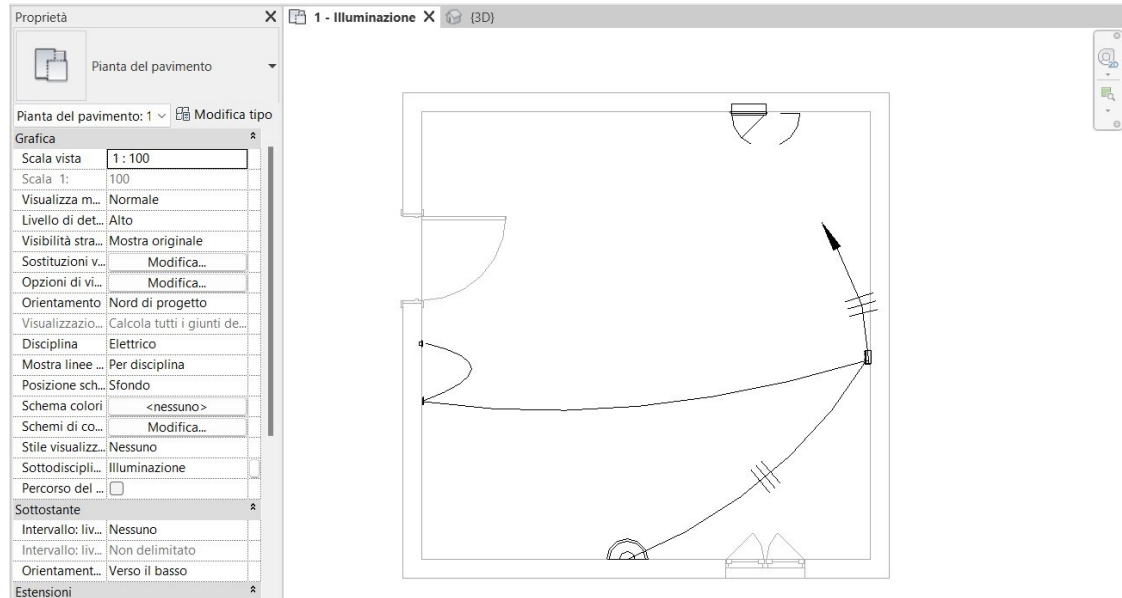


Figura 4.14: Pianta del modello BIM in Revit.

I dispositivi usati sono tutti prodotti da Schneider Electric, di seguito qui elencati con i riferimenti diretti alle proprie schede tecniche pubblicate nel sito ufficiale:

- Sensore: SE - SmartX - Optimum Housing - Touch Screen with Off-screen Control³⁹ (fig. 4.15)



Figura 4.15: Sensore SmartX di Schneider Electric.

³⁹ <https://www.se.com/ww/en/product/SXWSC2PSELXB/smartx-cover-plate-for-sensor-base-touchscreen-with-2-buttons-lighting-pir-optimum-black-cover/>

- Comando luce: KNX - System M - Push-button module - 1-gang⁴⁰ (fig. 4.16);

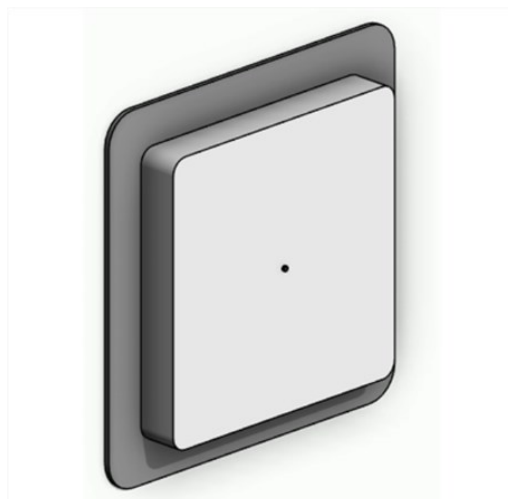


Figura 4.16: Comando luce SpaceLogic MTN625199 di Schneider Electric.

- Attuatore: Switch actuator REG-K/2x230/10 with manual mode, MTN649202⁴¹ (fig 4.17);

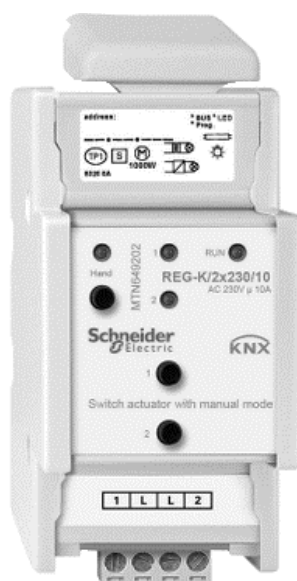


Figura 4.17: Attuatore SpaceLogic MTN649202 di Schneider Electric

- Quadro elettrico: Schneider-Electric Pragma flush mounted LV distribution board⁴² (fig. 4.18).

⁴⁰ <https://www.se.com/ww/en/product/MTN625199/knx-pushbutton-module-1gang-system-m/>

⁴¹ <https://www.se.com/ww/en/product/MTN649202/switch-actuator-regk-2x230-10-with-manual-mode-light-grey/>

⁴² <https://www.se.com/ww/en/product/PRA06118/pragma-interface-for-surface-enclosure-1-x-13-or-18-modules-without-door/>



Figura 4.18: Quadro elettrico e relativa interfaccia Pragma di Schneider Electric.

Per ogni dispositivo inserito è stata poi studiata l'appartenenza allo standard IFC, popolandoli delle entità e delle adeguate proprietà (Tabella 11). È stato scelto di implementare lo standard IFC4 Add2, ultimo ufficiale pubblicato da BuildingSMART.

Tabella 11: Dispositivi del modello con anagrafica IFC associata.

Dispositivi	Name	Classe IFC	TypeEnum	IfcGuid
Comando Luce	KNX - System M - Push-button module - 1-gang	7.4.3.47 IfcSwitchingDevice	7.4.2.24 TOGGLESWITCH	1S1zQCICPEhu732E3J_WXd
Sensore	SE - SmartX - Optimum Housing - Touch Screen with Off-screen Control	7.2.3.9 IfcSensor	7.2.2.5 MOVEMENTSENSOR	1S1zQCICPEhu732E3J_h2v
Attuatore	KNX - Actuator - Switch - 10 A - Manual mode	7.2.3.1 IfcActuator	7.2.2.1 ELECTRICACTUATOR	3CL4rYpDjBQOItXZgc5ZPT
Lampadina	M_Applique - Illuminazione indiretta 120V	7.4.3.33 IfcLightFixture	7.4.2.17 POINTSOURCE	2b00\$1j2X1deD5QdRz5yMh
Quadro elettrico	Schneider-Electric Pragma flush mounted LV distribution board	7.4.3.13 IfcElectricDistributionBoard	7.4.2.7 SWITCHBOARD	1S1zQCICPEhu732E3J_a2D

Per ogni classe IFC vengono ora descritte le specifiche per consentire una corretta comprensione del percorso di sperimentazione; dunque, si procede con la mappatura dei dispositivi inseriti, sottolineando la classe IFC di appartenenza, il TypeEnum e l'IfcGuid. Quest'ultimo è un codice unico generato dai software e assegnato ad ogni oggetto, indipendentemente dalla sua natura e non attribuibile manualmente dall'utente. La sua presenza rende più semplice il coordinamento del modello specialistico successivo, poiché i componenti sono distintamente riconoscibili.

Innanzitutto, è necessario specificare i domini di appartenenza dei vari device presenti, i quali sono due:

- 7.2 *IfcBuildingControlsDomain*, al quale appartengono sensore e attuatore;
- 7.4 *IfcElectricalDomain*, al quale appartengono il comando luce, la lampadina e il quadro elettrico.

Seguiranno poi le analisi delle loro classi di appartenenza, con i vari TypeEnum ad essi

associati, specificati in Tabella 11.

3.1 Classificazione IFC degli elementi del modello

3.1.1 *L'IfcBuildingControlsDomain e l'IfcElectricalDomain*

Il primo è già stato trattato al 1.1.1 del Capitolo Terzo.

L'IfcElectricalDomain, invece, anche lui facente parte del *Domain Layer*, definisce i concetti di sistemi cablati in cui il cablaggio trasporta alimentazione elettrica, dati, segnali telefonici o altre forme di trasmissione via cavo. Definisce, inoltre, vari dispositivi che sono collegati tramite cablaggio, protezione dei dispositivi elettrici, fornitura e concetti di apparecchi di illuminazione all'interno degli edifici, il cablaggio stesso e metodi per sostenere e portare i cavi. L'ambito di cui si occupa è composto come segue:

- sistemi cablati per la distribuzione di energia e illuminazione, dati, telecomunicazioni, sicurezza, segnalazione, controllo e scopi audiovisivi,
- apparecchiature utilizzate all'interno di tali sistemi,
- collegamento di apparecchiature a circuiti, punti di distribuzione ecc.,
- funzionamento a regime degli impianti elettrici,
- apparecchi di illuminazione, i loro tipi e la fornitura di tali informazioni necessarie per consentire calcoli di illuminazione che forniscano un illuminamento fisicamente accurato.

Si noti che per gli impianti elettrici lo schema ha il particolare campo di applicazione degli impianti elettrici in bassa tensione in conformità con le definizioni ISO/IEC. Altri sistemi elettrici, inclusi quelli a bassissima, media e alta tensione, possono essere specificati utilizzando questo schema, ma le disposizioni di tali sistemi non sono ancora state specificatamente previste.

Allo stesso modo, mentre lo schema del dominio elettrico può essere utilizzato per scopi dati, telecomunicazioni, sicurezza, segnalazione, controllo e audiovisivi, le disposizioni specifiche richieste a questi tipi di sistemi non sono ancora completamente catturate ed elaborate all'interno del modello.

Lo schema *IfcElectricalDomain* supporta idee che includono tipi di:

- apparecchio audiovisivo;
- raccordi per portacavi (per conduit, passerella portacavi, canalina portacavi e scala);
- apparecchio elettrico;
- motore elettrico;
- quadri di distribuzione;

- generatore;
- scatola di giunzione;
- lampadario;
- lampada;
- presa;
- dispositivo di protezione;
- unità di sgancio del dispositivo di protezione;
- dispositivo di commutazione;
- trasformatore.

I seguenti elementi sono nello stesso ambito, ma non elaborati:

- impianti di media e alta tensione oltre 1000V AC e 1500V DC;
- sistemi con tensione inferiore a 12 volt (AC/DC);
- sistemi audiovisivi;
- sistemi di telecomunicazioni;
- sistemi di dati;
- cablaggio per sistemi di segnalazione e controllo;
- router per cavi (passerelle per cavi);
- cablaggi di reti informatiche e dispositivi utilizzati nelle reti.

Quanto segue, invece, è considerato ancor oggi al di fuori dell'ambito dello schema in esame:

- fornitura di servizi elettrici di pubblica utilità;
- stati non stazionari/transitori di funzionamento degli impianti elettrici;
- dispositivi di sicurezza associati alla sicurezza degli impianti elettrici;
- segnali di comunicazione tra dispositivi di automazione o sistemi bus;
- sensori (che sono trattati nello schema di dominio *IfcBuildingControl*);
- sistemi di specchi a faretto in cui un faretto è diretto verso uno specchio distante che distribuisce l'energia luminosa (diverso da un gruppo di oggetti definiti separatamente);
- illuminazione per scopi specialistici (ad esempio, palcoscenico, pittura).

Passando ora alla descrizione dei device inseriti nel modello, si descrivono nell'ordine: IfcSensor, IfcActuator, IfcSwitchingDevice, IfcLightFixture, IfcDistributionboard.

3.1.2 IfcSensor

Trattato al 1.2.9 del Capitolo Terzo.

3.1.3 IfcActuator

Un attuatore è un dispositivo meccanico deputato alla movimentazione o al controllo di un meccanismo, che agisce prendendo l'energia generalmente creata da fonti come aria, elettricità o da un liquido e convertendola in un movimento. L'*IfcActuatorTypeEnum* definisce l'intervallo di diversi tipi di attuatore che possono essere specificati.

3.1.3.1 IfcActuatorTypeEnum

- ELECTRICACTUATOR: Dispositivo che attua un elemento di controllo elettricamente.
- HANDOPERATEDACTUATOR: Dispositivo che attua un elemento di controllo manualmente.
- HYDRAULICACTUATOR: Dispositivo che attua un elemento di controllo idraulicamente.
- PNEUMATICACTUATOR: Dispositivo che attua un elemento di controllo pneumaticamente (ad aria).
- THERMOSTATICACTUATOR: Dispositivo che attua un elemento di controllo termostaticamente.
- USERDEFINED: Tipo definito dall'utente.
- NOTDEFINED: Tipo non definito.

3.1.4 IfcSwitchingDevice

Questa classe fornisce una definizione di "interruttore", ossia un dispositivo usato in un sistema di distribuzione cablato per controllare o modulare il flusso di elettricità. Sono inclusi gli interruttori utilizzati per l'energia elettrica, per le comunicazioni, per i segnali audiovisivi, o per altri tipi di sistemi di distribuzione, in funzione da quanto determinato dalle porte delle entità connesse. L'*IfcSwitchingDeviceTypeEnum* definisce l'intervallo di diversi tipi di switch che possono essere specificati.

3.1.4.1 IfcSwitchingDeviceTypeEnum

- CONTACTOR: Dispositivo elettrico usato per controllare il flusso di potenza in un

circuito acceso o spento. Chiamato “contattore” o “teleruttore”.

- DIMMERSWITCH: Dispositivo capace di assumere posizioni variabili e di regolare la potenza elettrica o altre impostazioni (secondo il tipo di porta commutata). Comunemente chiamato “dimmer”.
- EMERGENCYSTOP: Dispositivo di arresto di emergenza, che agisce per rimuovere il più rapidamente possibile qualsiasi pericolo che possa essere sorto inaspettatamente.
- KEYPAD: Descrive un insieme di pulsanti o interruttori, ognuno potenzialmente applicabile a un dispositivo diverso, anche se letteralmente è traducibile come “tastierino”.
- MOMENTARYSWITCH: “Interruttore momentaneo” è un tipo di interruttore che non ha una posizione, ma che può innescare un'azione.
- SELECTORSWITCH: “Selettore” è un tipo di interruttore che ha più posizioni e può cambiare la sorgente della corrente, il livello di potenza o altre impostazioni (secondo il tipo di porta commutata).
- STARTER: Interruttore che in posizione chiusa controlla l'applicazione di potenza a un dispositivo elettrico, o “interruttore di avviamento”.
- SWITCHDISCONNECTOR: Interruttore che in posizione aperta soddisfa i requisiti di isolamento specificati per un “sezionatore”.
- TOGGLESWITCH: Interruttore a levetta, che può assumere due posizioni e conseguentemente abilitare o isolare l'alimentazione elettrica o altre impostazioni (secondo il tipo di porta commutata), chiamato anche “interruttore bistabile”.
- RELAY: Dispositivo progettato per produrre cambiamenti predeterminati e improvvisi in uno o più circuiti elettrici di uscita, quando certe condizioni sono soddisfatte nei circuiti elettrici di ingresso che controllano il dispositivo (come definito da IEC 60050 151-13-31).
- START_AND_STOP_EQUIPMENT: Interruttore che chiude e apre alternativamente uno o più circuiti elettrici, comunemente posizionato a bordo macchina per gestirne l'avvio o l'arresto.
- USERDEFINED: Tipo definito dall'utente.
- NOTDEFINED: Tipo non definito.

3.1.5 *IfcLightFixture*

Si definisce con *LightFixture* un corpo illuminante, ovvero un contenitore progettato allo

scopo di alloggiare una o più lampade ed eventualmente dispositivi che ne controllano, limitano o variano l'emissione. L'*IfcLightFixtureTypeEnum* definisce i diversi tipi di dispositivi di illuminazione.

3.1.5.1 IfcLightFixtureTypeEnum

- POINTSOURCE: traducibile come “sorgente puntiforme”. Si tratta di un apparecchio illuminante che si considera avere un'area trascurabile e che emette luce con intensità approssimativamente uguale in tutte le direzioni. Un esempio ne sono quei corpi che contengono una lampadina al tungsteno o alogena.
- DIRECTIONSOURCE: Traducibile come “sorgente direzionale”. Si tratta di un corpo illuminante che è considerato avere una lunghezza o una superficie da cui emette luce in una direzione. Un esempio ne sono quei corpi che contengono una o più lampadine a fluorescenza.
- SECURITYLIGHTING: Dispositivo luminoso che ha lo scopo specifico di dirigere, in caso di emergenza, gli occupanti di un edificio verso un luogo sicuro, come un segnale di uscita illuminato o una luce di emergenza.
- USERDEFINED: Tipo definito dall'utente.
- NOTDEFINED: Tipo non definito.

3.1.6 IfcElectricDistributionBoard

Un quadro di distribuzione è definibile come un regolatore di flusso in cui le istanze di dispositivi elettrici o di comunicazione sono riunite in un unico luogo per un particolare scopo. Il quadro fornisce alloggiamento per gli elementi di distribuzione collegati, affinché possano essere visualizzati e azionati da un unico punto. Ogni elemento collegato può avere la sua rappresentazione geometrica e la sua posizione. L'*IfcElectricDistributionBoardTypeEnum* definisce diversi tipi e le funzioni dei quadri elettrici di distribuzione.

3.1.6.1 IfcElectricDistibutionBoardTypeEnum

- CONSUMERUNIT: Contatore, punto di distribuzione dell'alimentazione elettrica in entrata dotato di dispositivi di protezione, tipicamente collocato in locali domestici.
- DISTRIBUTIONBOARD: Quadro elettrico in cui vengono effettuate le connessioni per la distribuzione dei circuiti elettrici di solito attraverso dispositivi di protezione (ad esempio quadri di zona).
- DISTRIBUTIONFRAME: Permutatore, quadro impiegato per interconnettere e gestire il cablaggio strutturato tra le apparecchiature attive e l'utenza. Può essere composto da più schede di distribuzione e altri componenti.

- MOTORCONTROLCENTER: Punto di distribuzione in cui si trovano i dispositivi di avviamento e di controllo dei principali elementi degli impianti meccanici.
- SWITCHBOARD: Punto di distribuzione in cui sono collocati i dispositivi di interruzione. Si riferisce a quadri di grandi dimensioni e che possono alimentare altri quadri di zona (ad esempio i QGBT, Quadri Generali di Bassa Tensione).
- USERDEFINED: Tipo definito dall'utente.
- NOTDEFINED: Tipo non definito.

3.2 Mappatura ed esportazione del modello IFC

Una volta compresa la classificazione IFC degli elementi presenti nel modello BIM, si associano ad ognuno di essi, tramite mappatura ifc, la classe e il tipo corrispondente in Revit, al fine di popolare il progetto nel modo più completo possibile. Questo sarà fondamentale per procedere con l'esportazione del modello IFC, il quale sarà successivamente implementato in Visual Studio Code per avanzare con la sperimentazione in atto.

A partire dalla versione di Revit2023, la mappatura ifc delle entità presenti in un progetto viene semplificata rispetto alle versioni precedenti. Il programma, infatti, ora contiene nativamente i parametri necessari e, pertanto, non è più indispensabile creare nuovi parametri condivisi "IfcExportAs" per le classi IFC e "IfcExportType" per i tipi, e classificarli come parametri IFC.

Per mappare un sensore, ad esempio, si procede come ora indicato:

- Selezionare il sensore che si vuole specificare e cliccare su "modifica tipo";
Tra i parametri di tipo, scendendo, si trovano i "Parametri IFC" i quali appariranno vuoti, sarà specificato solamente l'IfcGUID e il tipo di esportazione di Default (fig. 4.19);

Parametri IFC	
Tipo IFC predefinito	
Esporta tipo in formato IFC con nome	
Esporta tipo in IFC	Default
Tipo IfcGUID	1S1zQClcPEhu732E3J_h2v

Figura 4.19: Parametri IFC in Revit da definire.

- Cliccare su "esporta tipo in formato IFC con nome";
Selezionare la classe desiderata nell'elenco di sinistra e il tipo desiderato nell'elenco di destra mostrati in figura 4.20, nel caso in esame si tratta di un sensore di rilevamento presenza e dunque si seleziona "IfcSensorType" e successivamente "MOVEMENTSENSOR";

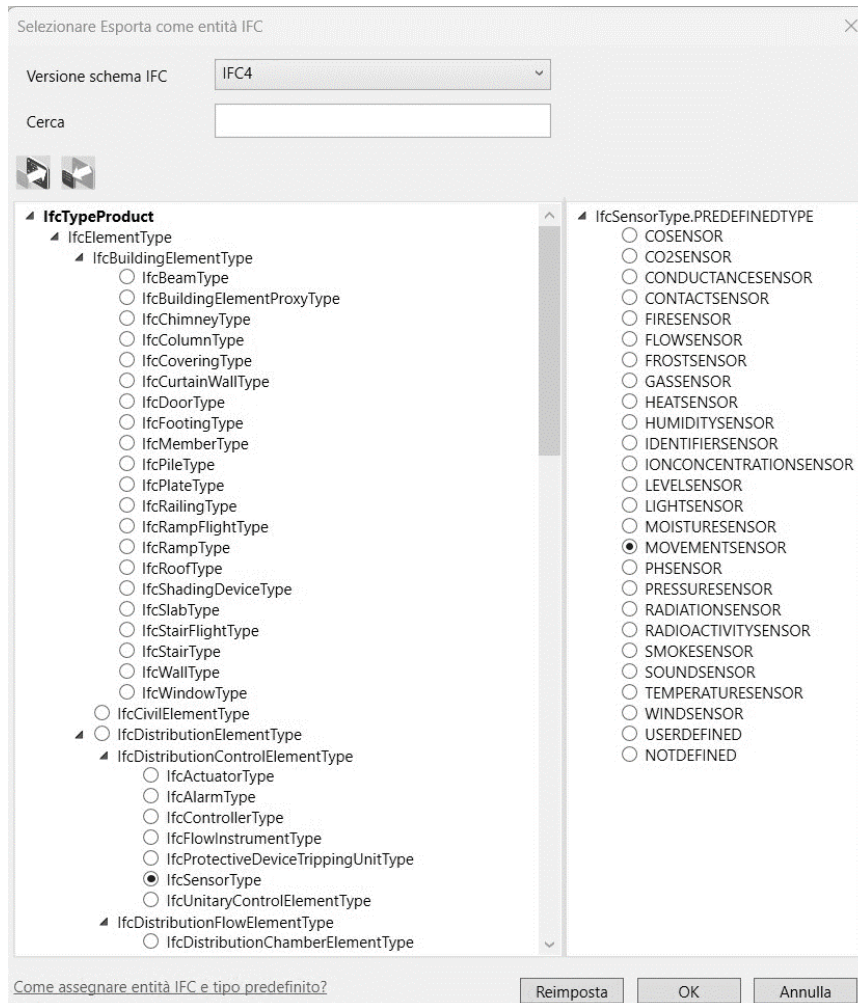


Figura 4.20: Processo di mappatura dei parametri IFC in Revit2023.

- I parametri appariranno ora popolati come visibile in figura 4.21.

Parametri IFC	
Tipo IFC predefinito	MOVEMENTSENSOR
Esporta tipo in formato IFC con nome	IfcSensorType
Esporta tipo in IFC	Default
Tipo IfcGUID	1S1zQC1cPEhu732E3J_h2v

Figura 4.21: Parametri IFC in Revit definiti.

La stessa procedura è da ripetere per ogni elemento presente nel modello. Nel caso in esame la mappatura IFC si è concentrata puntualmente per gli elementi appartenenti alla disciplina elettrica, come descritto in Tabella 11, ma è possibile per ogni categoria di oggetto, come da standard IFC.

Fatto ciò, si continua con l'esportazione nel progetto IFC in Revit2023 seguendo il suddetto percorso:

- File → Esporta → IFC

Qui si aprirà una finestra di dialogo che permette all'utente di determinare il nome del file IFC da esportare con il percorso di salvataggio e la configurazione IFC desiderata, scegliendo tra le versioni standard di IFC, come mostrato in figura 4.22.

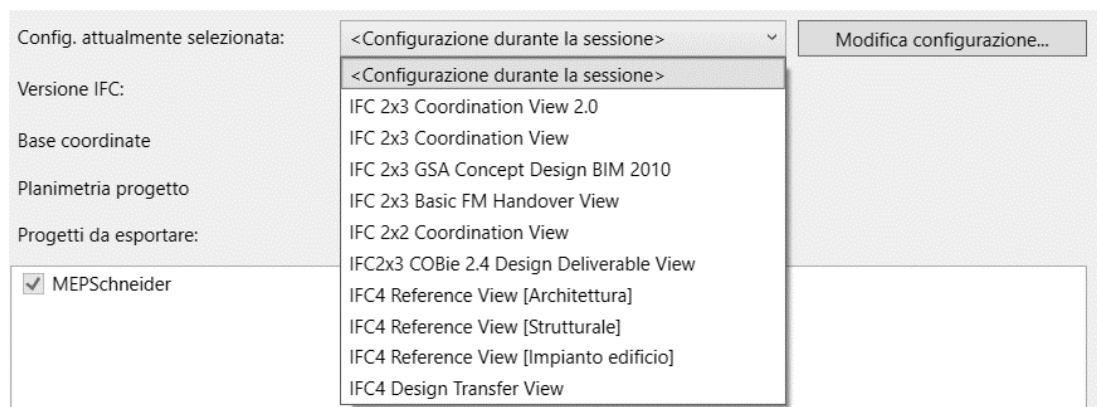


Figura 4.22: Versioni dello standard IFC selezionabili nell'esportazione di un file IFC.

Dopo di che, l'utente può intervenire modificando alcuni parametri come i livelli di dettaglio, i riferimenti geografici determinando le coordinate del progetto, gli elementi da esportare nella vista specifica e i gruppi di elementi. Popolato il progetto di tutte le informazioni qui descritte, è possibile procedere con l'esportazione cliccando appunto su "Esporta".

3.3 Creazione del modello in EcoStruxure Building Operation

Una volta progettato il modello BIM in Revit2023, viene creato il medesimo progetto nel software di Schneider Electric; dunque, si avanza seguendo la metodologia proposta al 2.4 e seguenti sottoparagrafi espressi in questo capitolo, la quale viene ora descritta nel dettaglio del caso in esame per permettere una maggiore comprensione. Si omette la creazione dell'interfaccia BACnet in quanto la procedura è la medesima vista nel 2.4.1.

3.3.1 Creazione dell'attuatore

All'interno dell'interfaccia BACnet creata, entrando nell'IP Network, si è creato un nuovo dispositivo secondo quanto esplicito al 2.4.2.

Si è scelto un dispositivo Schneider della serie SmartX, elencati di seguito in figura 4.23, in particolare il device RP-C-12A. Quest'ultimo rappresenta il controller SpaceLogic RP-C, ossia un controller da campo BACnet IP per ambienti, completamente programmabile, per applicazioni HVAC e soluzioni Connected Room. È adoperabile come controller BACnet/IP autonomo o come parte di un sistema di building automation programmato da EcoStruxure con un server AS-P, AS-B o Enterprise come server principale.

L'RP-C è dotato di un chip wireless che consente all'applicazione di messa in servizio mobile di connettersi direttamente al controller. Il controller RP-C è disponibile in tre modelli con diverso numero di I/O e, in particolare, quello in uso ha 8 I/O universali di tipo Ub e 4 uscite

relè a stato solido. (Schneider Electric, 2019)

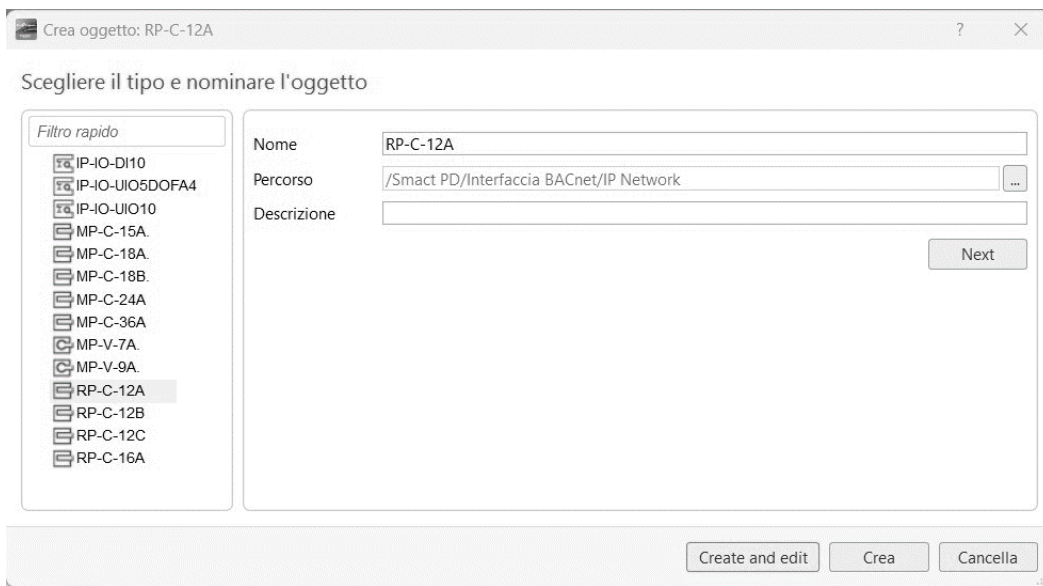


Figura 4.23: Elenco dei dispositivi Schneider Electric della Serie SmartX.

Per semplicità della codifica successiva, al posto del nome del dispositivo della serie di Schneider viene scritto “Attuatore”. Questo è il nome con cui ora il device viene mostrato nell’interfaccia della piattaforma EBO, una volta cliccato sul pulsante “Crea”.

Avanzando, ora che è presente l’attuatore è doveroso popolarlo degli oggetti BACnet necessari per renderlo operativo. Nella voce “Application” del device, dunque, si segue l’iter delineato al 2.4.3 e si crea un valore del tipo Digital Value, al quale è associato il nome “Richiesta accensione luce”.

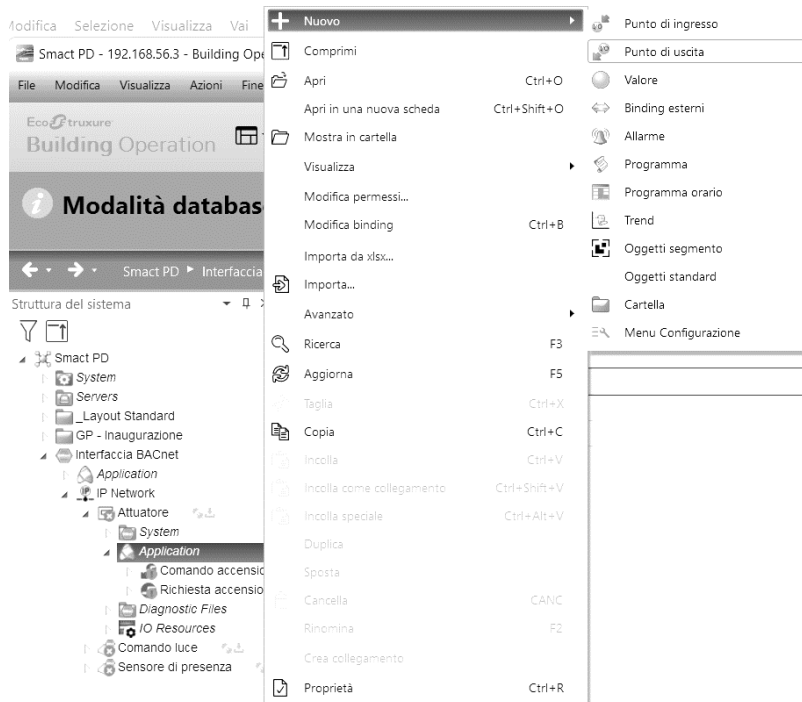


Figura 4.24: Creazione dell’oggetto “Comando accensione luce” all’interno dell’attuatore.

Viene poi creato un secondo oggetto, ovvero un punto di uscita, seguendo la procedura mostrata in figura 4.24. Tale punto d'uscita sarà del tipo Digital Output e, ad esso, viene associato il nome "Comando accensione luce" e prodotto esattamente come il caso precedente. Tale valore, assieme a quello precedente, sarà fondamentale per la comunicazione delle grandezze con il sensore e il comando luce.

3.3.2 Creazione del comando luce

Con tale dispositivo si vuole emulare un comando a pulsante atto all'accensione della luce in seguito ad un input ricevuto. All'interno dell'interfaccia di EcoStruxure Building Operation non si sono trovati dispositivi Schneider adatti al caso; pertanto, viene creato un generico dispositivo BACnet da programmare da parte dell'utente.

L'iter praticato è sempre il medesimo a partire dall'IP Network, selezionando ora la voce "Dispositivo BACnet" al quale sarà associato il nome "Comando luce".

Al suo interno viene plasmato un oggetto BACnet che in questo caso è un punto di entrata del tipo Digital Input (fig. 4.25). Ad esso è associato il nome "Comando accensione luce" il quale presenta lo stesso nome del caso precedente per l'attuatore, ma rappresenta qui un valore in ingresso e non in uscita.

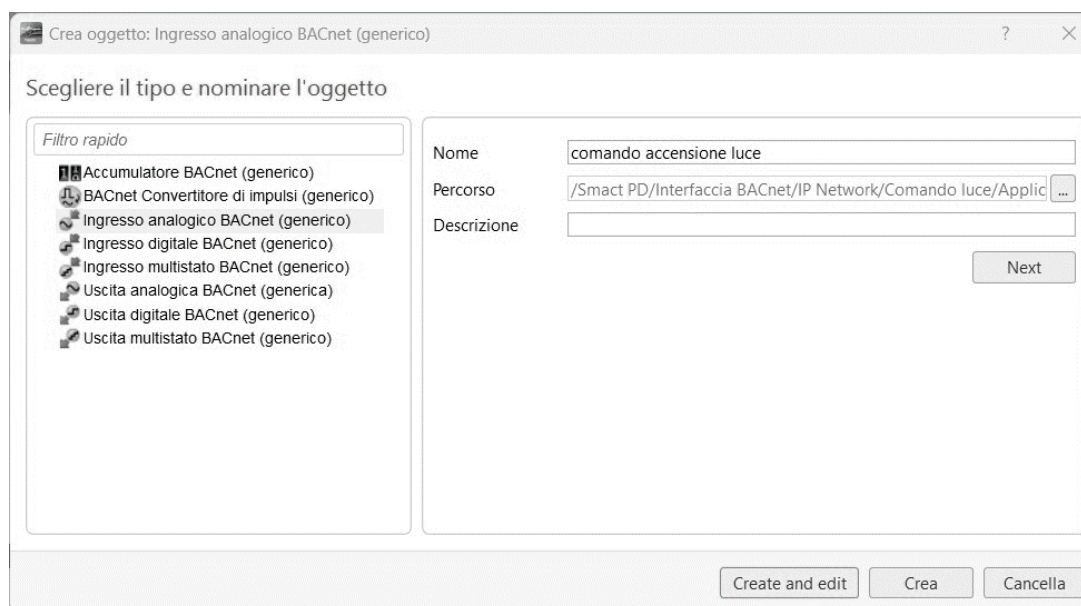


Figura 4.25: Creazione dell'oggetto "Comando accensione luce" all'interno del Comando luce.

3.3.3 Creazione del sensore di presenza

L'ultimo dispositivo da inserire all'interno del modello è il sensore, il quale è programmato per la rilevazione della presenza di persone nella stanza.

Anche in questo caso si è creato un generico dispositivo BACnet chiamato ora "Sensore di presenza", arricchito da un oggetto simile al caso precedente, dunque un Digital Input

chiamato per l'occasione "presenza persone".

Alla fine di tutto ciò è visibile nel cuore dell'interfaccia BACnet la seguente diramazione mostrata in figura 4.26.

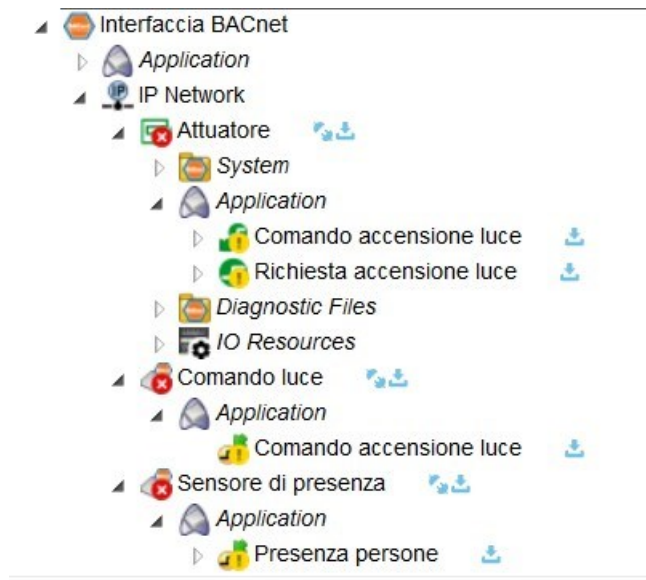


Figura 4.26: Interfaccia BACnet popolata di tutti i dispositivi e oggetti utili al progetto di sperimentazione.

3.3.4 Binding tra proprietà degli oggetti presenti nel modello

Creati tutti i dispositivi e popolati essi degli oggetti necessari per il funzionamento, si rivela fondamentale delineare le relazioni che intercorrono tra gli stessi al fine di far funzionare correttamente il sistema di automazione della stanza in esame.

Il concetto di relazione in EcoStruxure Building Operation si declina, come già accennato, nei binding tra proprietà e tra oggetti, di cui si rimanda al paragrafo 2.1 del capitolo ivi descritto.

Nella pratica del caso sperimentato, i binding vengono implementati all'interno dell'attuatore e, in particolare, nell'oggetto Digital Value "Richiesta accensione luce".

Per tracciare correttamente tale scambio di informazioni si seleziona l'oggetto desiderato e cliccando con il tasto destro del mouse, oppure digitando CTRL+B, si sceglie la voce "modifica binding". In risposta a ciò, si apre la finestra di dialogo relativa, mostrata in figura 4.27. Come si può vedere, si ha ora accesso alla trasformazione diverse variabili e parametri associati e associabili all'oggetto prescelto.

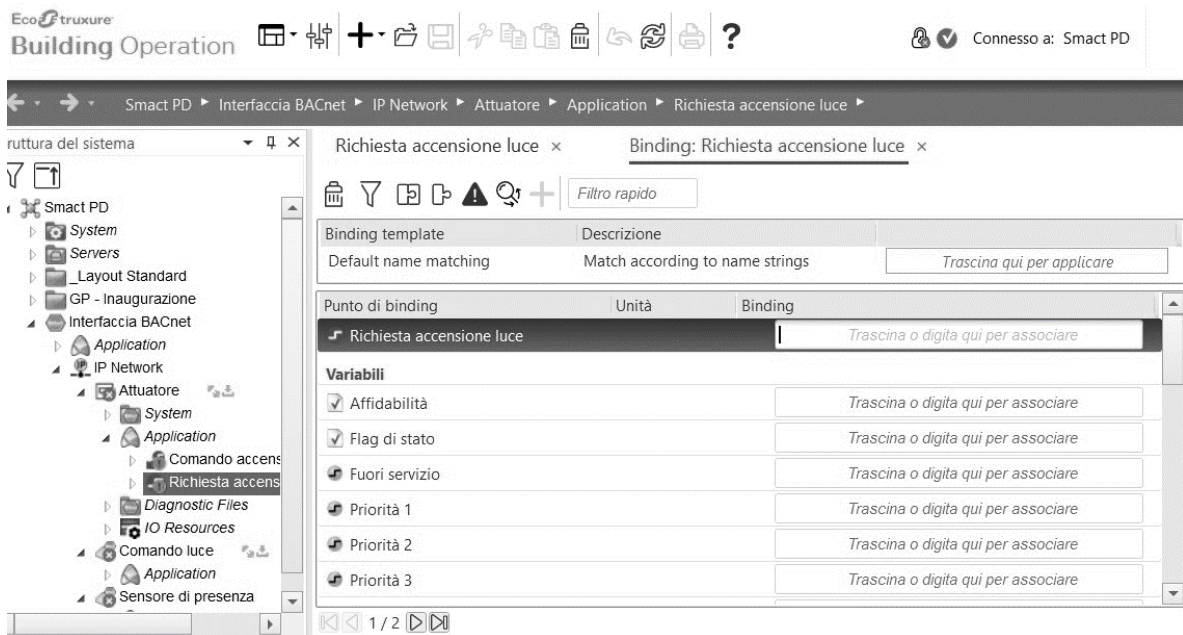


Figura 4.27: Finestra di dialogo in EBO dedicata ai binding.

In questo oggetto vengono eseguiti due binding (fig. 4.28), dunque alla richiesta di accensione luce sono associati:

- Il valore registrato in input della presenza di persone dal sensore;
- Il valore registrato in input dal comando di accensione luce.



Figura 4.28: valori associati all'oggetto "Richiesta accensione luce".

Il meccanismo di associazione dei valori è piuttosto semplice, poiché occorre solamente selezionare il valore desiderato da collegare e trascinarlo all'interno di una delle priorità visibili nella finestra di dialogo.

La priorità è a scelta dell'utente, a patto che i valori siano in accordo allo sviluppo logico delle informazioni che circolano in un circuito. In questo caso allora, prima il sistema riceve una comunicazione di rilevazione presenza da parte del sensore e successivamente entra in gioco il comando di accensione luce.

Altro binding imprescindibile è quello destinato al valore dell'oggetto in esame, al quale si associa il Digital output che comunica il "Comando accensione luce". (fig. 4.29)

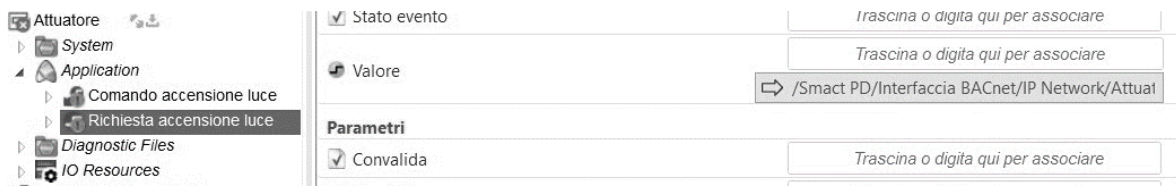


Figura 4.29: Binding del Valore dell'oggetto "Richiesta accensione luce" con il digital output.

Infine, all'interno delle risorse I/O dell'attuatore, in corrispondenza del Digital Output 1, visibile a schermo come DO1, viene associato l'oggetto "Richiesta accensione luce" (fig. 4.30).

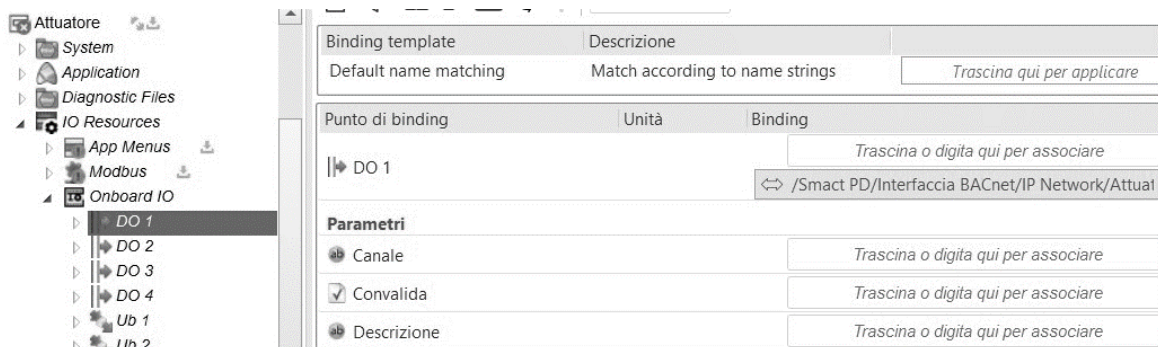


Figura 4.30: Binding con la porta DO1 dell'attuatore a cui è associata la "Richiesta accensione luce".

Tale parametro è usato poi come terminale del Digital Output "comando accensione luce" (fig. 4.31).



Figura 4.31: Porta DO1 usata come terminale del "Comando accensione luce".

3.4 Export del file XML da EBO

Il modello appena descritto viene esportato in formato xml, dal quale provengono le figure suggerite nel proseguo dello studio svolto. Si vede, pertanto, come sia stata creata per prima cosa un'interfaccia BACnet (fig. 4.32) contenente i seguenti dispositivi, popolati con i relativi oggetti:

- Un attuatore con le adeguate proprietà BACnet (fig. 4.32) che contiene in sé un oggetto Digital Value che comunica una richiesta di accensione luce (fig. 4.33) e un oggetto Digital Output che comunica un comando di accensione luce (fig. 4.34).

```

9  <ExportedObjects>
10  <OI NAME="Interfaccia BACnet" TYPE="bacnet.Device">
11    <OI NAME="IP Network" TYPE="bacnet.IPDataLink">
12      <PI Name="NetworkId" Value="1"/>
13    <OI NAME="Attuatore" TYPE="bacnet.mpx.udt.RPC12A">
14      <OI NAME="IO Resources" TYPE="bacnet.mpx.IOResources" declared="1">
15        <OI NAME="Ports" TYPE="bacnet.mpx.PortsFolder" declared="1">
16          <OI DESCR="Room Bus" NAME="RS485-COMB" TYPE="bacnet.mpx.ports.RS485" declared="1"/>
17          <OI DESCR="Sensor Bus" NAME="RS485-COMA" TYPE="bacnet.mpx.ports.RS485" declared="1"/>
18          <OI DESCR="USB" NAME="USB" TYPE="bacnet.mpx.ports.USBPort" declared="1"/>

```

Figura 4.32: Interfaccia BACnet (riga 10) con il relativo attuatore (riga 13).

```

52  <OI NAME="Application" TYPE="bacnet.mpx.RPXApplicationProxy" declared="1">
53    <OI NAME="Richiesta accensione luce" TYPE="bacnet.mpx.value.DigitalValue">
54      <PI Name="ActiveText" Value=""/>
55      <PI Name="ForeignAddress" Value="&lt;binary-value, 10&gt;"/>
56      <PI Name="InactiveText" Value=""/>
57      <PI Name="Priority1" Type="bacnet.pt.BACnetBinaryPV"/>
58      <PI Name="Priority10" Type="bacnet.pt.BACnetBinaryPV"/>
59      <PI Name="Priority11" Type="bacnet.pt.BACnetBinaryPV"/>
60      <PI Name="Priority12" Type="bacnet.pt.BACnetBinaryPV"/>
61      <PI Name="Priority13">

```

Figura 4.33: Oggetto "Richiesta accensione luce" (riga 53) contenuto nell'attuatore.

```

80  <OI NAME="Comando accensione luce" TYPE="bacnet.mpx.point.DigitalOutput">
81    <PI Name="ForeignAddress" Value="&lt;binary-output, 11&gt;"/>
82    <PI Name="Priority1" Type="bacnet.pt.BACnetBinaryPV"/>
83    <PI Name="Priority10" Type="bacnet.pt.BACnetBinaryPV"/>
84    <PI Name="Priority11" Type="bacnet.pt.BACnetBinaryPV"/>
85    <PI Name="Priority12" Type="bacnet.pt.BACnetBinaryPV"/>
86    <PI Name="Priority13" Type="bacnet.pt.BACnetBinaryPV"/>

```

Figura 4.34: Oggetto "Comando accensione luce" (riga 80) contenuto nell'attuatore.

- Un comando luce con le adeguate proprietà BACnet che contiene in sé un oggetto Digital Input che riceve le comunicazioni di comando di accensione luce prima citate. (fig. 4.35)

```
108 <OI NAME="Comando Luce" TYPE="bacnet.DeviceProxy">
109   <PI Name="APDUsegmentTimeout" Value="2000"/>
110   <PI Name="AlignIntervals" Value="0"/>
111   <PI Name="BackupFailureTimeout" Value="0"/>
112   <PI Name="IntervalOffset" Value="0"/>
113   <PI Name="Location" Value=""/>
114   <PI Name="MaxInfoFrames" Value="2"/>
115   <PI Name="MaxMaster" Value="127"/>
116   <PI Name="TimeSynchronizationInterval" Value="0"/>
117   <PI Name="UTCOffset" Value="0"/>
118   <OI NAME="Application" TYPE="bacnet.BACnetApplicationProxy" declared="1">
119     <OI NAME="Comando accensione luce" TYPE="bacnet.pointproxy.digital.Input">
120       <PI Name="ActiveText" Value=""/>
121       <PI Name="AlarmMessage" Value=""/>
122       <PI Name="AlarmValue" Type="bacnet.pt.BACnetBinaryPV" Value="1"/>
```

Figura 4.35: Dispositivo "Comando luce" (riga 108) con relativo oggetto "Comando accensione luce" (riga 119).

- Un sensore di presenza con le adeguate proprietà BACnet che contiene in sé un oggetto Digital Input che registra e comunica la presenza di persone all'interno della stanza. (fig. 4.36)

```
134 <OI NAME="Sensore di presenza" TYPE="bacnet.DeviceProxy">
135   <PI Name="APDUsegmentTimeout" Value="2000"/>
136   <PI Name="AlignIntervals" Value="0"/>
137   <PI Name="BackupFailureTimeout" Value="0"/>
138   <PI Name="IntervalOffset" Value="0"/>
139   <PI Name="Location" Value=""/>
140   <PI Name="MaxInfoFrames" Value="2"/>
141   <PI Name="MaxMaster" Value="127"/>
142   <PI Name="TimeSynchronizationInterval" Value="0"/>
143   <PI Name="UTCOffset" Value="0"/>
144   <OI NAME="Application" TYPE="bacnet.BACnetApplicationProxy" declared="1">
145     <OI NAME="Presenza persone" TYPE="bacnet.pointproxy.digital.Input">
146       <PI Name="ActiveText" Value=""/>
147       <PI Name="AlarmMessage" Value=""/>
148       <PI Name="AlarmValue" Type="bacnet.pt.BACnetBinaryPV" Value="1"/>
```

Figura 4.36: Dispositivo "Sensore di presenza" (riga 134) e relativo oggetto "presenza persone" (riga 145).

Si vede come la struttura sia ad albero, esattamente come si presenta all'interno della piattaforma EBO vista nella figura 4.26, e come ogni istanza creata all'interno di un'entità maggiore sia caratterizzata da un rientro verso sinistra. Ad esempio, l'oggetto "presenza persone" è associato ad un dispositivo "sensore di presenza" che nella piattaforma rappresenta un tipo di entità madre rispetto al primo, esattamente come, se si segue la linea grigia a salire nel file xml, si vede come lo stesso dispositivo citato sia figlio dell'IP Network, a sua volta figlio dell'interfaccia BACnet.

Studiata tale struttura del file xml esportato da EcoStruxure Building Operation, è possibile creare un algoritmo tramite Visual Studio Code partendo dal file IFC esportato da Revit2023, al fine di creare un file con estensione xml leggibile da EBO e che renda automatizzata la creazione di un progetto base come quello in analisi. Questa funzione risulterebbe una scoperta essenziale agli occhi degli esperti del settore allo scopo di semplificare e velocizzare il processo, senza perdita alcuna di informazioni, il quale si è visto essere piuttosto complesso e ricco di dettagli.

3.5 Visual Studio Code per parsing IFC to XML

Visual Studio Code è un leggero e gratuito editor di codici di risorsa, offerto da Microsoft, che si avvia sul desktop o su web, ed è disponibile per diversi sistemi operativi. Supporta nativamente diversi linguaggi di programmazione come, ad esempio, C++, C#, Java, Python, PHP e Go, runtime come .NET e Unity, ambienti come Docker e Kubernetes e cloud come Amazon Web Services, Microsoft Azure e Google Cloud Platform.

La sua interfaccia grafica per l'inserimento di script di codice permette una visualizzazione più accessibile in particolare ai neofiti giacché evidenzia con i medesimi colori parti dello script aventi la medesima funzione concettuale.

3.5.1 Impostazioni preliminari del codice in Python

Prima di cominciare una qualsiasi azione, l'editor richiede sia scelto un linguaggio di programmazione (fig. 4.37) e, per il caso studio applicativo in esame, si sceglie Python.

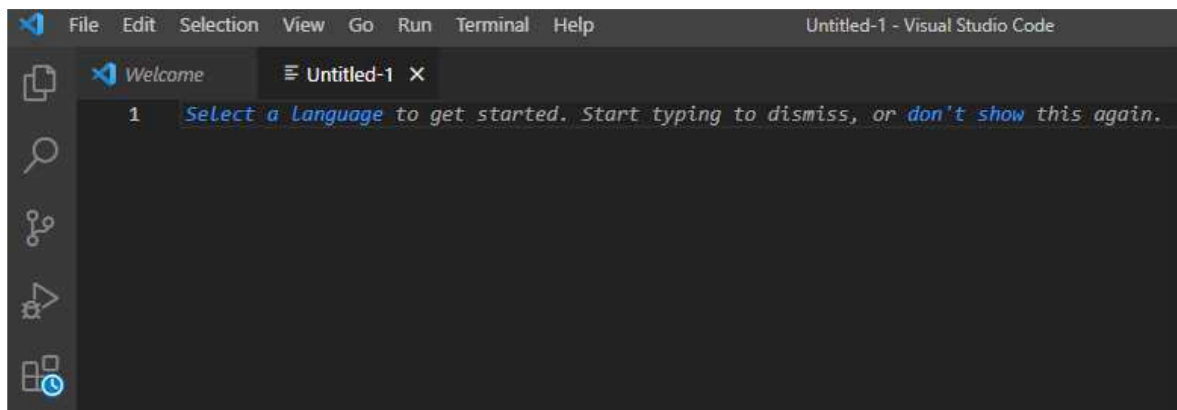


Figura 4.37: Interfaccia di Visual Studio Code, pagina iniziale.

Successivamente, si importano tutte le librerie necessarie per procedere con il parsing⁴³ del file ifc al fine di renderlo leggibile in xml. Le librerie importate dunque sono:

- "Ifcopenshell", libreria opensource per software, gratuita, aperta e

⁴³ In informatica, il parsing, è una sorta di analisi sintattica; pertanto, si definisce come un processo che analizza un flusso continuo di dati in ingresso, letti per esempio da un file, in modo da determinare la correttezza della sua struttura grazie ad una data grammatica formale.

modificabile da tutti, nata per aiutare gli utenti e gli sviluppatori di applicativi per operare con i file in formato IFC.

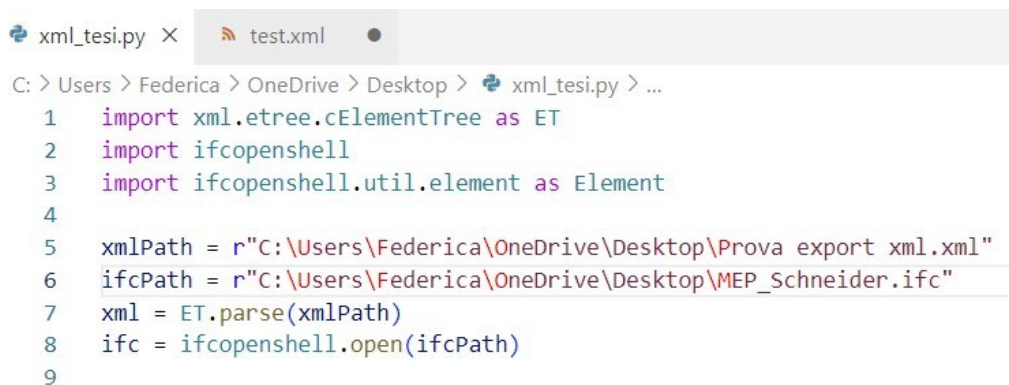
- La libreria ElementTree, nata come contributo alla libreria standard presente in Python. Comprende tutti gli strumenti atti per analizzare XML usando API basate su eventi e documenti, per cercare nei documenti analizzati tramite espressioni XPath e per creare o modificare documenti.

Il percorso di importazione di una libreria, ad esempio per ifcopenshell, è del tipo:

→ import ifcopenshell

Le librerie vengono importate come modulo di implementazione per Python. In questo modo il linguaggio di programmazione può accedere ai comandi offerti dalle stesse ed effettuare funzioni specifiche da loro definite. Una libreria è correttamente importata se il programma non restituisce alcun messaggio di errore.

Fatto questo, si procede inserendo come input il file xml esportato da EBO da avere come template e il file IFC esportato dal modello in Revit. Nella figura 4.38 si può vedere il codice fino ad ora realizzato, nelle prime righe la procedura di importazione delle librerie e a seguire, dalla riga 5, l'input dei file xml e ifc.



```
xml_tesi.py × test.xml
C: > Users > Federica > OneDrive > Desktop > xml_tesi.py > ...
1 import xml.etree.cElementTree as ET
2 import ifcopenshell
3 import ifcopenshell.util.element as Element
4
5 xmlPath = r"C:\Users\Federica\OneDrive\Desktop\Prova export xml.xml"
6 ifcPath = r"C:\Users\Federica\OneDrive\Desktop\MEP_Schneider.ifc"
7 xml = ET.parse(xmlPath)
8 ifc = ifcopenshell.open(ifcPath)
9
```

Figura 4.38: Procedura iniziale di scrittura codice per il parsing dei file.

In caso di progetti molto popolati è sicuramente utile comprendere quali dispositivi sono presenti e dividerli per categorie, così da rendere semplificato l'accostamento alle corrette sorgenti. Si interroga dunque il file IFC "by_type" al fine di avere una chiara visione di attuatori, sensori e comandi luce. Si veda la figura 4.39.

```
11 attuatori = ifc.by_type("IfcActuatorType")
12 sensori = ifc.by_type("IfcSensorType")
13 comandiLuce = ifc.by_type("IfcSwitchingDevice")
14 print (attuatori)
15 print (sensori)
16 print (comandiLuce)
```

Figura 4.39: Procedura di ricerca dei dispositivi domotici nel file IFC per categoria.

Va fatta particolare attenzione a questa procedura, poiché tra parentesi deve essere presente esattamente l'entità ifc così come risulta specificata nel sito ufficiale di buildingSMART nello standard IFC2x3, poiché ifcopenshell non è momentaneamente aggiornato oltre.

Avviando il debug del codice scritto fino ad ora, nel terminale della console python posizionato in basso, appariranno scritte tutte le entità indagate. Ogni categoria sarà divisa dall'altra da parentesi quadre secondo quanto mostrato in figura 4.40.



```

PROBLEMI OUTPUT CONSOLE DI DEBUG TERMINALE Python Debug Console + - □ 🗑️ ... ^ ×

[#1938=IfcActuatorType('3CL4rYpDjBQ0ItXZgc5ZPT',#20,'Electrical_Controls_Schneider-Electric_KNX-Actuator-Switch-10-A-Manual-Mode:MTN649202 - REG-K/2x230/10',,$,($,(#1973),(#1937),'924613',,$,.ELECTRICAUTATOR.))]
[#1292=IfcSensorType('1S1zQCICPEhu732E3J_h2v',#20,'Electrical_Controls_Schneider-Electric_SpaceLogic_Optimum-Housing_Touch-Screen-with-Off-screen-Control:SWWSC2PSELXW - Sensor Base + White Cover - Setpoint, Override, Touchscreen, Light Buttons, Occupancy',,$,($,(#1323),(#1291),'818917',,$,.MOVEMENTSENSOR.))]
[#1638=IfcSwitchingDevice('1S1zQCICPEhu732E3J_wmy',#20,'o 3D - Fixing frame:o 3D - Fixing frame:837216',,$,'o 3D - Fixing frame:o 3D - Fixing frame',#1637,#1634,'837216',.NOTDEFINED.),#1662=IfcSwitchingDevice('1S1zQCICPEhu732E3J_wwz',#20,'o 3D - Insert:o 3D - Insert:837217',,$,'o 3D - Insert:o 3D - Insert',#1661,#1658,'837217',.NOTDEFINED.), #1710=IfcSwitchingDevice('1S1zQCICPEhu732E3J_ww',#20,'SE - KNX - System M - Rockers for 1-gang push-button module:SE - KNX - System M - Rockers for 1-gang push-button module:837218',,$,'SE - KNX - System M - Rockers for 1-gang push-button module:SE - KNX - System M - Rockers for 1-gang push-button module',#1709,#1706,'837218',.NOTDEFINED.)]

```

Figura 4.40: Entità indagate “by_type” visualizzate per categoria nel terminale.

Si vede nella schermata riportata l'attuatore taggato con #1938, il sensore taggato con #1292 e il comando luce, con tutte le sue componenti a seguire, taggato con #1638. Tutte le entità sono denominate come la classe IFC a cui sono state associate e, dove specificato, riportano anche il TypeEnum dello standard IFC assegnato nella mappatura IFC in Revit.

Si intraprende ora la strada per la scrittura del file xml, iterando ogni dispositivo ifc individuato e istituendo la corrispettiva entità BACnet nell'xml tramite il comando “getroot” applicato al file xml e il comando “ExportedObjects”.

Si estrapola infine, dal template del file esportato da EBO, l'entità rete di BACnet “bacnet.IPDataLink” la quale sarà l'entità padre di tutti gli oggetti che si andranno a delineare in un momento secondario.

3.5.2 Realizzazione dei dispositivi necessari in Python

Prima di cominciare l'attività di scrittura del codice, si è rivelato fondamentale studiare dettagliatamente la struttura del file xml estratto da EcoStruxure Building Operation. Come si è visto nel paragrafo 3.4 del capitolo ivi descritto, infatti, ogni dispositivo è ricco di proprietà che derivano direttamente dalla morfologia della piattaforma di Schneider Electric. Queste, ad esempio, sono le priorità dei valori in binding nel dispositivo della serie SmartX di Schneider Electric, oppure le proprietà “TimeSynchronizationInterval”, “AlignIntervals” o ancora “Location”.

In questo percorso di sperimentazione non si è affrontata la scrittura in Python di tali parametri, nonostante esistano librerie di esportazione ifc in xml che ne consentano una scrittura esaustiva. Tale lavoro va affrontato seguendo un programma estensivo che contempli dettagliatamente le definizioni e i parametri richiesti da BACnet ed è destinato ad uno sviluppo posteriore.

Per delineare un qualsiasi dispositivo si è ricorso alla creazione di un ciclo “for” su Python, il quale permette di iterare su tutti gli elementi di un oggetto in grado di restituire le entità al suo interno una dopo l’altra, come ad esempio liste e set. In particolare, questo tipo di ciclo descritto, esegue il blocco di codice interessato per un numero specifico di cicli, ovvero i necessari affinché tutte le entità di quel tipo non vengano stampate dal programma.

Per il caso dell’attuatore, perciò, il ciclo creato è quello visualizzabile nella figura 4.41, il quale itera ogni attuatore presente nel gruppo stampato in precedenza dal file IFC (figure 4.39 e 4.40).

```
for attuatore in attuatori:
```

Figura 4.41: Ciclo for per la ricerca di attuatori.

Se ci si focalizza sulla struttura xml dei dispositivi stampata da EBO, si nota che l’attuatore, inserito come device specifico di Schneider Electric, ha delle proprietà differenti rispetto al sensore, il quale invece è stato inserito come dispositivo generico BACnet nel modello. A tal proposito, si è rivelato necessario scrivere dei cicli for specifici e differenti nel contenuto, a meno di alcuni parametri in comune.

Ogni dispositivo che esiste all’interno dell’interfaccia EBO, e dunque nel file xml esportato dallo stesso, è richiamato da due parametri comuni ovvero il “NAME” e il “TYPE”, come mostrato in figura 4.42. Il primo corrisponde al nome associato da parte dell’utente in fase di creazione del device e il secondo è il modello scelto tra quelli proposti nativamente da EBO. Ad esempio, per l’attuatore si era scelto il tipo RP-C-12A tra la lista di tutti i dispositivi appartenenti alla serie SmartX di Schneider Electric, visti in figura 4.23.

```
<OI NAME="Attuatore" TYPE="bacnet.mpx.udt.RPC12A">
```

Figura 4.42: NAME e TYPE relativi all’attuatore nel file xml esportato da EBO.

In Python, pertanto, si è creato un nuovo oggetto e tramite il comando “NewObj.set” gli si è assegnata la proprietà “NAME”. A seguire, per mezzo del comando “Element.get_type” si è letto il type associato all’oggetto nel file ifc e, usando poi lo stesso comando prima citato, si è assegnato tale parametro all’oggetto.

Ciò che verrà stampato dal codice è quanto riportato nella seguente immagine, figura 4.43.

```
<OI NAME="Attuatore" TYPE="Electrical Controls Schneider-Electric KNX-Actuator
```

Figura 4.43: Proprietà dell’attuatore stampate dal codice scritto in Python.

La stringa di caratteri stampata al parametro “TYPE” è stata riportata parzialmente, ma nella sua completezza riprende esattamente il tipo di attuatore scelto nel percorso di creazione dell’istanza in Revit, riportato in Tabella 11.

Si vede dunque come la struttura iniziale del device creata con Python a partire dal file IFC sia identica a quella ufficiale esportata da EBO. Per una più corretta scrittura sarà doveroso, in un momento postero, far collimare il type derivato da Revit con quello scritto in EBO, creando un nuovo parametro IFC in Revit in grado di restituire quanto desiderato o, meglio, istituendo una famiglia contenente tutte le proprietà BACnet ambite.

La medesima procedura di realizzazione va poi associata alle entità successive presenti nel file xml di EBO. Ad esempio, continuando la dimostrazione di scrittura relativa all’attuatore, la riga seguente di file vede estrapolato il parametro “IO Resources” all’interno del device. Viene stampata a schermo, nella figura 4.44, la parte di codice che interessa la sua programmazione, con la relativa descrizione dei passaggi visibile in verde.

```
#creo l'entità IO Resources in xml
risorse = ET.Element("OI")
#assegno i suoi valori
risorse.set("NAME", "IO Resources")
risorse.set("TYPE", "bacnet.mpx.IOResources")
risorse.set("declared", "1")
```

Figura 4.44: Procedura di realizzazione del parametro “IO Resources” con relativi valori associati.

Dove ET sta per ElementTree, ovvero la libreria importata all’origine del codice per rappresentare il documento xml con una struttura ad albero, ed Element consente di individuare il singolo nodo contestualizzato in tale struttura ad albero.

Alla fine della procedura in atto è opportuno inserire il comando “append” digitando, ad esempio seguendo il codice finora scritto, “NewObj.append(risorse)” allo scopo di inserire tale entità all’interno degli oggetti padre creati in precedenza, seguendo l’ordinamento ad albero necessario al file.

Il tutto è da ripetere per ogni parametro visibile, dunque nell’attuatore si identifica questa parte di codice anche per le entità che seguono: “Onboard I/O” che nel caso ivi delineato rappresenta le porte di ingresso e uscita dei valori, “Application” che rappresenta l’entità padre degli oggetti BACnet contenuti nel dispositivo.

Nell’eventualità di utilizzo dei dispositivi BACnet generici all’interno del software EcoStruxure Building Operation, il codice cambia sotto alcune peculiarità. Nel dettaglio di quanto applicato per il sensore, ad esempio, la conformazione iniziale della scrittura del device vede sempre i parametri “NAME” e “TYPE”, ma in questo caso il secondo nominato non raffigura più un parametro esistente definito come visto prima da Schneider Electric, ma un

generico "Device Proxy" (fig. 4.45). Tale parametro sarà il medesimo per ogni oggetto BACnet generico inserito nel modello in EBO.

```
<OI NAME="Sensore di presenza" TYPE="bacnet.DeviceProxy">
```

Figura 4.45: Parametri "NAME" e "TYPE" relativi al dispositivo sensore di presenza.

Pertanto, il percorso di struttura del codice sarà al minimo dissimile dal quanto visto anteriormente: una volta indagato il nome e il tipo del dispositivo ricercato, digitando "NewObj.set" gli si assegnerà come "TYPE" ciò che viene visualizzato in figura 4.45, ovvero "bacnet.DeviceProxy" e non più quanto letto dal file ifc tramite il comando "get_type".

Come già accennato, i dispositivi generici BACnet nella piattaforma EBO non presentano al loro interno particolari proprietà, ma sono da implementare da parte dell'utente. L'unica entità presente è "Application", la quale sarà creata come descritto nella procedura per "IO Resources".

3.5.3 Restituzione del nuovo file xml creato in Python

Per concludere il percorso di codifica dei dispositivi domotici inseriti nel modello con le annesse proprietà al loro interno, è doveroso popolare l'applicativo esterno. Questo è fattibile per mezzo di una funzione in grado di restituire l'entità xml con determinate proprietà e valori partendo dal suo *root*.⁴⁴

Per mezzo della parola chiave "def" in Python è possibile definire una specificata funzione, la quale prende in input gli argomenti passati durante la chiamata della funzione. Poi, la stessa elabora il blocco di istruzioni specificato con indentazione al suo interno e restituisce, alla fine, il risultato dell'elaborazione svolta tramite l'istruzione "return".

Si specifica che le variabili locali usate dentro la funzione sono separate da quelle globali del programma che richiama la funzione.

Per il caso in esame si è compilata la funzione come visibile in figura 4.46.

```
def TrovaByAttributo(root,attr,valore):  
    for t in root:  
        if attr in t.attrib:  
            if t.attrib[attr]==valore:  
                return t  
            else:  
                return "Nessun tag con quel nome e attributo"
```

Figura 4.46: Scrittura della funzione in grado di restituire proprietà e valori specificati all'interno del codice.

L'ultima riga del codice da scrivere è quella destinata alla codifica di un nuovo file xml contenente tutte le istanze plasmate all'interno dello stesso.

⁴⁴ Con l'espressione *root* si intende la radice più profonda, e comune, di tutte le entità.

Avvalendosi della funzione “write” e specificando l’estensione desiderata del file da stampare, il codice ha l’opportunità di creare un nuovo documento. L’utente può denominare a piacimento il suddetto, scrivendo tra parentesi tonde il nome desiderato (fig. 4.47).

```
xml.write("test.xml")
```

Figura 4.47: Funzione “write” per creare il nuovo documento xml.

Avviando il debug del codice, ora completo, in Python e attendendo la fine del processo, si salverà il documento ambito. Questo è reperibile dall’utente nella stessa posizione dei file specificati al codice in input (fig. 4.38).

Analisi del flusso di informazioni tra i modelli BIM e i sistemi di automazione e controllo degli edifici (BACS) tramite BACnet e IFC
Santinato Federica

Capitolo Quinto

ANALISI CRITICA DEI RISULTATI E SVILUPPI FUTURI

1 ANALISI DEI RISULTATI

Il codice descritto al Capitolo Quattro crea un nuovo file xml che riprende pari passo l'architettura di sistema stampata nel file di esportazione proveniente da EcoStruxure Building Operation. Di seguito vengono perciò riportati gli esempi di tutti i dispositivi codificati con il programma creato in Python nelle figure 5.1, 5.2 e 5.3.

```
<OI NAME="Attuatore"
  TYPE="Electrical_Controls_Schneider-Electric_KNX-Actuator-Switch-10-A-Manual-Mode:MTN649202 - REG-K/2x230/10">
  <OI NAME="IO Resources" TYPE="bacnet.mpx.IOResources" declared="1" />
  <OI NAME="Onboard IO" TYPE="bacnet.mpx.io.IOModule" declared="1" />
  <OI NAME="Application" TYPE="bacnet.mpx.RPXApplicationProxy" declared="1" />
</OI>
```

Figura 5.1: Nuovo attuatore creato dal programma scritto in Python.

```
<OI NAME="Comando luce" TYPE="bacnet.DeviceProxy">
  <OI NAME="Application" TYPE="bacnet.BACnetApplicationProxy" declared="1" />
</OI>
```

Figura 5.2: Nuovo comando luce creato dal programma scritto in Python.

```
<OI NAME="Sensore di presenza" TYPE="bacnet.DeviceProxy">
  <OI NAME="Application" TYPE="bacnet.BACnetApplicationProxy" declared="1" />
</OI>
```

Figura 5.3: Nuovo sensore di presenza creato dal programma scritto in Python.

Per un più rapido confronto d'esempio si riportano anche alcune parti di file xml esportata da EBO riguardante l'attuatore.

```
<OI NAME="Attuatore" TYPE="bacnet.mpx.udt.RPC12A">
  <OI NAME="IO Resources" TYPE="bacnet.mpx.IOResources" declared="1">
    <OI NAME="Ports" TYPE="bacnet.mpx.PortsFolder" declared="1">
      <OI DESCR="Room Bus" NAME="RS485-COMB" TYPE="bacnet.mpx.ports.RS485" declared="1" />
      <OI DESCR="Sensor Bus" NAME="RS485-COMA" TYPE="bacnet.mpx.ports.RS485" declared="1" />
      <OI DESCR="USB" NAME="USB" TYPE="bacnet.mpx.ports.USBPort" declared="1" />
    </OI>
    <OI NAME="Sensor Bus" TYPE="bacnet.mpx.ru.SensorBus" declared="1">
      <PI Name="PortReference">
        <Reference DeltaFilter="0" Object="../Ports/RS485-COMA" Retransmit="0"
          TransferRate="10" />
      </PI>
    </OI>
  </OI>
  <OI NAME="Room Bus" TYPE="bacnet.mpx.irc.IRCBus" declared="1">
    <PI Name="ForeignAddress" Value="&lt;structured-view 64020&gt;" />
  </OI>
```

Figura 5.4: Attuatore inizialmente esportato da EBO, proprietà, prima parte.

```
<OI NAME="Onboard IO" TYPE="bacnet.mpx.io.IOModule" declared="1">
  <OI DESCR="Solid state relay output terminal" NAME="DO 1" TYPE="bacnet.mpx.udt.DOT"
  | declared="1" />
  <OI DESCR="Solid state relay output terminal" NAME="DO 2" TYPE="bacnet.mpx.udt.DOT"
  | declared="1" />
  <OI DESCR="Solid state relay output terminal" NAME="DO 3" TYPE="bacnet.mpx.udt.DOT"
  | declared="1" />
  <OI DESCR="Solid state relay output terminal" NAME="DO 4" TYPE="bacnet.mpx.udt.DOT"
  | declared="1" />
  <OI DESCR="Universal input/output terminal, type B" NAME="Ub 1"
  | TYPE="bacnet.mpx.udt.Ub" declared="1" />
  <OI DESCR="Universal input/output terminal, type B" NAME="Ub 2"
  | TYPE="bacnet.mpx.udt.Ub" declared="1" />
  <OI DESCR="Universal input/output terminal, type B" NAME="Ub 3"
  | TYPE="bacnet.mpx.udt.Ub" declared="1" />
  <OI DESCR="Universal input/output terminal, type B" NAME="Ub 4"
  | TYPE="bacnet.mpx.udt.Ub" declared="1" />
  <OI DESCR="Universal input/output terminal, type B" NAME="Ub 5"
  | TYPE="bacnet.mpx.udt.Ub" declared="1" />
  <OI DESCR="Universal input/output terminal, type B" NAME="Ub 6"
  | TYPE="bacnet.mpx.udt.Ub" declared="1" />
  <OI DESCR="Universal input/output terminal, type B" NAME="Ub 7"
  | TYPE="bacnet.mpx.udt.Ub" declared="1" />
  <OI DESCR="Universal input/output terminal, type B" NAME="Ub 8"
  | TYPE="bacnet.mpx.udt.Ub" declared="1" />
</OI>
<OI NAME="Modbus" TYPE="bacnet.mpx.modbus.network.Modbus" declared="1">
  <PI Name="ForeignAddress" Value="&lt;structured-view, 64035&gt;" />
</OI>
```

Figura 5.5: Attuatore inizialmente esportato da EBO, proprietà, seconda parte.

Si vede, pertanto, come dalla scrittura del codice su Python provengano le proprietà principali di un dispositivo qual è l'attuatore, le quali rappresentano il template di base sulla piattaforma di Schneider Electric, come si può notare in figura 4.26.

In questo percorso di sperimentazione volutamente non sono stati implementati nel codice i parametri secondari che discendono dalle relative entità padre, come ad esempio le porte di ingresso e uscita digitali e universali dell'attuatore in esame. Tali parametri potevano essere codificati in Python e stampati poi nel file XML tramite linguaggio meramente informatico. Ciò comportava però la creazione di entità senza che queste avessero una base fondata sul progetto modellato in Revit2023, il che non collima adeguatamente con l'ideologia che sta alla base di questa tesi.

La stessa filosofia viene seguita per quanto concerne gli oggetti BACnet contenuti all'interno dei dispositivi domotici. Di questi si riporta comunque una parte di codice implementata durante la sperimentazione in Python e afferente all'oggetto Digital Value "Richiesta accensione luce" visto precedentemente nella figura 4.24. Il procedimento di creazione è riportato nella figura 5.6.

```
#creo l'entità "richiesta accensione luce" in xml
richiestaaccensioneluce = ET.Element("OI")

#assegno i suoi valori
richiestaaccensioneluce.set("NAME","Richiesta accensione luce")
richiestaaccensioneluce.set("TYPE","bacnet.mpx.value.DigitalValue")
```

Figura 5.6: Creazione in Python dell'oggetto BACnet "Richiesta accensione luce".

Si ritiene che a tale entità sia destinata una procedura di implementazione simile a quella delineata per la creazione dei dispositivi domotici vista in precedenza e che, dunque, preveda la creazione di un ciclo di iterazione in grado poi di restituire delle entità. A queste ultime è poi doveroso associare le funzioni "New.Object.set" e "Element.get_type" attraverso le quali si indagano delle proprietà già presenti all'interno di un file IFC.

Il codice creato in questo percorso di sperimentazione, ad ora, grazie alle librerie presenti in Python, interroga tutti gli elementi domotici presenti all'interno di un progetto del tipo IFC, li raggruppa in una lista e li restituisce in un formato di file leggibile e direttamente integrabile all'interno del panorama del software EcoStruxure building Operation. A prescindere dalla complessità dei dispositivi e degli oggetti, il codice può leggere gli elementi e tradurli in entità BACnet.

Da questa base l'obiettivo è quello di arrivare ad una scrittura completa, e non necessariamente esaustiva, di un codice importabile nella piattaforma di Schneider Electric.

Alcuni parametri presenti di default all'interno degli elementi in EBO, e comuni in modo continuativo, infatti possono comunque essere generati all'interno del codice informatico. Ciò significa che in Python si crea una stringa che rimane di fatto senza alcun valore, la quale verrà popolata di tutti i valori necessari in un momento secondario.

La modellazione di entità di questo genere può rappresentare una base di scrittura completa e semplificativa nel processo di progettazione di sistemi di automazione e controllo degli edifici. Si pensi infatti ad un progetto BACS di un ospedale o un complesso adibito ad uffici dove sono presenti migliaia di dispositivi da programmare dal principio, avere a disposizione in automatico un'architettura di sistema già formata facilita e velocizza notevolmente la procedura di progettazione.

2 SVILUPPI FUTURI

Nel percorso di modellazione di impianti elettrici in Revit, è possibile creare dei circuiti elettrici relazionando tra di loro vari elementi compatibili e desiderati dall'utente. Durante l'esportazione del file in IFC, tale entità astratta viene automaticamente restituita in IFC per mezzo della relazione oggettivata "IfcRelAssignsToGroup", la quale di per sé gestisce l'assegnazione di oggetti ad un gruppo.

Il codice scritto in Python è in grado di intraprendere e recepire tali relazioni interrogando "by_type" il file IFC e recependo come argomento, scritto tra parentesi tonde, "IfcRelAssignsToGroup". Per avere un'idea più chiara di quanto appena eseguito si rivela appropriato stampare quanto appena restituito dalla funzione, usando il comando "print" in Python e associandogli un ciclo for in grado di iterare tutte le entità cercate, come fatto in figura 5.7

```
#visualizzo gruppi creati tramite relazioni in revit
distributionSystems = ifc.by_type("IfcRelAssignsToGroup")
for ds in distributionSystems:
    print ("-----GRUPPO-----")
    for ob in ds.RelatedObjects:
        entità = ob.is_a()
        print(f" {entità} : {ob.Name}")
    print("")
```

Figura 5.7: Procedura di stampa in Python delle relazioni tra dispositivi di un circuito in Revit.

Ciò che verrà restituito nel terminale della console di Python avviando il debug è quanto segue nella figura 5.8, in cui viene riportato come esempio la relazione che intercorre tra attuatore e lampadina e quella tra sensore e attuatore.

```
-----GRUPPO-----
IfcActuator : Electrical_Controls_Schneider-Electric_KNX-Actuator-Switch-10-A-Manual-Mode:MTN649202 - REG-
K/2x230/10:852348
IfcLightFixture : M_Applique - Illuminazione indiretta:100W - 120V:894341

-----GRUPPO-----
IfcSensor : Electrical_Controls_Schneider-Electric_SpaceLogic_Optimum-Housing_Touch-Screen-with-Off-screen
-Control: SXWSC2PSELXW - Sensor Base + White Cover - Setpoint, Override, Touchscreen, Light Buttons, Occupan
cy:818966
IfcActuator : Electrical_Controls_Schneider-Electric_KNX-Actuator-Switch-10-A-Manual-Mode:MTN649202 - REG-
K/2x230/10:852348
```

Figura 5.8: Esempio di gruppo di relazione tra dispositivi in Revit, stampata nel terminale in Python.

Data questa premessa si propone, per uno sviluppo futuro, di esaminare minuziosamente le relazioni che intercorrono tra dispositivi domotici, considerando anche che essi comunicano tra di loro via bus. Questo per consentire al codice di implementare soluzioni per la traduzione delle connessioni provenienti da ifc da restituire in relazioni BACnet espresse tramite binding.

Appurato che il programma creato restituisce le proprietà padre presenti nell'interfaccia BACnet all'interno di EcoStruxure Building Operation, è opportuno implementarle arricchendo innanzitutto le entità in Revit con più informazioni e parametri possibili, in maniera tale da poter popolare agevolmente e il più esaustivamente possibile i rispettivi dispositivi e oggetti BACnet. Idealmente, le famiglie ifc da editare sono quelle offerte da Schneider Electric, le quali sono nativamente programmate dall'azienda per restituire i parametri adeguati a seconda dell'elemento indagato.

È da valutare anche la possibilità di usufruire di database esterni, offerti direttamente dalle case produttrici delle apparecchiature, per arricchire i dispositivi di tutte quelle caratteristiche proprie e originarie come possono essere, ad esempio, le porte di input e output di un attuatore. L'utilizzo di documenti simili potrebbe semplificare la scrittura del codice, attribuendo all'elemento in Revit una determinata proprietà presente in maniera costante in oggetti dello stesso tipo.

Ad oggi il codice prende spunto dalla struttura del file XML precedentemente esportato da EBO, ma il processo può essere standardizzato e automatizzato per rendere agevole la comunicazione dei dati e ancor più immediata la modellazione di un impianto BMS. A tal proposito si rivela confacente l'implementazione di un'interfaccia grafica al programma creato in Python al fine di renderlo più maneggevole da parte dell'utente. In un secondo momento, idealmente, è possibile implementare la stessa all'interno del software EBO rendendola una parte integrante applicativa.

Analisi del flusso di informazioni tra i modelli BIM e i sistemi di automazione e controllo degli edifici (BACS) tramite BACnet e IFC
Santinato Federica

Capitolo Sesto

CONCLUSIONI

1 SINTESI DELLA RICERCA

Il presente percorso di tesi ha approfondito il tema del dialogo tra i sistemi informativi digitali e i sistemi di controllo e di automazione degli edifici, dato il loro ruolo via via più strategico e fondamentale nel settore costruttivo.

Gli standard BIM aperti non proprietari, come l'IFC ad esempio, si posizionano come perfetti candidati per la modellazione e lo scambio di informazioni riguardanti l'automazione, permettendo di conservare e scambiare dati coerenti tra i membri di un team di progetto, sia tra piattaforme con software simili che differenti, uniformando i flussi di lavoro.

D'altra parte, dall'analisi di letteratura si è notato come BACnet sia l'unico standard che permette la modellazione del sistema BMS mediante la selezione dei dispositivi in base al loro profilo standard e in base alle caratteristiche di interoperabilità da implementare. Pertanto, consente la progettazione senza la necessità di dichiarare specifici dispositivi reali.

Con la sperimentazione associata a tale studio si è dimostrato come questi standard abbiano un'effettiva affinità e come sia possibile trasferire in modo soddisfacente determinati dati a partire da un modello in formato ifc ad un file in formato xml, al fine di automatizzare il percorso di modellazione degli impianti domotici in software specializzati che applicano lo standard BACnet.

È stato possibile, infatti, partire da un modello ifc ricavato da Revit e, per mezzo di determinate librerie di Python, scrivere un codice in xml e renderlo integrabile naturalmente all'interno del software EcoStruxure Building Operation. Tale codice plasma un template di esportazione degli oggetti BACnet usufruendo di entità ifc editate correttamente in Revit in maniera tale da trasferire agevolmente i dati. Si creano, pertanto, in questa fase embrionale di sviluppo, le basi per creare un'architettura di sistema completa ed esaustiva a partire da un software parametrico legato all'edilizia.

BIBLIOGRAFIA

-
- Aste, N., Manfren, M., & Marenzi, G. (2017). Building Automation and Control Systems and performance optimization: A framework for analysis. *Renewable and Sustainable Energy Reviews*, 75, 313–330. <https://doi.org/10.1016/J.RSER.2016.10.072>
- Atzori, L., Iera, A., & Morabito, G. (2010). The Internet of Things: A survey. *Computer Networks*, 54(15), 2787–2805. <https://doi.org/10.1016/J.COMNET.2010.05.010>
- BACnet International. (2021, October 25). *BACnet conformance certificate*. https://www.bacnetinternational.net/catalog/manu/schneider%20electric/BTL_Cert_30791_SchneiderElectric_SmartFieldUC.pdf
- BACnet International. (2022). *BACnet TESTING LABORATORIES TESTING POLICIES*. https://btl.org/wp-content/uploads/sites/3/2022/10/BTL-Testing-Policies_2022-10-18-1.pdf
- BuildingSMART International. (2020). https://standards.buildingsmart.org/IFC/RELEASE/IFC4/ADD2_TC1/HTML/. <https://github.com/buildingSMART/technical.buildingsmart.org/blob/main/IFC-Formats.md>
- Bushby T, S. (1997). *BACnet - A standard communication infrastructure for intelligent buildings*. Automation in Construction. <http://www.bacnet.org/Bibliography/AIC-97/AIC1997.htm>
- Domingues, P., Carreira, P., Vieira, R., & Kastner, W. (2016). Building automation systems: Concepts and technology review. *Computer Standards & Interfaces*, 45, 1–12. <https://doi.org/10.1016/J.CSI.2015.11.005>
- Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, 29(7), 1645–1660. <https://doi.org/10.1016/J.FUTURE.2013.01.010>
- Intelligence Council, N. (2008). *Six Technologies With Potential Impacts on US Interests Out to 2025*.
- Jia, M., Komeily, A., Wang, Y., & Srinivasan, R. S. (2019). Adopting IoT for the development of smart buildings: A review of enabling technologies and applications. *Automation in Construction*, 101, 111–126. <https://doi.org/10.1016/J.AUTCON.2019.01.023>
- Kastner, W., Neugschwandtner, G., Soucek, S., & Newman, H. M. (2005). Communication systems for building automation and control. *Proceedings of the IEEE*, 93(6), 1178–1203. <https://doi.org/10.1109/JPROC.2005.849726>
- Malinowsky, B., & Kastner, W. (2010). Integrating process communication in building

- information models with IFC and LON. *IEEE International Workshop on Factory Communication Systems - Proceedings, WFCS*, 221–230. <https://doi.org/10.1109/WFCS.2010.5548630>
- Merz, H., Hansemann, T., & Hübner, C. (2009). *Building Automation: Communication systems with EIB/KNX, LON and BACnet - Hermann Merz, Thomas Hansemann, Christof Hübner - Google Libri*. <https://books.google.it/books?id=pxPq2CZSVooC&printsec=frontcover&hl=it#v=onepage&q&f=false>
- Newman, H. M. (2013). BACnet Today and the Smart Grid - BACnet Explained part one. *ASHRAE Journal*. <https://www.ashrae.org/File%20Library/Technical%20Resources/Bookstore/BACnet-Explained-Pt1.pdf>
- Schneider Electric. (2013). *Andover Continuum bCX1-CR Protocol Implementation Conformance Statement*. www.schneider-electric.com/building
- Schneider Electric. (2019). *Product Environmental Profile SmartX IP Controller RP-C-12C-F-24V*. <http://www2.schneider-electric.com/sites/corporate/en/products-services/green-premium/green-premium.page>
- Schneider Electric. (2021, July 23). *Document Policy*. <https://ecostruxure-building-help.se.com/bms/topics/show.castle?id=10952&locale=it-IT&productversion=3.3>
- Schneider Electric. (2022, December 2). *EcoStruxure™ Building Soluzione integrata per la gestione degli edifici*. https://download.schneider-electric.com/files?p_Doc_Ref=LEESGTT327CI&p_enDocType=User+guide&p_File_Name=LEESGTT327CI.pdf
- Swan, B. (1996). *Il linguaggio degli oggetti, delle proprietà e dei servizi BACnet*. <http://www.bacnet.org/Bibliography/ES-7-96/ES-7-96.htm>
- Tang, S., Shelden, D. R., Eastman, C. M., Pishdad-Bozorgi, P., & Gao, X. (2020). BIM assisted Building Automation System information exchange using BACnet and IFC. *Automation in Construction*, 110, 103049. <https://doi.org/10.1016/J.AUTCON.2019.103049>
- van Berlo, L. (2022). *IFC Schema Specifications - buildingSMART Technical*. <https://technical.buildingsmart.org/standards/ifc/ifc-schema-specifications/>
- Vasseur, J.-P., & Dunkels, A. (2010). Non-IP Smart Object Technologies. *Interconnecting Smart Objects with IP*, 295–302. <https://doi.org/10.1016/B978-0-12-375165-2.00019-3>
- Vieira, R., Carreira, P., Domingues, P., & Costa, A. A. (2020). Supporting building automation systems in BIM/IFC: reviewing the existing information gap. *Engineering, Construction and Architectural Management*, 27(6), 1357–1375. <https://doi.org/10.1108/ECAM-07-2018-0294/FULL/PDF>

- Wang, S. (2009). Intelligent buildings and building automation. *Intelligent Buildings and Building Automation*, 9780203890813, 1–248. <https://doi.org/10.4324/9780203890813>
- Wichenko, G. (2013). BACnet Today and the Smart Grid - BACnet Explained part two. *ASHRAE Journal*.
<https://www.ashrae.org/File%20Library/Technical%20Resources/Bookstore/BACnet-Explained-Pt2.pdf>

Analisi del flusso di informazioni tra i modelli BIM e i sistemi di automazione e controllo degli edifici (BACS) tramite BACnet e IFC
Santinato Federica

INDICE DELLE TABELLE

Tabella 1: Elenco dei tipi di oggetti standard BACnet divisi per categorie. (Newman, 2013).....	53
Tabella 2: Esempio delle proprietà BACnet obbligatorie e facoltative in un BACnet Analog Input Object. (Swan, 1996)	67
Tabella 3: Servizi di allarme ed eventi in un dispositivo BACnet con relativa descrizione. (Swan, 1996)	70
Tabella 4: Servizi di accesso ai file BACnet con relativa descrizione. (Swan, 1996)	70
Tabella 5: Servizi di accesso agli oggetti BACnet con relativa descrizione. (Swan, 1996)	71
Tabella 6: Servizi di gestione dei dispositivi remoti con relativa descrizione. (Swan, 1996)	72
Tabella 7: Servizi di Virtual Terminal BACnet con relativa descrizione. (Swan, 1996)...	72
Tabella 8: Classificazioni di indirizzo per IfcDistributionControlElement. (BuildingSMART International, 2020)	91
Tabella 9: Mappatura degli oggetti BACnet nello standard IFC.	95
Tabella 10: Elenco completo delle Property Set e delle Property Enumeration user-defined. (Tang et al., 2020)	118
Tabella 11: Dispositivi del modello con anagrafica IFC associata.....	143

Analisi del flusso di informazioni tra i modelli BIM e i sistemi di automazione e controllo degli edifici (BACS) tramite BACnet e IFC
Santinato Federica

INDICE ICONOGRAFICO

Figura 1.1: Percorso evolutivo dei Sistemi di Automazione e Controllo degli Edifici. (Aste et al., 2017)	8
Figura 1.2: Database aggiornato delle release IFC. (van Berlo, 2022).....	11
Figura 1.3: Il logo di IFC (Fonte: BIS-LAB® building innovation & skills-lab, 2017)	12
Figura 1.4: Sezioni e relativi capitoli di riferimento dello schema IFC. (BuildingSMART International, 2020)	12
Figura 1.5: Organizzazione delle classi di IFC. (BuildingSMART International, 2020)....	16
Figura 1.6: Elenco dei formati ufficiali di IFC e loro caratteristiche. (BuildingSMART International, 2020)	17
Figura 1.7: Paradigma “Internet of Things” come risultato della convergenza di visioni diverse (Atzori et al., 2010)	21
Figura 1.8: Domini applicativi e scenari principali rilevanti dell’IoT. (Atzori et al., 2010)	23
Figura 1.9: Componenti di un sistema IoT. (Jia et al., 2019)	25
Figura 1.10: Architettura IoT a tre strati (Jia et al., 2019)	26
Figura 2.1: Stacking dei livelli gerarchici di BACS (Vieira et al., 2020)	29
Figura 2.2: Gerarchia funzionale a tre livelli di un sistema di automazione. (Kastner et al., 2005)	31
Figura 2.3: Topologia di rete EIB/KNX. (Kastner et al., 2005).....	37
Figura 2.4: Tipica architettura di un nodo di rete in LonWorks.	39
Figura 2.5: Tipica gerarchia logica delle sottoreti in LonWorks. (Kastner et al., 2005) .	41
Figura 2.6: Stack del protocollo ZigBee. (Vasseur & Dunkels, 2010).....	43
Figura 2.7: Esempio di (B-OVS), lo StruxureWare Building Operation - Automation Server di Schneider Electric.	48
Figura 2.8: Esempio di B-OD della serie SmartX di Schneider Electric.	48
Figura 2.9: Esempio di B-BC, l’Andover Continuum™ bCX1 di Schneider Electric.	49
Figura 2.10: Esempio di B-AAC, gli SmartX IP Controllers MP-C and RP-C di Schneider Electric.	50
Figura 2.11: Esempio di B-ASC, lo SmartX IP Controllers IP-IO Modules di Schneider	

Electric.....	50
Figura 2.12: Esempio di B-SA della serie EM4800 multi-circuit energy meter di Schneider Electric.....	51
Figura 2.13: Esempi di B-SS, gli SXWS sensors di Schneider Electric.	51
Figura 2.14: Un oggetto standard BACnet, in particolare un ingresso analogico. (Swan, 1996)	54
Figura 2.15: Programma giornaliero di gestione illuminazione utilizzando Lighting Output.(Wichenko, 2013).....	61
Figura 2.16: Esempio di utilizzo di un Pulse Converter associato ad un Accumulator..	64
Figura 2.17: Elenco delle proprietà degli oggetti BACnet. (Swan, 1996).....	66
Figura 2.18: Schema esemplificativo di un dispositivo BACnet, con la sua aggregazione interna di BACnet Objects e le relative proprietà. (Tang et al., 2020).....	68
Figura 2.19: Elenco dei BIBBs per i singoli dispositivi standard BACnet.	73
Figura 2.20: Interfaccia della pagina web di BACnet International dedicata ai PICs&BIBB's per Schneider Electric. (Fonte: https://www.bacnetinternational.net/btl/)	75
Figura 2.21: Proprietà supportate da oggetti appartenenti ad un controller B-AAC. (Schneider Electric, 2013).....	76
Figura 2.22: BIBBs supportati da un Advanced application Controller B-AAC.(Schneider Electric, 2013).....	77
Figura 2.23: Marchio BTL. (Fonte: https://btl.org/)	78
Figura 2.24: Esempio di certificazione BTL per un controller B-AAC.(BACnet International, 2021).....	79
Figura 2.25: Lo stack di BACnet e gli strati equivalenti dello stack ISO-OSI. (Bushby T, 1997)	80
Figura 3.1: Indicazioni dalla versione IFC2x3 per una mappatura tra entità dello standard IFC e tipi di oggetti definiti negli standard per l'automazione degli edifici. (BuildingSMART International, 2020).....	88
Figura 3.2: Specificazioni dell' <i>IfcBuildingControlsDomain</i> e descrizione delle entità riguardanti l'automazione nella versione IFC4ADD2 al capitolo 7.2. (BuildingSMART International, 2022).....	89
Figura 3.3: Diagramma di istanza per la classificazione di un <i>IfcObjectDefinition</i> . (BuildingSMART International, 2020)	92
Figura 3.4: Processo di implementazione del BACnet MVD. (Tang et al., 2020)	115

Figura 3.5: Mappatura dei processi per la progettazione di BACS su BIM e rappresentazione dello scambio di informazioni operative. (Tang et al., 2020).....	116
Figura 3.6: Requisiti di esportazione e importazione per sistemi BACS nella MVD BACnet (Tang et al., 2020)	117
Figura 4.1: Struttura del sistema EcoStruxure di Schneider Electric. (Schneider Electric, 2022).....	124
Figura 4.2: Architettura completa di EcoStruxure™ Building. (Schneider Electric, 2022)	128
Figura 4.3: A sinistra concetto ed esempi di Binding da proprietà a proprietà, a destra concetto ed esempio di Link tra proprietà ed oggetto.	129
Figura 4.4: Visualizzazione di Binding e Link nell'interfaccia di EBO.	129
Figura 4.5: Esempio di binding di oggetto di invio e ricezione appartenenti a server differenti.	130
Figura 4.6: Finestra di dialogo dedicata alle impostazioni di trasferimento valori.....	131
Figura 4.7: Esempio di binding di oggetti di invio e ricezione appartenenti al medesimo server.	131
Figura 4.8: Le tre topologie di rete implementate in EBO.	133
Figura 4.9: BACnet/IP con Automation Server (sinistra) ed Enterprise Server (destra).	136
Figura 4.10: Finestra di dialogo in EBO dedicata alle proprietà di una BACnet Interface, a sinistra il caso con un Enterprise Server e a destra quello con un Automation Server.	137
Figura 4.11: Finestra di dialogo in EBO dedicata alle proprietà dell'IP Network.....	138
Figura 4.12: Finestra di dialogo in EBO dedicata alla creazione di dispositivi BACnet.	139
Figura 4.13: Elenco ad albero degli oggetti BACnet in EBO.	140
Figura 4.14: Pianta del modello BIM in Revit.	141
Figura 4.15: Sensore SmartX di Schneider Electric.....	141
Figura 4.16: Comando luce SpaceLogic MTN625199 di Schneider Electric.	142
Figura 4.17: Attuatore SpaceLogic MTN649202 di Schneider Electric.....	142
Figura 4.18: Quadro elettrico e relativa interfaccia Pragma di Schneider Electric.....	143
Figura 4.19: Parametri IFC in Revit da definire.....	149
Figura 4.20: Processo di mappatura dei parametri IFC in Revit2023.	150
Figura 4.21: Parametri IFC in Revit definiti.....	150

Figura 4.22: Versioni dello standard IFC selezionabili nell'esportazione di un file IFC.	151
Figura 4.23: Elenco dei dispositivi Schneider Electric della Serie SmartX.....	152
Figura 4.24: Creazione dell'oggetto "Comando accensione luce" all'interno dell'attuatore.....	152
Figura 4.25: Creazione dell'oggetto "Comando accensione luce" all'interno del Comando luce.....	153
Figura 4.26: Interfaccia BACnet popolata di tutti i dispositivi e oggetti utili al progetto di sperimentazione.....	154
Figura 4.27: Finestra di dialogo in EBO dedicata ai binding.....	155
Figura 4.28: valori associati all'oggetto "Richiesta accensione luce".	155
Figura 4.29: Binding del Valore dell'oggetto "Richiesta accensione luce" con il digital output.....	156
Figura 4.30: Binding con la porta DO1 dell'attuatore a cui è associata la "Richiesta accensione luce".....	156
Figura 4.31: Porta DO1 usata come terminale del "Comando accensione luce".	156
Figura 4.32: Interfaccia BACnet (riga 10) con il relativo attuatore (riga 13).....	157
Figura 4.33: Oggetto "Richiesta accensione luce" (riga 53) contenuto nell'attuatore.	157
Figura 4.34: Oggetto "Comando accensione luce" (riga 80) contenuto nell'attuatore.	157
Figura 4.35: Dispositivo "Comando luce" (riga 108) con relativo oggetto "Comando accensione luce" (riga 119).	158
Figura 4.36: Dispositivo "Sensore di presenza" (riga 134) e relativo oggetto "presenza persone" (riga 145).....	158
Figura 4.37: Interfaccia di Visual Studio Code, pagina iniziale.....	159
Figura 4.38: Procedura iniziale di scrittura codice per il parsing dei file.....	160
Figura 4.39: Procedura di ricerca dei dispositivi domotici nel file IFC per categoria. .	160
Figura 4.40: Entità indagate "by_type" visualizzate per categoria nel terminale.....	161
Figura 4.41: Ciclo for per la ricerca di attuatori.....	162
Figura 4.42: NAME e TYPE relativi all'attuatore nel file xml esportato da EBO.	162
Figura 4.43: Proprietà dell'attuatore stampate dal codice scritto in Python.	162

Figura 4.44: Procedura di realizzazione del parametro “IO Resources” con relativi valori associati.	163
Figura 4.45: Parametri “NAME” e “TYPE” relativi al dispositivo sensore di presenza.	164
Figura 4.46: Scrittura della funzione in grado di restituire proprietà e valori specificati all’interno del codice.....	164
Figura 4.47: Funzione “write” per creare il nuovo documento xml.	165
Figura 5.1: Nuovo attuatore creato dal programma scritto in Python.....	167
Figura 5.2: Nuovo comando luce creato dal programma scritto in Python.	167
Figura 5.3: Nuovo sensore di presenza creato dal programma scritto in Python.	167
Figura 5.4: Attuatore inizialmente esportato da EBO, proprietà, prima parte.	167
Figura 5.5: Attuatore inizialmente esportato da EBO, proprietà, seconda parte.....	168
Figura 5.6: Creazione in Python dell’oggetto BACnet “Richiesta accensione luce”. ...	169
Figura 5.7: Procedura di stampa in Python delle relazioni tra dispositivi di un circuito in Revit.	170
Figura 5.8: Esempio di gruppo di relazione tra dispositivi in Revit, stampata nel terminale in Python.	170

