LexFusion AI Full-Lifecycle Patent Management System

Field of the Invention

This invention relates generally to computer-implemented systems for intellectual property management. In particular, it concerns an integrated platform that automates the full patent lifecycle – from drafting patent applications, through prosecuting applications before patent offices, to post-grant enforcement – using advanced artificial intelligence (AI) modules and unified docketing workflows.

Cross-Reference to Related Applications

This application claims priority to and builds upon the co-pending patent application titled *"System and Method for Automated Office Action Response Generation and Submission"* (incorporated herein by reference in its entirety). The present invention extends that framework to additional phases of the patent lifecycle, introducing new modules and functionalities without duplicating the claims of the prior application.

Background of the Invention

Obtaining and enforcing patent rights is a complex, multi-stage process involving distinct tasks: drafting an initial patent application, prosecuting the application (responding to patent office communications such as Office Actions), and monitoring and enforcing the granted patent against infringers. Traditionally, each stage has been handled in isolation by patent professionals, often using separate tools or manual effort. This disjointed approach leads to inefficiency, increased cost, and risk of error.

Application Drafting: Preparing a patent application is labor-intensive and requires careful claim drafting, adherence to jurisdiction-specific formalisms, and consistency between the claims, specification, and figures. Conventionally, attorneys draft claims and specifications manually with general-purpose editors or basic templates. Some AI-assisted drafting tools have recently emerged to aid in generating patent text, but they typically operate as standalone aids. For example, **AI-based patent drafting assistants** can produce preliminary claim language or boilerplate text, but these tools function as isolated suggestion engines within word processors, lacking integration with formal patent templates or downstream processes. They still rely on the practitioner to select appropriate legal templates for each jurisdiction and to manually ensure that the claims, description, and drawings are consistent. In short, while AI can now help with initial patent drafting, existing solutions remain point tools that require significant manual oversight and integration.

Prosecution Automation: Once an application is filed, prosecuting it involves tracking official communications (e.g. Office Actions), meeting deadlines, and drafting responses to examiner rejections. The prior application (incorporated by reference) addressed automation of Office Action response generation and submission. By way of background, various preexisting software tools handled fragments of the prosecution process, but none provided end-to-end automation[1][2]. For example, docket management systems like FoundationIP® or CPI® automatically calendar due dates from patent office emails, but do not generate response documents or analyze rejection content[1]. On the other hand, drafting aids such as Patent

ClaimMaster® (a Word™ plugin) help format responses or check claim numbering, yet they require the attorney to manually input details and do not ingest Office Actions automatically[3][4]. Even newer AI-based tools that suggest Office Action response text (integrated as plugins to word processors) still operate in a *partial* workflow – the practitioner must copy-and-paste the examiner's remarks into the tool and later manually upload the finished response to the USPTO[2]. In summary, prior to the earlier patent, **no system existed that could seamlessly retrieve an Office Action, parse its rejections, draft a complete reply with claim amendments and supporting documents, and electronically file the response with minimal human intervention**[5]. The prior patent filled this gap for the prosecution stage, introducing a secure, end-to-end Office Action response automation platform with AI-generated responses, compliance checks, and one-click USPTO filing.

Post-Grant Enforcement: After a patent is granted, patent owners must monitor the market for potential infringement and manage enforcement actions (such as licensing discussions or litigation) when unauthorized use is detected. Historically, this enforcement phase is handled manually or with disparate tools. Patent attorneys and in-house counsel typically rely on their own market intelligence, industry news, or analyst reports to identify infringing products. Specialized patent analytics services can assist by correlating patents to products, but these require expert users and are not integrated into the owner's workflow. For instance, companies may use spreadsheets or separate docketing systems to track enforcement actions (cease-and-desist letters, lawsuit filings, licensing negotiations), which are disconnected from the prosecution docket. While there are **AI tools that can help identify possible infringement by analyzing product information against patent claims**[6], these tools operate in isolation, focusing on data analysis without automatically initiating enforcement workflows. No existing solution ties automated infringement detection into a docketing system that triggers and tracks enforcement steps through resolution. In practice, this means if an AI flags a potential infringing product, human experts must still manually confirm the infringement, draft notification letters or licensing offers, and create calendar entries for follow-ups or legal deadlines. The lack of integration here can lead to missed opportunities for timely enforcement and increased administrative overhead.

Figure Drafting and Specification Consistency: Another challenge throughout the patent lifecycle is the creation of patent drawings (figures) and maintaining consistency between those figures and the written specification and claims. Patent drawings are typically produced by illustrators based on the text description, a process that can be slow and prone to miscommunication. Each component shown in the figures must be described and numbered in the specification. Keeping track of reference numerals and updating the text as drawings evolve is tedious and error-prone when done manually. Some software tools assist with generating simple flowcharts or ensure that reference numbers are used consistently, but **no available tool can automatically generate candidate patent figures from the written description and synchronize those figures with the specification in real-time**. Typically, attorneys must manually insert reference number placeholders in the text, then separately coordinate with draftspeople to produce corresponding figures. If a figure is revised (for example, a part is added or renumbered), the burden is on the drafter to update the specification and claims accordingly. This creates potential for inconsistencies and omissions.

In sum, **the current state of the art provides only piecemeal assistance at different stages of the patent process**. There are point solutions for initial patent drafting, docketing, prior art

searching, Office Action responses, portfolio analytics, and litigation management, but these are not unified. Data and work products are fragmented across tools, requiring manual transfer and increasing the risk of mistakes. Even emerging "end-to-end" IP platforms in the industry are limited. For example, a recent entrant claims to offer an AI-powered patent platform covering steps from invention disclosure to prosecution and portfolio management[7], reflecting the demand for integration. However, such platforms are nascent and, to the extent they exist, **they lack key automation features and fine-grained workflow integration**. In particular, none provide the comprehensive combination of AI-assisted claim drafting with jurisdiction-specific templates, automated Office Action response generation (with one-click USPTO filing), proactive infringement monitoring with claim-to-product matching, and AI-driven figure creation with automatic spec synchronization. This gap in the art results in inefficiencies and inconsistencies across the patent lifecycle.

Novelty and Differentiators

The present invention provides a **unified, full-lifecycle patent management system** that overcomes the above limitations. Key novel features and advantages include:

- **End-to-End Integration:** Unlike prior art point solutions, the system seamlessly connects all phases of the patent lifecycle in one platform[8]. Data flows from the patent drafting stage into prosecution and enforcement, eliminating manual transfer and ensuring consistency.

- **AI-Assisted Drafting with Templates:** The invention introduces an **AI-driven patent application creation module** that not only generates draft claims and specifications, but does so in compliance with jurisdiction-specific formatting rules. This combination of generative AI with built-in legal templates and semantic suggestions is unique; prior drafting aids provide either generic text generation or template libraries, but not an integrated system that intelligently adapts to different patent office requirements while suggesting optimized language.

- **Automated Prosecution Workflow:** Building on the earlier Office Action response automation, the new system embeds that functionality into a larger workflow. The prosecution module automatically dockets incoming Office Actions, parses them, drafts responses with AI, prepares ancillary forms, and submits filings. The novelty here is the **tight coupling with the drafting and enforcement stages** – for example, claim amendments proposed during prosecution are reconciled with the original application text, and docket data (deadlines, filings) flows into the enforcement timeline if a patent issues, without user re-entry.

- **Proactive Enforcement and Licensing Automation:** A groundbreaking aspect of the invention is the **enforcement docketing module** that continuously monitors external data (market news, product releases, etc.) for potential infringement triggers. Upon detecting a product or service that likely falls within the claims of the patent, the system automatically generates an evidence-of-use analysis (e.g., a claim chart mapping claim elements to the suspect product) and can initiate a recommended action, such as drafting a licensing offer or alerting legal counsel. This level of automated infringement detection and action-triggering, integrated with the patent's prosecution history and docket, is not

found in prior systems. It transforms enforcement from a reactive, manual task into a proactive, semi-automated workflow.

- **AI-Generated Figures and Specification Synchronization:** The platform uniquely features a **figure drafting assistant** that uses AI to propose patent drawings (e.g., flowcharts or system diagrams) based on the written disclosure. The user can iteratively refine these suggestions, and once a figure is approved, the system automatically inserts and updates corresponding part reference numbers in the specification and claims. This ensures perfect consistency between drawings and text, something no prior tool achieves automatically. This innovation significantly reduces the time and errors involved in preparing patent figures and aligning them with the narrative.

Overall, the LexFusion AI system differentiates itself by providing **comprehensive automation across the patent lifecycle**. It is not merely a sum of known tools, but a new synergistic combination of modules that share data and context from the invention conception stage all the way to patent enforcement. By preserving attorney oversight at key decision points (to maintain legal and ethical standards), the system offers dramatically improved efficiency while ensuring quality and compliance. No single prior art reference or combination of references teaches or suggests the full scope of this integrated, AI-driven approach.

Summary of the Invention

The invention provides a computer-implemented **full-lifecycle patent management system** that automates and supports patent work from application drafting through prosecution and enforcement. In one aspect, the invention is realized as a **software platform comprising multiple interconnected modules** (see FIG. 1A) that correspond to different stages of the patent process. These modules work in concert and share a common data repository or docketing database to ensure continuity of information. High-level components of the system include:

- **Patent Application Creation Module (111):** An AI-assisted drafting environment that helps users create patent applications. This module includes a **claim drafting engine** driven by a trained language model to generate initial claims and descriptions based on an invention disclosure. It leverages **jurisdiction-specific templates** for formatting (for example, to automatically produce a U.S. patent application format vs. an EPO format) and incorporates a **semantic suggestion engine** that analyzes the drafted text to suggest alternative phrasing, synonyms, or claim scope adjustments. The result is a complete draft specification and claim set, optionally with initial figures, which can be further edited by the practitioner. The module ensures that the draft adheres to formal requirements of the target patent office (such as section headings, claim ordering, abstract length, etc.). Importantly, the drafting module interfaces with the figure generation module (113) to include reference numerals and descriptions of parts consistently between the text and any figures.

- **USPTO Prosecution Automation Module (101–110):** A prosecution management subsystem that automates responding to patent office communications, particularly Office Actions, and handles electronic filings. This module may incorporate the features of the prior Office Action response system (modules 101–110 as described in the incorporated application). It includes a **secure communication sub-module (101)** to retrieve official

Patent Office notices (e.g., via email or API), a **parser (102)** to extract rejection reasons and cited references from Office Actions, an **AI response generator (103)** to draft reply arguments and claim amendments, a **compliance checker (104)** to verify adherence to format rules and completeness, a **deadline calculator (109)** for tracking response due dates, a **forms generator (110)** to prepare any required USPTO forms (e.g., Information Disclosure Statements, Request for Continued Examination, etc.), and a **submission adapter (105)** for electronically filing documents through the Patent Office's online systems. A **user review interface (106)** allows the practitioner to supervise and approve the draft response before filing, and a **credential vault (107)** and **audit log (108)** provide security and accountability (storing login tokens and recording all actions). This prosecution module seamlessly integrates with the drafting module 111: for instance, claim amendments made in response to an Office Action can be traced back to the original claims from module 111, allowing the system to ensure consistency and possibly update the application text if needed. Likewise, any documents generated (responses or new applications) are tracked in a unified docket so that due dates and status are visible platform-wide.

- **Enforcement Docketing and Monitoring Module (112):** A post-grant management component that monitors markets and triggers enforcement workflows. The enforcement module includes an **automated monitoring pipeline** that continuously scans various data sources (e.g., web news, product releases, databases of products, industry publications, etc.) for indications that a third-party product or service might fall within the scope of the user's patent claims. It uses AI-based infringement detection algorithms – for example, natural language processing to compare product descriptions with the patent's claim language – to flag potential matches. When a potential infringement is detected, a **claim-product matching engine** analyzes each element of at least one representative claim against information about the target product, generating a preliminary **claim chart** or evidence-of-use document. A **licensing workflow sub-module** can then automatically draft a notification or licensing offer, populated with details of the patent and evidence of use, and docket a sequence of follow-up tasks (e.g., a reminder to check for response in 30 days, a deadline to file a legal action if not resolved by a certain date). This module essentially provides an **enforcement docket**, meaning it creates and manages calendar entries and action items for enforcement activities just as traditional docket systems do for prosecution deadlines. If the issue proceeds to litigation, the system can integrate with or export data to litigation management tools, and docket key litigation dates (like complaint filing, answer due dates, trial dates, etc.) in the same unified system. The enforcement module ensures that once a patent is granted, the owner's obligations (such as paying maintenance fees, which can also be tracked) and opportunities (licensing or litigation) are monitored and acted upon in a timely, automated fashion.

- **Figure Drafting and Specification Synchronization Module (113):** An AI-powered figure assistance component that aids in creating patent drawings and aligning them with the written disclosure. This module can interpret the text of the patent application (from module 111) to suggest one or more candidate figures. For example, if the invention is a process, the module might generate a **flowchart diagram** identifying the steps as labeled blocks; if the invention is a system or apparatus, the module could produce a **schematic block diagram** showing major components. The figure drafting module utilizes a

combination of trained image generation models and rule-based drawing generation (ensuring compliance with patent drawing conventions, such as black-and-white line drawings, reference numeral lead lines, etc.). It presents the user with an interactive GUI where each suggested figure can be reviewed. The user can **iteratively accept or refine the figure** – for instance, choosing between alternate depictions or adjusting labels. During this process, the module assigns **reference numerals** to each element of the figure and maintains a mapping of those numerals to element descriptions. Once a figure is confirmed by the user, the system automatically **injects the reference numbers and corresponding part names into the specification text and claims**. For example, if the figure has labeled a component as *"processing module (305)"*, the specification's description of that component is updated to include "(305)" after its name, and any mention in the claims can also include the numeral for clarity. The module ensures that all references are consistent and that each figure element is described in the text, vastly reducing the chance of missing an element or mis-numbering. Additionally, if the user later modifies the figure (e.g., adding a new part), the system can update the text accordingly. This two-way synchronization between drawings and text is a unique feature of the platform.

These modules are unified within a single platform so that they share information. For example, the *invention disclosure* initially provided to module 111 not only yields a draft application, but its key concepts are stored and can later be used by the enforcement module 112 to focus monitoring on relevant technology areas or competitors. Similarly, the *docket* of deadlines and actions is unified: when an application is filed and prosecution begins, the response deadlines (from module 109) and eventual patent issuance date flow into the enforcement docket to start the monitoring process. Throughout, a central database stores all patent documents, correspondence, and metadata, enabling advanced analytics (the system can provide reports or dashboards about the portfolio status, which is an added benefit but ancillary to the core invention).

In another aspect, the invention can be described as a **computer-implemented method** for managing a patent lifecycle. This method involves steps such as: drafting a patent application with AI assistance; filing the application; automatically handling patent office communications by generating and filing responses; monitoring for potential infringement of the resulting patent; and initiating enforcement actions when appropriate. In yet another aspect, the invention encompasses a **non-transitory computer-readable medium** containing instructions that cause a computing system to perform the aforementioned method.

By integrating these capabilities, the platform dramatically reduces the time and effort required at each stage of the patent process while maintaining a high level of accuracy and compliance. Practitioners remain in control – the system is designed with *human-in-the-loop* checkpoints (e.g., final review of AI-generated content, approval before filing or sending an enforcement letter) to ensure that critical legal judgments are made by qualified individuals. All automated actions are transparent and traceable via the audit logging (module 108), providing accountability. The result is a **comprehensive patent lifecycle management system** that accelerates workflows, improves consistency (e.g., the same terminology and data carry through drafting, prosecution, and enforcement), and enhances an organization's ability to secure and leverage intellectual property rights.

Brief Description of the Drawings

FIG. 1A is a high-level block diagram of the full-lifecycle patent management system architecture, illustrating the major modules and data flows between patent drafting, prosecution, and enforcement components.

FIG. 1B is a schematic diagram of the Patent Application Creation Module (111) in one embodiment, showing sub-components including an AI claim drafting engine, template and rules library, semantic suggestion engine, and user interface for drafting and editing.

FIG. 1C is a block diagram of the USPTO Prosecution Automation Module, detailing how incoming Office Action communications are processed through various sub-modules (101–110) – including secure intake, parsing, AI response generation, compliance validation, and automated submission – with practitioner oversight.

FIG. 1D is a flowchart depicting an example Enforcement Docketing process. It illustrates the sequence of operations in module 112, from continuous market monitoring and AI-based infringement detection to generating a claim chart and triggering follow-up enforcement actions (licensing offers or litigation steps), all tied into a docket timeline.

FIG. 1E is an illustration of the Figure Drafting and Synchronization Module's interface. It shows an example user interface where an AI-generated figure is presented with numbered elements, and how those elements correspond to highlighted sections of the specification text. Iterative user feedback controls and the automatic injection of reference numerals into the text are highlighted.

Detailed Description of Embodiments

Overall System Architecture (FIG. 1A)

Referring to FIG. 1A, the full-lifecycle patent management system 100 is depicted in an overview form. The system 100 includes several primary modules: an application drafting module 111, a prosecution automation module 115 (which encompasses sub-modules 101–110), an enforcement docketing module 112, and a figure drafting/sync module 113. These modules are implemented in software (and possibly supporting hardware for acceleration) and communicate via one or more data buses or network connections. In some embodiments, the entire system is hosted on a cloud platform (e.g., a secure server environment), accessible through a web-based interface by users (attorneys, agents, or inventors). In other embodiments, certain modules (like drafting 111) may run on a local workstation for responsiveness, while others (like monitoring 112) run on cloud servers continuously – the architecture is flexible, provided the modules can exchange necessary information.

A **common data repository 114** (e.g., a database or knowledge base) is preferably shared among the modules. Repository 114 stores data such as: invention disclosures and draft applications, official communications and responses, docketing information (deadlines, reminders), patent grant information, and external monitoring data. This unified storage ensures that, for example, the enforcement module 112 has access to the final granted patent text and claims (originating from module 111 and possibly amended via module 115) for use in infringement analysis. The

repository 114 can also log all actions (functioning in part as the audit log corresponding to module 108 in the prosecution subsystem).

The modules in FIG. 1A operate in sequence or in parallel at different times in the patent lifecycle. **During drafting (phase 1)**, module 111 is active: it ingests an inventor's input and produces a draft application. Once the draft is approved by the user, the system may directly file it with the patent office – either by passing it to the submission adapter (105) within the prosecution module or via a dedicated filing function. **During prosecution (phase 2)**, module 115 (101–110) comes into play whenever an Office Action or other office communication is received. It interacts with the patent office systems (via secure email or APIs) to fetch communications and then processes them as described below, ultimately docketing any deadlines and preparing responses. **After the patent is granted (phase 3, enforcement)**, module 112 continuously runs to safeguard the patent. It monitors the environment and when a trigger event (potential infringement) is found, it creates an enforcement case in the docket and assists the user in taking action.

Notably, the **user interface layer** (UI) spans across all modules (illustrated conceptually as the top layer of FIG. 1A, though not given a specific reference numeral in the figure). The user interface provides a unified portal where the practitioner can switch between drafting mode, prosecution status, and enforcement alerts. It ensures that, for example, when the user is drafting the application, they can also see any relevant prior art references or similar inventions (fetched via suggestion engine 111), and later, when in enforcement mode, they can review the prosecution history (like file wrapper) or original claim scope easily. This unified UI significantly improves the user experience and reduces context-switching errors.

Security and authentication are handled globally by the system as well. FIG. 1A shows that secure credential storage 107 (from the prior system) is accessible for any operation requiring authentication with external systems (e.g., logging into the USPTO for either filing an application or a response, or accessing subscription-based data sources for monitoring). The communications with external systems (Patent Office servers, web data sources for monitoring) are encrypted and go through secure channels, as represented by the secure communication module 101 at the boundary between the system and external networks.

In summary, FIG. 1A provides the big picture: the inventive system 100 is an integrated platform with modules 111, 115, 112, and 113 covering drafting, prosecution, enforcement, and figure management, all tied together by a common data framework 114 and user interface. The subsequent figures and description will detail each major component.

Patent Application Creation Module (FIG. 1B)

FIG. 1B illustrates an embodiment of the Patent Application Creation Module 111 in greater detail. In this figure, the internal sub-components of module 111 are shown and labeled for clarity. The process flow within this module is as follows.

An **Invention Intake Interface (120)** allows the user (e.g., an inventor or attorney) to input the core details of the invention. This could be in the form of a written description, bullet points of key features, or even an audio transcription or interview answers. The interface 120 may prompt the user with questions to flesh out the disclosure (for example, asking about the problem being

solved, embodiments, advantages, etc.). It serves as the starting point for the AI to understand the invention.

Connected to the intake is the **AI Drafting Engine (122)**. This engine incorporates a trained language model specialized for patent drafting. In some embodiments, the language model is a large language model (LLM) fine-tuned on patent texts, enabling it to generate patent-like prose. The drafting engine 122 uses the inventor's input from 120 and possibly additional context (such as known prior art or classification information) to produce **draft claims (122a)** and a **draft specification (122b)**. It typically begins by formulating a set of independent claims and dependent claims that capture the invention's scope, then generates a corresponding detailed description that supports those claims. The generation can be iterative – for example, the engine might propose multiple variants of a particular claim or description paragraph for the user to choose from.

To ensure that the draft conforms to formal requirements, the module includes a **Template & Rules Library (124)**. This library 124 stores jurisdiction-specific templates and guidelines. For instance, it contains the proper section headings and ordering for a U.S. patent application (Title, Technical Field, Background, Summary, Brief Description of Drawings, Detailed Description, Claims, Abstract), as well as certain boilerplate language (like statements of government support or cross-reference to related applications if needed). If the user intends to file in another jurisdiction (for example, an International PCT application or a European application), the library provides the appropriate format (e.g., EPC rule-compliant formats, reference numeral requirements, etc.). The AI Drafting Engine 122 makes use of this library when constructing the document. The result is that the output is not free-form, but structured in a legally acceptable way from the start.

During drafting, a **Semantic Suggestion Engine (126)** continuously analyzes the text being produced. This engine 126 uses natural language processing to detect potential issues or improvements in the draft. For example, it might identify that a term lacks antecedent basis in the claims, or that a particular phrase could be broadened or clarified. It might suggest synonyms for overly repetitive terms, propose alternative claim phrasing to avoid common §112 pitfalls (indefiniteness), or highlight where an acronym is used without definition. The semantic suggestion engine can also draw from a knowledge base of prior patents to suggest related terminology: for instance, if the invention is in the field of machine learning, the engine might suggest including the term "neural network" or "model training" if not already present, based on similar patents. These suggestions are presented to the user (for example, as underlined recommendations or as a sidebar list). The user can accept or reject each suggestion. This interactive feedback loop helps refine the application's language and scope efficiently, leveraging both the AI's broad knowledge and the attorney's judgment.

The module 111 also integrates with the figure generation module 113 (though in FIG. 1B this is represented conceptually, the actual figure generation process is detailed in FIG. 1E). When the drafting engine 122 produces technical descriptions that imply a figure (for example, if the invention has steps A, B, C, or components 10, 20, 30), it can send a request to module 113 to generate a candidate figure. In practice, the user at interface 120/130 might click a "Generate Figure" button at a certain point in the description. The result (an AI-drafted figure) is returned and can be embedded in the draft. The system then labels parts in the text accordingly.

All through the drafting process, the **User Drafting Interface (130)** allows the practitioner to supervise and edit content. Interface 130 is typically a rich text editor customized for patent drafting. It might display the claims in one panel and the specification in another, highlighting consistency issues. It shows suggestions from engine 126 and allows the user to interact with the figure previews from module 113. The interface ensures that if the user edits a claim, the engine 126 re-checks dependencies and consistency (and possibly updates the description if a key term changed). There may also be features like **real-time claim tree visualization** (to see independent vs dependent relationships) and **word count checks** (important for abstract length or claim length constraints in some jurisdictions).

Once the user is satisfied with the draft application, the module 111 can finalize the documents. This includes compiling the specification, claims, abstract, drawings, and any required forms or annexes (like sequence listings or computer program listings, if applicable). At this point, the system can proceed to filing: it generates a filing-ready package. The user can invoke the integrated filing function, which hands off the finalized documents to the **submission adapter (105)** (as part of the prosecution subsystem) to electronically file the new application with the patent office. In some embodiments, the system can automatically populate an Application Data Sheet (ADS) with bibliographic data provided in the intake (such as inventor names, addresses, etc.) and prepare other needed forms (power of attorney, etc.), similar to how the forms generator 110 handles Office Action response forms. By doing so, the creation module 111 doesn't end with just a draft – it can carry the application through to actual filing, thereby placing the application into the prosecution pipeline.

FIG. 1B thus shows a self-contained but connected workflow: The invention intake 120 flows into AI drafting 122, guided by templates 124 and suggestions 126, presented via interface 130 to the user. The output is a polished patent application ready for filing. This module greatly accelerates the drafting process and ensures high-quality output by combining AI generation with human expertise and robust templates.

USPTO Prosecution Automation Module (FIG. 1C)

FIG. 1C focuses on the prosecution automation stage, which corresponds largely to the technology of the incorporated prior application. The drawing illustrates how an incoming Office Action is processed by the system and a response is generated and filed. The modules are numbered 101–110 as in the prior system.

At the left of FIG. 1C, the process begins with the **Secure Communication Module (101)** receiving an official communication from a patent office. Typically, this is triggered by an Office Action issued for the patent application. Module 101 securely checks for new communications – for example, it may monitor an email inbox that the USPTO uses to send alerts (via the Patent Center system). Alternatively or additionally, it may use an API to query the status of the application periodically. When an Office Action is detected, module 101 downloads the documents (which could include a PDF of the Office Action and an accompanying XML or text summary of rejections).

The downloaded Office Action is passed to the **Parser Module (102)**. In FIG. 1C, 102 is shown analyzing the document's content. The parser extracts key information: identification of the application (application number, examiner name, art unit, etc.), and a structured breakdown of

each rejection or objection in the Office Action. For example, the parser will identify that claims 1-5 are rejected under 35 U.S.C. §102 over Reference A, claim 6 is rejected under §103 over A+B, etc., along with the examiner's arguments. Cited prior art references are also extracted. This parsed data is essentially a machine-readable representation of the Office Action, which will be input to the AI generator.

Next, the **Drafting Engine (103)** for responses (which can be the same or a similar language model as used in module 122, but fine-tuned for argumentative writing) takes the parsed data and generates a **draft Office Action response**. FIG. 1C shows this as creating a document 103a that includes, for each rejection, a section with a suggested argument or amendment. For example, for a §102 rejection, the draft might argue that the cited reference fails to disclose a particular feature of claim 1; for a §103 rejection, it might argue lack of motivation or that the combination changes the principle of operation, etc. If claim amendments are likely needed, the engine will propose amended claim text (e.g., adding a new limitation to overcome the prior art). It uses the context of the original claims and specification (which it can retrieve from the repository 114 or from module 111's stored data) to ensure any amendments have support. At this stage, the draft response is comprehensive: it typically includes an introduction (identifying the application, the Office Action date, etc.), a summary of the amendments (if any), argument sections for each rejection, and a conclusion. It may also generate ancillary documents if needed – for example, if prior art was cited, an **Information Disclosure Statement (IDS)** listing those references can be prepared; if an amendment necessitates excess claims fees or the like, it notes that.

The **Compliance/Validation Module (104)** then automatically checks the draft response for any issues. This includes formatting checks (proper heading formats, margins, fonts as per Patent Office rules), completeness checks (did it address every rejection listed? did it include required statements like indicating no new matter was added?), and legal compliance (for instance, ensuring amendments are underlined, citations are properly formatted, etc.). Module 104 cross-references the Office Action to confirm that each rejection has a corresponding rebuttal in the draft. It also checks the claim amendments against the rules (e.g., no prohibited multiple dependent claim forms, antecedent basis issues resolved, etc.). If any compliance issue is found, 104 either fixes it automatically (for minor issues like formatting) or flags it for the user.

An optional sub-module in this stage is the **Deadline Calculator (109)** (depicted in FIG. 1C as a clock or calendar icon associated with the workflow). Module 109 computes the response deadline based on the Office Action's mail date and type. For example, it determines the statutory 3-month deadline and any extension possibilities up to 6 months (for USPTO practice). It logs this deadline in the docket and may also update the user's calendar or send reminders. In the integrated system, this deadline information is also important for the enforcement module if, for instance, a continuation application is planned or if abandonment triggers certain monitoring. The deadline calculator 109 ensures no Office Action is missed, and if the system generates an RCE (Request for Continued Examination) in response, it calculates new deadlines accordingly.

Another sub-module is the **Forms & Document Assembler (110)**. This component gathers all pieces of the response package. For example, if claim amendments are made, module 110 generates the clean and marked-up versions of the claims as required. It also fills out any necessary USPTO forms: an RCE form if an RCE is to be filed, an IDS form listing references (prefilled with citation information), extension of time fee calculation sheets, and so on. It essentially prepares a complete filing packet that includes the response letter (from 103),

amended claims (if any), and any supporting documents. These documents are in PDF or the required format ready for upload.

Now, the **User Review Interface (106)** comes into play. FIG. 1C indicates an interface where the attorney or agent can review the AI-generated draft and all documents. This is a crucial ethical checkpoint: the practitioner must approve and possibly edit the response. The interface may present a comparison of the current draft against the previous response or original claims, highlight what changes the AI is proposing, and allow line-by-line edits. The user can accept the draft as-is, or modify wording, add manual arguments, etc. The system tracks these edits (feeding them into the audit log 108). The user interface 106 also likely shows a checklist (e.g., "All rejections addressed?", "IDS included?") to help ensure nothing is overlooked. Once the practitioner is satisfied, they indicate approval (e.g., by applying a digital signature or clicking an "Approve for Filing" button).

Upon approval, the **Submission Adapter (105)** takes over. This module 105 uses the stored USPTO credentials (from vault 107) to log in to the Patent Office's electronic filing system (Patent Center or EFS-Web). It programmatically uploads each document in the package, selecting the appropriate document codes (e.g., "Amendment/Argument", "Claim Amendment", "Information Disclosure Statement" etc.), pays any required fees (using saved payment information if available), and submits the filing. Because it operates via the Patent Office's API or web interface, it mimics what a human filer would do but in an automated fashion. The submission adapter then waits for confirmation – it captures the electronic acknowledgement and the official **Filing Receipt** or confirmation number and stores that back into the system. FIG. 1C shows the flow from user approval to automated submission to the Patent Office, and then an acknowledgment coming back.

Throughout this process, **Credential Vault (107)** and **Audit Log (108)** underpin security and accountability. Module 107 ensures that the login credentials and any digital certificates needed for filing are stored securely (e.g., encrypted) and accessed only when the submission occurs, thereby protecting sensitive information. Module 108 records each significant action: when the Office Action was received, what data was parsed, the content of the AI draft (and even previous draft versions if the user iterated), the changes the user made, the final approval timestamp, and the confirmation of successful submission. This log 108 can be critical for later review or evidence, showing that a licensed practitioner ultimately approved the content, satisfying legal and ethical requirements.

The integration of module 115 with the rest of the platform offers additional advantages in context. For example, because the drafting module 111 knows the original disclosure, if an examiner cites new prior art, the system could automatically link that reference to the technical context in the original application (perhaps suggesting amendments or arguments referencing how the original spec distinguishes the new reference). Also, any bibliographic data (inventor names, title) parsed from the Office Action can update the central repository, ensuring consistency. If the application is allowed and a patent issues, module 115 can seamlessly pass that baton to module 112 (e.g., sending a message that patent number X has issued, which cues the enforcement monitoring to start).

In summary, FIG. 1C depicts an advanced prosecution automation subsystem where Office Action responses are handled end-to-end. This subsystem significantly reduces the attorney's

burden during prosecution, handling routine tasks swiftly while keeping the attorney in control of substantive decisions. It functions within the larger system by taking outputs from the drafting phase and feeding inputs to the enforcement phase, exemplifying the continuous lifecycle coverage.

Enforcement Docketing and Infringement Detection (FIG. 1D)

Turning to FIG. 1D, the enforcement docketing module 112 is shown as a flowchart or sequence diagram highlighting how the system monitors for infringement and handles enforcement actions. The process can be broken into several stages:

1. Market Monitoring (Stage 310): The system begins continuous monitoring once a patent (or even a published application, if the user desires pre-grant monitoring) is in force. A **Monitoring Pipeline (311)** is depicted in FIG. 1D. This pipeline may consist of web crawlers, data feeds, and APIs that gather information about new products, services, publications, or other companies' activities in the relevant technical field. The system can use various sources: for instance, RSS feeds of press releases, product catalogs, industry news sites, patent databases (to see if others cite the user's patent or file for similar inventions), and even social media or technical forums where product features might be discussed. The monitoring pipeline 311 likely has filters or agents configured specifically for the user's portfolio – for example, keywords related to the patented invention or names of known competitor companies. It continuously (or periodically) collects this data and stores it in a raw form in the repository.

2. AI Infringement Analysis (Stage 320): Collected data flows into an **AI Infringement Detection Engine (312)**. This corresponds to an AI model or set of heuristics that can screen the data for potential patent infringements. The engine 312 works by comparing the features described in the external data to the patented claims. For example, if the patent is about a specific type of battery technology, the engine will look for descriptions of products that mention similar technical features (materials, performance attributes, designs). It uses natural language processing to handle linguistic differences – the product brochure might not use the exact wording of the claim, but the AI can infer similarity (e.g., "multi-layer cell" vs "stacked battery layers"). In some embodiments, the engine might also take into account the **file history** (from module 115) to know which aspects of the claim were emphasized as novel, focusing the search on those aspects. The result of this analysis is a set of **Potential Matches (312a)** – basically flags that "Product X announced on date Y appears to include element A and B as in claim 1 of Patent Z."

To improve accuracy, the system may implement a scoring mechanism. Each potential match could get a confidence score indicating how likely it is that the product infringes the claim. The engine can be trained on examples (if available) of known infringements vs non-infringements to calibrate these scores. High-scoring matches will proceed to the next stage.

3. Claim-Product Mapping (Stage 330): For each high-confidence potential infringement, the system generates a detailed **Claim Chart (314)**. FIG. 1D shows a **Claim-Product Matching Module (313)** that takes a representative claim (often the broadest independent claim, or a set of key claims) and maps each limitation to evidence found about the product. For example, if claim 1 has elements [1a], [1b], [1c], the module 313 will extract snippets from the product description that correspond to each element and lay them out side by side. This results in a claim chart 314

(also known as an "Evidence of Use" chart). The claim chart might be formatted as a table, where one column lists each claim element and another column provides the matching product description or image citation. The matching module can use more than text; if, for instance, the product information includes images or diagrams (like from a product manual), the system could attempt to interpret those images or use associated captions to bolster the evidence. Generating such claim charts is traditionally a time-consuming task for attorneys or analysts, often taking days of work, but here the AI automates the initial draft in minutes[9].

The claim chart is then stored and presented to the user as part of an **Enforcement Alert (315)**. At this point, module 112 notifies the relevant users (e.g., the patent attorney or the in-house IP manager) that a potential infringement has been detected. The alert can be via email, dashboard notification, or even an automatic entry in the docket indicating an "Infringement Review" task.

4. Automated Enforcement Actions (Stage 340): Once an infringement alert is confirmed (either automatically if above a certain confidence, or after the user reviews the claim chart and agrees there is likely infringement), the system can initiate enforcement actions. FIG. 1D illustrates a **Licensing/Enforcement Workflow Module (316)**. This module 316 uses predefined rules or user settings to decide on next steps. For example, the user might have set a policy that for high-confidence infringements, the system should prepare a draft **Cease-and-Desist letter** or a **License Offer** letter to the offending company. In such a scenario, module 316 will auto-generate a communication: it will pull in the patent details (patent number, title, relevant claims), insert the company's name and product, and include an attachment or appendix (the claim chart 314) as evidence. The tone and content might follow a template (stored in the system) but can be adjusted by AI for the context (e.g., a friendly licensing invitation vs a stern infringement notice).

Simultaneously, the module docketing aspect kicks in: the system creates a case file in the docket for this enforcement action. It logs key dates, such as when the notice letter is sent, and sets a **follow-up reminder** (for instance, 30 days out to check if there's a response). This is indicated in FIG. 1D by an arrow from the workflow to a **Docketing Database (317)**. If the alleged infringer responds and perhaps negotiations begin, the user can log that in the system; otherwise, if no satisfactory response by the deadline, the system escalates.

5. Litigation Support (Stage 350, optional): If the enforcement escalates to a formal legal action (litigation), the system supports that as well. While the actual conduct of litigation is outside the scope of automation (lawyers will draft complaints, etc.), the system can facilitate by providing all the gathered evidence in an organized fashion and continuing to track deadlines. For example, if the decision is made to file a lawsuit, the system could generate a draft **Complaint outline** by inserting the patent details, a summary of the infringement (based on the claim chart), and jurisdiction information (perhaps suggesting suitable venues based on where the infringement occurs). This draft would still require attorney completion, but gives a head start. The system would then add the litigation deadlines (once known, like answer due date, discovery cutoff, etc.) into the docket 317. FIG. 1D notes a **Litigation Docketing** step, meaning integration with court dates.

At any time, the user can intervene in the enforcement process via the UI. For instance, the attorney might get the alert, review the claim chart, and decide not to pursue (maybe they disagree that the product infringes or it's a minor competitor). The system would then mark that

case as closed or set a low priority. Or the user might adjust the drafted notice letter before sending. The platform is flexible to allow human control.

One particularly powerful aspect of this integrated enforcement system is feedback and learning. Over time, as the user labels some alerts as false positives or identifies certain product descriptions as always irrelevant, the AI engine 312 can learn and improve its filtering. Conversely, if a particular infringement scenario was missed and later manually discovered by the user, they could feed that info back to train the system to catch similar cases.

In the context of the full lifecycle, the enforcement module 112 can also loop results back to earlier stages. For example, if many potential infringers are observed designing around a particular claim limitation, the platform could suggest (to the portfolio manager) that new patent applications (via module 111) be filed to cover those design-arounds, thereby continuously strengthening the portfolio. While this is an advanced feature outside the immediate enforcement workflow, it highlights how having all stages integrated opens up possibilities for strategic insights that standalone systems wouldn't provide.

In summary, FIG. 1D shows the enforcement module 112 as a proactive system that not only monitors and detects infringement using AI (block 312) but also initiates and manages the enforcement process via automated claim charts (314), docketing (317), and document generation (316). This ensures that once a patent is granted, its value is protected with minimal delay and oversight, enabling patent owners to respond swiftly to unauthorized use of their inventions.

Figure Drafting and Specification Synchronization (FIG. 1E)

FIG. 1E illustrates the Figure Drafting and Spec Synchronization Module 113 in action, particularly emphasizing the user interface and workflow for generating and integrating patent figures. In this depiction, we see a composite view: on one side, an example figure being generated, and on the other side, the corresponding text from the specification with integrated reference numerals.

At the core of module 113 is the **AI Figure Generator (402)**. This component accepts input in various forms to produce candidate drawings. One primary input is the **text of the specification or claim** that describes an embodiment (for example, if the spec describes "Figure 2 is a block diagram of the system architecture including an intake module, a processing module, and an output module…" the AI will pick out "intake module," "processing module," etc., and attempt to illustrate them). Another possible input is a **user-provided sketch or diagram** (401). Often, inventors or drafters have a rough hand-drawn sketch or a concept diagram; the system can take such an image and apply transformations to convert it into a formal patent drawing style. This might involve using computer vision to identify shapes and text in the sketch, then redrawing it with proper line art, reference numeral callouts, and without any extraneous marks (effectively "inkifying" a pencil sketch into a clean figure).

FIG. 1E shows an **Interactive Figure Editor UI (410)** where the user can see the generated figure. In this example, let's assume the invention is a system of modules (similar to our architecture). The AI generator 402 has created a block diagram: rectangles representing modules, each labeled generically or with a number, and arrows showing their interactions. Each block has an associated reference numeral (e.g., 101 for intake module, 103 for AI engine, etc.,

as in our earlier description). The figure might have been generated in draft form first with placeholders. The **user interface 410** allows the user to click on any element of the figure and see options or properties. For instance, clicking on a block might show a text field to edit its label. The user might change a label "Module A" to "Secure Communication Module". The moment they do that, the system can automatically update the numeral mapping such that "Secure Communication Module (101)" is recognized.

The UI also provides **iteration controls (412)** – perhaps a toolbar or dialog where the user can request alternate figure layouts or styles. For example, the user could ask for a different orientation (horizontal vs vertical flowchart), or if the first suggestion had three modules in one figure but the user wants each in separate sub-figures, they can adjust that. The AI can generate new variants accordingly. This iterative approach continues until the user is satisfied with the figure's content and appearance.

On the right side of FIG. 1E, the **Specification Text Panel (420)** is shown. This panel displays the portion of the patent specification corresponding to the figure. For example, it might show a paragraph describing FIG. 1E: "Referring to FIG. 1E, a system diagram is shown comprising an intake module 101, a processing engine 103, and an output interface 105…". The synchronization module actively links this text with the figure. In the UI, this can be visualized by highlighting – if the user clicks on reference number **101** in the text, the intake module box in the figure will highlight, and vice versa. This ensures the user can easily verify that each part is accounted for.

The **Reference Numeral Injector (404)** automatically places these reference numbers in the text. Initially, the text might have been written without numerals (especially if it came from AI drafting module 111). Once the figure is approved and finalized, module 404 will go through the spec and find where each component is described and insert "(###)" after the first occurrence of that component name. For example, if the text says "intake module" the first time, it becomes "intake module **(101)**". In subsequent mentions, some drafting conventions also include the numeral, and the system can do so or the user might choose to include numerals only at first mention – the system can follow the user's preferred convention. The injector also checks that every numeral used in the figure appears at least once in the text; if not, it can insert a brief description of that part to satisfy disclosure requirements.

The module 113 also looks at **claims** for synchronization (though typically, patent claims do not require reference numerals, sometimes practitioners include them for clarity). The system can optionally insert the same numerals into the claims as well – e.g., "a secure communication module (101) configured to…". These numerals in claims are usually put in parentheses and don't limit the claim, but they help map the claim to the figure for the reader. The inclusion of reference numerals in claims is often a stylistic choice; the system can do it automatically if the user opts in.

Beyond insertion, synchronization means if a numeral changes in the figure (say the user renumbers something or deletes a component), the text is immediately updated. If the user deletes "module 105" from the figure, the system will remove "(105)" from the text (and perhaps flag that the textual description might need revision). Conversely, if text changes – e.g., the user adds a new component description in the spec – the system can suggest adding a corresponding shape to the figure.

FIG. 1E also depicts possibly multiple figures, like FIG. 1A, 1B, etc., each with their own subparts. The module 113 can manage multiple drawings and their relationships. It ensures, for example, that numbering is consistent across figures when appropriate. (Common practice is to use a certain range of numbers per figure or continue numbering across all figures. The system could follow a rule like: start Figure 1A at 100, Figure 1B at 200, etc., or maintain a single sequence. The user could configure this.)

An important benefit of this module is maintaining **version control** between figures and text. The audit log (108) will record changes such as "Figure 1 revised, part 203 added" and "Spec paragraph [0051] updated to include (203)". This makes it easier later to see what changed when – for example, if during prosecution the attorney decides to add a new figure and description, the system can show those changes clearly.

From a technical standpoint, implementing the AI figure generation might involve using specialized frameworks (for instance, graph drawing algorithms for flowcharts, or using text-to-image models for schematic generation). The module might also incorporate libraries for specific diagram types (like circuit diagrams vs flow diagrams) depending on the technology domain. The invention doesn't require one specific technique – it broadly covers using AI to assist in figure creation in any suitable manner, combined with the unique integration to the spec.

By automating figure drafting and tying it directly to the specification, module 113 saves significant time and ensures accuracy. In conventional practice, figures and text can get out of sync if, say, a figure is modified late in the drafting process but the text isn't fully updated. Here, such inconsistencies are virtually eliminated by design. Additionally, the presence of figures early (even during initial drafting) can improve the quality of the patent application, since the visual thinking can influence better writing, and vice versa. It also impresses a synergy where the AI that helped write the spec (module 111) and the AI that draws the figures (113) effectively collaborate, mediated by the user.

Overall, FIG. 1E demonstrates an example embodiment of the figure support interface: The AI suggests a figure, the user refines it, and the system automatically propagates the final figure's details into the textual narrative of the patent. This component of the system is a novel assistive tool that brings the often siloed task of patent illustration into the same workflow as drafting, thereby streamlining the end-to-end preparation of a patent application.