Revista de Științe Politice. Revue des Sciences Politiques • No. 92 • 2026: 185 – 199

ORIGINAL PAPER

Pragmatic Aspects of Irrealis Tools in the Language of Information Technology

Irina Janina Boncea[1)]

Abstract
In post-communist EFL instruction, focus has shifted from the grammar translation approach to the communicative approach in order to satisfy the increasing need for globalisation in trade and communication. The current article seeks to investigate a set of pragmatic aspects regarding the way in which the English language associated with the Information Technology field makes use of the wide range of irrealis tools and hedging to convey non-factual information. The need for the current research derives from the inherent factuality engineering fields, i.e. from the declared focus on **what is** rather than **what is desirable** or **possible**. Nonetheless, a stringent need for non-factuality is manifesting through a wide range of presumptive tools, both epistemic and deontic, which the current research will attempt to analyse pragmatically within the framework of native Romanian students studying English for Computer Science.

Keywords: *conditional, EFL, ESP, ICT English, Irrealis, non-factual, subjunctive*.

[1)] Lecturer, PhD, University of Craiova, Faculty of Letters, Department of Applied Foreign Languages, specialisation: English for Specific Purposes, Phone 0040724083333, E-mail: irina.boncea@edu.ucv.ro ORCID ID 0000-0002-9277-0234

1. Introduction

As the field of technology advances and permeates every aspect of modern living, students enrolled in English study programs related to information and computer technology are experiencing a dire need to master English as it will become part and parcel of their daily communicative exchanges in the field of work. Post-communist EFL instruction saw the rise of English for Specific Purposes as a newly emerging necessity with an array of challenges for students and teachers alike. Technical English instruction, particularly specific of the Information and Computer Technology field, has been developing alongside technology itself and has provided a wide variety of learning and teaching tools and aids in the attempt to keep students up-to-date with what in new in the field. What seems to be obvious about the ICT field is that it is primarily concerned with what **is** rather than what **might be**, with the factual rather than the non-factual. The present paper seeks to clarify some aspects regarding non-factual grammaticalisation in English and how it lends itself to the teaching and learning of Computer English.

The linguistic manifestations of **irrealis** are often overlooked during ESP classes, resulting in the incomplete or incorrect understanding of irrealis tools (moods and other markers) particularly subjunctives and conditionals, which the current paper will investigate more in depth. It is often the case that students are not aware of the difference between the factuality of the indicative and the counter-factuality of the irrealis moods, which often results in misunderstanding the message or in ill-formed statements. Consequently, it is of great didactic interest that the source of misunderstanding be clarified and tackled during ESP classes.

In linguistics, **irrealis** is represented by the set of grammatical tools (moods and markers) indicating that a particular situation or action is not known to be factual or is explored as hypothetical at the moment of the utterance, in contrast with **realis**, which refers to past or present factual occurrences and events. An irrealis category has been defined as mapping "potential and counterfactual meanings and no actual meanings" onto an utterance, or it acts as a semantic domain describing possible and counterfactual worlds (Prince, Krajinović, Krifka, 2022:221). The term was traced back to the late nineteenth century, to the German literature based on Greek and Latin academic writings. According to Autenrieth (1875:5) '***irrealis*** means that, for the speaker, the action is excluded from reality, probability and possibility, that it is untrue or impossible or is thought of that way'. At the beginning of the 20th century, the term is used by Sapir to make reference to a suffix in Paiute rendering ***irrealis*** as follows: 'This element indicates that the activity expressed by the verb is unreal, i.e. either merely potential or contrary to fact (potential in past time)' (Sapir 1930:168). In subsequent years, the term ***irrealis*** was also further employed by Bickerton (1975), Dixon (1980), Johnston (1980), Givón (1982), Chung & Timberlake (1985), Foley (1986), Blewett (1991), Kimball (1991) and Mithun (1995) to makes reference to "constructions encoding some type of unrealized states of affairs, e.g. possible or future states of affairs... roughly equivalent to more traditional terms such as, e.g., future, subjunctive, or conditional" (Kimball, 1991: 190). Irrealis tools are used in statements about events that are desired, conditioned, mandated and hoped rather than factual (in conditionals, subjunctives, future, imperatives, interrogatives etc.). Though every natural language displays grammatical ways of expressing unreality and counter-factuality, literature seems to reserve the term ***irrealis*** for particular morphological markers or clause types that introduce non-factuality into a statement.

A variety of elements across languages have been argued to act as proof of a grammatical category of ***irrealis***, directly referring to the way in which speakers render the states of affairs in terms of actuality, such as the marking of TAM (tense, aspect, mood), polarity (positive vs. negative sentences) and illocutionary force (declarative in opposition to imperative and interrogative sentences). While traditional grammars examine ***irrealis*** as a strict morphological or syntactic category (like verb inflections for the subjunctive or conditional), pragmatics is more focused on how speakers use irrealis structures to convey social, epistemic, transactional and interactional meanings (McGregor, Wagner, 2006:340):

• **Speaker Stance/Attitude**: Irrealis markers reflect a speaker's subjective degree of low certainty, doubt, hope, or fear regarding a statement. For instance, an utterance such as "If I were you..." uses a subjunctive form not just to state a condition, but to pragmatically signal distance and detachment from the situation, a polite word of advice or an encouragement to pursue a particular course of action.

• **Contextual Inferences**: Irrealis meanings are evidently contextually-bound, i.e. they do not always render a singular rigid truth-value. Instead, listeners calculate pragmatic inferences (**implicatures**) within the framework of a particular context— an irrealis form may imply a polite request, a remote possibility, a warning, or a counterfactual impossibility. For instance, a statement such as "If I knew who hacked into the website I would go to the police myself" can be 1.a request for a name, 2. possibility that I might find the name one day 3. warning to anyone who might attempt this again 4. statement of impossibility on account of lack of knowledge.

• **Epistemic** vs. **Deotic** overlap: the Epistemic notions of probability, possibility and deontic notions such as obligation, permission, often merge with irrealis marking. Pragmatics is often concerned with distinguishing between the illocution and the illocutionary effect (intention) and the perlocution (actual effect) an utterance has on the other participants when "unrealized" event shifts in conversation between "you should do this" or "it might be true."

Example:

"It is expected that the project **be finished** by noon."

-Deontic reading (Obligation / Rule): It is a requirement, duty, or rule set by someone in authority that the deadline be at noon (team members are required by rule to complete the project by noon).

-Epistemic reading (Inference / Likelihood): Based on the evidence or logic available, we conclude that the project will be already finished by noon (Given the facts, tem members will probably finish)

2. Irrealis marking and contextualisation of irrealis meaning

A variety of meaning clusters becomes apparent when counterfactual utterances are made depending on the pragmatic intentionality of the speaker, as identified in the following list with examples specific to English for Computer contexts:

Irrealis marking	Meaning, as intended by speaker	Example
Subjunctive • **Present** • **Past** • **Perfect**	Event is considered unlikely (mainly used in dependent clauses).	"It is essential that he **finish** the debug on time." "The manager recommended/advised/ urged that the debug **be** done."

Irrealis marking	Meaning, as intended by speaker	Example
		"I wish this update **weren't** necessary." "It's time they **had finished** the update."
Conditional Type 0,1,2,3, mixed	Event depends on a stated condition.	0 "If we enter the password, we log in." 1 "If I enter the password, we'll log in." 2 "If entered the password, we'd log in." 3 "If I had entered the password, we would have logged in." Mixed "If I **had entered** the password, we would be logged in."
Imperative Prohibitive	Event is directly ordered, requested or prohibited by the speaker.	"Start the update!" "Do not reboot now!"
The following meaning clusters are not grammaticalised in English and are often induced by contextual elements such as modal adverbs or verb semantics that render the state of affairs desirable, doubtful, presupposed therefore non-factual:		
Desiderative	Event is desired/ wished by a participant in the state of affairs referred to in the utterance.	"More time was needed for developing that feature." "Hopefully the updates resolve the issue!"
Dubitative	Event is uncertain, doubtful	"She seems to ignore the viral threat messages." "The software appears to have been updated."
Hypothetical	Event is hypothetical or counterfactual, but possible.	"I might be able to solve this issue." "The key should be in that folder"
Presumptive	Event is assumed, presupposed by the speaker based on known information	"Knowing the way this code works, I wouldn't bet on it." "That code alteration might/can't work."
Permissive	Event is permitted by the speaker.	"You may [not] reboot the system now!" "The system can be rebooted now."
Mirative Admirative	Event is surprising or amazing (literally or sarcastically).	"Wow! They might actually know programming!" "Apparently, they know programming."
Inferential	Event is not witnessed and not confirmed.	"Something tells me the problem lies between desktop and chair." "The issue seems to be that code line."

Irrealis marking	Meaning, as intended by speaker	Example
		“It looks as if there is a problem with the CPU.”
Interrogative	Event is asked or questioned by the speaker	"Does the program do its own updates?" “Is there smoke coming from the CPU?”
Concessive	Event is presupposed or admitted as part of a refutation.	"Even if they troubleshoot the CPU, the issue will persist."; "Although she solves the key issue, it will not work."

2. The Subjunctive in English for Computer Science

The **subjunctive mood** represents a particularly powerful irrealis tool in technical writing and is often used in computer science documentation. While the indicative mood states factual information ("The server *is* active"), the subjunctive renders hypothetical scenarios, i.e. contrary to fact situations, explorations, necessary requirements, dependencies as well as ideal system states. ESP classes often include technical writing tasks that might benefit from clarifications regarding the variety of tools for expressing irrealis that students can employ to render their technical writing more powerful.

The semantic charge of irrealis for the **present subjunctive** is generated by the occurrence of impersonal constructions with adjectives and nouns instantiating urgency or necessity but also by verbs within the semantic field of persuasion (to recommend, advise, urge, suggest, propose, advise etc.) followed by a that-clause. In these structures, the base form of the verb is used and no agreement is made between subject and verb.

Examples:

It is critical/mandatory/necessary that the server **be** active when the update starts.

It is of great/the utmost importance that the server **be** active.

They recommend/ advise that the server **be** active.

The protocol requires that each client **send** an authentication token before initiating a stream.

It is essential that every payload **contain** a valid checksum field.

The absence of a that-clause following adjective, nouns in impersonal constructions or mandating verbs suspends the need for a subjunctive.

Examples:

Subjunctive: It is vital **that** the migration script **run** inside a transaction block to prevent partial database updates.

No subjunctive: It is vital **for** the migration script **to run** inside a transaction block to prevent partial database updates.

Subjunctive: We suggest that the function **return** an empty array rather than null on failure.

No subjunctive: We suggest the function **to return** an empty array rather than null on failure.

When using the subjunctive construction, what often baffles students is the instinctive need to form the indicative subject-verb agreement as they do not realize the presence of the ***irrealis*** in the occurrence of the subjunctive and interpret the statement as factual rather than desirable or hypothetical. An essential idea to be retained is that the subjunctive, be it present, past or perfect, illustrates the idea that the speaker is actively trying to mandate or alter a course of action or state of affairs through the statement that tries to contrast with what actually is.

Contrast: **Subjunctive** (non-factual= desirable) vs. **Indicative** (factual- as is)

Example 1

Subjunctive (Rule/Requirement): "The configuration specifies that the timeout **be** set to 30 seconds." (Meaning: This is the mandated rule) **Indicative** (Statement of fact): "The configuration specifies that the timeout **is** set to 30 seconds." (Meaning: It is currently a known fact that it is set to 30 seconds)

Example 2

Subjunctive: The security team insist that the application **not store** plaintext passwords in local storage. (Meaning: it is important that the feature of not storing be implemented into the app)

Indicative: The security team insist that the application **does not store** plaintext passwords in local storage. (Meaning: the application does not store plaintext passwords at present but some participants are not convinced that this is true).

More examples in contrast:

Indicative (Fact/Reality) = what is	**Subjunctive (Requirement/Hypothetical)=what might/should be**
The algorithm **processes** data in real-time.	It is vital that the algorithm **process** data in real-time.
The user **toggles** the dark mode switch.	The UI design demands that the user **toggle** the switch to proceed.
The secondary server **is** ready.	It is of dire need that the second server **be** ready so we can run the test.

The use of the subjunctive allows technical writing to be more polished and oftentimes clearer in establishing rapport, drafting responsibilities and formulating tasks.

Area of computing	**Subjunctive – formal**	**Alternative- no subjunctive- informal**
System Requirements and Limitations, Constraints	The protocol requires that the client **send** an authentication token with every request. It is recommended that antivirus **be** installed.	The protocol requires the client to send an authentication token with the request. Antivirus must/ should be installed.

System Criticality	It is essential that the database **be** backed up before the migration begins.	It is essential for the database to be backed up before the migration begins.
User Mandates	We recommend that the administrator **restrict** root access to unauthorized IP addresses.	We recommend the administrator to restrict root access to unauthorized IP addresses.
Software Specs	The specification demands that the payload **conform** to JSON standards.	The specification demands the payload to conform to JSON standards.
Error Handling	Safety guidelines insist that the application **terminate** the process if a memory leak is detected.	Safety guidelines insist on the application terminating the process if a memory leak is detected.

Past and **perfect subjunctive** are instantiated by the same set of expressions, but while the past subjunctive indicates a present hypothesis, the perfect subjunctive renders a hypothesis about a past that is parallel to the factual past, therefore unreal.

1. **Past Subjunctive** hypothetical **present** – contrary to facts Looks like **Past tense** of the indicative To be= WERE in all persons Often occurs in 2-nd type conditionals	2. **Perfect subjunctive**- hypothetical **past** – contrary to what really happened Looks like **past perfect** of the indicative Often occurs in 3-rd type conditionals
1. and 2. occur after *the same phrases*, with a difference in meaning: IF ONLY.. IT'S (HIGH TIME)... I'D SOONER +different subject !(same subject calls for short infinitive: I'd sooner/rather (not) go.) I'D RATHER + different subject ! I'D BETTER (I/YOU HAD BETTER) + SHORT INF.: I/you'd better go. I'D PREFER IT IF YOU... ...AS IF ... I WISH...	

PRESENT MEANING	**PAST MEANING**
If only they solved the issue!	If only they had solved the issue!
It's time we finished the meeting. (the meeting is now over)	It's time we had finished the meeting.(the meeting is long overdue)
I'd rather we installed the app.	I'd rather we had installed the app.

I'd sooner he wrote the code...	I'd sooner he had written the code...
He acts as if he owned the place. (he doesn't own it)	He acts/acted as if he had owned the place. (he didn't own it)
I'd prefer it if you didn't reboot the computer.	I'd prefer it if you hadn't rebooted the computer.
They wish the computer didn't break down so often.	They wish the computer hadn't broken down so often.

Area of computing	**Past subjunctive-hypothetical present**
Simulations and Testing	**I wish server were** offline so the API **would not return** a 503 error. *Meaning:* The server is online right now, but we are testing or describing a hypothetical alternative.
Configuration and functionality drawbacks	The app behaves as if it **ran** in a sandboxed environment." *Meaning:* The app does not run in a sandbox, but its behavior is awkward.
User Preferences and State Checks	**I wish this database supported** real-time streaming. *Meaning:* The database currently lacks this feature.
	Perfect Subjunctive – hypothetical past
Post-Mortem Analysis & Log Review	**I wish the packet had arrived** on time so the transaction wouldn't have timed out. Meaning: The packet arrived late, so the transaction timed out.
Error Handling & State Reversals	The compiler acts **as though it had parsed** the syntax tree already. Meaning: It has not parsed it yet.
Configuration Regrets in Documentation	**I'd rather we had enabled debug mode** before the crash occurred. Meaning: We did not enable debug mode, and the crash occurred.

3. The Conditional

During technical English classes, it is often overlooked that the verbal mood in conditional subordinate clauses is, in fact, the subjunctive. That is probably why students are rarely aware of its ***irrealis*** nature and often mistake it for the indicative and its factual interpretation. Moreover, failure to identify the subjunctive as an irrealis tool is possibly

the reason why students have difficulty remembering the tenses they are supposed to use when marking conditionals.

0 type If the server ***is*** active it updates its parameters.
1st type If the server ***is*** active, it will update its parameters.
2nd type If the server ***were*** active when it would update its parameters.
3rd type ***Had*** the server ***been*** active, it would have been able to update its parameters.

Employing subjunctives allows engineers and technical writers to describe how systems are expected to behave under specific, ideal conditions without claiming that they are currently doing so. What is interesting about the subjunctives occurring in conditionals is the fact that they are often mistaken with indicatives (present, past and past perfect of the indicative).

3.1 Type 0 conditionals

Example: If the server ***is*** active, it updates its parameters.

0 type conditionals are used to depict what is expected to happen **every time** the condition is met. Though this may seem factual, it is not. It is certain to become factual, though, as soon as and only if the condition is met. 0 type conditionals are particularly useful in describing what is to be expected if the condition is met.

More examples:

• If you **press** Ctrl + S, the file **saves** automatically. =what to expect if you press…
• The system triggers a low-power warning if the battery level drops below 5%,. =hypothetical scenario conditioned by a drop below 5%
• If the input is null, the function returns an error.= what to expect when/if the input is null

3.2 Type 1 conditionals

Example: If the server ***is*** active, it will update its parameters= the server is not active, only if it becomes active will it update its parameters.

The subjunctive explores what is likely to happen in the future when and if the server becomes active, which it is currently not. In fact, what the subjunctive does is to borrow the future meaning from the main clause in exploring what is to be expected in the future if/when the condition is met. 1st type conditionals act as promises or likely-to-happen projections into the future.

More examples:

• The app will grant access if the user **enters** a valid password.=promise
• If the server response **takes** longer than 30 seconds, the client will timeout.= warning
• If this script **is** compiled, a syntax warning will appear.=likely scenario
• The script will fail unless you **install** the required dependencies.=warning

An alternative type one conditional includes instructions formulated with an imperative in the main clause replacing the future. In opposition to the previous exmples, the imperative urges action to prevent potential negative scenarios depicted by the conditional or to enable action following a sequence of hypothetical events:

Examples:

- Always **back up** your database in case the migration crashes.=prevent hypothetical mishap
- **Compile** the script now if the error appears on the screen.= if clause triggers action
- If the application crashes on launch, **clear** the cache and restart your device.= the hypothesis in the conditional clause triggers a sequence of actions
- Should a 401 error occur, **redirect** the user back to the login screen immediately.=hypothetical unlikely scenario

3.3 Type 2 conditionals

Example: If the server ***were*** active, it would update is parameters= the reality at present is that the server is not active.

The past subjunctive looks like past tense but it is not past (in fact it is often called **unreal past**), but rather borrows the meaning of present conditional in the main clause to explore an alternative reality, a parallel world that is:

a. purely hypothetical at present, i.e. in contrast to factual reality.
b. more tentative than 1st type likely conditionals

More examples:

• **If we had** more RAM, this virtual machine would run much faster.= we don't have more RAM, it is also unlikely we'll get more, just exploring a hypothetical scenario=complaining

• **If I were** the system administrator, I would disable guest access immediately.= word of advice

• **If the API supported** GraphQL, we wouldn't need so many REST endpoints.=wishful thinking/complaining

Inverted type 2 conditionals lose the explicit IF and can only occur with **were** positioned before the subject. These structures render an increased level of improbability.

Standard Type 2 Conditional Exploratory/hypothetical	**Inverted Type 2 Conditional Highly unlikely scenario**
If a data breach exposed customer PII, the company would face substantial regulatory penalties.	**Were a data breach to expose** customer PII, the company would face substantial regulatory penalties.
If we migrated our entire database infrastructure to a distributed ledger, latency would increase significantly.	**Were we to migrate** our entire database infrastructure to a distributed ledger, latency would increase significantly.
If the integer were larger than 64 bits, it would cause a buffer overflow. (Implies it currently fits within the boundary, explores hypothetical event).	**Were the integer larger than 64 bits,** it would cause a buffer overflow. (Implies it is unlikely it will ever exceed the boundary).

Further examples of computing situations in which second conditional sentences sound natural in discussing hypothetical, imaginary or currently unfeasible scenarios:

a. Project Management & Deadlines (The "If We Had More Time/Money")

Software Development teams frequently use the second conditional to discuss ideal features or scope changes that cannot happen due to strict deadlines or budgets.

- **If we had a bigger budget**, we would implement machine learning algorithms.
- **If the deadline weren't** next week, we would rewrite this entire module.

b. Cybersecurity & Threat Modeling (The "What If" Breach Scenario)

Security analysts use the second conditional to discuss theoretical vulnerabilities and the hypothetical fallout of cyber-attacks.

• **If a hacker gained root access**, they would delete the entire database.

• **If users chose stronger passwords**, phishing attacks wouldn't be as successful.

c. System Architecture & Tech Stack Choices (The "Alternative ")

Engineers use second conditionals when comparing their current tech stack to an alternative architecture they aren't actually using.

• **If we hosted the app** on our own hardware, maintenance would take too much time.

• **If the platform used a microservices architecture**, scaling specific features would be much easier.

d. Code Debugging & Troubleshooting (The "Missing Pieces" Problem)

Software developers use it during code reviews or debugging sessions when they lack the tools, permissions, or data needed to fix a bug immediately.

• **If we could replicate this bug locally**, we would fix it in five minutes.

• **If the error logs were more detailed**, we would identify the root cause instantly.

e. User Interface Design & User Behaviour (The "Ideal User/Machine" Assumption)

Product designers often use second conditionals to explore how users might interact with an interface if it were designed differently.

• **If the checkout button were brighter**, more users would complete their purchases.

• **If the app supported dark mode**, it would look much better at night.

3.4 Type 3 conditionals for unreal past

The 3rd type conditional explores an alternative past world that might have been if the condition had been met, therefore it projects the imaginary exploration of some events subsequent to the hypothetical fulfillment of a past condition.

Example: If the server ***had been*** active, it would have updated its parameters.= The server was not active therefore it failed to update its parameters.

In programming logic and architecture planning, students often explore "what-if" scenarios that are contrary to the reality of the indicative. The perfect subjunctive is used to introduce events that never took place, realities that could have been but never were. The challenge in these cases is double fold: to acknowledge the fact that the unrealized scenario is an exploratory past reality and that the apparent past perfect tense is not genuinely an anterior past. 3rd type conditionals often display the illocutionary effect of **regret** or **reproach**.

Making a Reproach

• **If the DevOps team had configured** the monitoring alerts correctly, they would have detected the memory leak before the server crashed.

• **If the security patch had been deployed** to the staging environment first, the QA team would have noticed the dependency conflict.

Expressing Regret (Ist person singular or plural)

• **If we had automated our database** backups daily, we would not have lost three days of production data during the ransomware attack.

• **If our team had limited incoming requests**, the infrastructure would have survived the unexpected traffic spike.

Inverted Type 3 Conditionals (Highly Formal Technical Writing)

Just as Type 2 conditionals can be inverted, Type 3 conditionals can drop the IF and instantiate HAD in initial position, frequently used in formal executive summaries of technical incidents.

Examples:

• **Had** the load balancer **distributed** the traffic evenly, the primary node would not have become overwhelmed.

• **Had** the user **updated** their browser to the latest version, the cross-site scripting (XSS) vulnerability would not have executed.

• **Had** we **refactored** the monolith earlier, scaling the micro-services would have been much easier during the product launch.

Common computing contexts for 3rd type conditionals:

- **Post-Event & Error Analysis:**

If the developer **had sanitized** the input, the server would have survived the injection attack. (Reality: The developer didn't sanitize it, so the server crashed).

If we **had updated** the firmware, the memory leak wouldn't have occurred.(Reality: We didn't update it, so it leaked).

- **Performance & Optimization Regrets, cause/effect=regret:**

If the query **had been indexed** properly, the database could have finished the lookup in milliseconds. (Reality: It wasn't indexed so it timed out).

If we **had allocated** enough RAM, the container might not have thrown an OutOfMemory error. (Reality: RAM wasn't allocated so it crashed).

- **Version Control / Deployment Scenarios:**

If CI **had caught** the syntax error, we wouldn't have pushed the broken build to production. (Reality: CI/CD missed it so we pushed the product into production).

3.5 Mixed conditionals

A mixed conditional combines a second conditional and a third conditional to connect a hypothetical past event to a present result, or an unreal present state or situation with a past result with irrealis embedded in both situations.

a. Past Condition → Present Result

Meaning: speaker imagines an alternative past event and how it would change the present non-factual situation.

Example

If he had backed up my database yesterday, **he would not be** panicking right now.

Reality: He didn't back it up yesterday, so he is in panic mode now.

b. Present/General Condition → Past Result

Meaning: speaker imagines a different present state or habit and how it would have changed a past event.

Example

If I were a better programmer, **I would have solved** that memory leak last week.

Reality: I am not a good programmer, so I didn't solve the memory leak last week.

In the computing world, mixed conditionals are frequently used in post-mortem analyses, debugging sessions, user stories and system architecture reviews.

Computing Context	**Unreal Condition (If-Clause)**	**Unreal Result (Main Clause)**	**Example Scenario**
Reporting Incidents	Past Action	Present State	If we **had run** the load tests, the site **would be** online.
Creating Technical Debt	Present Reality	Past Outcome	If the code **were** documented, onboarding **would have taken** less time.
Debugging	Past Action	Present Error	If the container **had started**, the logs **would be printing** data.
Product Strategy	Present Capability	Past Milestone	If we **owned** the IP, we **could have integrated** it sooner.

A Structural Comparison between conditional types

The following table explains how a hypothetical past incident (Type 3) differs from future planning (Type 1) or hypothetical present scenario (Type 2):

Conditional Type	Computing Context	Example
Type 1 (Likely Future)	Planning ahead	If we **implement** a firewall, it **will block** malicious traffic.
Type 2 (Hypothetical Present)	Brainstorming	If we **implemented** a firewall right now, it **would block** this traffic.

Type 3 (Past Post-Mortem)	Analyzing the error	If we **had implemented** a firewall, it **would have blocked t**he traffic.
Mixed type (3+2)	Hypothetical Past affecting the present	If we **had implemented** a firewall, the traffic **wouldn't be blocked**.

4. Conclusions

The current paper has attempted to investigate manifestations of the grammatical category of ***irrealis*** ant its pragmatic implications in the ESP area of English for Computers and Information Technology with a special focus on the subjunctive and conditional clauses. The peculiarities of the aforementioned verbal moods in contrast with the factuality of the indicative, the most commonly used and taught verbal mood, have constituted the subject of interest for the present paper as they are often a source of misconception for ESP students and are of particular importance to technical writing. As students migrate into computing professions that entail the writing of software code and technical documentation to accompany it, they are required to acknowledge and correctly employ the manifestations of ***irrealis*** in formulating technical specifications with power and grit. Hence the need for the present research that might prove useful to ESP teachers and students in clarifying and refining the semantic realms of possibilities instantiated by ***irrealis*** marked statements.

References:

Bickerton, D. (1975). *Dynamics of a creole system*. Cambridge University Press Cambridge.

Blewett, S.C. (1991). Irrealis in Manam discourse. *Language and Linguistics in Melanesia*, 22(1-2),1–20.

Chung, S., & Timberlake, A. (1985). *Tense, aspect, and mood. language typology and syntactic description: Grammatical categories and the lexicon, vol. by Timothy Shopen, 202-258*.

Dixon, R. MW. (1980). *The Languages of Australia*. Cambridge: Cambridge University Press.

Downing, A. & P. Locke (1992). A University Course in English Grammar. Hemel Hempstead: Phoenix ELT.

Foley, W.A. (1986). *The Papuan Languages of New Guinea*. Cambridge University Press.

Givón, T. (1982). Tense-aspect-modality: The creole prototype and beyond. Pp. 115–163, in Hopper, Paul J. (ed), *Tense-aspect: Between semantics and pragmatics*. Amsterdam: John Benjamins.

Gosden, H. (1995). Success in research article writing and revision: A social constructionist perspective. *English for Specific Purposes* 14: 37-57.

Grabe, W., Kaplan R. B. (1997). On the writing of science and the science of writing: Hedging in science text and elsewhere. In *Markkanen & Schröder* (eds.), pp. 151-167.

Holmes, R. (1997). Genre analysis and the social sciences: An investigation of the structure of research article discussion sections in three disciplines. *English for Specific Purposes* 16: 321-337.

Hyland, K. (1994). Hedging in academic writing and EAP textbooks. *English for Specific Purposes* 13: 239-256.
Huddleston, R. D.; Pullum, G. K. (2002). *The Cambridge Grammar of the English Language*. Cambridge: Cambridge University Press.
Johnston, R. L. (1980). *Nakanai of New Britain: The grammar of an Oceanic language*. The Australian National University: Pacific Linguistics.
Kimball, G. D. (1991). *Koasati Grammar*. University of Nebraska Press.
Lyons, J. (1977). *Semantics.* Cambridge: Cambridge University Press.
McGregor, W. and Wagner, T.(2006). The Semantics and Pragmatics of Irrealis Mood in Nyulnyulan Languages. *Oceanic Linguistics,* vol. 45 no. 2, 2006, p. 339-379. Project MUSE, https://dx.doi.org/10.1353/ol.2007.0005.
Meyer, P. G. (1997). Hedging strategies in written academic discourse: Strengthening the argument by weakening the claim. In *Markkanen & Schröder* (eds.), pp. 21-40.
Mithun, M. (1995). On the relativity of irreality. In*: Modality in grammar and discourse*. John Benjamins.
Von Prince, K., Krajinović, A., and Krifka, M. (2022). Irrealis is real. *Language*, 98(2), 221–249.

Article Info

Received: October 09 2026
Accepted: November 06 2026

How to cite this article:

Boncea, I. J. (2026). Pragmatic Aspects of Irrealis Tools in the Language of Information Technology. *Revista de Științe Politice. Revue des Sciences Politiques,* no. 92, pp. 185 - 199.